Zur Abbildung unscharfer Wissensbasen innerhalb geschlossener Gefechtssimulationssysteme

Ron Alexander Wörsdörfer

Vollständiger Abdruck der von der Fakultät für Informatik von der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Gutachter/Gutachterin:

1. Prof. Dr. rer. nat. habil. Marko Hofmann
2. Prof. Dr. rer. nat. habil. Oliver Rose

Die Dissertation wurde am 26.11.2019 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Informatik am 06.12.2019 angenommen. Die mündliche Prüfung fand am 23.09.2020 statt.

Übersicht

Als Hauptproblem im taktischen Führungsprozess erweist sich die generell vorherrschende Ungewissheit, welche die Feststellung der eigenen und feindliche Lage erschwert. Der militärische Führer hat diesem Problem stets zu begegnen und insbesondere dann Entscheidungen zu treffen, wenn diesem nur ein teilweise vorhandenes, d. h. mitunter auch unvollständiges Lagebild vorliegt. Dem entsprechend hat die realistische Abbildung des taktischen Führungsprozesses innerhalb einer Computersimulation auch den Umgang mit Ungewissheit zu umfassen. Sogenannte Führungsautomaten dienen hierbei zur modellbasierten Abbildung militärischer Führer und Führungsprozesse, die ihren Haupteinsatz innerhalb der Simulation militärischer Auseinandersetzungen, auch bekannt als Gefechtssimulation finden.

Die heutzutage hauptsächlich in Nutzung befindlichen Führungsautomaten bedienen sich zur Abbildung der Führungsprozesse fast auschließlich regelbasierter Mechanismen. Sie ziehen dadurch deduktive Schlüsse auf Basis der ihnen in der Simulation vorliegenden Lageinformationen. Dies befähigt derartige Führungsautomaten grundsätzlich vordefinierte und auch lagebezogene Führungsentscheidungen zu simulieren. Problematisch ist jedoch, dass den Automaten hierfür entsprechende Regeln einprogrammiert werden müssen, welche mitunter auf eine exakte Definition der vorherrschenden Lage und damit paradoxerweise auf den Ausschluss von Ungewissheit angewiesen sind. Dies führt mitunter dazu, dass bereits definierte Regeln bei abweichenden und unvollständigen Lagebildern angepasst werden müssen, damit diese weiterhin nachvollziehbare und kontextadäquate Führungsentscheidungen abbilden.

Die Eigenschaft eines Führungsautomaten unter abweichenden und unvollständigen Lagebildern nachvollziehbare Entscheidungen zu simulieren wird auch als Robustheit bezeichnet. Inhalt dieser Disseration ist die programmatische Darstellung und Untersuchung bisher geläufiger Umsetzungen von Führungsautomaten, welche diese insbesondere auf das Vermögen zur Verarbeitung von Ungewissheit hin betrachtet. Das generelle Unvermögen der momentan noch in Nutzung befindlichen regelbasierten Mechanismen ist dabei der Abholpunkt, welche den Einsatz wissensbasierter Systeme und künstlicher Intelligenz für Führungsautomaten untersucht. Speziell kann dabei gezeigt werden, dass diagnostische – d. h. medizinische Experten- – Systeme Mechanismen aufweisen, welche für Führungsautomaten den Einbezug von Ungewissheit erlauben.

Mit sog. kausalen Netzwerken wird dabei eine neue Form von Führungsautomat entworfen. Diese weist im experimentellen Vergleich zu regelbasierten Mechanismen tatsächlich durch den Einbezug von Ungewissheit eine höhere Robustheit vor. Der Nachweis hierfür wird in einem eigens entwickelten Experimentalsystem aufgezeigt, welches gemeinsam mit den in Zusammenarbeit mit militärischen Experten entwickelten Regelsätzen, Wissensbasen und einer umfassenden Experimentlage im Detail beschrieben ist. Dem aufgeworfenen Ergebnis folgend, findet ferner eine abschließende Rückführung der erörterten Erkenntnisse auf die vergangener wissenschaftlicher Arbeiten und konkrete Vorschläge für zukünftige Entwicklungen im Bereich der Führungsautomaten statt.

Abstract

One of the main problems in tactical leadership is uncertainty. Military leaders have to cope with uncertainty and most importantly make decisions even when they are affected by the fog of war. Hence a realistic computer simulation of military leadership has to cope with uncertainty as well. This is the domain of so called Command & Control (C2) automata, formal models of military leaders and leadership processes. Such automata are commonly used in simulations of warfare - often called combat simulations.

In combat simulation most of today's commonly deployed C2 automata make use of rule based mechanisms. These allow automata to judge their current state and conclude certain decisions based upon deductive reasoning. Such deductive decision making is pre-defined and state-dependant. Paradoxically this means for C2 automata that they have to be build in foresight for certain intended decisions and thus diverge from reality by excluding uncertainty by design.

The ability of C2 automata to make reasonable decisions in changing and uncertain environments is called robustness. This thesis takes a look at commonly used patterns for C2 automata and their ability to compute and cope with uncertainty. Starting with the aforementioned exclusion of uncertainy in rule based mechanisms, alternative formal models are investigated that are able to improve robustness. Such alternatives present themselves in knowledge based systems and - more generally - artificial intelligence. Evidently diagnostic, i. e. medical expert systems provide a fruitful basis to the inclusion of uncertainty in C2 models.

One of the most dominant variations of diagnostic systems are causal networks. This marks the starting point for a new class of C2 automata called Causal Automata. Causal Automata show an increased robustness in an experimental comparison with rule based automata. The proof is given within an experimental simulation environment. The said environment, knowledge bases and the utilized scenario have been developed in collaboration with military experts and are described in detail. Finally, a reflection of past research and the relevance of causal networks for the future of C2 automata is discussed.

Inhaltsverzeichnis

Abbildungsverzeichnis V

Tabellenverzeichnis VII

Abkürzungsverzeichnis VIII

1 Einleitung 1

2 Grundlagen 7
2.1 Grundlegendes zur (Gefechts-)Simulation . . . 7
2.2 Künstliche Intelligenz im Kontext dieser Arbeit . . . 12
2.3 Führungsautomaten . . . 17
2.3.1 Regelsteuerungen . . . 19
2.3.2 Algorithmische Ansätze . . . 24
2.3.3 Wissensbasierte Ansätze . . . 29
2.3.4 Kognitive Ansätze . . . 31

3 Unschärfen und Unsicherheiten 36
3.1 Extensionale Ansätze . . . 40
3.1.1 Beispiel eines Führungsautomaten . . . 42
3.1.2 Schwächen des Ansatzes . . . 47
3.2 Intensionale Ansätze . . . 52
3.2.1 Beispiel eines Führungsautomaten . . . 57
3.2.2 Schwächen des Ansatzes . . . 64
3.3 Konklusion . . . 69

4 Kausale Führungsautomaten 73
4.1 Propagierung des Belief State . . . 78
4.2 Anwendungsbeispiel . . . 86

5 Experimentalsystem 93
5.1 Gelände, Bebauung und Bewuchs . . . 94
5.2 Auflösung und Repräsentation militärischer Akteure . . . 95
5.3 Sensorik . . . 98
5.4 Logikkomponente und Wissensbasen . . . 101
5.4.1 Binärlogische Zustände und regelbasierte Wissensbasis . . . 102
5.4.2 Probabilistische Zustände und kausale Wissensbasis . . . 103

6 Experiment 105
6.1 Beschreibung des Szenars . . . 105
6.2 Inhalt der verwendeten Wissensbasen . . . 107
6.2.1 Regelbasierter Führungsautomat . . . 108
6.2.2 Kausaler Führungsautomat . . . 111
6.3 Experimentaufbau . . . 118
6.3.1 Versuchsreihe 1: Verhalten bei voller Stärke . . . 119
6.3.2 Versuchsreihe 2: Verhalten bei reduzierter Stärke . . . 125
6.3.3 Versuchsreihe 3: Verhalten bei Aufreiben bzw. Zerschlagen . . . 131
6.3.4 Versuchsreihe 4: Verhalten bei ebenbürtigem Feind . . . 137
6.3.5 Versuchsreihe 5: Verhalten bei fortschreitender Lageentwicklung . . . 142
6.4 Auswertung des Experiments . . . 148

7 Interpretation der Ergebnisse 152

8 Schlussbemerkungen 159

A Anhang 162
A.1 Maximum Cardinality Search . . . 162
A.2 Ermittlung von Laufzeit und Speicherbedarf . . . 164
A.3 Beispielanwendung des Belief Propagation Algorithmus . . . 168
A.4 Umfrageergebnis . . . 176

Literaturverzeichnis 178

Abbildungsverzeichnis

2.1 Beispiel einer deklarativen Wissensbasis . . . 15
2.2 Beispiel einer prozeduralen Wissensbasis . . . 16
2.3 Beispiel einer Drehbuchsteuerung . . . 20
2.4 Beispiel eines regelbasierten Automaten . . . 22
2.5 Beispiel eines algorithmischen Automaten . . . 26
2.6 Beispiel eines hybriden wissensbasierten Automaten . . . 30
2.7 Beispiel eines kognitiven Automaten mit vier Produktionsregeln . . . 33
2.8 Produktionsregeln als Entscheidungsbaum . . . 35

3.1 Ausgangslage mit Deckungsinformationen . . . 43
3.2 Gesamtgraph der extensionalen deklarativen Wissensbasis . . . 44
3.3 Beispielwerte in der extensionalen deklarativen Wissensbasis . . . 45
3.4 Extensionaler Ansatz eines hybriden wissensbasierten Automaten . . . 47
3.5 Beispielgraph (a) und dessen zugehöriger Cliquengraph (b) . . . 57
3.6 Gesamtgraph der intensionalen deklarativen Wissensbasis . . . 60
3.7 Intensionaler Ansatz eines hybriden wissensbasierten Automaten . . . 63
3.8 Beispiel zur Verdeutlichung des Problems der engsten Bezugsklasse . . . 67
3.9 Unscharfe Ausgangslage . . . 70

4.1 Beispielhaftes Fragment eines Chow-Liu Baumes . . . 81
4.2 Nachrichtenfluss im Beispielnetz . . . 85
4.3 Kim und Pearls Belief Propagation Algorithmus . . . 86
4.4 Initialisiertes (a) und instantiiertes (b) Beispielnetz . . . 88
4.5 Änderung der Lage und die resultierende Parameterbelegung . . . 90
4.6 Beispiel eines kausalen Führungsautomaten . . . 92

5.1 Bilden des Sichtfeldes (a) ohne und (b) mit Einschränkung . . . 100
5.2 Relative Positionen eines Agenten . . . 100

6.1 Gesamtlagebild des Experiments . . . 106
6.2 Kausales Netzwerk des Experiments . . . 115
6.3 Initialisiertes kausales Netzwerk des Experiments . . . 117
6.4 Aufbau und Ablauf der ersten Versuchsreihe . . . 120
6.5 Aufbau und Ablauf der zweiten Versuchsreihe . . . 126
6.6 Aufbau und Ablauf der dritten Versuchsreihe . . . 131
6.7 Aufbau und Ablauf der vierten Versuchsreihe . . . 137
6.8 Aufbau und Ablauf der fünften Versuchsreihe . . . 142
6.9 Umfrageergebnisse zu den Versuchsreihen 4 und 5 . . . 150

7.1 Belegung des kausalen Netzwerks in Versuch 5 II 154
7.2 Belegung des kausalen Netzwerks in Versuch 5 III 154
7.3 Relevante Markov-Decken des kausalen Netzwerks in Versuchsreihe 5 156

A.1 Maximum Cardinality Search Algorithmus 162

Tabellenverzeichnis

3.1 Zuweisung der Gewissheit . . . 45
3.2 Eliminations- und Cliquenordnung des Beispielgraphen . . . 60
3.3 Wahrscheinlichkeitsverteilungen des intensionalen Führungsautomaten . . . 61

4.1 Formeln des Belief Propagation Algorithmus . . . 85

5.1 Deckungswerte je Gelände- und Einheitentyp . . . 95

6.1 Entitäten des Experiments . . . 107
6.2 Binärlogische Zustände des Experiments . . . 110
6.3 Prozedurale Wissensbasis des regelbasierten Führungsautomaten . . . 110
6.4 Probabilistische Zustände des Experiments . . . 113
6.5 Wahrscheinlichkeitsverteilungen der deklarativen Wissensbasis . . . 116
6.6 Prozedurale Wissensbasis des kausalen Führungsautomaten . . . 118
6.7 Ergebnis Versuchsreihe 1, Versuch I . . . 120
6.8 Ergebnis Versuchsreihe 1, Versuch II . . . 122
6.9 Ergebnis Versuchsreihe 1, Versuch III . . . 123
6.10 Ergebnis Versuchsreihe 1, Versuch IV . . . 124
6.11 Ergebnis Versuchsreihe 2, Versuch I . . . 126
6.12 Ergebnis Versuchsreihe 2, Versuch II . . . 128
6.13 Ergebnis Versuchsreihe 2, Versuch III . . . 129
6.14 Ergebnis Versuchsreihe 2, Versuch IV . . . 130
6.15 Ergebnis Versuchsreihe 3, Versuch I . . . 132
6.16 Ergebnis Versuchsreihe 3, Versuch II . . . 133
6.17 Ergebnis Versuchsreihe 3, Versuch III . . . 135
6.18 Ergebnis Versuchsreihe 3, Versuch IV . . . 136
6.19 Ergebnis Versuchsreihe 4, Versuch I . . . 138
6.20 Ergebnis Versuchsreihe 4, Versuch II . . . 139
6.21 Ergebnis Versuchsreihe 4, Versuch III . . . 141
6.22 Ergebnis Versuchsreihe 5, Versuch I . . . 143
6.23 Ergebnis Versuchsreihe 5, Versuch II . . . 144
6.24 Ergebnis Versuchsreihe 5, Versuch III . . . 145
6.25 Ergebnis Versuchsreihe 5, Versuch IV . . . 147
6.26 Bewertungsschlüssel zur Auswertung der Umfrage . . . 150

A.1 Anwendung des MCS-Algorithmus . . . 163
A.2 Datenquelle des Umfrageergebnis . . . 177

Abkürzungsverzeichnis

Abkürzung	Bedeutung
ACOMSIM	**A**symmetric **com**bat **sim**ulation
ABSEM	**A**genten-**B**asiertes **S**ensor-**E**ffektor-**M**odell
AHEntwg	**A**mt für **He**eres**entw**icklun**g**
CF	**C**ertainty **F**actors (Sicherheitsfaktoren)
EINSTein	**E**nhanced **I**SAAC **N**eural **S**imulation **T**oolkit
GBL	**G**eneric **B**ehaviour **L**anguage
GESI	**Ge**fechts**si**mulationssystem (Produkt)
GSS	**G**efechts**s**imulations**s**ystem (Abk.)
GUPPIS	**G**efechtssimulationssystem zur **U**nterstützung von **P**lan- bzw. Stabsübungen und **P**lanuntersuchtungen **i**n **S**täben
HORUS	**H**eeresmodell für **O**perations **R**esearch **u**nd zur Ausbildung von **S**täben
ITSimBw	**IT-Sim**ulation der **B**undes**w**ehr
JASS	**J**oint **A**gent-based **S**imulation **S**ystem
JOANA	**J**oint **O**perations **A**rmy, **N**avy, **A**irforce
KI	**K**ünstliche **I**ntelligenz
KonzSimLdschH 2020+	**Konz**ept zur **Sim**ulations**l**an**dsch**aft des **H**eeres **2020+**
KORA/OA	**Ko**rps**r**ahmenmodell für die **A**usbildung von **O**ffiziers**a**nwärtern
Lex-BFS	**L**exicographic **B**readth-**F**irst **S**earch
MANA	**M**ap-**A**ware **N**on-uniform **A**utomata
MEU	**M**aximum **E**xpected **U**tility
MCS	**M**aximum **C**ardinality **S**earch
NDM	**N**atural **D**ecision **M**aking
PAXSEM	**P**eace-support **A**nd e**X**tendable **S**ensor **E**ffector **M**odel
SIRA	**Si**mulationssystem zur Unterstützung von **Ra**hmenübungen
US DoD	**U**nited **S**tates **D**epartment **o**f **D**efense
VBS	**V**irtual **B**attle**S**pace

1 Einleitung

Das Führen von Menschen auf dem Gefechtsfeld wird von vielen Faktoren bestimmt. Gleichwohl Doktrin und Ausbildung dem militärischen Führer ein grundsätzliches Handlungswerkzeug zur Hand geben, ist es stets die nicht vollständig erfassbare Natur des Gefechts, welche sorgfältige Planung und Handlungssicherheit auf die Probe stellen. Scheinbar sicheres Wissen über Absicht und Lage des Feindes erweisen sich hierbei oftmals nur als grobe Anhaltspunkte, da die persönliche Wahrnehmung oder die Wahrnehmung eines Personenkreises oftmals von nicht voraussehbaren, extrinsischen Faktoren beeinflusst wird. Beispiele wie die Kundus-Affäre[1] zeigen dabei, dass speziell Ungewissheit innerhalb taktischer Führungsentscheidungen nach wie vor einen großen Einfluss besitzt. Dadurch bestätigt sich auch heute noch Carl von Clausewitz (1832, 45) These eines „Nebel des Krieges“:

> *„Der Krieg ist das Gebiet der Ungewißheit; drei Viertheile [!sic] derjenigen Dinge, auf welche das Handeln im Kriege gebaut wird, liegen im Nebel einer mehr oder weniger großen Ungewißheit. Hier also zuerst wird ein feiner, durchdringender Verstand in Anspruch genommen, um mit dem Takte seines Urtheils [!sic] die Wahrheit heraus zu fühlen.“*

Dabei ist es nach wie vor ein Bestreben moderner Streitkräfte, den Nebel des Krieges zu durchdringen. Technologische Errungenschaften und Weiterentwicklungen ermöglichen heutzutage Echtzeit- und Fernaufklärung, sind jedoch nicht in der Lage, Ungewissheit und Unvorhersehbarkeit im militärischen Führungsprozess zu beseitigen. Dies lässt sich dadurch begründen, dass Gefechte nicht punktuell feststellbare und statische Situationen darstellen. Einen besonderen Stellenwert nimmt dabei die Dynamik des Gefechts ein, da diese dem militärischen Führer ein ständiges Abwägen der eigenen Alternativen und der vermuteten feindlichen Absicht abverlangt. Damit mag der Erfolg im Gefecht einerseits, ganz im Geiste des o. a. Zitats von Clausewitz, vom Vermögen des militärischen Führers abhängen, durch den Nebel des Krieges zu blicken. Im modernen Gefechtsfeld wird dies jedoch andererseits insbesondere durch ständig wechselnde Gefechtssituationen erschwert, wie bspw. durch Betros (1991, 3) festgestellt wird:

> *„Changing situations introduce added uncertainty that may confound the effort to see through the fog of war. The environment of wartime uncertainty leaves commanders but one choice: they must structure their organizations to cope with incomplete information; those who excel at it improve their chances of success in battle.“*

Diese durch Betros beschriebene Fertigkeit zum Umgang mit unvollständigen Lagebildern und wechselnden Situationen ist es letztlich auch, welche eine der zentralen Herausforderungen des militärischen Führers darstellt. Ein modernes Ausbildungsmittel zur Anerziehung dieser Fertigkeit stellt die Gefechtssimulation dar. Ziel ist es dort, ganz nach der Devise "train as you fight", dem militärischen Führer in der Dynamik des Gefechts zu schulen und diesen im weiteren Sinne darauf vorzubereiten, auch in unbekannten Situationen

1 Vergleiche hierzu (Deutscher Bundestag, 2011). Die durch einen deutschen Oberst angewiesene Bombardierung eines Tanklastzugs in Afghanistan forderte einen in den deutschen Medien kontrovers diskutierten Kollateralschaden.

Handlungssicherheit zu besitzen. Die hierfür in der Bundeswehr genutzten Simulationssysteme GUPPIS[2], SIRA[3] und VBS[4] entsprechen dabei diesem Bedarf. Größtes Manko dieser Systeme stellt jedoch der hohe Personalbedarf für die Bedienung dar[5]. Um diesen Personalbedarf zu verringern, etablierten sich ab den 1970er Jahren automatisierte Steuerungsmechanismen, welche an Stelle eines menschlichen Bedieners militärische Einheiten oder Verbände innerhalb eines Simulationslaufs kontrollieren und dadurch den benötigten Personalbedarf reduzieren.

Dies stellte gewissermaßen die Geburtsstunde der sog. Führungsautomaten dar. Führungsautomaten fanden dabei ihren Weg in viele moderne Gefechtssimulationssysteme, wo diese gewinnbringend eingesetzt werden konnten. Diesem Erfolg folgte jedoch zu Beginn des 21. Jahrhunderts eine wissenschaftliche Stagnation, da den Führungsautomaten als etablierten Mechanismen im militärischen Gebrauch kein fortwährender Innovationsbedarf zugeschrieben wurde und bald innerhalb der marktverfügbaren Simulationssysteme zu monolithischen Systemen heranwuchsen. Bestes Beispiel hierfür ist der Führungsautomat von GUPPIS, welcher zuletzt auf dem Fähigkeitsstand der 1990er Jahre verblieb und seitdem – trotz intensiver Nutzung des Simulationssystems – nicht mehr weiterentwickelt wurde. Die Bundeswehr war sich dieses Umstandes durchaus bewusst, wie beispielsweise aus einem Vortrag eines ehemaligen Gruppenleiters des Heeresamtes V (3) hervorgeht:

> *„Im Anwendungsbereich Analyse und Planung könnten wir* [das Heer; Anm. d. Verf.] *mit dem gegenwärtigen Stand der Modelle ohne weitere Automatisierung alle wesentlichen Analysen für das Heer durchführen. Hier sehe ich daher keinen dringenden Handlungsbedarf.* [...] *Im Bereich der Ausbildungssimulationen haben wir demgegenüber dringenden Handlungsbedarf.“*(Konert, 2000, 38)

Nichtsdestotrotz blieb die Entwicklung der Führungsautomaten zu diesem Zeitpunkt – zumindest nach Kenntnis des Autors – stehen. Dieser offensichtliche Innovationsstop ist jedoch kein deutsches Phänomen, wie bpsw. durch Tolk (2017) mit Blick auf die USA bestätigt wird:

> *„The development of command and control automata for higher echelons has been dormant for quite some time. Most approaches utilize still the semi-automated forces approach, which requires a significant part of human decision support. In higher analysis tool, the aggregation level is so high that attrition and movement is modelled on such high aggregation levels that command agents make no sense. Only with the recent events* [...] *the interest in tactical and operational command agents gains some speed again, but in a nutshell: we are more or less where we left 10 years ago.“*

Dies ist aus zweierlei Hinsicht problematisch. Einerseits erfolgt durch die ehemals für Führungsautomaten wissenschaftlich führenden Nationen keine weitergehende Forschung und Entwicklung neuartiger Ansätze. Andererseits werden militärische Führer am modernen Ausbildungsgerät mit veralteter Technik ausgebildet. Beide Punkte wiegen hierbei mit Blick auf die Umsetzung der in den Führungsautomaten verwendeten

2 Ein bekannterer Bestandteil von GUPPIS ist KORA/OA, welches zur Ausbildung für den Einsatz von Landstreitkräften bzw. deren Unterstützung durch Luft- und Seestreitkräfte innerhalb eines Korps bzw. eines Land Component Command im Divisions- und Brigaderahmen dient.

3 SIRA, auch bekannt als GESI, wird oftmals auch als Gefechtsübungssimulationssystem klassifiziert und dient vorrangig der Ausbildung von Battalionsführern und Kompanieführern.

4 VBS wird zur taktischen Ausbildung bis zur Bataillonsebene und Einsatzauswertung in den Domänen Land, Luft und See verwendet.

5 In Vergangenheit führte dieser Umstand bspw. dazu, dass für die simulationsgestützte Ausbildung eines einzelnen Brigadestabes ein komplettes Bataillon zur Bedienung des Systems vonnöten war

künstlichen Intelligenz schwer: Alle sich momentan in Nutzung befindlichen Führungsautomaten sind nach wie vor – so lässt sich zeigen – unfähig zur adäquaten Verarbeitung von Ungewissheit, wodurch eine elementare Charakteristik[6] des militärischen Führungsprozesses nach wie vor nicht umgesetzt wird. Dieses in Ausbildung und Übung noch mittels Bedienpersonal ausgleichbare Manko, wiegt jedoch ungleich schwerer in den Anwendungsbereichen der Analyse und Planung. So bedarf insbesondere die Gefechtssimulation zu analytischen Zwecken einem hohen Grad der Automatisierung, da diese nicht nur verstärkt stochastische Modelle beinhaltet, sondern auch hierfür eine hohe Anzahl an Experimentdurchläufen benötigt. Damit hängen Aussagefähigkeit und Nutzen in den Anwendungsbereichen der Analyse und Planung auch stark vom Leistungsvermögen der automatisierten Steuerungsmechanismen ab. Hier ist insbesondere der Innovationsstop festzustellen, da in einem Großteil der momentan in diesem Feld in Nutzung befindlichen Simulationssysteme Führungsautomaten entweder gar nicht oder nur mit veralteten Mechanismen implementiert wurden.

In diesem Zusammenhang muss es Erwähnung finden, dass bis auf wenige Ausnahmen alle heute gebräuchlichen analytischen Simulationssysteme dem Ende der 1990er Jahre aufgekommenen Trend der agentenbasierten Gefechtssimulation entsprechen[7]. Die zur Behandlung analytischer Fragestellungen durch die Bundeswehr genutzten konstruktiven Simulationssysteme JASS[8] (ca. 2009), PAXSEM[9] (2008-2010) und ITSimBw (ca. 2005) weisen daher an sich einen grundlegend anderen internen Aufbau als die zur Ausbildung und Übung verwendeten, teils weitaus älteren Simulationssysteme wie GUPPIS (ca. 1978) und SIRA (ca. 1988) auf. Führungsautomaten werden hierbei in veralteter Form (JASS[10]), mit dem Versuch alte Mechanismen auf die moderne Struktur anzupassen (ITSimBw[11]), oder überhaupt nicht (PAXSEM[12]) eingebunden. Letzteres stellt auch ein international nachzuvollziehendes Phänomen dar, welches entsprechend in vergleichbaren Systemen wie z. B. dem neuseeländische Simulationssystem MANA (vgl. (Lauren et al., 2001)), dem türkischen ACOMSIM (vgl. (Cil und Mala, 2010)), oder den US-Systemen Pythagoras (vgl. (Bitinas et al., 2003)) und EINSTein (vgl. (Ilachinski, 1999)) ebenfalls vorgefunden werden kann. Es stellt sich daher in Summe durchaus die Frage, wie und in welchem Umfang momentane und zukünftige Führungsautomaten weiterentwickelt werden müssen, um der Ungewissheit innerhalb des modellierten Führungsprozesses in modernen Simulationssystemen angemessen Rechnung tragen zu können. Die Antwort zu dieser Frage

6 Vergleiche hierzu auch die Ausführungen von Reidelhuber (1979), welcher u. a. die adäquate Adressierung von Unsicherheit innerhalb der Informations- und Führungsvorgänge als zwingende Voraussetzung für ein Modell des militärischen Führungsprozesses beschreibt.

7 Der Trend der agentenbasierten Gefechtssimulation begründet sich in großen Teilen auf den Erfolg und das Aufkommen des sog. Data Farming – vgl. (Horne et al., 2014). Data Farming etablierte die agentenbasierte Simulation innerhalb des militärischen Kontexts. Agentenbasierte Simulation ist heute (2019) ein integraler Bestandteil zur Ergründung simulationsbasierter analytischer Fragestellungen innerhalb der Bundeswehr und NATO.

8 Ehemals bekannt unter den Namen JOANA, vereint dieses Simulationssystem fünf Teilmodelle, wobei das Modell HORUS zur Abbildung von Gefechten eingesetzt wird.

9 Hervorgegangen aus dem Prototypen ABSEM und PAX, einem Testsystem für agentenbasierte Sensor-Effektor-Modelle und einem Modell für Analysen zur Unterstützung von Friedensmissionen.

10 Der Führungsautomat von JASS arbeitet auf Basis einer hausintern entwickelten Skriptsprache, der sog. GBL. Ein wesentlicher Unterschied lässt sich zwischen dem in JASS und dem in GUPPIS verwendeten Führungsautomaten bis auf die einfachere Bedienung nicht ableiten.

11 Vergleiche die Beschreibung des Automaten in (Hugelmeyer et al., 2006). Tatsächlich wurde 2011 in ITSimBw bis auf einfache Lanchaster-Ansätze die Abbildungsmöglichkeiten von Gefechtsprozessen entfernt, womit das ursprünglich für die Simulation von IT-Netzwerken entworfene System nur noch selten Anwendung findet.

12 Im Simulationsverbund der Bundeswehr stellt PAXSEM das bis dato höchstauflösende Simulationssystem dar. Die Abbildung des Individualverhaltens erfolgt auf Basis von Verhaltensbäumen, einem Abkömmling endlicher Zustandsautomaten. Diese Sonderform der Verhaltensabbildung sorgt einerseits für eine leichte Handhabung in der Generierung neuen Verhaltens, ermangelt jedoch andererseits eines entsprechenden Konzepts für die Einbindung von Führungsautomaten.

lässt sich aus zwei Blickwinkeln heraus betrachten: Aus Sicht des militärischen Bedarfsträgers und aus akademischer Sicht. Nachfolgend wird dies nun näher beleuchtet.

Begonnen wird hierbei mit den konkreten Anforderungen des militärischen Bedarfsträgers an die Führungsautomaten eines agentenbasierten Gefechtssimulationssystems. Hierfür lohnt ein einleitender Blick auf die eigentliche militärische Definition des Begriffs Führung, wie sie bspw. gemäß der HDv 100/200 festgelegt wurde:

> *„Führung ist ein Prozess steuernden Einwirkens auf das Verhalten von Menschen, um ein Ziel zu erreichen. Sie umfasst den zielgerichteten Einsatz von Kräften und Mitteln sowie Information nach Raum und Zeit."(BMVg, 2010, 12, Absatz 101)*

Hieraus lassen sich die entscheidenden Schlüsselelemente eines Führungsautomaten innerhalb der agentenbasierten Gefechtssimulation extrahieren: Die Möglichkeit zur Steuerung von Agenten, konkrete Zielstellungen, sowie die Möglichkeit zur Auswertung von Informationen über einen gewissen Raum und eine gewisse Zeit. In Summe soll daher ein Führungsautomat die Möglichkeit bieten

(1) zur externen Beeinflussung der internen Verhaltenssteuerung eines Agenten,

(2) zum Setzen einer konkreten Zielstellung,

(3) zur Überprüfung, ob eine konkrete Zielstellung erreicht wurde (bzw. zu welchem Grad diese erreicht wurde) und

(4) zur Übertragung eines Systemzustands und der korrespondierenden Umweltinformationen an ein externes Steuerelement zu einer beliebigen Simulationszeit.

Mit besonderem Augenmerk auf ein geschlossenes Simulationssystem erweitern sich diese Anforderungen noch zusätzlich um einen weiteren Punkt, namentlich der selbstständigen Ablauffähigkeit der internen Steuerungsmechanismen. Die fünfte Anforderung lautet daher:

(5) Die Möglichkeit zur eigenständigen[13] Durchführung aller für den realisierten Führungsprozess relevanten Punkte.

Wie es bereits zu Beginn dieses Kapitels aufgeführt wurde, stellt es darüber hinaus eine Kernkompetenz moderner Führung dar, mit wechselnden Situationen und unvollständigen Informationen umzugehen. Dies umschreibt die sog. Robustheit des Führungsprozesses, d. h. die Fähigkeit zum Erhalt der Effektivität bei der Führung in ausgewählten Aufträgen, Situationen und Lagen (Alberts und Hayes, 2003, 128). Hieraus ergibt sich die finale sechste Anforderung:

(6) Die Möglichkeit zum Umgang mit wechselnden Situationen und unvollständigen Informationsständen.

13 Dieser Punkt adressiert die systeminterne Eigenständigkeit. Offene Gefechtssimulationssysteme besitzen zwar durchaus auch die Möglichkeit eigenständig Führungsprozesse entsprechend zu verarbeiten, benötigen hierfür jedoch einen externen Input, zumeist durch einen Bediener eingegeben, welcher den eigentlichen Prozess initiiert. Dies ist in der geschlossenen Gefechtssimulation per Definition nicht beabsichtigt.

Aus akademischer Sicht gibt dies einen Anhaltspunkt dafür, inwiefern zukünftige Führungsautomaten der Ungewissheit im Führungsprozess adäquat Rechnung tragen können. Wie es bereits innerhalb dieses Kapitels dargestellt (und durch Tolk (2017) nochmals aus Sicht der USA bestätigt) wurde, sind momentan gebräuchliche Ansätze für Führungsautomaten veraltet. Daher erscheint eine Modernisierung der Ansätze für Führungsautomaten unter Berücksichtigung der Forderungen des militärischen Bedarfsträgers nur dann als sinnvoll, wenn diese die hintergründigen Formalismen zur Modellierung der Entscheidungsprozesse ebenfalls erneuert. Erste Anhaltspunkte zu möglichen Lösungsansätzen lieferte hierzu bspw. Tolk (1995), welcher Techniken der künstlichen Intelligenz (KI) zur Optimierung von Führungsautomaten anwandte. Die hierbei gewonnen Rückschlüsse für die Eignung der KI zur Modellierung von militärischen Entscheidungsprozessen ist in anderer Form auch Hofmann (2000b) zu entnehmen, der diesen Schluss auf Basis der Forschungsergebnisse des US Army Artificial Intelligence Center (1996) zog. Im Geiste dieser Erkenntnisse erscheint daher die Verwendung der künstlichen Intelligenz zur Erneuerung der Führungsautomaten als probates Mittel zur Modernisierung. Dies eröffnet insbesondere Möglichkeiten zur Abbildung und Verarbeitung von Unschärfen im Entscheidungsprozess, die, anders umschrieben, nichts anderes als einen Angriffspunkt zur Berücksichtigung von Ungewissheit im Führungsprozess darstellen.

Führungsautomaten können dabei von der Einbindung von Ungewissheit im Entscheidungsprozess profitieren. So kann bspw. durch die Einbindung unscharfer Wissensverarbeitung der Wartungsaufwand von Wissensbasen für Führungsautomaten zusätzlich deutlich reduziert werden. In diesem Zusammenhang ist es zu betonen, dass die hierfür notwendigen Mechanismen nicht nur ausschließlich auf die Abbildung von Unschärfen im militärischen Entscheidungsprozess beschränkt sind. Innerhalb dieser Arbeit sei jedoch der Fokus auf den militärischen Anwendungsbereich gelegt, obgleich eine Ableitung in jedwede weitere verwandte Domäne möglich wäre. Im Speziellen bedeutet dieser Umstand aber auch im Umkehrschluss, dass die auch für nicht-militärische Anwendungszwecke geeigneten Ansätze der künstlichen Intelligenz u. a. im weiteren Verlauf dorthingehend untersucht werden müssen, ob diese zur Abbildung unscharfer Wissensverarbeitung respektive zur Modellierung von militärischen Führungsprozessen tauglich sind. Einer der entscheidensten Aspekte ist dabei, ob hierdurch überhaupt eine Verbesserung in der Abbildung der Entscheidungsprozesse herbeigeführt werden kann. Insbesondere muss dabei das Augenmerk auf die Herbeiführung einer gesteigerten Robustheit im modellierten Führungsprozess gelegt werden, wodurch in Summe die Forschungsfrage dieser Arbeit erwächst:

- Worin liegen die Unzulänglichkeiten bisheriger Ansätze von Führungsautomaten in der Abbildung von Ungewissheit begründet,
- welche Verfahren der künstlichen Intelligenz sind für die Abbildung von Ungewissheit innerhalb von Führungsautomaten geeignet und
- lässt sich durch die Abbildung von Ungewissheit die Robustheit von Führungsautomaten erhöhen?

Das Ziel dieser Arbeit ist es aufzuzeigen, dass die momentan noch in Nutzung befindlichen Ansätze für Verhaltens- und Führungsautomaten Ungewissheit – und damit speziell Unschärfen im Entscheidungsprozess – unzulänglich abzubilden vermögen. Im Vergleich mit alternativen Ansätzen der künstlichen Intelligenz soll gezeigt werden, dass momentan noch geläufige Umsetzungen hier in der Wissensrepräsentation einem Defizit unterliegen. Die Möglichkeiten in der Abbildung von Führungsentscheidungen

unter der Nutzung unscharfer Wissensverarbeitung werden ausgewertet, ein geeigneter Formalismus zur Verbesserung des vorherrschenden Defizits hergeleitet und ein neuartiger Führungsautomat erarbeitet. Letzterer wird in ein Testsystem implementiert, wo dieser in einem konkreten Vergleich mit einem archetypischen Führungsautomaten bisheriger Machart gestellt wird. Damit soll nachgewiesen werden, dass der neuartige Führungsautomat unter Sicherheit nicht nur gleichwertig zum bisherigen Führungsautomaten agiert, sondern darüber hinaus durch den Einbezug von Ungewissheit eine erhöhte Robustheit vorweist. Die Arbeit gliedert sich hierzu wie folgt:

- Kapitel 1 gibt einen groben Überblick über die Problemstellung und formuliert die Forschungsfrage auf Grundlage des Problems.
- Kapitel 2 befasst sich mit den notwendigen Grundlagen für die folgende Diskussion. Grundsätzliche Fachbegriffe der Simulation (Kapitel 2.1), die benötigten Fundamente der mathematischen Logik bzw. der künstlichen Intelligenz (Kapitel 2.2) und notwendiges Wissen über Führungsautomaten (Kapitel 2.3) werden erläutert. Die Abschnitte 2.3.2 bis 2.3.4 befassen sich dabei im Detail mit möglichen Ansätzen für Führungsautomaten und zeigen jeweils auf Basis eines konkreten Beispiels, wie diese praktisch umgesetzt werden.
- Kapitel 3 diskutiert Mechanismen zur Abbildung von Unschärfen. Extensionale (Kapitel 3.1) und intensionale (Kapitel 3.2) Ansätze werden hier grundlegend und mithilfe eines Anwendungsbeispiels beschrieben. In Konklusion (Kapitel 3.3) folgt hier die auf Basis der Vor- und Nachteile beider Ansätze ermittelte Festlegung auf intensionale Ansätze, welche in Folge als Ausgangspunkt für die Entwicklung eines Mechanismus für unscharfe Wissensverarbeitung gesetzt werden.
- Kapitel 4 adressiert mit kausalen Netzwerken die verwendete Ausprägung eines intensionalen Ansatzes. Hier werden die Grundlagen der Belief Propagation (Kapitel 4.1) und eines hierauf basierenden Führungsautomaten (Kapitel 4.2) erläutert.
- Kapitel 5 stellt das Experimentalsystem vor, welches für den Vergleich von Führungsautomaten speziell für diese Arbeit entwickelt wurde.
- Kapitel 6 beschreibt das Experiment. Auf einer konkreten Lage (Kapitel 6.1) aufbauend, werden folgend sowohl ein regelbasierter, als auch ein kausaler Führungsautomat (Kapitel 6.2) definiert. Die Führungsentscheidungen der Automaten werden für unterschiedliche Situationen ermittelt und hinsichtlich ihrer Plausibilität bemessen (Kapitel 6.3). Das Kapitel schließt mit einer Auswertung des Experiments ab (Kapitel 6.4).
- Kapitel 7 interpretiert die Experimentergebnisse und zieht entsprechende Rückschlüsse zur geäußerten Forschungsfrage.
- Kapitel 8 vergleicht das Ergebnis dieser Arbeit mit dem vergangener Arbeiten und zieht ein Resümee für zukünftige Forschungsansätze im Kontext von Führungsautomaten.

2 Grundlagen

2.1 Grundlegendes zur (Gefechts-)Simulation

Dieses Unterkapitel gibt eine Übersicht über die in der Arbeit verwendete Nomenklatur der Simulation. Beginnend mit der theoretischen Betrachtung des abstrakten Simulationsbegriffs, wird in den nachstehenden Abschnitten noch der Sonderfall der Gefechtssimulation nähere Betrachtung finden. Hierdurch soll im weiteren Sinne ein vokabularer Grundstock geschaffen werden, welcher zusammen mit den folgenden Unterkapiteln zur Beschreibung und Bewertung der Ansätze für Führungsautomaten dient.

Simulation, Systeme, Modelle und Agenten

Simulation beschreibt die Durchführung von Experimenten, um Sachverhalte respektive Vorgänge zu Übungs- oder Erkenntniszwecken modellhaft – möglichst wirklichkeitsgetreu – nachzubilden oder nachzuahmen (Dudenredaktion, o. J.). Gleichwohl sich eine Simulation dabei auch auf die Durchführung mit analogen Modellen abstützen kann, findet diese zur heutigen Zeit vorrangig digital, d. h. mithilfe eines Computers auf Basis virtueller Modelle und Experimente statt. Im Folgenden wird unter Simulation die synthetische Durchführung von Experimenten auf Basis eines auf Computern ausführbaren Modells verstanden. Vereinfacht kann ein derartiges Experiment in drei konsekutive Phasen aufgeteilt werden:

1. Die **Systemanalyse**, welche den zu betrachtenden Sachverhalt auf Basis systemtheoretischer Prinzipien dekomponiert,
2. die **Modellbildung**, die auf Basis des Ergebnisses der Dekomposition ein Modell des Sachverhalts mit hinreichender Genauigkeit abbildet und
3. das **Experiment**, welches das Verhaltens des Modells über einen festgelegten Zeitraum innerhalb einer definierten Umgebung betrachtet.

Ein System sei in dieser Arbeit dabei als ein Tripel $S = (U, M, R)$ – frei nach Forrester (1972, 9) und Law und Kelton (2000, 3) – definiert, wobei U die Umwelt beschreibt, in welcher das System sich befindet und M eine endliche Menge von Systemelementen $e \in M$ ist, die in einer Nachbarschaftsrelation R derart zueinander stehen, dass diese von ihrer Umwelt als einheitliches Ganzes unterschieden werden können.

Ein **Modell** K stelle darauf aufbauend ein vereinfachtes homomorphes oder isomorphes Abbild $k : M \to N$ der dekomponierten Systemelemente $m \subset M$ auf Modellelemente $n \in N$ dar (Domschke et al., 2015, 3). Es muss hierzu angemerkt werden, dass die Abbildung der dekomponierten Systemelemente innerhalb des Modells aufgrund einer z. B. homomorphen Abbildung nur bedingt die realen (oftmals dynamischen)

Zusammenhänge zwischen den Elementen zu erfassen vermag. Um die hierzu benötigten und im Vorhinein getätigten Vereinfachungen bzgl. der internen Zustandsübergänge und Verhaltensweisen einheitlich benennen zu können, ist es üblich, zwischen grundsätzlichen Begriffen zur Klassifikation dieser Modellannahmen zu unterscheiden. Die für den weiteren Verlauf in dieser Arbeit wichtigen Begriffe werden folgend nun näher erläutert – darüber hinausgehende Klassifikationen können bspw. Law und Kelton (2000, 5f), Bossel (2004, 21ff), oder Bungartz et al. (2009, 11f) entnommen werden.

Ein häufig im Kontext der Simulation zu findender Begriff ist der des **Determinismus**. Dieser aus Sicht der Algorithmik wohldefinierte Begriff beschreibt eine streng sequentielle Abarbeitung einer Kette von Anweisungen, welche bei gleicher (gültiger) Eingabe stets zu einem (gültigen) definierten Endzustand führen[1]. **Nichtdeterminismus** steht in diesem Kontext dafür, dass bei gleicher (gültiger) Eingabe kein (gültiger) Endzustand erreicht werden kann[2]. Des Öfteren wird im Rahmen des Nichtdeterminismus die Abweichung von der streng sequentiellen Abarbeitung im Falle einer wiederholten (gültigen) Eingabe verstanden. Eine in diesem Kontext häufig gewählte Repräsentation zur Abbildung dieses Phänomens stellen Ansätze auf Basis probabilistischer Zustandsübergänge dar, weswegen auch als Gegenpart des Determinismus innerhalb der Simulation die **Stochastik** Erwähnung findet. Daher wird vorrangig zur Klassifikation zwischen deterministischen und probabilistischen Zustandsübergängen unterschieden.

Nicht zu vernachlässigen ist die Repräsentation der Zeit innerhalb einer Simulation. Während reale Systeme gewissermaßen naturgemäß dem Diktat einer **kontinuierlich** fortschreitenden Zeit unterworfen sind, können Modelle hiervon gewollt abweichen und während einer real weiterlaufenden Zeit innerhalb der Simulation die Zeitfortschreitung verlangsamen, beschleunigen oder sogar unterbrechen. Eine unter diesem Kontext geläufige Notation stellen **diskrete** Zeitschritte[3] dar. Die Festlegung der gewählten Zeitrepräsentation ist dabei maßgeblich entscheidend für die Modellierung der Zustandsübergänge inner- und außerhalb des abzubildenden Systems. Ein speziell für die Abbildung zeitdiskreter Zustandsübergänge benutzter Mechanismus stellen **Ereignisse** (engl. events) dar, welche als Auslöser für einen Zustandswechsel und damit zum Fortschreiten der Simulationszeit dienen. Es ist somit zwischen zeitkontinuierlicher und zeitdiskreter Zustandsübergänge zu unterscheiden.

Ferner können in einer Simulation ein oder mehrere Systemelente zusammengefasst abgebildet werden, was eine vereinfachte Abbildung des Modellverhaltens erlaubt. Ein derartig zusammengefasstes Modellverhalten wird auch als **aggregiertes** (von lat. *aggregare*, ansammeln) Verhalten bezeichnet. Aggregation veringert dabei den Detailgrad des Modells, sorgt jedoch auf lange Sicht dafür, dass sich bestimmte Eigenheiten eines Systems vereinfacht und dadurch ressourcenschonender innerhalb einer Simulation darstellen lassen[4]. Derartige Vereinfachungen sind jedoch nicht immer zweckmäßig, insbesondere wenn diese zu einem unverhältnismäßig großem Verlust des Detailgrades führen, oder aufgrund einer Aggregation das abzubil-

1 Als Grundlage hierfür gelte in dieser Arbeit die Definition des Determinsmus auf Grundlage eines deterministischen endlichen Automaten - vgl. (Hopcroft et al., 2011, 71f).

2 Auch hier gelte als Grundlage die gültige Definition des Nichtdeterminismus auf Basis nichtdeterministischer endlicher Automaten - vgl. (Hopcroft et al., 2011, 85).

3 Diskrete Zeitschritte legen bspw. feste Zeitintervalle fest, in welchen die Simulationszeit voranschreitet. Näheres kann hierzu u. a. (Hedtstück, 2013, 10ff) entnommen werden

4 Wie Bossel (2004, 24) hierzu u. a. aufführt, lassen sich speziell in deterministischen Systemen unter Einsatz statistischer Mittelwerte zur Aggregation des Verhaltens von Einzelelementen bestimmte Realsysteme gut annähern.

dende Realsystem trivialisiert[5] wird. „*Hier muss das Verhalten der Schlüsselakteure („Agenten") explizit simuliert werden, oft mit Verfahren der nichtnumerischen Wissensverarbeitung, die menschliche* [oder eine vergleichbare] *Wissensverarbeitung prinzipiell abbilden kann.*"(Bossel, 2004, 24) Dies wird auch als **individuelles** Verhalten oder auch als **Individualverhalten** bezeichnet.

In diesem Zusammenhang lohnt es zu verdeutlichen, was unter dem Begriff eines **Agenten** und damit in der Abbildung von Individualverhalten grundsätzlich zu verstehen ist. Ein hierbei hilfreicher Schluss stellt die Tatsache dar, dass sich ein simuliertes Verhalten grundsätzlich nicht ohne eine konkrete simulierte Tat umsetzen lässt. Im einfachsten Fall bedeutet dies, dass ein Agent also, wie durch Russell und Norvig (2010, 4) dargestellt wird, lediglich ein Neutrum ist, welches innerhalb einer definierten Umwelt handelt (und daher Agent von lat. *agere* - tun /ausführen abgeleitet werden kann). Wird dieser Gedanke erweitert und diese Handlung als nachvollziehbar betrachtet, d. h. wird diese durch den Agenten so durchgeführt, dass ein bestimmtes Ziel (bestmöglich) erreicht werden kann, so heißt ein Agent rational (Russell und Norvig, 2010, 4). Dies wirft wiederum zwei Kernfragen auf, namentlich wie es einem Agenten überhaupt möglich ist zu handeln bzw. wie ein adäquates Ziel für eben jene Handlung ermittelt wird.

Tatsächlich benötigt ein Agent zusätzlich zur bloßen Anweisung zur Handlung auch eine Möglichkeit zur Beeinflussung seiner Umwelt, sowie eine Möglichkeit diese Umwelt entsprechend nach Informationen zu durchsuchen, um diese Handlung den äußeren Umständen entsprechend anpassen zu können. Dies wird innerhalb einer Simulation mittels sog. **Aktuatoren** zur Beeinflussung und **Sensoren** zur Wahrnehmung der Umwelt umgesetzt. Russell und Norvig (2010, 34) definieren einen Agenten daher informell: *„An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators."* Formal gelte jedoch die folgende Definition nach Ferber (1999, 9), da diese den Grundstein für weitere Betrachtungen im Rahmen dieser Arbeit legt:

> *„An agent is a physical or virtual entity*
>
> *a. which is capable of **acting in an environment**,*
> *b. which can communicate directly with other agents,*
> *c. which is driven by a set of tendencies (in the form of **individual objectives** or of a **satisfaction/ survival function** which it tries to optimise),*
> *d. which possesses resources of its own,*
> *e. which is capable of **perceiving its environment (but to a limited extent)**,*
> *f. which has only a partial representation of this environment (and perhaps none at all),*
> *g. which possesses skills and can offer services,*
> *h. which may be able to reproduce itself,*

5 Beispiele hierfür sind die Abbildung des Einsatzes von Reserven durch eine Permutation der verfügbaren Kräfte, die Abbildung des Einsatzes von Pionierkräften zur Gewässerüberquerung als verminderte Bewegungsgeschwindigkeit, usw. Es muss in diesem Falle jedoch klar festgestellt werden: Die Trivialisierung eines Modells ist kein Zeichen eines schlechten Modells, da diese im Rahmen der Modellbildung durchaus dem Zweck angemessene Abstraktionen darstellt. Im schlimmsten Fall führt eine zu starke Trivialisierung jedoch zur Verfälschung des Simulationsergebnisses.

> i. *whose behaviour* ***tends toward satisfying its objectives****, taking account of the resources and skills available to it and* ***depending on its perception, its representations and the communications it receives****."*
>
> [Hervorhebungen d. d. Verf.]

Wie insbesondere durch den Aufzählungspunkt (i) hierbei hervorgeht, stellt die Umsetzung eines Individualverhaltens eine Funktion der Wahrnehmung (und damit des Sensorinputs) eines Agenten dar, welche die individuelle Wahrnehmung einer passenden Handlung zuordnet[6].

Gefechtssimulation, Auflösung und Computer Generated Forces

Gefechtssimulation beschreibt die Simulation von bewaffneten Auseinandersetzungen zur Darstellung elementarer bzw. komplexer Gefechtsprozesse und deren operativen Abläufe über einen definierten Zeitraum. Ein Gefechtssimulationssystem[7] umfasst daher mannigfaltige Ausprägungsmöglichkeiten, die sich hauptsächlich am jeweilig beabsichtigten Einsatzzweck orientieren – siehe hierfür (Hofmann, 2000b, 32ff) für Näheres. Eine moderne Abgrenzung zur Unterscheidung der Gefechtssimulation stellt die Unterteilung nach dem übergeordneten Zweck dar. Zeitgenössisch ist hier zwischen **Ausbildungssimulation** und **Simulation zur Analyse- und Entscheidungsunterstützung** zu unterscheiden – vgl. hierfür u. a. die entsprechende Einteilung in (AHEntwg, 2016). Feiner wird hier unter den sog. **live-**, **virtuellen-** oder **konstruktiven** Simulationssystemen unterschieden. Diese Einteilung – zurückzuführen auf das US DoD (1995) – konzentriert sich insbesondere auf die Interaktionsmöglichkeiten von und mit dem zu betrachtenden System. Sie beruht auf einen steigenden Anteil der Automatisierung im Kern der Simulation, d. h. während bei einer live Simulation reale Bediener reale Systeme bedienen, bedienen innerhalb der virtuellen Simulation reale Bediener simulierte Systeme[8]. Einen Sonderstatus nimmt die konstruktive Simulation ein, da in eben jenem Fall simulierte Bediener simulierte Systeme bedienen. Dass diese Festlegung jedoch nicht immer trennscharf ist, verdeutlicht die folgende Definition des US DoD (1995, 57):

> „[Constructive models or simulations are] [m]*odels and simulations that involve simulated people operating simulated systems. Real people stimulate (make inputs) to such simulations, but are not involved in determining the outcome."*

Wie diese Definition bereits erwähnt, schließen konstruktive Gefechtssimulationssysteme auch externe Stimuli durch Bediener mit ein. Um hier dennoch eine scharfe Trennung zur Vollautomatisierung zu generieren, hat es sich bewährt Simulationssysteme – auf Grundlage systemtheoretischer Ansätze – ferner noch in **geschlossene** und **offene** Systeme zu unterteilen. Diese Unterteilung schließt implizit die durch einen externen Stimulus eingegebene Änderung des Simulationsverlaufs mit ein, wie beispielsweise von L. von Bertalanffy (1968, 40) gefolgert werden kann:

6 Näheres zur Umsetzung derartiger Agentenfunktionen, wird in Kapitel 2.2 nähere Erläuterung finden.
7 Manchmal auch als GSS abekürzt vorzufinden; dieser Terminus ist nicht zu verwechseln mit dem gleichnamigen rechnergestützten Simulationsmodell GESI.
8 Auch als human-in-the-loop bezeichnet.

> *„In any closed system, the final state is unequivocally determined by the initial conditions: e. g., the motion in a planetary system where the positions of the planets at a time t are unequivocally determined by their positions at a time t_0.* [...] *If either the initial conditions or the process is altered, the final state will also be changed. This is not so in open systems. Here, the same final state may be reached from different initial conditions and in different ways.“*

Mit besonderer Berücksichtigung möglicher Einsatzszenarien der Gefechtssimulation ist es ferner noch üblich, diese bezüglich der **Auflösung** der Simulation zu unterscheiden. Unterscheidungen finden hierbei hinsichtlich der expliziten Abbildung der simulierten Entitäten und Prozesse statt, d. h. mit welchem Detaillierungsgrad Simulationsobjekte und deren Verhalten abgebildet werden. Eine Simulation gilt dabei als **hochauflösend** (bzw. höchstauflösend)[9], wenn diese reale Systeme mit größtmöglicher Genauigkeit abbildet. Im Kontext der Gefechtssimulation bedeutet dies zumeist die explizite Abbildung einzelner Waffensysteme bis hin zum Einzelschützen, die feingranulare Abbildung elementarer Gefechtsprozesse und physikalisch korrekter Waffenwirkung. Dem entgegen steht die **aggregierte** Gefechtssimulation, welche vorrangig zur Abbildung komplexer Systeme wie zur Simulation von Abnutzungsgefechten zwischen Großverbänden und zeitkritischer militärischer Fragestellungen[10] Anwendung findet. Da innerhalb dieser Arbeit lediglich formale Modelle zur Abbildung des Führungsprozesses im Allgemeinen betrachtet werden, ist die Auflösung der jeweils untergeordneten Strukturen zu vernachlässigen. Zur Auflösung von Gefechtssimulationssystemen, insbesondere unter dem Aspekt der taktischen Führung, sei für Weiteres auf Pötzsch (2003) verwiesen.

Ein insbesondere in der Abbildung hochauflösender und konstruktiver Simulation häufig anzutreffender Begriff stellen ferner die sog. **Computer Generated Forces** (kurz: CGF) dar. Computer Generated Forces umschreiben eine Oberklasse teil- und vollautomatisierter Abbildungen des Modellverhaltens innerhalb von Konfliktsimulationen, was entsprechend auch einer Definition des US DoD (1995, 54) zu entnehmen ist:

> „[CGF is] [a] *generic term used to refer to computer representations of entities in simulations which attempt to model human behaviour sufficiently so that the forces will take some actions automatically (without requiring* [hu]*man-in-the-loop interaction).“*

CGF sind häufig als Synonym für Gefechtssimulation in der Wissenschaft anzutreffen, obgleich die Reduzierung von Computer Generated Forces auf die Gefechtssimulation – wie auch die oben aufgeführte Definition zeigt – den Begriff an sich stark eingrenzt. Nichtsdestotrotz stellen CGF wertvolle Innovationsträger für militärische Simulationen dar. So bedienen sich diese zur Abbildung der automatisierten Durchführung von Aktionen vorrangig der Ansätze der künstlichen Intelligenz, auf die im Schwerpunkt nun nachfolgend näher eingegangen wird.

9 Sowohl der Begriff hochauflösend, als auch höchstauflösend entstammen dem gebräuchlichen Vokabular der Geoinformationswissenschaften und dienen zur Klassifikation des Detailgrades von Kartenmaterial. Die Grenze zwischen hoch- und höchstauflösend kann nicht als trennscharf betrachtet werden, da diese sowohl durch die technischen Voraussetzungen, als auch durch die eigene Interpretation des Detailgrades abhängt. Im Sinne dieser Arbeit sind diese beiden Begriffe als gleichbedeutend zu werten.

10 Beispiele hierfür sind z. B. die simulative Unterstützung im Rahmen eines Wargames, der Eventualfallplanung im Rahmen des operativen Planungsprozesses, etc.

2.2 Künstliche Intelligenz im Kontext dieser Arbeit

Dieses Unterkapitel gibt eine grobe Übersicht über die im weiteren Verlauf verwendete Notation und Nomenklatur zur künstlichen Intelligenz. Von der in dieser Arbeit gültigen Definition ausgehend, werden im ersten Abschnitt die Grundlagen der Wissensverarbeitung und der hintergründigen mathematischen Logik aufgeführt. Der zweite Abschnitt befasst sich mit der konkreten Umsetzung von Wissensbasen und zeigt insbesondere auf, wie Programme zur künstlichen Intelligenz prinzipiell aufgebaut sind. Auf dieser Basis wird es anschließend möglich sein, Ansätze für Führungsautomaten formal darzustellen und mögliche Ansatzpunkte zur Verarbeitung unscharfen Wissens zu untersuchen.

Intelligenzbegriff und mathematische Logik

Während die Abbildung des Verhaltens eines technischen Systems innerhalb einer Simulation eindeutigen schematischen Abläufen folgt, erweist sich dies insbesondere in der Abbildung „denkender" Systeme als ungleich schwieriger. Komplex erweist sich dabei jedoch nicht die Ableitung des Denkprozesses per se, sondern der nicht eindeutig definierbare Begriff der hierfür zugrundeliegenden Intelligenz. Tatsächlich mag Intelligenz nicht ausschließlich für reines Denkvermögen sprechen, sie lässt jedoch eine entscheidende Folgerung zu: Sie ermöglicht strukturiertes und rationales Handeln. Soll letzteres von einer Maschine bewerkstelligt werden, so benötigt diese in gewisser Weise also auch eine Form von Intelligenz. Dies wird, insbesondere in Anwendungen innerhalb einer Simulation, auch als künstliche Intelligenz bezeichnet (Jackson, 1985, 29). Die Definition von KI leidet dabei gewissermaßen an denselben Grundproblem wie eigentliche Intelligenz, namentlich der nicht eindeutigen Bestimmbarkeit des Begriffs an sich. So folgert bspw. auch Luger (2005, 1): „*Thus the problem of defining artificial intelligence becomes one of defining intelligence itself: is intelligence a single faculty, or is it just a name for a collection of distinct and unrelated abilities?*" Eine Antwort auf diese Frage verwehrt sich diese Arbeit. Es sei stattdessen in vielerlei Hinsicht auf die durch Russell und Norvig (2010, 1) gesammelten Antwortmöglichkeiten verwiesen.

Fernab einer Definition der KI sei an dieser Stelle jedoch entscheidend, was eine Maschine – im weiteren Verlauf gleichzusetzen mit einem Computer – überhaupt dazu befähigt (künstlich) intelligentes Verhalten abzubilden. Wichtig ist hierbei der Grundsatz der Wissensverarbeitung, wortwörtlich der Wahrnehmung von Eindrücken (Designation) und der Verarbeitung dieser Eindrücke (Interpretation). **Designation** beschreibt im Falle einer Maschine dabei die Möglichkeit, zu jeder Zeit ein oder mehrere vertraute Ausdrücke aufzunehmen, welche an sich aus einer endlichen Anzahl von strukturierten und der Maschine vertrauten Symbolen bestehen. **Interpretation** beschreibe dazu passend das Vermögen, gegeben eines entsprechend vertrauten Ausdruckes einen indizierten (Verarbeitungs-)Prozess zu starten. Newell und Simon (1976) definieren dies – obgleich in abstrakterer Form – als ein physikalisches Symbolsystem, welches hypothetisch ausreichende Voraussetzungen für grundsätzlich intelligente Handlungen besitzt[11]. Dies stellt auch das weitergehende Verständnis künstlicher Intelligenz innerhalb dieser Arbeit dar.

11 Newell und Simon (1976, 116) erläuterten hierzu: „*By „general intelligent action" we wish to indicate the same scope of intelligence as we see in human action: that in any real situation behaviour appropriate to the ends of the system and adaptive to the demands of the environment can occur [...].*"

Designation beschreibt nicht ausschließlich die reine Fähigkeit zur Wahrnehmung, sondern gleichermaßen auch die Wahrnehmung in einer entsprechenden Form verarbeitungsfähig bereitzustellen. Ein wichtiger Begriff ist dabei die Wissensrepräsentation, welche im Falle der Wissensmodellierung für eine Maschine dazu dient, Wissen formal abzubilden. Der Hypothese von Simon und Newell folgend, bedeutet dies Symbole (**Semantik**) in strukturierten Ausdrücken (**Syntax**) so zusammenzufassen, dass diese Umfang und Zusammenhang der bekannten Umweltzustände erfassen können. Mathematisch kann dies mittels eines Axiomensystems respektive einem dem Axiomensystem begründenden logischen Kalkül abgebildet werden. Ein **Axiomensystem** sei dabei – frei der mathematischen Definition von Ebbinghaus et al. (1996, 7) folgend – als ein System von Aussagen Φ einer zugrundeliegenden Theorie[12] definiert. Dabei folge eine Aussage Ψ_i aus Φ, wenn in jeder Struktur, in der alle Aussagen aus Φ erfüllt sind, auch Ψ_i gelte. In diesem Zusammenhang sei auch eine Aussage naiv als ein deklarativer Satz definiert, welcher entweder wahr $(\top)$ oder falsch $(\bot)$ ist.

Die Syntax und Semantik dieser Aussagen werden – wie bereits im vorherigen Absatz angedeutet wurde – durch ein das Axiomensystem begründenden logischen Kalkül bestimmt. Im weiteren Verlauf beschränkt sich diese Arbeit hierbei auf das aussagenlogische Kalkül und die Prädikatenlogik erster Stufe. Die Zuordnung eines Wahrheitswerts, hier in Form einer Abbildungsfunktion $B : \mathcal{S} \to \mathcal{B}$ der Menge aller gültigen Symbole $\mathcal{S}$ auf eine zwei-elementige Menge $\mathcal{B} := \{\bot, \top\}$ von Wahrheitswerten dargestellt, sei im weiteren Verlauf auch als Belegung eines Symbols bezeichnet. Eine Formel $f \in \mathcal{F}$ im Sinne des aussagenlogischen Kalküls bezeichnet dabei eine endliche Menge von Symbolen $s \in \mathcal{S}$, welche durch logische Operatoren der Menge $\mathcal{N} := \{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$ miteinander verbunden sind. Die Abbildung $I : f \times \Sigma \to \mathcal{B}$ heiße dabei Interpretation oder (formale) Semantik über die Menge aller gültigen Belegungen Σ[13] (Schenke, 2013, 11).

Eine Erweiterung dieses Formalismus stellt die zusätzliche Einführung von Variablen als Platzhalter möglicher Symbolbelegungen dar[14]. Die bisher bekannte Menge logischer Operatoren muss hierfür durch drei weitere Operatoren zu $\mathcal{N}_{Prd} := \{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \forall, \exists, \equiv\}$ ergänzt werden. Anders als in der bisher betrachteten Aussagenlogik, stellen in der hierdurch gewonnenen Mächtigkeit nicht (zusammengesetzte) atomare Aussagen aus Symbolen, sondern (zusammengesetzte) atomare Ausdrücke von Thermen symbolischer Aussagen die Syntax dar. Die Abbildung $I_{Prd} : a \times \sigma \to \mathcal{B}$ heiße dabei Interpretation oder Modellbeziehung[15] für alle gültigen Aussagen $a \in \mathcal{A}$ innerhalb einer σ-Signatur, mit $\sigma = (\mathcal{N}, \mathcal{P})$ und $\mathcal{N} := \{f_1(\vec{v}_1), f_2(\vec{v}_2), ...\}$, $\mathcal{P} := \{p_1(x_1), p_2(x_2), ...\}$. Es sei jedoch an dieser Stelle zum besseren Verständnis und zur vorrangigen Vermeidung gedoppelter Notationen angemerkt, dass im weiteren Verlauf auf die Notation der hierdurch grob beschriebenen Prädikatenlogik (vorerst) keine Rücksicht genommen werden wird. Insbesondere da sowohl der Funktions-, als auch Prädikatbegriff induktiv definiert sind, sei für Näheres auf die Ausführungen von Schenke (2013, 31ff) und Ebbinghaus et al. (1996, 11ff) verwiesen.

12 Eine Theorie stellt eine endliche Anzahl von Aussagen über eine Signatur dar (Ebbinghaus et al., 1996, 7).
13 Zur Vollständigkeit sei hierzu aufgeführt, dass grundsätzlich $\mathcal{B} \subset \Sigma$ gilt.
14 Hierdurch entstehen sog. Prädikate: *„A **predicate** is a statement that may be true or false depending on the values of its variables.“*(Cunningham, 2012, 29)
15 Dies stellt eine starke Vereinfachung des tatsächlichen mathematischen Modellbegriffs dar. Genaueres zum Modellbegriff kann Ebbinghaus et al. (1996, 39ff) entnommen werden.

Folgernd lässt sich nun mathematisch formulieren, wie ein Schluss innerhalb einer Maschine tatsächlich realisiert werden kann. Ein Argument bestehe hierzu aus einer endlichen Liste von Aussagen $\Psi_1, \ldots, \Psi_n$ (genannt Prämissen), welche durch deren Gültigkeit daher ($\therefore$) zur Konklusion φ führen – siehe (Cunningham, 2012, 19):

$$\frac{\Psi_1 \ldots \Psi_n}{\therefore \varphi}$$

Hierbei gilt, dass ein Schluss und damit die Konklusion wahr (d. h. **valide**) ist, wenn auch dessen Prämissen wahr sind. Folgt dieser Schluss durch Ableitung des eigenen Wissens, wird dies auch als Inferenz bezeichnet. Ein Beispiel stellt hierfür der sog. **Modus Ponens**[16] dar, eine Schlussregel auf die Konsequenz einer Implikation auf Basis einer gültigen Prämisse:

$$\frac{\begin{array}{cc}[A] & [B] \\ \Psi \Rightarrow \varphi & \Psi\end{array}}{\therefore \varphi}\ (MP)$$

Systeme die Schlüsse auf Basis derartiger Inferenzregeln ziehen, werden auch als deduktive Systeme bezeichnet. Ein hierfür wichtiger Vertreter sind sog. Expertensysteme[17], die valide Schlüsse auf Grundlage des axiomatischen Wissens eines Domänenexperten[18] ziehen. Expertensysteme stellen die am häufigsten anzutreffende Umsetzung für Führungsautomaten innerhalb der CGF dar, nähere Ausführungen sind Kapitel 2.3 zu entnehmen. Anderweitig schließende Systeme – bspw. induktive Systeme (auch algorithmisch schließende Systeme genannt) – werden ebenfalls im Verlauf dieses Kapitels betrachtet.

Die Ausführungen dieses Abschnittes bilden nun die Grundlage, auf welcher die konkrete Implementierung künstlich intelligenter Steuerungsmechanismen erfolgen kann. Hierzu befasst sich der nachfolgende Abschnitt mit den unterschiedlichen Möglichkeiten eines Systems, kontextbezogenes Wissen repräsentativ zu speichern und einzusetzen.

Wissensbasen und wissensbasierte Agenten

Wie bereits im letzten Abschnitt aufgeführt wurde, ist es das Ziel innerhalb eines Axiomensystems Φ valide Schlüsse $\Phi \models \Psi_i$ ableitbarer Aussagen Ψ_i zu ziehen, die in Φ gültig sind ($\models$). Die Befähigung einer Maschine bedingt hierfür in der Praxis einer sog. Wissensbasis (engl. Knowledge Base), welche entsprechend in einer für die Maschine aufbereiteten Form, d. h. einer durch die Maschine interpretierbaren Sprache, bereitgestellt werden muss. Sie enthält zumeist in Sätzen formulierte Aussagen, welche den für die Maschine interpretierbaren Bereich der (beschränkten) Realität abbilden. Einen entscheidenden Einfluss hierauf hat die Interdependenz der Wahrnehmung und des Schlusses, d. h. in welcher Form von der Wahrnehmung

16 Der Modus Ponens ist nur eine von mehreren sog. Inferenzregeln. Für eine Übersicht weiterer Regeln siehe bspw. die entsprechenden Ausführungen von Cunningham (2012, 24).

17 *„An expert system is a computer program that represents and reasons with knowledge of some specialists subject with a view to solving problems or giving advice."* (Jackson, 1986, 2)

18 *„The **domain expert** is a knowledgeable and skilled person capable of solving problems in a specific area or **domain**."* (Negnevitsky, 2005, 29)

auf den konkreten Schluss gefolgert wird. Hier finden im Kern klassischerweise regelbasierte Wissensbasen Anwendung, welche sich prinzipiell auf Inferenz und damit auf Inferenzregeln abstützen. Eine Regel R_i stellt dabei entweder ein Axiom oder eine Folgerung aus einer oder mehreren anderen (axiomatischen) Regeln R_j dar. Das folgende Beispiel in Abbildung 2.1 diene hierfür zur besseren Ersichtlichkeit. Folgende Symbole finden darin Verwendung:

X beschreibe einen beliebigen Plattformabschnitt.

P_X sei wahr, wenn an der Stelle X ein Hindernis (Loch, Hügel) ist.

R_X sei wahr, wenn die Plattform an der Stelle X passierbar ist.

F_X sei wahr, wenn an der Stelle X die Zielflagge steht.

$$R_1 : \neg F_1$$
$$R_2 : \neg P_X \Leftrightarrow R_X$$
$$R_3 : \neg F_X \Rightarrow (R_X \vee P_X)$$

Abbildung 2.1: Beispiel einer deklarativen Wissensbasis

Wie sich hieraus zeigen lässt, können vermeintlich einfache Regeln bereits mannigfaltige Systemzustände beschreiben. So ließen sich beispielsweise durch Auflösung der Biimplikation und der Inferenzregel „Konjunktive-Eliminierung“ aus R_2 zwei weitere Regeln R_4 und R_5 folgern:

$$\dfrac{\dfrac{[R_2]}{\neg P_X \Leftrightarrow R_X}\,(\equiv)}{\dfrac{(\neg P_X \Rightarrow R_X) \wedge (R_X \Rightarrow \neg P_X)}{\therefore \neg P_X \Rightarrow R_X}\,(\wedge^-)}\,[R_4] \qquad \dfrac{\dfrac{[R_2]}{\neg P_X \Leftrightarrow R_X}\,(\equiv)}{\dfrac{(\neg P_X \Rightarrow R_X) \wedge (R_X \Rightarrow \neg P_X)}{\therefore R_X \Rightarrow \neg P_X}\,(\wedge^-)}\,[R_5]$$

Die Wissensbasis ermöglicht es daher aus „*Kein Hindernis an der Stelle X ist äquivalent zu die Stelle X ist passierbar*“ die Schlüsse „*Kein Hindernis an der Stelle X impliziert die Stelle X ist passierbar*“ und „*Die Stelle X ist passierbar impliziert es ist kein Hindernis an der Stelle X*“ ziehen zu können. Wie aus diesem Beispiel zu schließen ist, beschreibt eine derartig aufgebaute Wissensbasis einem System nachvollziehbar alles nötige Wissen, was zur Erfüllung der ihm gestellten Aufgabe (und damit zum Erreichen des Zieles) über die ihm wahrnehmbare Umwelt bekannt sein sollte. Dies wird auch als **deklarativer Ansatz** bezeichnet. Eine explizite Folgerung für zu tätigendes Verhalten ist diesen sog. deklarativen Wissensbasen jedoch nicht zu entnehmen, wodurch Folgeschlüsse auf Basis des abgebildeten Wissens anderweitig in das System eingebunden werden müssen. Eine Möglichkeit stellt in diesem Zusammenhang der sog. **prozedurale Ansatz** dar, welcher auf Basis der innerhalb der Wissensbasis enthaltenen logischen Verknüpfungen ein gefordertes Systemverhalten explizit folgert. Zum Vergleich mit dem deklarativen Ansatz, sei auf die folgende Abbildung 2.2 zur Verdeutlichung der Unterschiede verwiesen. Es gelten hierbei folgende Änderungen bzgl. der verwendeten Symbole:

X beschreibe den Plattformabschnitt, auf dem sich die Figur aktuell befindet.

$f(X)$ sei eine Funktion, welche die Figur um ein Feld nach rechts bewegt

$R_1 : F_X \Rightarrow \top$
$R_2 : R_{X+1} \Rightarrow f(X)$
$R_3 : P_{X+1} \Rightarrow f(f(X))$

Abbildung 2.2: Beispiel einer prozeduralen Wissensbasis

Prozeduralen Wissensbasen stellen programmnahe Steuerungsmechanismen dar, wodurch diese meist als Teil des Programmcodes ein fester Bestandteil des zu steuernden Systems sind. Mit besonderem Blick auf das vorangestellte Beispiel ist zu erkennen, dass eine prozedurale Wissensbasis ein ihr gestelltes Problem durch die Ausführung bestimmter Verhaltensweisen zu lösen vermag. Jedoch ist festzuhalten, dass dieses Vorgehen keine weitergehenden Schlüsse wie im Falle ihres deklarativen Pendant zulässt. Hybride Wissensbasen – d. h. Mischformen deklarativer und prozeduraler Wissensbasen – bieten hier einen Mittelweg, da diese die Ausdrucksstärke deklarativer Ansätze mit der expliziten Verhaltensadressierung prozeduraler Ansätze kombinieren und somit die Schwäche der einzelnen Ansätze ausgleichen.

Wie zuvor beschrieben wurde, weisen insbesondere prozedurale Wissensbasen konkrete Verhaltensanweisungen für ein System auf. Dies adressiert jedoch nicht, wie ein künstlich intelligentes Programm nun in Gänze konzipiert ist. Innerhalb dieser Arbeit sei ein entsprechendes Programm auf Basis eines Agenten definiert, folgernd der in Kapitel 2.1 aufgestellten Definition. Wie im vorigen Unterkapitel aufgeführt ist, besitzen Agenten sowohl die Möglichkeit ihre Umwelt über Sensoren wahrzunehmen, als auch mithilfe von Aktuatoren zu beeinflussen. Ziehen Agenten hierbei Schlüsse mithilfe einer Wissensbasis, so ist auch die Rede von wissensbasierten Agenten (Russell und Norvig, 2010, 235). Der Kern eines Agentenverhalten – auch als Agentenprogramm oder Verhaltensfunktion bezeichnet (Ferber, 1999, 24; Russell und Norvig, 2010, 35) – sei dabei als sog. **Sense-Decide-Act**-Schleife definiert. **Sense** stehe hier im wortwörtlichen Sinne für die Wahrnehmung der Umwelt durch die Sensoren des Agenten. **Decide** beschreibe die Entscheidungsfindung auf Basis der Rückschlüsse der wahrgenommenen Umwelt und **Act** eine entsprechende Beeinflussung der Umwelt gemäß der getroffenen Entscheidung. Dabei ist die Reihenfolge des Einbezugs eines jeweiligen Ansatzes für Wissensbasen durchaus entscheidend. So ist eine deklarative Wissensbasis bestenfalls im Schritt Sense beheimatet, während eine prozeduralen Wissensbasis im Rahmen des Decide-Schrittes eher ihren Platz findet. Dies zeigt bereits, dass das Agentenprogramm von wissensbasierte Agenten im optimalen Fall als hybride Wissensbasis zu konzipieren ist, da dies eine umfängliche Umsetzung der Sense-Decide-Act-Schleife darstellt. Wissensbasen und deren praktische Umsetzung innerhalb eines Agentenprogramms werden im weiteren Verlauf dieser Arbeit im Kontext der Führungsautomaten nähere Betrachtung finden. Zuvor bedarf es jedoch einer grundlegenden Betrachtung der Umsetzung von Führungsprozessen innerhalb der Gefechtssimulation, welche dem folgenden Unterkapitel zu entnehmen ist.

2.3 Führungsautomaten

Das folgende Unterkapitel definiert den Begriff der Führungsautomaten, die zur Abbildung von Führungsprozessen innerhalb der Gefechtssimulation Anwendung finden. Die konkreten Anforderungen hierzu werden dargestellt und hierauf aufbauend Ansätze im Detail anhand einer vorher getroffenen Einteilung in Automaten-Klassen erläutert. Jeder Ansatz ist dabei mit einem Beispiel versehen, welches nicht nur die grundsätzliche Funktionsweise, sondern auch ggf. dessen Unzulänglichkeiten aufzeigt.

Begriffsbestimmung und Anforderungen

Sollen innerhalb eines Gefechtssimulationssystems Führungsentscheidungen abgebildet werden, so geschieht dies in der Regel durch einen entsprechenden Automatismus, welcher entweder die Befehlseingabe eines externen Bedieners interpretiert oder einen autonomen Führungsprozess initiiert. Dies ist Aufgabe von Führungsautomaten, welche im engere Sinne formale Modelle militärischer Führungsprozesse bzw. militärischer Führer sind und den Ablauf realer oder simulierter Prozesse bzw. militärischer Operationen steuern (Huber, 2000, 263). Sie verringern den personellen Aufwand zur Einbringung von taktischen Entscheidungen in ein ausführliches Simulationsmodell – eine zwingende Voraussetzung, um derartige Entscheidungen bspw. im Rahmen analytischer Simulation reproduzierbar und mit vertretbarem Aufwand abzubilden. Ein Führungsautomat unterliegt hierbei dem Diktat der Auflösung des übergeordneten Simulationssystems, da hierunter nicht nur implizit die kleinste abzubildende Führungsebene, sondern auch die für die Führung relevanten Metadaten des Geländes überhaupt erst identifiziert werden können. Führungsautomaten stellen aus diesem Grund hochspezialisierte Steuerungsmechanismen dar, die zuweilen keine Übertragung der implementierten Mechanismen zwischen Simulationssystemen unterschiedlicher Auflösung erlauben. Zusätzlich herrscht eine starke Abhängigkeit des implementierten Mechanismus von der systemspezifischen Verhaltensteuerung der Simulationsobjekte vor, da die Umsetzung des Führungsprozesses in geeigneter Form die Semantik des oftmals systemeigenen Verhaltensformalismus adressieren muss. Führungsautomaten bedürfen somit eines nicht unerheblichen Entwicklungsaufwands, dessen Resultate sich an den systemspezifischen Limitierungen orientieren. In ihrem Einsatz werden Führungsautomaten an qualitativen Forderungen bemessen, welche in gleicher oder ähnlicher Form auch für reale militärische Führungsprozesse gelten. In Folgerung zu den durch Hofmann (2000b, 227ff) und Junge (2008, 6ff) gesammelten Forderungen, werden die für weitere Betrachtungen wichtigsten Begriffe nun nachstehend näher erläutert.

Erstes und mitunter entscheidendes Kriterium für die letztendliche Aussagefähigkeit der Simulationsläufe stellt die **Plausibilität** der durch den Führungsautomaten getroffenen Entscheidungen dar. Dies adressiert nicht nur die reine Abbildungstreue der inhärenten Prozessabläufe, sondern auch eine im Rahmen der gegenwärtigen Lage angemessene und zweckmäßige Führungsentscheidung. Während ersteres aufgrund erlassener Dienstvorschriften und Doktrinen klar definiert ist, stellt speziell die Bewertung einer Führungsentscheidung eine oftmals durchaus subjektive Einschätzung dar. Hier ist eine klare Trennlinie zur eigentlich wünschenswerten Validität zu ziehen, da es mitunter durch unterschiedliche Experten zu unterschiedlichen Ergebnissen ob der Bewertung auf Passgenauigkeit der Entscheidung in der gegenwärtigen

Situation kommen kann. Als hinreichendes Kriterium wird daher zusätzlich die **Nachvollziehbarkeit** der Führungsentscheidung mit der Plausibilität der Entscheidung gleichgestellt. Sind die Führungsentscheidungen in diesem Sinne nachvollziehbar, so sind diese im weiteren Sinne auch als plausibel anzusehen. Dies birgt jedoch eine nicht zu unterschätzende Komplexität, da im weitergehenden Sinne nicht nur die Nachvollziehbarkeit der letztendlichen Entscheidung in deren Ausführung, sondern auch durch deren Wissensbasis gegeben sein muss. Hier kann es mitunter – je nach tatsächlicher Realisierung der Wissensbasis – zu Problemen hinsichtlich einer Interpretierbarkeit durch externe Experten kommen. Es ist dabei beispielsweise durch die Augen eines unbedarften Domänenexperten leichter innerhalb einer prozeduralen Wissensbasis als innerhalb einer deklarativen Wissensbasis zu erkennen, wie ein konkreter Schluss nun tatsächlich zustande gekommen sein mag. Diskussionen ob der Nachvollziehbarkeit stellt das in der Praxis mitunter am häufigsten anzutreffende Problem in der Legitimation der Entscheidungen eines Führungsautomaten dar. Hier hat es sich bewährt vorrangig mit prozeduralen Ansätzen zu arbeiten, da diese nicht nur einfach vermittelt, sondern auch direkt sichtbare Ergebnisse erzeugen können. Dem entsprechend herrschen häufig Präferenzen für prozedurale Ansätze vor, welche im nächsten Unterabschnitt nähere Erläuterung finden.

Auf der Plausibilität aufbauend stellt sich die Forderung nach der **Robustheit** eines Führungsautomaten. Hierunter versteht sich die Fähigkeit eines Automaten in möglichst vielen wechselnden Situationen, d. h. bei wechselnder Geländedatenbasis bzw. wechselnder eigener Lage und Feindlage stabil zu nachvollziehbaren Ergebnissen zu kommen. Die Herbeiführung der Robustheit von Führungsautomaten erfolgt in der Regel durch Einfügen zusätzlichem kontextsensitiven Wissens. Ein mitunter dabei auftretendes Phänomen sind übersättigte Wissensbasen, d. h. Wissensbasen eines Automaten, die durch ständig wechselnde Anforderungen und für den aktuell beabsichtigten Zweck um relevante Symbole und Sätze übermäßig angereichert wurden. Dies birgt in der Praxis zweierlei Gefahren – einerseits steigt hierbei die Gefahr des Verlusts der Interpretierbarkeit und damit der Nachvollziehbarkeit der Wissensbasis. Andererseits kann es aufgrund der Übersättigung zu Inkonsistenzen, bis hin zum Auftreten struktureller Varianzen[19] kommen. Eine in diesem Sinne erweiterte Forderung ist daher auch die nach **Konsistenz** der Wissensbasis, wobei u. a. Antinomie und das Zutreffen unterschiedlicher Schlüsse zur selben Zeit die Funktionsfähigkeit des Automaten erheblich zu beeinflussen vermögen. Dies mahnt insbesondere zur Vorsicht in der Erstellung der Wissensbasen, wobei Widersprüche und logische Überlappungen sich zumeist nur durch intensives Testen, seltener durch einen formalen Vergleich der Aussagen klar erkennen und beseitigen lassen. Die Möglichkeiten zu Konsistenzprüfungen orientieren sich dabei am jeweiligen gewählten Ansatz zur Wissensrepräsentation. Da diese Ansätze selbst wiederum mannigfaltige Möglichkeiten zur Umsetzung von Führungsprozessen bieten, ist in diesem Zusammenhang eine tiefergehende Betrachtung bisher üblicher Mechanismen lohnenswert. Der folgende Unterabschnitt adressiert in diesem Sinne vier Archetypen, welche gem. der durch Hofmann (2000a, 18ff) vorgenommenen Einteilung nun nachfolgend diskutiert werden.

19 Als strukturelle Varianzen werden u. a. Diskontinuitäten im Ergebnis des simulierten Gefechtsverlaufs bezeichnet, die häufig auf Fehler im Führungsprozess begründet sind. Ein Beispiel wäre z. B. dass ein Führungsautomat ausreichendes Wissen für eine Gefechtsführung von Nord nach Süd, jedoch nicht von Ost nach West besitzt und infolgedessen bei einem Angriff aus östlicher Richtung den Befehl zum Ausweichen nach Osten evaluiert. Mehr zu strukturellen Varianzen und insbesondere zur Reduzierung dieser Randerscheinung kann Tolk (1995) entnommen werden.

2.3.1 Regelsteuerungen

Regelsteuerungen stellen den ältesten und nach wie vor auch gebräuchlichsten Ansatz zur Implementierung von Führungsprozessen innerhalb der Gefechtssimulation dar. Obgleich sich diese in Form und Aufbau im Detail unterscheiden, sind alle Umsetzungen im Kern grundsätzlich gleich, da sie Handlungsanweisungen in Form deduktiver Schlüsse abbilden. Ausgangspunkt dieser Anweisungen sind implikative Aussagen der Form $A \Rightarrow B$, welche oftmals auch mit einem Alternativfall zu $(A \Rightarrow B) \wedge C$ erweitert werden. Algorithmisch können derartige Schlüsse als „**IF** A **THEN** B [**ELSE** C]“ interpretiert werden, wodurch sich eine direkte Hinterlegung in Form von prozeduralen Programmcode anbietet. Derartig einprogrammierte Aussagen werden auch als **Regeln** bezeichnet, da diese dem Automaten de facto vorgeben, in welcher Situation (A) welches Verhalten (B) zu folgern ist. Formal lassen sich Regelsteuerungen als disjunktiv verknüpfte Implikationen definieren, wobei immer dann ein valides Verhalten gefolgert werden kann, wenn eine der Implikationen logisch wahr ist. Klassisch wird hier unter zwei unterschiedlichen Ausprägungen – den **Drehbuchsteuerungen** und den **regelbasierten Automaten** – unterschieden, welche nun folgend nähere Betrachtung finden.

2.3.1.1 Drehbuchsteuerungen

Wie schon aus dem Namen an sich zu folgern ist, folgen **Drehbuchsteuerungen** dem Prinzip einer festen Abfolge von Handlungen (Drehbuch, engl. *script*). Träger dieser Ablaufsteuerung stellt die Zeit, seltener jedoch auch ein vorgegebener Prozessablauf dar. In der Regel werden dabei zu einem fest vorgegebenen Zeitpunkt durch den Führungsautomaten vordefinierte Führungsentscheidungen getroffen. Aufgrund dieser Tatsache müssen Drehbuchsteuerungen für jedes neue Simulationsmodell bis zu einer gewissen Führungsebene immer wieder neu entwickelt werden, da der Steuerungsmechanismus sich streng an der gegebenen Ausgangslage und der sich durch die getroffenen Entscheidungen entwickelnden Gefechtsverläufe orientiert. Entscheidender Vorteil ist dabei, dass die explizite und zeitlich benannte Abfolge von Führungsentscheidungen dadurch nur geringen Freiraum für interne logische Inkonsistenzen zulässt. Nichtsdestotrotz stellt der für jeden neuen Anwendungsfall vorherrschende Anpassungsbedarf ein klares Manko dar, wodurch reine Drehbuchsteuerungen heutzutage hauptsächlich nur noch in virtuellen Simulationssystemen anzutreffen sind[20]. Zur Verdeutlichung der Funktionsweise sei auf Abbildung 2.3 als beispielhafte Umsetzung dieses Mechanismus nachfolgend verwiesen.

20 Siehe bspw. die Umsetzung in VBS. In der Regel erfolgt die Einbindung von Drehbuchsteuerungen in Form eingebetteter Skriptsprachen. Bekannte Vertreter sind u. a. die Skriptsprachen Lua und Python.

Programm 1 Drehbuchsteuerung

```
procedure C2(blue, red: entity)
    while true do
        if time == 0 then
            blue.engage(tPos);
            red.engage(tPos)
        else if time == 15 then
            blue.defend(tPos);
            break
        end if
    end while
end procedure
```

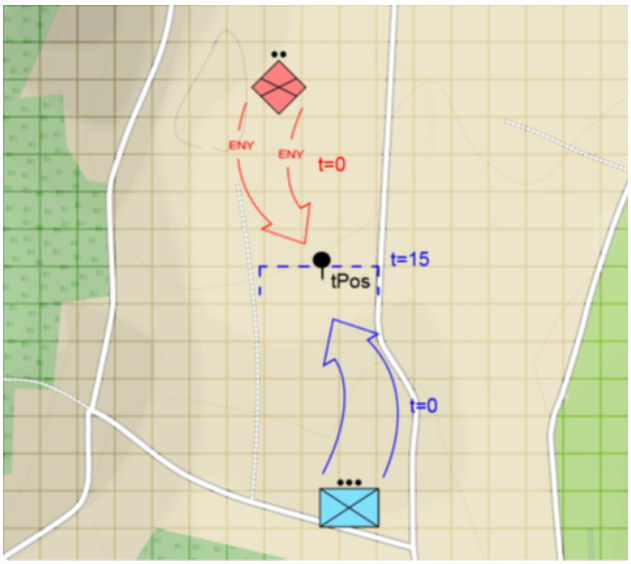


Abbildung 2.3: Beispiel einer Drehbuchsteuerung

Erläuterung des Beispiels Das Beispiel weist in Abhängigkeit der verstrichenen Simulationszeit den einzelnen Entitäten, hier einem blauen Jägerzug (blue) und einer roten Infanteriegruppe (red) Verhaltensprozeduren zu. Die Verhaltensprozeduren sind dabei als Black Box Implementierungen zu betrachten, d. h. zur vereinfachten Darstellung wird auf die Detailimplementierung an dieser Stelle nicht näher eingegangen. Zum Startzeitpunkt $t = 0$ beginnt Blau mit dem Angriff auf den designierten Zielpunkt **tPos** (Anweisung: *engage*). Zur selben Zeit beginnt Rot ebenfalls mit einem Angriff auf den Zielpunkt. Ziel des Jägerzuges ist gem. der taktischen Vorausplanung die Errichtung einer Verteidigungslinie (Anweisung: *defend*), was zum Zeitpunkt $t = 15$ am Zielpunkt begonnen werden soll.

Da Führungsentscheidungen bei Drehbuchsteuerungen für jede Entität explizit implementiert werden müssen, lässt sich aus diesem Beispiel bereits folgern, dass das Hinzukommen weiterer Entitäten für eine Erhöhung der Anzahl benötigter Zuweisungen von Verhaltensprozeduren sorgt. Besonders in der Abbildung komplexer taktischer Manöver, ggf. auch unter hochauflösender Abbildung von Großverbänden, sorgt dies für einen sprunghaften Aufwuchs benötigter Anweisungen. Wie dem Beispiel zu entnehmen ist, erlauben Drehbuchsteuerungen keine Einbindung von Ungewissheit, da alle Entscheidungen bereits im Vorhinein durch den Programmierer vorgenommen werden. Die einzige Möglichkeit besteht hier im simulieren einer auf Basis von Unschärfen getroffenen Führungsentscheidung, die anhand von Fallunterscheidungen eingebunden wird. Die ohnehin schon mikroskopische Zuweisung von Verhaltensprozeduren muss hierzu zusätzlich durch weitere **IF-THEN-ELSE**-Blöcke aufgeteilt werden. Vordefinierte Wahrscheinlichkeiten dienen hierbei als Auslöser der jeweiligen Fälle. Beispielhaft lässt sich dies durch eine Erweiterung des bisher betrachteten Beispiels verdeutlichen. Eine exemplarische Abweichung in der Befehlsgebung soll hierbei nun entweder einen Marsch oder einen Angriff zum Zeitpunkt $t = 0$ mit gleichbleibender Wahrscheinlichkeit $P(X) = 0.5$ vorsehen. Dies führt zur folgenden Änderung der prozeduralen Ablaufsteuerung:

Abweichende Befehlsgebung

```
⋮
if time == 0 then
    if (P(X)) then
        blue.move(tPos);
    else
        blue.engage(tPos);
    end if
    red.engage(tPos)
end if
⋮
```

Wie hieraus gefolgert werden kann, erweist sich diese Form der Abbildung von Unschärfen im Führungsprozess als sichtbar ineffizient. Unschärfen müssen nicht nur einerseits explizit per Fallunterscheidungen eingebracht werden, worunter die spätere Nachvollziehbarkeit aufgrund des entstehenden „Spaghetticodes“ sichtlich erschwert wird. Sie müssen andererseits auch für jeden definierten Zeitpunkt und für jede Entität im prozeduralen Programmcode hinterlegt werden, was bei wechselnden Situationen und Szenaren sowohl arbeits-, als auch zeitintensiv in der Pflege und Wartung des Führungsautomaten ist.

2.3.1.2 Regelbasierte Automaten

Rekapitulierend stellen Drehbuchsteuerungen einen kontextabhängigen, deterministischen Steuerungsmechanismus dar. Wie bereits anhand des zugehörigen Beispiels aufgezeigt wurde, resultiert dies jedoch in einem vergleichsweise hohen Aufwand zur Pflege und Wartung in Gefechtssimulationssystemen[21], weswegen eine Erweiterung des Mechanismus zur Erhöhung der Robustheit und Verringerung des Wartungsaufwands notwendig ist. Dies stellt den Ansatzpunkt der sog. **regelbasierte Automaten** dar. Grundintention ist es dabei, die zeitlich-motivierten Prämissen eines Drehbuches durch fallbasierte Prämissen zu ersetzen und damit eine zeitliche und prozedurale Unabhängigkeit des Steuerungsmechanismus vom Simulationsablauf selbst zu schaffen. Hierzu beinhaltet die Wissensbasis des Führungsautomaten Regelsätze, welche im Falle des Eintretens konkreter Situationen zu einer vordefinierten Folgerung seitens des Automaten führen. Klassischerweise werden als Entscheidungsgrundlagen dieser Regelsätze die Zustände der Simulationsobjekte und Informationen des eingebetteten Geländemodells verwendet. Die Entscheidung für ein bestimmtes Verhalten fällt dabei nach Überschreiten einer Prüfgrenze – auch Schwellwert genannt – und wird an der momentanen Lage der jeweilig zu steuernden Objekte gemessen.

Dies sorgt in Summe dafür, dass die Ablaufsteuerung per se nicht mehr zeitorient, dennoch aber von den eingestellten Schwellwerten des fallbasierten Schlusses abhängig ist. Regelbasierte Automaten sind aus

21 Vergleiche hierfür zusätzlich auch die Beispiele und Aussagen von Hofmann (2000a, 6).

diesem Grund anfällig für Inkonsistenzen, welche nur durch Feinjustierung der Schwellwerte und Konsistenzprüfung der Regeln auf ein Minimum zu reduzieren sind. Eine hierzu häufig getroffene Maßnahme stellt die Einbindung von Experten in der Generierung der jeweiligen Regeln dar, wodurch regelbasierte Ansätze zwar große Ähnlichkeit zu Expertensystemen besitzen, jedoch nicht den wissensbasierten Ansätzen zuzuordnen sind. Nichtsdestotrotz weisen prozedurale Wissensbasen häufig einen ähnlichen Aufbau wie regelbasierte Automaten auf. Hier ist zu betonen, dass diese Ähnlichkeit in weitesten Sinne dem identischen logischen Fundament zuzuschreiben ist. Sinngemäß kann daher auch, aufgrund des ähnlichen Aufbaus, eine Regelsteuerung als Synonym für eine prozedurale Wissensbasis verwendet werden. Dessen (vorläufig) ungeachtet diene die folgende Abbildung 2.4 fortfahrend als Anschauungsbeispiel regelbasierter Ansätze.

Programm 2 Regelbasierter Automat

```
procedure C2(blue, red: Entity)
    y1 = blue.getWpnRng();
    y2 = red.getWpnRng();
    x  = c2.getTPos(blue);
    for Integer: time = 1 to n do
        if blue.getPos() != x then                  ▷ R1
            blue.move(x);
        else
            blue.defend(x);
        end if
        if blue.getDistTo(red) ≤ y1 then            ▷ R2
            blue.engage(red);
        end if
        if red.getDistTo(blue) ≤ y2 then            ▷ R3
            red.engange(blue);
        end if
    end for
end procedure
```

Abbildung 2.4: Beispiel eines regelbasierten Automaten

Erläuterung des Beispiels In Anlehnung an Abb. 2.3 orientiert sich das Beispiel an derselben Ausgangslage, welche nun mit einer Regelsteuerung versehen wird. Der Beispielregelsatz umfasst dafür zwei Regeln (R_1, R_2) zur Steuerung des blauen Jägerzugs (blue) und eine Regel (R_3) zur Steuerung der roten Infanteriegruppe (red). Blau und Rot besitzen innerhalb dieses Beispiels identische Parameter: $WpnRng$, die Reichweite der eigenen Waffen. Als Abstraktion wird davon ausgegangen, dass diese Waffenreichweite – wie eingezeichnet – von Gelände und Bewuchs unbeeinflusst bleibt. Darüber hinaus wird davon ausgegangen, dass sowohl Blau als auch Rot permanent aufgeklärt sind und ihre Position eindeutig durch die andere Entität nachvollzogen werden kann.

Das Beispielprogramm skizziert eine diskrete Regelsteuerung, d. h. einen Steuerungsmechanismus, welcher zu jedem Zeitschritt eines fest definierten Zeitraumes im Intervall $t \in [1, n]$, in sequenzieller Reihenfolge die programmatisch implementierten Prämissen der einzelnen Regeln prüft. Die Regel R_1 dient hierbei zur Prüfung, ob sich der blaue Jägerzug bereits an der ermittelten Zielposition ($getPos = x = TPos$) befindet. Ist dies nicht der Fall, so soll sich Blau einen Schritt in Richtung des Zieles bewegen. Andernfalls hat Blau diese Stelle zu verteidigen. Befindet sich Rot ungeachtet der Position innerhalb der Waffenreichweite von Blau ($DistTo < WpnRng$), so ist folgernd der Regel R_2 der Feuerkampf aufzunehmen. Die Regel R_3 dient aus Sicht der roten Infanteriegruppe für den Umkehrfall der Regel R_2.

Aus dem Programm heraus wird ersichtlich, dass die im Vergleich zu einer reinen Drehbuchsteuerung gewonnene zeitliche Unabhängigkeit innerhalb dieses Ansatzes durch fallbasierte Handlungsanweisungen in diskreten Zeitschritten herbeigeführt wird. Die im Beispiel gewählte Reihenfolge der Regeln ist subjektiv festgelegt worden und offenbart damit eine große Schwachstelle innerhalb dieses Steuerungsmechanismus[22], da Reihenfolge und Zutreffen der Regeln zu strukturellen Varianzen führen können. Über diese Schwachstelle hinaus, fällen regelbasierte Automaten dichotome Entscheidungen auf Basis der eingebetteten Parameter. Wie exemplarisch aus der Visualisierung des Beispiels durch Nachmessen von der Zielposition aus festzustellen ist, beinhaltet der dargestellte Regelsatz eine nicht sofort ersichtliche strukturelle Varianz: Es kommt aufgrund der eingestellten Waffenreichweite trotz vorhandener Regeln zu keinem Gefecht zwischen Rot und Blau. Damit bedürfte der Regelsatz eine Feinjustierung, welche fernab der durch die in der Abbildung 2.4 zu Anschauungszwecken eingezeichneten Waffenwirkungskreise nur durch fallbasierte Validierung zu folgern gewesen wäre.

Da regelbasierte Automaten Entscheidungen primär auf Grundlage im Vorhinein spezifizierter Situationen ziehen, sind diese an sich nicht in der Lage, etwaige nicht oder nur unvollständig vorhandene Umweltinformationen zu verarbeiten[23]. Dies bringt regelbasierte Automaten in de facto dieselbe Ausgangsposition wie Drehbuchsteuerungen, denen zur Einbindung von Unschärfen lediglich die Simulation derer Auswirkungen verbleibt. Ähnlich dem bereits aufgeführten Ansatz zur Simulation unscharfer Entscheidungsfindung in Skripten, führt die Umsetzung durch Einführung fallbasierter Unterscheidungen in regelbasierten Ansätzen zu zusätzlichen Problemen. Als Anschauungsbeispiel diene hierfür der folgende, schon aus dem vorherigen Beispiel bekannte Mechanismus auf Grundlage von Wahrscheinlichkeiten. Eine Veränderung der Regel R_2 soll hierbei zusätzlich Feinderkennung mit gleichbleibender Aufklärungswahrscheinlichkeit $P(X) = 0.5$ indirekt abbilden:

22 Zweifellos existieren im Beispielfall maximal sechs mögliche Permutationen. Es sollte erwähnt werden, dass die für das Beispiel getroffene Festlegung in der Ablaufreihenfolge nur eine von vielen möglichen Umsetzungen regelbasierter Automaten adressiert. Andere Ansätze konzentrieren sich z. B. darauf, inkrementelle Zeitabfragen und Festlegungen der Reihenfolge entweder durch ereignisorientierung der Regeln, oder individueller Einbettung der Steuerung in selbstständig agierenden Agenten weitestgehend zu umgehen.

23 Grund hierfür ist die sog. Modularität deduktiver Systeme. Den nachfolgenden Kapiteln ist hierzu genaueres zu entnehmen.

Eingebettete Aufklärungswahrscheinlichkeiten

```
⋮
if blue.getDistTo(red) ≤ y1 then
    if flag then
        blue.engage(red);
    else if P(X) then
        flag = true;
        blue.engage(red);
    end if
end if
⋮
```

Parallel zu den bereits bekannten Problemen – schlechte Nachvollziehbarkeit, benötigter Arbeits- bzw. Zeitaufwand und Wartbarkeit – führt diese Einführung von Unschärfen zusätzlich zu einer Erschwernis der Konsistenzprüfung. So kann auf Basis der Verkettung von Wahrscheinlichkeiten argumentiert werden, dass es aufgrund einer nicht stattgefundenen Aufklärung zu keinem Feuergefecht kam. Dies ignoriert jedoch die tatsächliche Inkonsistenz der Regelsätze. Viel mehr noch wird hier die Frage nach der Eignung der gewählten Wahrscheinlichkeit aufgeworfen werden, was sich klar vom eigentlichen Problem des Regelsatzes entfernt. In der Praxis wird aus diesen Gründen innerhalb regelbasierter Automaten auf die Verwendung von durch Wahrscheinlichkeiten ausgelösten Fallunterscheidungen verzichtet, da diese nicht nur ineffizient sind, sondern auch im schlimmsten Fall ein erhöhtes Fehlerpotenzial in der Wissensrepräsentation verursachen.

2.3.2 Algorithmische Ansätze

Algorithmische Ansätze stellen Umsetzungen von Führungsautomaten dar, deren Einsatzbandbreite sich von der Suche nach der Lösung eines konkreten Entscheidungsproblems, bis hin zur Planung ganzer Handlungssequenzen zum Erreichen einer konkrete Zielstellung erstreckt. Taktische militärische Entscheidungen und Zielstellungen muten im ersten Moment recht trivial an, da diese allzu oft auf die Bekämpfung eines Feindes oder das Einnehmen und Halten einer spezifischen Zielposition verallgemeinert werden. Diese Verallgemeinerung stellt jedoch einen Trugschluss dar, da sich Zielstellungen und Entscheidungen während eines Gefechtes lagespezifisch verändern können und dem militärischen Führer sich somit häufig keine eindeutige Lösung präsentiert. Algorithmische Ansätze adressieren diesen Umstand durch Generierung heuristisch bestimmter Entscheidungen, die – obgleich diese im seltesten Fall optimal sind – zumindest eine mögliche Lösung eines Entscheidungsproblems und damit einen groben Schritt in Richtung der gewünschten Zielstellung darstellen. Sie bedingen hierfür einer vorab erfolgten Definition der Zielstellung auf Basis der übergeordneten Absicht, welche durch den Automaten entsprechend anhand fest vorgegebener Kriterien verarbeitet werden kann. Mithilfe von Lösungsalgorithmen wird dabei, mit jedem Verarbeitungsschritt der Algorithmen, der momentane Systemzustand der zu führenden Elemente mit dem in der Zielstellung vorgegebenen Zielzustand abgeglichen. Hierdurch ist der Führungsautomat in der Lage die Abweichung

von der definierten Zielstellung zu bestimmen und entsprechende Schritte zur Einnahme des gewünschten Zielzustands vorauszuplanen.

Formal können Entscheidungsprobleme für algorithmische Ansätze – frei gefolgert aus der Notation von Krumke und Noltemeier (2012, 7ff) – im Falle einer Einzelentscheidung über einen gerichteten Graphen $G_E := (V, E, \alpha, \omega)$ abgebildet werden, wobei V die Menge aller bekannter Zustände, E die Menge aller möglichen Zustandsübergänge und $\alpha, \omega : E \to V$ Abbildungsfunktionen für den Ausgangs- (α) bzw. Zielzustand (ω) eines Zustandsüberganges ist. Die Lösung des Entscheidungsproblems stellt hierbei den Übergang in einen der Lage angemessenen Zielzustand $v \in V$ dar. Sequenzen von Handlungen können in Erweiterung hierzu als Bäume $G_S := (V, E, \gamma)$ abgebildet werden, wobei V erneut die Menge aller bekannten Zustände, E die Menge aller möglichen Zustandsübergänge und $\gamma : E \to \{X : X \subset V \text{ mit } 1 < |X| < 2\}$ eine Abbildungsfunktion zur Zuordnung ein oder mehrerer Zustandsübergänge zu einem konkreten Zielzustand $\gamma(e) \in V$ ist. Eine Sequenz von Handlungsanweisungen zur Erfüllung einer Zielstellung ist dabei eine endliche Menge $M \subset V$, welche eine die Zielstellung erfüllende Abfolge von Zustandsübergängen beschreibt – zumeist realisiert mittels eines minimal spannenden Baumes bzw. Waldes[24].

Eine derartige Formalisierung ermöglicht in der Praxis den Einsatz bewährter Verfahren des Operations Research zur Lösung kombinatorischer Probleme, wie beispielsweise Branch and Bound, dem Minimax-Algorithmus, Rückwärtsinduktion etc. Auch metaheuristische Verfahren wie simuliertes Abkühlen, evolutionäre Algorithmen und neuronale Netze sind üblich. Zielfunktionen (oder Fitnessfunktionen respektive) stellen im konkreten Fall eines Führungsautomaten die Minimierung der eigenen Kosten (z. B. eigene Verluste) und die Maximierung des eigenen Nutzens (z. B. feindliche Verluste) zum Zeitpunkt der Abfrage dar. Es muss hierbei erwähnt werden, dass Nutzen und Kosten üblicherweise entweder über im Vorhinein festgelegte Funktionen, beispielsweise zur dynamischen Berechnung von Kantengewichten in den Entscheidungsgraphen, oder mittels Voraussimulation (vgl. Junge (2008)) abgebildet werden können. Algorithmische Ansätze ziehen damit immer bestmögliche Schlüsse, was im Allgemeinen auf Basis der Rationalität nachvollziehbar, dennoch in besonderen Situationen wie im Falle von Gefechtshandlungen selten der Realität entsprechen mag. Sie werden daher weniger für Steuerungsmechanismen der analytischen Simulation genutzt, können jedoch in Kombination mit Regelsteuerungen effektiv strukturelle Varianzen verringern – siehe (Tolk, 1995) für Näheres. Zur besseren Ersichtlichkeit algorithmischer Ansätze folgt nun, parallel zu den Beispielen der Regelsteuerungen, eine beispielhafte Implementierung. Die nachstehende Abb. 2.5 diene hierfür als Anschauungsbeispiel algorithmischer Ansätze.

24 Für Bäume und Wälder vergleiche die hierfür gebräuchliche Notation von Krumke und Noltemeier (2012, 104).

Programm 3 Algorithmische Entscheidungsfunktion

```
Array of State: seq;                          ▷ globale Befehlssequenz
function C2(blue: Entity, order: Order) returns State
    State : initial, goal, dec;
    initial ← blue.getState();
    if seq = ∅ then
        goal← blue.formGoal(order.intent, order.pos);
        computeGoal(initial, goal);
        seq ← search(goal.stategraph) ;
        if seq.first() == null then
            return initial;
        end if
    end if
    dec ← seq.pop();
    return dec;
end function
```

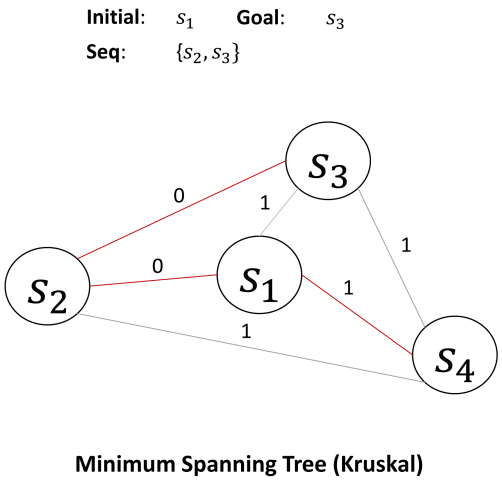


Abbildung 2.5: Beispiel eines algorithmischen Automaten

Erläuterung des Beispiels Entgegen der bisher in Abb. 2.3 und 2.4 dargestellten Automaten, stellt der abgebildete Ansatz Führungsentscheidungen nicht explizit anhand prozeduraler Entscheidungsregeln dar. Um die Wirkungsweise des Automaten dennoch anschaulich verdeutlichen zu können, ist es notwendig tiefergehend die exemplarische Funktionsweise darzustellen. Zur Verringerung der Komplexität wird dabei im Folgenden nur der blaue Jägerzug betrachtet werden. Ausgangspunkt des exemplarischen Automaten stellt die Festlegung auf vier mögliche Systemzustände einer endlichen Zustandsmenge $S := \{s_1, s_2, s_3, s_4\}$ dar. Der Zustand s_1 beschreibt hierbei den Zustand „**Wartend**“, s_2 den Zustand „**Bewegung**“, s_3 den Zustand „**Verteidigung**“und s_4 den Zustand „**Angriff**“. Für die interne Verarbeitung ist ferner ein Zustandsdatentyp (*State*) als Tripel $\mathfrak{Z} = (Z, C, G)$ definiert. Z ist dabei ein momentaner bzw. anzustrebender Systemzustand mit $Z \subseteq S$, C eine Rasterkoordinate an welchen sich der Jägerzug befindet bzw. befinden soll und G ein vollvermaschter ungerichter Graph, welcher zur Ermittlung der nötigen Zustandsübergänge dient. Der algorithmische Automat arbeite darüber hinaus mit einer festen Befehlsfolge (*Order*), welche aus einer String-Referenz der übergeordneten Absicht in Form fest vorgeschriebener Schlüsselworte und ggf. einer Zielkoordinate besteht.

Es wird nun im weiteren Verlauf davon ausgegangen, dass die übergeordnete Absicht - wie auch in Abb. 2.5 rechts dargestellt- zu Beginn der Simulation in den Automaten eingespeist wird. Der Automat dient damit zur Abbildung eines zielorientierten Führungsmechanismus. Standardmäßig bauen sich derartige Mechanismen auf drei konsekutiven Phasen auf: Der **Zielformulierung** (*formGoal*), der **Zielverarbeitung** (*computeGoal*) und der **Planung** (*search*). Die einleitende Zielformulierung umfasst dabei im Allgemeinen die Entschlüsselung der übergeordneten Absicht. Für den hier aufgezeigten Beispielfall interpretiert diese die eingespeiste Befehlsfolge und ermittelt den gewünschten Zielzustand in Abhängigkeit der eigenen, möglichen Systemzustände. Konkret kann dies dem nachfolgenden Programm entnommen werden.

Zielformulierung

```
function FORMGOAL(intent: String, pos: Coord) returns State
    State: goal;
    case intent of
        move  : goal.setState(s_2);
        defend: goal.setState(s_3);
        engage: goal.setState(s_4);
        others:
        goal.setState(s_1);
    end case
    goal.setPos(pos);
    return goal;
end function
```

Nachdem durch die Zielformulierung der gewünschte Zielzustand ermittelt wurde, muss dieser im Folgeschritt zu einer Zielstellung verarbeitet werden. Wie bereits erläutert wurde, lassen sich die dazu möglichen Zustandsräume und Zustandsübergänge anhand eines Graphen beschreiben. Der für diesen Beispielfall gewählte vollvermaschte Graph $G = (V, E, \gamma)$ besitzt vier Knoten der Menge $V \subseteq S$, welche zusätzlich zu den Kanten $e_{i,j} \in E$ des Graphen jeweils ein zugehöriges Kantengewicht $\omega_{i,j} \in [0, 1]$ besitzen. Das Kantengewicht entspreche hierbei der Zuordnung von Kosten im Sinne der Zielerreichung, wobei das System bestrebt ist, zum Erreichen der Zielstellung die entstehenden Kosten zu minimieren. Zur vereinfachten Erläuterung dieses Ansatzes gelte dabei, dass immer nur dann Kosten entstehen, wenn die Transition in einen Nicht-Zielzustand nicht unmittelbar mit der Auftragserfüllung in Relation steht. Dies bedeutet beispielsweise, dass zum Angriff auf die Position X eine Verteidigung der Position X nicht zweckdienlich ist und ein direkter Zustandswechsel in die Verteidigung die Kosten erhöhen würde. Darüber hinaus ist auch eine direkte Transition in den Angriff auf die Position X nicht zweckdienlich, sofern diese noch nicht hinreichend erschlossen wurde. Der Automat berechnet daher innerhalb der Zielverarbeitung entsprechende Kantengewichte auf Basis des aktuellen Zustandsgraphen G, welche in Abhängigkeit der Zielstellung in der Planung als Kosten interpretiert werden können.

Weicht der intendierte Zielzustand dabei vom Ausgangszustand ab, so wird durch den Algorithmus ein Pfad innerhalb des Graphen zum Zielzustand x generiert. In einem ersten Schritt werden hierfür alle direkten Kanten zum Zielzustand mit dem Kantengewicht $\omega_{x,j} = 0$ versehen. Befindet sich die Entität zum Zeitpunkt der Berechnung nicht an der Zielposition, so wird die direkte Kante mit dem Kantengewicht $\omega_{x,i} = 1$ versehen, da diese Transition nicht zielführend ist. Da sich folglich die Entität erst zur Zielposition bewegen muss, muss ein Pfad von Ausgangs- und Zielzustand zum Zustand s_2 generiert werden. Zu diesem Zweck werden alle Kanten von Ausgangs- und Zielzustand zum Zustand s_2 mit dem Kantengewicht $\omega_{2,i} = 0$ versehen. Die nachfolgende Prozedur diene zur Verdeutlichung dieses Vorgehens in der Zielverarbeitung.

Zielverarbeitung

```
procedure COMPUTEGOAL(current, goal: State)
    Graph graph = current.getStateGraph();
    Integer: x = goal.getState().getNum();        ▷ Index des Zielzustandes
    if (¬equals(current.getState(), goal.getState())) then
        for Integer i = 1 to x do
            graph.edgeWeight(x,i) = 0;            ▷ Kante zum Zielzustand
        end for
    end if
    if (¬equals(current.getPos(), goal.getPos())) then
        for Integer: i = 1 to x do
            graph.edgeWeight(i,x) = 1;            ▷ Kante zum Zielzustand
            graph.edgeWeight(2,i) = 0;            ▷ Kante zum Zustand S_2
        end for
    end if
    goal.setGraph(graph);
end procedure
```

Das Ergebnis der Ausführung der dargestellten Prozedur stellt einen aktualisierten Zustandsgraphen dar, dessen Kantengewichte entsprechend der prozedural implementierten Maßgaben zur Zielerreichung modifiziert wurden. Zur Bestimmung der benötigten Zustandssequenz wird nun als erster Schritt zur Verhaltensplanung der Algorithmus von Kruskal[25] ausgeführt, welcher einen minimalen Spannbaum des Graphen ermittelt. Das in Abbildung 2.5 rechts dargestellte Ergebnis dieses Algorithmus muss jedoch folgend noch – ausgehend vom Initialzustand – nach dem kürzesten direkten Pfad zum Zielzustand durchsucht werden. Ergebnis dieser Suche ist anschließend eine Zustandssequenz, welche elementweise als Handlungsanweisung an den blauen Jägerzug übergeben werden kann.

Die aus dem Beispiel folgende Zustandssequenz ist dabei eindeutig. Mit steigender Anzahl an Zuständen und/oder nicht-dichotomen Kantengewichten kann die demonstrierte Umsetzung jedoch dazu führen, dass konkurrierende – d. h. kostentechnisch gleichwertige, aber dennoch in der Zustandssequenz abweichende – Pfade innerhalb der Zustandsgraphen entstehen. Die zwangsläufig benötigte Auswahl eines Pfades kann hierbei lagebedingt, wenn Sie z. B. auf Basis zufälligen Entscheidens umgesetzt wurde, ungünstigen Zustandssequenzen hervorrufen, die im schlimmsten Fall in strukturellen Varianzen resultieren. Daher ist es geläufige Praxis durch eine prozedural verankerte Zustandspriorisierung die Möglichkeit zur Abweichung innerhalb der Zustandssequenzen a priori zu eliminieren. Da algorithmische Ansätze stark vom extern eingespeisten Systemzustand abhängig sind, sind diese nicht selbst in der Lage unscharfe Lagebilder zu verarbeiten. Ähnlich wie Regelsteuerungen ist es zwar auch hier möglich die Auswirkungen eines unscharfen Lagebildes zu simulieren. Jedoch erweist sich dies in diesem Fall auch als nicht zielführend, da bspw. eine etwaige Umsetzung auf Basis durch Wahrscheinlichkeiten ausgelöster Fälle zu zusätzlichen fehlerhaften Sequenzen respektive den bereits erwähnten strukturellen Varianzen führen kann.

25 Näheres zum Algorithmus ist den Darstellungen von Krumke und Noltemeier (2012, 104) zu entnehmen.

2.3.3 Wissensbasierte Ansätze

Wie der Name bereits suggeriert, stellen **wissensbasierte Ansätze** Steuerungsmechanismen auf Basis deklarativer und prozeduraler Wissensbasen dar. Fundament wissensbasierter Führungsautomaten ist ein dem System präsentes kontextadäquates Wissen, welches in Form von axiomatischen Aussagen den Zustand der Umwelt beurteilt oder anhand prozeduraler Entscheidungsregeln konkrete Handlungsanweisungen folgert. Ausschlaggebend für die Umsetzung ist dabei die Art und Weise der Wissensrepräsentation und des verwendeten logischen Kalküls. Hierbei ist es für das repräsentierte Wissen maßgeblich, auf welchem Träger dieses definiert ist. Hier wird in der Praxis darunter unterschieden, ob der Träger eingebettet in das oder modular an das zu steuernde System angefügt wurde. Eingebettete Wissensbasen sind in diesem Sinne häufiger anstelle programmatischer Ablaufsteuerungen, bspw. innerhalb eines Agentenprogramms wiederzufinden. Sie sind bewusst relativ einfach gehalten, da die Entstehung sog. emergenten Verhaltens in Kauf genommen wird. Modulare Wissensbasen sind im Gegenzug zumeist ungleich komplexer, da diese extern an ein bestehendes System angebunden und deswegen nicht-systemspezifisch konzipiert werden. Die Kehrseite hieraus stellt eine erhöhte Flexibilität hinsichtlich der Einsatzmöglichkeiten dar - ein Vorteil, den es gegenüber der Komplexität wohl abzuwägen gilt. Insbesondere ist hierbei darauf zu achten, dass eine zu hohe Komplexität bei der Prüfung der Nachvollziehbarkeit getroffener Schlüsse zu Problemen führen kann.

Sowohl die entstehende Komplexität, als auch der Zwang das eingebettete Wissen nachvollziehbar abzulegen stellt auch das größte Manko wissensbasierte Ansätze dar. Die Nachvollziehbarkeit wird dabei in der Praxis vorrangig durch das Urteil eines unabhängigen Domänenexperten bewertet, was sich als zweischneidiges Schwert erweist. Einerseits mag die Nachvollziehbarkeit eine für die Plausibilitätsprüfung entscheidende Instanz sein. Im Entwurf wissensbasierter Steuerungsmechanismen wird dies andererseits aber häufig – aufgrund der hintergründigen Komplexität – bewusst vernachlässigt, da logische Wissensbasen für Domänenexperten meist zu obskur sind. Das Resultat stellen zeitaufwendige nachträgliche Kalibrierungen und Plausbilitätsprüfungen auf Grundlage der sichtbaren Auswirkungen dar, welche in der Regel nur eine symptomatische Verbesserung herbeiführen. Die zielgerichtete Einbindung von Domänenexperten in der Entwicklung des Steuerungsmechanismus ist daher anzustreben, da somit deren vorherrschende Expertise kontextadäquat in den zugrundeliegenden Entscheidungsmechanismen implementiert wird. Dies bedingt einerseits Aussagen und Absicht eines oder mehrerer Experten in eine Wissensbasis derart einzutragen, dass diese im Anschluss innerhalb einer geschlossenen Simulation zur Steuerung von Simulationsobjekten verwendet werden kann. Andererseits ist das implementierte Wissen derart greifbar für den Experten zu formulieren, dass dieser eine nachträgliche Prüfung ohne konkretes Verständnis mathematischer Logik durchzuführen in der Lage ist. Als mögliche Umsetzungen erwiesen sich hierfür in der Praxis logisch programmierte Wissensbasen (z. B. mit Prolog oder Lisp) oder deduktive Regelsysteme in Form hybrider wissensbasierter Regelsteuerungen[26], deren Wissen mit semantischen Ausdrücken hinterlegt wurde. Im Kontext der Gefechtssimulation sind es eben diese Umsetzungen, welche weitläufig als **Expertensysteme** anerkannt sind.

In diesem Zusammenhang ist es dennoch erwähnenswert, dass sämtliche agentenbasierten Gefechtssi-

26 Entschlussregeln für ein wissensbasiertes System werden in diesem Kontext auch als Regelbasis bezeichnet, ein Hinweis auf deren Verwandschaft mit Regelsteuerungen.

mulationssysteme grundsätzlich auf wissensbasierten Ansätze fußen, obgleich diese sich in der konkreten Umsetzung des Agentenprogramms von klassischen Expertensystemen unterscheiden[27]. Hybride Wissensbasen stellen nach wie vor in der Gefechtssimulation eine Randerscheinung dar, obgleich diese zu rein prozeduralen Wissensbasen – wie sie ähnlich in Regelsteuerungen zu finden sind – entscheidende Vorteile besitzen. Die Fähigkeit zur Einbindung von Lernalgorithmen[28] und die durch Abwandlung des verwendeten logischen Kalküls zu gewinnende Möglichkeit zur Verarbeitung unscharfen Wissens[29] sind hierbei besonders hervorzuheben. Dieser Vorteile zum Trotz stellen wissensbasierte Ansätze außerhalb der reinen Forschung eine Randerscheinung dar. Grund hierfür sind einerseits die angesprochenen Probleme in der Einbindung von Experten in der Generierung der benötigten Wissensbasen. Andererseits zeigten sich Expertensysteme in Vergangenheit als äußerst Laufzeitineffizient, da insbesondere die hauptsächlich angestrebte externe Ankopplung an bereits bestehende prozedurale Komponenten eines Simulationssystems aufgrund der vorzunehmenden Interpretation mitunter Schwierigkeiten erzeugte (Dompke, 1992, 72). Moderne Ansätze der Expertensysteme unterliegen jedoch nicht mehr den in Vergangenheit vorherrschenden Hardware-Limitierungen, wodurch eine Laufzeitineffizienz heutzutage praktisch nicht mehr anzutreffen ist. Zur allgemeinen Betrachtung wissensbasierter Automaten sei abschließend auf nachfolgendes Beispiel in Abb.2.6 zum besseren Verständnis der Funktionsweise verwiesen.

Programm 4 Hybride wissensbasierte Steuerung

```
Database : KB;
Array of State: seq;
function C2(blue: Entity) returns Action
    State: current;
    Action: next;
    current ← KB.queryPerc(blue.getCPerception());
    if seq == ∅ then
        if current == „enemy in sight“ then
            seq ← planAttack(blue, „engage“);
        else if current == „under fire“ then
            seq ← planAttack(blue, „fire superiority“);
        else if current == „pinned down“ then
            seq ← planRoute(blue, „avoid“);
            ⋮
        end if
    end if
    next ← KB.queryAct(seq.pop());
    return next
end function
```

Abbildung 2.6: Beispiel eines hybriden wissensbasierten Automaten

27 Die in den Systemen verwendeten wissensbasierten Agenten können Paradigmen der Expertensysteme umfassen. Häufig erweitern diese jedoch nur die Mächtigkeit des Systems durch die mögliche Entstehung emergenter Strukturen. Näheres zu wissensbasierten Agenten und die Herbeiführung emergenter Strukturen kann grundsätzlich den Ausführungen von Ferber (1999) entnommen werden.

28 Innerhalb dieser Arbeit erfolgt keine nähere Betrachtung von Lernalgorithmen für wissensbasierte Ansätze.

29 Siehe das folgende Kapitel 3 für Näheres.

Erläuterung des Beispiels Der dargestellte Steuerungsmechanismus wird erneut aus dem Kontext der Lage aus Abbildung 2.3 heraus betrachtet. Kern des dargestellten wissensbasierten Automaten stellt die namensgebende Wissensbasis (hier eine Datenbank undefinierten Inhalts, Deklaration: **KB**) dar, welche global über der eigentlichen Steuerung initialisiert wird. Aus Gründen der leichteren Verständlichkeit wird erneut darauf verzichtet, parallel zum Beispiel der algorithmischen Ansätze, beide Steuerungsmechanismen der Konfliktparteien zu beschreiben. Darüber hinaus soll es aus Gründen der einfacheren Verständlichkeit hierbei genügen, dass die Wissensbasis, weil hybrid, sowohl aus einem nicht näher erläuterten deklarativen und prozeduralen Anteil besteht. Wie bereits im Verlauf des Kapitels 2.2 erläutert wurde, dient der deklarative Anteil der Wissensbasis dabei zur Auswertung des aktuellen Umweltzustands, während der prozedurale Anteil konkrete Handlungsanweisungen an den blauen Jägerzug gemäß im Vorhinein definierter Verhaltensregeln umfasst.

Intention des wissensbasierten Automaten stellt es in diesem Sinne dar, dass mithilfe der momentanen Wahrnehmung (**CPerception**) auf Basis von Abfragen (engl. queries) über den deklarativen Anteil der Wissensbasis sowohl die momentane Lage (**current**), als auch eine für die Lage adäquate Aktion (**next**) ermittelt wird. Für letzteres plant der Automat Zustandsketten (**seq**), die im Grunde in gleicher Form wie die Zustandssequenzen des in Abbildung 2.5 dargestellten algorithmischen Ansatzes produziert werden. Wie sich hieraus bereits folgern lässt, können wissensbasierte Ansätze daher durchaus auch die Funktionalitäten anderer Ansätze umfassen. Sie erweitern diese Ansätze jedoch durch eine eigenständige Erfassung der Lage auf Basis der Wahrnehmung einer Entität und generieren hieraus Folgerungen (Anweisungsdeklaration: **queryPerc()**), die wiederum auch als Ausgangspunkt für andere Ansätze dienen können. Im Beispielfall verarbeitet der Automat die Auswertung der eigenen Lage, indem er diese fallbasiert mit einer Zielstellung (z. B. *engage*, *avoid*, etc.) verknüpft. Auf dieser Zielstellung aufbauend wird in Folge eine Zustandskette zur Erreichung der konkreten Zielstellung (*planAttack*, *planRoute*, etc.) erarbeitet. Das Wissen um die konkrete Umsetzung der einzelnen Zustände der resultierenden Zustandsketten ist wiederum Teil der prozeduralen Wissensbasis, welche anschließend schrittweise die jeweilig auszuführende Einzelaktionen per regelbasierterr Abfrage (*queryAct*) auf Grundlage des ersten Folgezustands in der Kette (*seq.pop*) ermittelt.

Wie aus diesem Beispiel hervorgeht, besitzen wissensbasierte Ansätze ein herausragendes Alleinstellungsmerkmal: Sie ziehen vor Bestimmung konkreter Aktionen Rückschlüsse aus dem Umweltzustand. Im Gegensatz zu Regelsteuerungen und algorithmischen Ansätzen befähigt dieser Umstand den Automaten dazu, den Umfang der Umweltinformationen und den Mechanismus zur Verarbeitung dieser Informationen festzulegen. Dies erlaubt es auch unvollständiges Wissen und Unschärfen im Lagebild zu adressieren, obgleich die tatsächliche Umsetzung hierfür aufgrund der oberflächlichen Betrachtung innerhalb dieses Beispiels nicht adressiert werden kann. In späteren Kapiteln dieser Arbeit wird dieser Umstand jedoch tiefergehend untersucht.

2.3.4 Kognitive Ansätze

Kognitive Ansätze bezeichnen die Abbildung menschenähnlicher Entscheidungsprozesse in Verhaltens- und Führungsautomaten. Menschenähnlich umschreibt in diesem Kontext einen von der perfekten Rationalität

abweichenden, bis hin durch subjektive Faktoren beeinflussten Steuerungsmechanismus. Dies adressiert in gewisser Hinsicht eine „natürlichere“ Entscheidungsfindung, da diese naturgemäß von Individuum zu Individuum durch deren Erfahrungsschatz und Persönlichkeit beeinflusst wird. Häufig werden kognitive Ansätze daher mit dem **Natural Decision Making** (kurz: NDM) gleichgesetzt:

> *„The study of NDM asks how experienced people, working as individuals or groups in dynamic, uncertain, and often fast-paced environments, identify and assess their situation, make decisions and take actions whose consequences are meaningful to them and to the larger organization in which they operate.“* (Zsambok und Klein, 2014, 5)

Besondere Aufmerksamkeit wird hierbei der Repräsentation menschlichem Verhaltens[30] zuteil. Bestimmendes Merkmal ist in diesem Zusammenhang u. a. die Einbindung von Humanfaktoren (engl. human factors), welche zu typisierten Veränderungen innerhalb der Entscheidungsfindung führen. Humanfaktoren umfassen nur schwer zu quantifizierende Faktoren, wie den Erfahrungsstand einer Person, deren Stresspegel, Müdigkeit, etc. Diese mögen dabei zwar zweifelsohne eine Auswirkung auf den Entscheidungsprozess haben, lassen sich häufig jedoch nur anhand geschätzter quantitativer Werte repräsentieren und finden daher lediglich auf Basis pauschalisierter Experimente Einbindung[31]. Da klassische – regel- oder wissensbasierte – Ansätze bereits eine gewissermaßen menschennahe Schlussbildung abbilden, erweitern kognitive Ansätze häufig nur die implementierte Verhaltenslogik um Meta-Regeln oder natürlich inspirierte Verarbeitungsalgorithmen. Diese werden dem klassischen Schluss entweder vorangestellt oder beeinflussen die tatsächliche Durchführung einer Aktion, bspw. durch Einbringen von stochastischen Abweichungen oder einer Mustererkennung. Insbesondere sind kognitive Ansätze innerhalb der Umsetzung von Individualverhalten und damit vorrangig in der agentenbasierten Modellierung vorzufinden.

Ein beherrschender Punkt zur Kritik kognitiver Ansätze stellt deren empirische Glaubwürdigkeit in der Abbildung menschenähnlicher Entscheidungsprozesse, vorrangig durch den Einsatz von Humanfaktoren dar. So führen bspw. Ranyard et al. (1997, 51) in Bezug auf durch individuelle Faktoren beeinflusste Abbildungen von Entscheidungsprozessen auf: „*The ideas* [...] *concerning possible effects of value-, impression- and outcome-relevant involvement on decision making processes are tentative, and need to be empirically tested*.“ Bestätigt wird diese These durch Ergebnisse aus Studien der Bundeswehr [32], wo die Einbindung von Humanfaktoren zu zweifelhaften und oftmals obskuren Verhaltensweisen innerhalb der Simulationsläufe führte. Insbesondere zeigte sich hier, dass eine Vielzahl von Faktoren – wie beispielsweise Trainingsstand, Angst und Stresspegel – sich gegenseitig stark beeinflussen, wobei bereits die Änderung einzelner Faktoren ganze Ketten von Nachprüfungen mit sich zogen. Zusätzlich litt die Plausibilität des Verhaltensmodels, da keine empirisch haltbare Datenbasis für das Zusammenwirken der Faktoren im Entscheidungsprozess gefunden werden konnte. In Konklusion waren sich daher militärische Experten einig, dass Humanfaktoren zwar durchaus im Geiste der Realitätsnähe Einbindung finden sollten, jedoch die insbesondere schwierige empirische Nachweisführung, hohe Abhängigkeiten zwischen den Faktoren und die in Summe dieser Erscheinungen

30 *„Human behaviour (B) is a purposive reaction of a human being (P) to an idiosyncratic meaningful situation (S).“*(Dompke, 2000, 189)
31 Vgl. hierzu bspw. Lundin (2004); Driskell und Salas (1991), oder die Folgerungen von Gillis et al. (2000).
32 Vgl. hierzu u. a. die methodisch-fachliche Stellungnahme des Zentrums für Transformation der Bundeswehr im Abschlussbericht der Studie „Anpassung und Validierung PAX, Teil II“, SKZ CE 21P 9 232 D, VS-NfD.

problematischen Plausibilitätsüberprüfungen sich negativ auf die resultierenden Steuerungsmechanismen auswirken.

Kognitive Ansätze verloren aufgrund dieser Tatsache Anfang/Mitte der 2000er Jahre zusehends das Vertrauen der militärischen Bedarfsträger und besitzen zum momentanen Zeitpunkt innerhalb der Gefechtssimulation keine praktische Relevanz. Die folgende durch Hofmann (2000a, 20) getroffene Feststellung besitzt daher auch heute noch Gültigkeit:

> „***Insgesamt ist festzustellen, daß alle dem Verfasser bekannten größeren Gefechtssimulationsmodelle bislang nahezu ausschließlich auf Regelsteuerungen in Verbindung mit algorithmischen Ansätzen beruhen.*** *Demgegenüber stehen auf dem mehr forschungsmäßig orientierten Konferenzen und wissenschaftlichen Symposien naturgemäß mehr die wissensbasierten und/oder kognitiven Ansätze im Vordergrund, wobei jedoch - wenn überhaupt - meist nur kleinere Beispiele bzw. Anwendungen präsentiert werden.*“
>
> [Hervorhebungen aus dem Original übernommen]

Diesem Schattendasein zum Trotz, diene das nachfolgende Beispiel in Abb. 2.7 dennoch zur Verdeutlichung, dass kognitive Ansätze nach wie vor zur Automatisierung von Führungsprozessen in Simulationen geeignet sind.

Programm 5 Beispielhafter kognitiver Automat

```
procedure C2
    Element elem;
    Rule[] rhs;
    Heap work_mem;
    tGraph prod_mem;
    while true do
        elem ← buildElem(blue.getCState());
        for all Object o ∈ elem do
            work_mem.push(o);
        end for
        while ¬(working_mem.isEmpty()) do
            rhs ← prod_mem.match(work_mem.pop());
        end while
        rhconflictRes(elem);
        if rhhasFirst() then
            act(rhs[0]);
        end if
        sleep();
    end while
end procedure
```

```
(P Rule1
(Sensor ↑Position Target  ↑Sight <X>)
- ->(call defend() )

(P Rule2
(Sensor ↑Position Target  ↑Sight Enemy)
(Effector ↑Reach In)
- ->(call openFire() )

(P Rule3
(Sensor ↑Position ¬Target  ↑Sight <X>)
- ->(call move() )

(P Rule4
(Sensor ↑Position ¬Target  ↑Sight Enemy)
(Effector ↑Reach In)
- ->(call engage() )
```

Abbildung 2.7: Beispiel eines kognitiven Automaten mit vier Produktionsregeln

Erläuterung des Beispiels Zur Verdeutlichung kognitiver Ansätze zeigt das Beispiel ein sog. Produktionssystem. Den beiden vorangegangenen Beispielen folgend wird hierfür ebenfalls die Lage aus Abbildung 2.3 als Grundlage dienen und lediglich die Steuerung des blauen Jägerzugs näher erläutert werden. Der abgebildete Steuerungsmechanismus basiert hierbei auf der grundsätzlichen Annahme, dass – wie es bei Produktionssystemen üblich ist – sich die hintergründigen menschlich-kognitiven Prozesse in der Entscheidungsfindung als Menge von Bedingung-Aktion-Paaren darstellen lassen (Anderson, 1983, 5). Bedingungen spezifizieren dabei in diesem Sinne Datenmuster (engl. patterns), die, sofern wahrgenommene Elemente diesen Mustern entsprechen, zur Anwendung von Aktionen (sog. Produktionen) führen. Es sei im Kontext der Führungsautomaten hierzu jedoch festgestellt, dass Produktionen generell Zielstellungen oder Verhaltensanweisungen beschreiben.

Produktionssysteme stellen eine Sonderform regelbasierter Mechanismen dar. Aufgrund dessen sind dem Beispiel zur Verdeutlichung dieses Mechanismus vier Produktionsregeln beigefügt, die in der hierfür üblichen OPS5-Notation (vgl. (Forgy, 1981)) verfasst wurden. Zur besseren Verständlichkeit werden diese nun nachfolgend kurz erläutert:

- Im Falle der eingenommenen Zielposition (↑Position Target) soll, gleichgültig dessen ob ein Feind wahrgenommen wurde oder nicht (↑Sight <X>), die eigene Position verteidigt werden (**Rule1**).
- Bei eingenommener Zielposition und aufgeklärtem Feind (↑Sight Enemy) in Waffenreichweite (↑Reach In), soll das Feuer eröffnet werden (**Rule2**).
- Befindet sich der Jägerzug nicht in Zielposition (↑Position ¬Target), so soll dieser ungeachtet der Feindlage zum Ziel marschieren (**Rule3**).
- Wird jedoch in derselben Situation ein Feind in Waffenreichweite aufgeklärt, so soll dieser bekämpft werden (**Rule4**).

Produktionssysteme speichern Prouktionsregeln in einem persistenten Regelspeicher (engl. production memory, hier: *prod_mem*), der sich auch in Form eines Entscheidungsbaumes darstellen lässt – vgl. hierfür Abb. 2.8. Zu verarbeitende Daten werden in Form von aneinandergeketteten Objekten des Systemzustands als Zustandselemente (*element*, hier aus dem momentanen Systemzustand *CState* der Entität generiert) von der Wurzel ausgehend in den Entscheidungsbaum eingesickert.

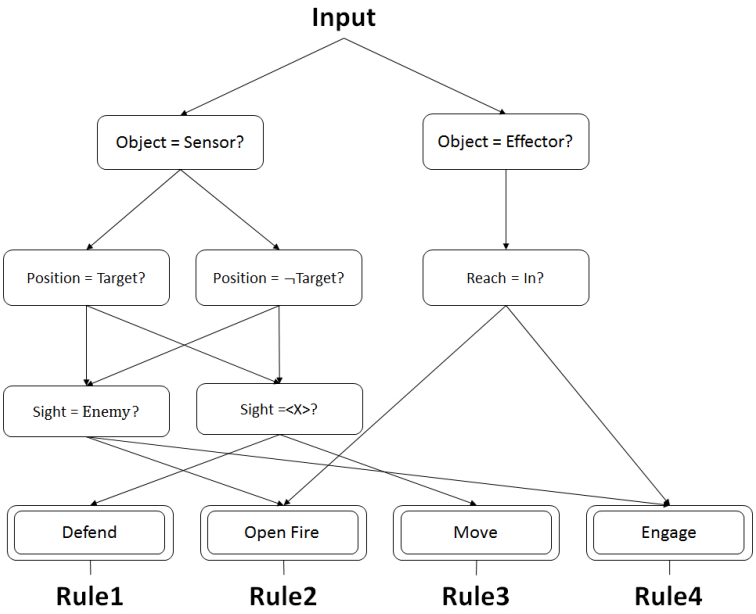


Abbildung 2.8: Produktionsregeln als Entscheidungsbaum

Einzusickernde Daten werden hierzu in einen temporären Arbeitsspeicher (working memory, hier: *work_mem*) geschrieben, auf dessen Basis ein Abgleich des Systemzustands mit den Mustern des Regelspeichers (*match*) durchgeführt wird. Auslösende – d. h. dem individuellen Regelmuster entsprechende – Konsequenzen führen dazu, dass die Regel im Folgeschritt weiter zu betrachten ist. Hierfür werden im Beispielfall entsprechende Regeln in ein Array (*rhs*) geschrieben. Da es in Folge des Abgleichs zu konkurrierenden, d. h. zeitgleich erfüllten Mustern kommen kann, müssen in einem Folgeschritt noch potenzielle Regelkonflikte aufgelöst werden (*conflictRes*). Dies reduziert den Inhalt des Arrays standardmäßig auf eine Regel[33], deren Konsequenz dann als Verhaltensanweisung ausgeführt (*act*) werden kann. Der Automat begibt sich abschließend in den Ruhezustand (sleep), bis dieser durch einen externen Auslöser – bspw. eine Lageänderung – eine neue Iteration der Programmschleife durchführt.

Wie sich aus diesem Beispiel schon erahnen lässt, erben kognitive Ansätze sowohl Stärken, als auch Schwächen derjenigen Ansätze, die sie erweitern. Da das in diesem Beispiel dargestellte Produktionssystem ebenfalls im Kern einen regelbasierten Ansatz darstellt, besitzt dieses auch – trotz der besonderen Form der Sensor-Datenverarbeitung als Muster – die in Kap. 2.3.1.2 aufgeführten Schwächen. Speziell die dort aufgeführte Problematik, dass die Einbindung von Unschärfen zu einer Erschwernis der Konsistenzprüfung führt, gilt auch in diesem Fall. Trotzdem ermöglicht die vorläufige Verarbeitung der Sensor-Inputs die Möglichkeit zur Einbindung unscharfer Lageaquisition. Dass diese in deduktiven Schlussmechanismen jedoch grundsätzlich kritisch zu bewerten ist, wird nun im nachfolgenden Kapitel diskutiert.

33 „*The conflict resolution pricinples in the OPS system* [...] *involve a combination of refractoriness, recency, specificy, and production ordering. Refractoriness prevents a production from repeating if it matches the same data structure, thus preventing most accidental looping. Recency* [...] *selects the production that matches the data element most recently added to the working memory.* [...] *If there still are competing productions (and this is very seldom the case), then the conflict is resolved by ordering the productions.*“ (Anderson, 1983, 15)

3 Unschärfen und Unsicherheiten

Wie im vorhergehenden Kapitel dargestellt wurde, entscheiden Führungsautomaten auf Basis der Verarbeitung der ihnen präsenten Umweltinformationen, welche entweder in Form konkreter wahrnehmbarer Zustände oder durch externe Einsteuerung seitens des Simulationssystems vorliegen. Insbesondere Regelsteuerungen, die den Hauptanteil marktverfügbarer Umsetzungen darstellen, ignorieren hierbei jene Umweltinformationen, welche nicht explizit an diese weitergetragen wurden. Weitergehende Schlüsse und Ableitungen sind dem Automaten hieraus somit nicht möglich, da die jeweiligen Steuerungsmechanismen lediglich über fallbasierte Abfragen ein eingeschränktes aktuelles Lagebild reflektieren. Im Rahmen taktischer Entscheidungen ist es jedoch durchaus entscheidend, die Auswirkungen des eigenen Handelns zu bemessen respektive die konkret eingetretene Lage einzuschätzen und hierauf aufbauend passende Folgeschritte abzuleiten. Da fallbasierte Auswertungen reaktiv in einer vorherrschenden Lage agieren, sind diese nativ nicht befähigt diagnostische oder prädiktive Schlüsse zuzulassen. Wissensbasierte Ansätze wiederum erlauben es – wie bereits dargestellt wurde – diesem Umstand durch Änderung in der Verarbeitung der Umweltinformationen entgegenzuwirken und mit Hilfe deklarativer Wissensbasen die gegenwärtige Umwelt selbstständig zu bemessen.

Die bisher vorgestellten Wissensbasen ziehen dabei logische Schlussfolgerungen primär auf der Grundlage deduktiver Schlüsse. Dieser Umstand bedingt, dass das Wissen um die Auswirkungen spezifischer Aktionen – bspw. zur Einbindung prädiktiver Schlüsse – im Vorhinein in axiomatischer Form einprogrammiert wurde. Eine Möglichkeit hierzu stellen Effekt-Axiome dar, d. h. Axiome des erwarteten Effekts auf Grundlage des eigenen Zustandes oder des Zustandes anderer Entitäten. Entsprechende Ableitungen möglicher Effekte werden hierfür in der deklarativen Wissensbasis hinterlegt und anschließend mittels Inferenz abgerufen. In einfachster Form entspricht dies der Form $A \Rightarrow B$, d. h. dass bei zutreffender Aussage A der Effekt B als Konklusion gefolgert wird. Dass dies jedoch logische Hürden mit sich bringt, sei nun durch das folgende Beispiel erläutert.

Beispiel Es ist zu erwarten, dass der Angriff einer Jägerkompanie auf eine in befestigter Stellung befindliche (feindliche) Infanteriekompanie zu hohen Verlusten der Jäger führt. Um trotz dieser entstehenden Verluste das Momentum des Angriffes nicht zu verlieren, soll ein Führungsautomat zum Entschluss kommen, die Bataillonsreserve als Unterstützungskräfte einzusetzen. Der deklarativen Wissensbasis wird hierfür ein Effekt-Axiom vorgegeben, welches ausgehend von den Aussagen A: „*Eigener Zustand = Angriff*“ und B: „*Gegnerischer Zustand = In Stellung*“ den zukünftigen Effekt C: „*Effekt = Hohe Verluste*“ voraussieht:

F_1 : [A] Action: Attack $\wedge$ [B] E_State: Covered $\Rightarrow$ [C] Effect: H_Casualties

Hierauf aufbauend wird der prozeduralen Wissensbasis vorgegeben, dass bei anzunehmenden oder tatsächlich vorherrschenden hohen Verlusten, d. h. wenn C: „*Effekt = Hohe Verluste*“ gilt, D: „*Goal: Reserven Einsetzen*“ gefolgert werden soll:

F_2 : [C] Effect: H_Casualties ⇒ [D] Goal: Reserves

Führt nun wider Erwarten ein Angriff auf die Infanteriekompanie nicht zu hohen Verlusten, folgert der Automat auf Basis des Modus Ponens trotzdem den Effekt C, weil beide Prämissen A und B erfüllt wurden. Da jedoch in diesem Falle der tatsächliche Zustand der Jägerkompanie $\neg C$ darstellt, folgt $C \wedge \neg C$ als Resultat, was neben der logischen Inkonsistenz fälschlicherweise zu einem unnötigen Einsatz der Reserven führt.

Problematisch erweist sich insbesondere im geschilderten Beispielfall, dass der gefolgerte Effekt in Form einer Tautologie formuliert wurde. Da dies wie dargestellt unter Umständen zu Konflikten führen kann, müssen Effekt-Axiome in ihrer Aussagekraft derart abgeschwächt werden, damit diese die Allgemeingültigkeit verlieren. Ein Ansatz dafür stellt z. B. eine zeitliche Bindung der Aussagen dar, so dass $A(t) \wedge B(t) \Rightarrow C(t)$ gilt und hierdurch $C(t)$ nur zum Zeitpunkt t inferiert werden kann. Hierauf aufbauend muss dem Automaten zusätzliches Wissen zur korrekten Ableitung der zukünftigen Zustände einprogrammiert werden, so dass die deklarative Wissensbasis im Zeitschritt $t + 1$ konfliktfreie Vorhersagen treffen kann:

F_1 : [$A(t)$] Action(t): Attack ∧ [$B(t)$] E_State(t): Covered ⇒ [$C(t)$] Effect(t): H_Casualties
F_2 : [$C(t)$] Effect(t): H_Casualties ⇒ [$C(t+1)$] Effect(t+1): H_Casualties
F_3 : [$C(t+1)$] Effect(t+1): H_Casualties ⇒ [$D(t)$] Goal(t): Reserves

Diese Axiome sind nun im gegebenen Beispiel und dem Wissen um die konkrete Lage bzw. dem bekannten Zustand der Umwelt in ihrer Abfrage logisch konfliktfrei. Aus taktischer Sicht heraus stellt dies jedoch weiterhin eine tautologische Verallgemeinerung der Realität dar, da die zeitliche Abfolge nicht in jeder Situation und Konstellation zutrifft. So wäre ein zeitverzögertes Auftreten von hohen Verlusten zum Zeitpunkt t+2 immer noch außerhalb des Fokus des Führungsautomaten, wodurch dieser mit dem gegebenen Wissen weiterhin nur in der Lage ist, in spezfisichen Situationen zu einem korrekten Schluss zu kommen.

Tatsächlich stellt diese Abhängigkeit von spezifischen Situationen ein logisch nicht zu überwindendes Problem dar. Dies im gegebenen Beispielfall zu beweisen bedarf der Hilfe sog. Fluents (von lat. *Fluens*: fließen), logisch definierter Funktionen $\mathcal{F}$, deren Definitionsbereich der Raum Sit aller möglichen Situationen ist. Man nennt diese **Aussagen-Fluent**, wenn $\mathcal{F}_{prop} : Sit \rightarrow \mathcal{B}$ mit $\mathcal{B} := \{\bot, \top\}$ gilt und **Situations-Fluent**, wenn $\mathcal{F}_{Sit} : Sit \rightarrow Sit$ gilt (McCarthy und Hayes, 1969, 478). Fluents können als Annahmen (engl. *assertions*) über mögliche Systemzustände interpretiert werden. Geläufig ist hierbei der Einsatz unter Zuhilfenahme der Notation der Prädikatenlogik erster Stufe. So ist es nun möglich beispielsweise die Axiome F_1 und F_2 des letzten Absatzes für Entitäten x, y in einer konkreten Situation s als Fluent

$$F : \forall x \forall y \exists s : Attacks(x, y, s) \wedge Covered(y, s) \Rightarrow H_Casualties(x, s)$$

zu formulieren. Dabei beschreibe der Aussagen-Fluent $[\mathbf{A}]Attacks(x, y, s)$, dass Entität x einen Angriff auf Entität y in einer genau spezifizierten Situation s ausführt und $[\mathbf{B}]Covered(y, s)$, dass sich y in s in einer Stellung befindet. Die deduktive Folgerung daraus stellt $[\mathbf{C}]H_Casualties(x, s)$ dar, d. h. dass x in s hohe Verluste hat. Die allgemeine Gültigkeit von **F** folgt dann auf Grundlage des Reductio ad absurdum:

z. Z.: $\models \forall x \forall y \exists s : A(x,y,s) \wedge B(y,s) \Rightarrow C(x,s)$

Ann.: $\neg(\forall x \forall y \exists s : A(x,y,s) \wedge B(y,s) \Rightarrow C(x,s))$

$\exists x \exists y \forall s : A(x,y,s) \wedge B(y,s) \wedge \neg C(x,s) \quad [\equiv]$

$\exists y \forall s : A(x,y,s)\frac{X}{x} \wedge B(y,s) \wedge \neg C(x,s)\frac{X}{x} \quad [\exists^-]$

$\forall s : A(X,y,s)\frac{Y}{y} \wedge B(y,s)\frac{Y}{y} \wedge \neg C(X,s) \quad [\exists^-]$

$\forall s : A(X,Y,s) \wedge B(Y,s) \wedge \neg C(X,s)$

Sei $\mathcal{I}_s$ eine Interpretation mit $\mathcal{I}_s = (\mathcal{A}_s, \beta_s)$, $\mathcal{A}_s = (Sit, \varnothing, \wedge, \vee, \neg)$ und $\beta_s : s \to Sit$. Dann existiert mindestens ein $s \in Sit$, so dass gilt $\mathcal{I}_s \frac{X}{x}\frac{Y}{y} \not\models \neg F$. Somit gilt $\not\models \neg F$, da es mindestens ein Modell gibt, welches der Annahme widerspricht. Insbesondere aber gilt hierdurch $\models \neg\neg F$ und daher $\models F$.

Wie dieser Beweis zeigt, ist es folglich für deduktive Systeme zur Einbindung konfliktfreier Schlüsse zusätzlich zur zeitlichen Bindung immer notwendig, innerhalb einer deklarativen Wissensbasis sämtliche entsprechende Situationen zu spezifizieren, in denen die gewollte Schlussfolgerung zutrifft. Diese umfängliche Adressierung aller möglichen und zugleich gültigen Situationen ist jedoch per se nicht umsetzbar, was den Kern des sog. **repräsentativen Rahmenproblems** (engl. *representative frame*[1] *problem*) beschreibt. Eine mögliche und häufig gewählte Umgehung dieses Problems stellt die explizite Aufzählung all jener Ausnahmesituationen dar, in der die eigentlich intendierte Schlussfolgerung nicht zutrifft. Hier ist es geläufig entsprechende Ausnahmefälle innerhalb der Schlussfolgerungen zu klassifizieren, und dem Automaten somit nur vorzugeben, wie dieser sich in abweichenden oder unvollständig beschriebenen Fällen zu verhalten hat. Wie jedoch Pearl (1988, 1) entnommen werden kann, ist dieses Vorgehen ebenfalls nicht zwangsweise zielführend:

> „*The very act of preparing knowledge to support reasoning requires that we leave many facts unknown, unsaid, or crudely summarized. For example, if we choose to encode knowledge and behavior in rules* [...], *the rules will have many exceptions which we cannot afford to enumerate, and the conditions under which the rules apply* [...] *are usually ambiguously defined or difficult to satisfy precisly in real life.*“

Dies stellt die Basis des sog. **Qualifikationsproblems** dar, welches ohne weiterer Anpassungen mit der bisher aufgezeigten deduktiven Logik – simultan zum repräsentativen Rahmenproblem – nicht vollends gelöst werden kann (McCarthy, 1987, 461).

Das repräsentative Rahmenproblem und das Qualifikationsproblem erweisen sich hinderlich für die explizite Einbindung von Unschärfen, da zweifellos nicht alle möglichen (Ausnahme-)Situationen für einen Führungsautomaten innerhalb einer Wissensbasis explizit addressiert werden können. Dies wird von Russell und Norvig (2010, 481) durch die nachfolgenden Gründe entsprechend konstatiert:

- **Faulheit**

  Das menscheneigene Streben nach Einfachheit behindert aktiv die Benennung und Erzeugung vollständiger Mengen von Effekt-Axiomen.

1 Das Wort frame bedeutet in diesem Sinne den Rahmen, d. h. die umfassenden Zustände eines referenzierten Bezugsystems - vgl. Russell und Norvig (2010, 266).

- **Theoretische Ignoranz**

  Die zu beschreibende Domäne besitzt keine oder nur unzureichend durch Daten gestützte theoretische Abbildungsmöglichkeit.

- **Praktische Ignoranz**

  Selbst im Vorhandensein vollständigen Wissens herrscht möglicherweise Unsicherheit in bestimmten Situationen vor, die dem Erzeuger der Axiome unbekannt ist, oder durch das Modell im besten Fall nur begrenzt mittels der Sensoren erfasst werden kann.

Da eine explizite Adressierung aller Ausnahmefälle und konfliktfreier Situationen ergo schon allein aus praktischen Gründen nicht möglich ist, müssen diese implizit eingebunden werden. Eine Möglichkeit besteht dabei in der Einführung eines numerischen Maßes, welches die innerhalb der Wissensbasis enthaltenen Unschärfen aufsummiert und mittels einheitlicher Prinzipien miteinander kombiniert. Dies befähigt ein wissensbasiertes System dahingehend, unvollständige Informationen innerhalb der Abarbeitung des axiomatischen Wissens als Grad des Vertrauens in jenes Wissen zu beschreiben. Mittels semantischer Kombination des Vertrauens in einzelne Zustände (Ursache) lässt sich dann das Gesamtvertrauen in einen konkreten Effekt (Wirkung) berechnen[2]. Dieses skizzierte Vorgehen weist große Parallelen zur Diagnose von Krankheiten auf, da auch dort mit entsprechend vorliegenden Ursachen (Symptome) auf bestimmte Effekte (Krankheit) geschlossen wird. Derartig operierende wissensbasierte Systeme werden daher auch als diagnostische Systeme bezeichnet (Weichselberger und Pöhlmann, 1990, 1). Diagnostische Systeme stellen eine Sonderform der Expertensysteme dar, welche vorrangig zur Diagnose von Blutkrankheiten und den zielgerichteten Einsatz von Medikamenten in den 1970er und 1980er Jahren entwickelt wurden.

Ihrem herkömmlichen Anwendungszweck innerhalb der Medizinwissenschaften sind diagnostische Systeme jedoch längst entwachsen. Die grundlegenden Mechanismen fanden ihren Weg in eben jene klassischen Expertensysteme, die für ihre Schlussfolgerungen auf empirischen Interdependenzen basieren und hierfür konsequenterweise die Behandlung von Unsicherheiten im Entscheidungsprozess voraussetzen (Weichselberger und Pöhlmann, 1990, 1). Dass dies Führungsautomaten per se mit einschließt, soll nun im Laufe der folgenden Unterkapitel aufgezeigt werden. Die weiteren Betrachtungen werden dabei auf die durch Pearl (1988, 3) verwendete – von Perez und Jiroušek (1985) ursprünglich entnommenene – Unterscheidung der diagnostischen Systeme in **extensionale** und **intensionale** Ansätze aufgebaut. Die folgenden Passagen adressieren hierfür im Detail Ideen und Umsetzungsmöglichkeiten derartiger Ansätze für Führungsautomaten anhand zweier konkreter Beispiele. Jeweilig den Beispielen nachgestellt ist eine Aufstellung der Schwächen, welche mit Blick auf die Tauglichkeit der Ansätze zur Realisierung von Führungsautomaten zu erkennen sind. Den Abschluss dieses Kapitels stellt eine Konklusion dar, welche die für die folgenden Kapitel getroffene Festlegung für einen Ansatz zur Abbildung von Unschärfen im Führungsprozess begründet.

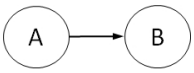
2 Dies sorgt tatsächlich für eine Umgehung des repräsentativen Rahmenproblems und des Qualifikationsproblems, da hierdurch die Allgemeingültigkeit des inhärenten Wissens deutlich entschärft bzw. Ausnahmesituationen indirekt schon mit adressiert werden.

3.1 Extensionale Ansätze

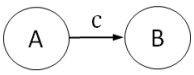
Als erste Möglichkeit zur Umsetzung unscharfer Wissensverarbeitung werden nachfolgend extensionale Ansätze näher beschrieben. Diese umfassen weithin eine Menge unterschiedlicher Realisierungen [MYCIN (Shortliffe, 1976), EMYCIN (Van Melle, 1980), CADUCEUS (Winston und Prendergast, 1986), etc.], welche die ältesten und prominentesten Vertreter medizinischer Expertensysteme darstellen. Dieses Kapitel beschreibt dazu die prinzipielle Funktionsweise extensionaler Ansätze auf Basis von Gewissheitsfaktoren[3]. Hierdurch ist es im Anschluss möglich, einen extensionalen Führungsautomaten zu entwerfen und dessen Nachteile näher zu untersuchen.

Extensionale Ansätze (von lat. *extensio*: Erweiterung, Ausdehnung) erweitern klassische deduktive Systeme, indem diese den Formeln individuelle Werte für deren Gewissheit hinzufügen. Sie verändern dadurch die grundsätzliche Syntax der einzelnen Formeln nicht, sondern führen auf der Grundlage der Verknüpfung extern angehangener Gewissheitsfaktoren zur Abbildung von Unschärfen. Um dieses Prinzip näher zu erläutern, wird nun mit dem Gewissheitsfaktorkalkül des Expertensystems *MYCIN* die grundsätzliche Funktionsweise näher erläutert, da MYCIN eine adäquate Beschreibung der Vor- und Nachteile erlaubt. Wie der Name des Kalküls dabei bereits suggeriert, werden Unschärfen auf Basis sog. Gewissheitsfaktoren (engl. *Certainty Factors*, kurz: **CF**) abgebildet. Da die CF sich in Form und Funktion jedoch von den binärlogischen Werten eines aussagenlogischen Kalkül unterscheiden, ist eine vorangestellte Erläuterung der prinzipiellen Funktionsweise notwendig. Zur besseren Ersichtlichkeit erfolgt diese auf Grundlage von Graphen, da dies ein anschauliches Bild der Verknüpfungen und Weitergabe der CF gibt.

Ausgangspunkt dieser Erläuterungen stellt die Darstellung deduktiver Schlüsse als gerichteter Graph nach Duda et al. (1976) dar. So kann bspw. die Formel „$A \Rightarrow B$“ („IF A THEN B“) mittels



dargestellt werden. Das Antezedenz und die Konsequenz der Formel werden hierbei als Knoten, die Implikation als gerichtete Kante vom Knoten A nach B interpretiert. Semantisch gesehen ist das Hinzufügen von Gewissheitsfaktoren eine Erweiterung der normalen Formel, wonach nun z. B. die bisher bekannte Formel um den Faktor c („…WITH c“) zu „IF A THEN B WITH c“ ergänzt wird. Graphisch gesehen ist dies durch die Einführung eines Kantengewichts c zu



zu folgern[4]. Als Gewissheitsfaktoren werden dabei Werte $c \in W := [-1, 1]$ definiert, wobei positive Werte eine Bestärkung, negative Werte eine Abschwächung der Gewissheit für das Zutreffen eines Systemzustandes (engl. Belief State) repräsentieren. Je stärker dabei die Gewissheitsfaktoren der einzelnen Aussagen gegen die jeweiligen äußeren Grenzen des Wertebereichs streben, desto höher oder niedriger ist die zugesprochene

3 In der Literatur u. a. auch als **Certainty-Faktoren** bezeichnet - vgl. hierzu (Spreckelsen und Spitzer, 2008, 154).

4 Pearl (1988) nutzt hierfür u. a. auch $A \xrightarrow{c} B$ als Vereinfachung dieser graphischen Beschreibung. Um triviale Verbindungen platzsparend darzustellen, wird im Folgenden hierauf zurückgegriffen werden.

Gewissheit des Zutreffens zu interpretieren[5]. Die Auswirkung einer Inferenz auf den Belief State wird als Funktion $cf : W \to W$ bzw. $cf : W \times W \to W$ berechnet. So könnte beispielsweise für die Durchführung eines deduktiven Schlusses mit einer Formel $A \xrightarrow{c} B$ die Funktion $cf_B(c_A) = c_A * c$ für die resultierende Veränderung der Gewissheit c_B gesetzt werden, wobei c_A die mittels cf_A berechnete Gewissheit für die Aussage A darstelle. Dies wird auch als **serielle Kombination** bezeichnet (Heckerman und Shortliffe, 1992, 39).

Besitzt ein Knoten A dabei keine eingehenden Kanten, so ist dessen Gewissheitsfunktion konstant $cf_A() = 1$ unter der Prämisse, dass das schließende System die zur Erfüllung der Aussage A notwendigen Sensorinputs wahrgenommen hat. Ansonsten ist $cf_A() = 0$, da keine konkrete Aussage über das Zutreffen von A getroffen werden kann. Die Wahrnehmung aller notwendigen Sensorinputs für A wird auch mit „A ist evident“ bezeichnet, was dem eigentlichen Ursprung des Wortes Evidenz (von lat. *evidens*: ersichtlich; bzw. lat. *videre*: sehen) entliehen ist. Im weiteren Verlauf der Betrachtung unscharfer Schlüsse wird in dieser Arbeit diese Bezeichnung synonym für die Erfüllung der Prämissen auf Basis des Sensorinputs verwendet werden. Unter Berücksichtigung dieser Notation folgt demnach im gegebenen Beispielfall $A \xrightarrow{c} B$ und vorliegender Evidenz für A, dass B mit Gewissheit c zutrifft. Auf dieser Basis lassen sich in Verbindung mit der seriellen Kombination Berechnungsketten $A \xrightarrow{c_1} B_1 \xrightarrow{c_2} \ldots \xrightarrow{c_n} B_n$ bilden, die auf Grundlage der vorliegender Evidenz einer Prämisse A die hypothetische Gewissheit $c_{B_1}, \ldots, c_{B_n}$ für Folgezustände $B_1, \ldots, B_n$ errechnen.

Sonderformen stellen in diesem Zusammenhang Schlüsse multipler Evidenzen, z. B. für $A_1 \wedge A_2$ auf einen Zustand B (**parallele Kombination**) bzw. Schlüsse von einer Evidenz, z. B. für A auf multiple Zustände $B_1 \vee B_2$ (**divergente Propagierung**) dar, was graphisch als

gefolgert werden kann Die Besonderheit dieser Schlüsse erklärt sich dadurch, dass diese sich u. a. in der weiterlaufenden Kombination der CF von einer seriellen Kombination unterscheiden. So teilt eine divergente Propagierung die Berechnungskette in mehrere Teilketten, die simultan jeweils auf Basis einer seriellen Kombination weiterbearbeitet werden: $cf_{B_1}(c_A) = c_A * c_1; \quad cf_{B_2}(c_A) = c_A * c_2$. Eine parallele Kombination bedingt dem entgegen eine eigens zugeschnittene Funktion $cf_B(c_{A_1}, c_{A_2})$, welche die Kombination der eingehenden Gewissheit definiert. Diese Funktion begründet sich dabei entweder auf einer subjektiven Festlegung oder statistischen Erhebungen, welche im Zweifelsfall aus dem Kontext und der Zielstellung heraus zu folgern sind. Da parallele Kombinationen somit höchst kontextsensitiv sind, wird zur besseren Ersichtlichkeit auf die Anschauungsbeispiele in (Heckerman, 1992, 132ff), (Heckerman und Shortliffe, 1992, 37ff) und auf das folgende Unterkapitel 3.1.1 verwiesen.

Dies beschreibt in Summe die die drei hauptsächlichen Kombinationsarten des Gewissheitsfaktorkalküls, was im weiteren Sinne zur Betrachtung extensionaler Ansätze in dieser Arbeit ausreicht. Im nachfolgenden

5 Ein Wert $c = 1$ stellt in diesem Sinne ein Schluss unter absoluter Gewissheit dar. Der Wert $c = -1$ gilt in diesem Sinne als komplette Aufhebung der Gewissheit für einen Belief State.

Abschnitt wird nun diese bisher rein formale Beschreibung der Funktionsweise anhand eines konkreten Beispiels für extensionale Führungsautomaten näher erläutert werden.

3.1.1 Beispiel eines Führungsautomaten

Als Ausgangslage soll erneut wieder die bereits bekannte Situation aus Kapitel 2.3 dienen, welche nun mit zusätzlichen Deckungsinformationen Abb. 3.1 zu entnehmen ist. Der Führungsautomat soll hierbei nun befähigt werden, in der gegebenen Lage einen Angriffsbefehl für den blauen Jägerzug zu bewerten, um adäquate Folgeschritte auf Basis des wahrgenommenen Systemzustands zu generieren. Für eine Auswertung der Lage benötigt der Automat dafür vorrangig ein Datenmodell, welches die möglichen Sensorinputs aufführt, die zur Definition einer deklarativen Wissensbasis benötigt werden. Anhaltspunkte lassen sich hierzu aus den allgemeinen Faktoren eines militärischen Kräfte- und Fähigkeitsvergleichs zur Beurteilung der Erfolgsaussichten der Möglichkeiten des eigenen Handelns extrahieren. Im Rahmen dieses Vorgehens wird sowohl die eigene Lage, als auch die Lage des Feindes mit der übergeordneten Intention verknüpft, um hierauf aufbauend den möglichen eigenen Erfolg zu bemessen. Im Beispielfall genüge es dabei, dass eine Bewertung der Lage auf Basis der **Deckung**, der beobachteten **Bewegungen**, des **Kräfteansatzes** und der erwarteten **Waffenwirkung** im Ziel getroffen wird. Eine in diesem Zusammenhang getroffene Grundannahme stelle zur Vereinfachung die Tatsache dar, dass innerhalb des Beispielsystems alle Kräfte zu jeder Zeit aufgeklärt sind.

Gemäß gängiger Konvention ist es üblich, die Bewertung der Faktoren – im Angriff – von der eigenen Truppe her zu beginnen. Aus Sicht des Führungsautomaten bedeutet dies, dass die vier Faktoren vom Bezugspunkt des blauen Jägerzugs aus beginnend zu bemessen sind. Zur Vereinheitlichung dieser Bemessungen werden diese hypothetischen Beurteilungen des Systems nun folgend in semantische Ausdrücke A_x, B_x, C_x und D_x mit $x \in \{blue, red\}$ unterteilt, deren mögliche Belegungen in der Wissensbasis hinterlegt werden:

- **[$\mathbf{A_x}$]** Bewegung $\rightarrow$ [$\mathbf{A_1}$] Keine | [$\mathbf{A_2}$] Front | [$\mathbf{A_3}$] Flanke | [$\mathbf{A_4}$] Rücken
- **[$\mathbf{B_x}$]** Waffenwirkung $\rightarrow$ [$\mathbf{B_1}$] Keine | [$\mathbf{B_2}$] Gering | [$\mathbf{B_3}$] Gut
- **[$\mathbf{C_x}$]** Deckung $\rightarrow$ [$\mathbf{C_1}$] Ungedeckt | [$\mathbf{C_2}$] Teilgedeckt | [$\mathbf{C_3}$] Gedeckt
- **[$\mathbf{D_x}$]** Kräfteansatz $\rightarrow$ [$\mathbf{D_1}$] Unterlegen | [$\mathbf{D_2}$] Ausgeglichen | [$\mathbf{D_3}$] Überlegen

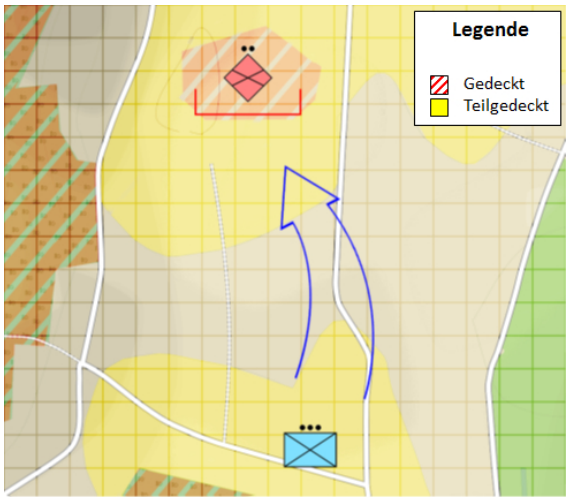
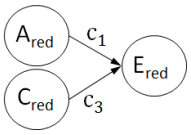
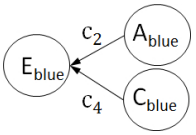


Abbildung 3.1: Ausgangslage mit Deckungsinformationen

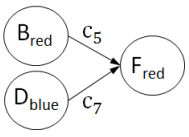
Mit Hilfe einer logischen Verknüpfung können diese Faktoren nun zu einer Bewertung der Lage eingesetzt werden. Die erste Bewertung stelle hierbei dar, dass die Bewegung des Feindes eine Aussage darüber zulässt, ob sich dieser in einer vorteilhaften Stellung (E_x) befinden möge. Dieser Zusammenhang wird über $A_{red} \xrightarrow{c_1} E_{red}$ bzw. $A_{blue} \xrightarrow{c_1} E_{blue}$ formal dargestellt, wobei $\mathbf{E_{red}}$ eine vorteilhafte Stellung der roten Gruppe aus blauer Sicht und $\mathbf{E_{blue}}$ eine vorteilhafte Stellung des blauen Zuges aus roter Sicht beschreibe. Diese Annahme wird zusätzlich dadurch beeinflusst, ob und wie der Feind gedeckt ist. Dies lässt sich zu $C_{red} \xrightarrow{c_3} E_{red}$ bzw. $C_{blue} \xrightarrow{c_4} E_{blue}$ formalisieren. Da somit sowohl $A_x \xrightarrow{c_\alpha} E_x$, als auch $C_x \xrightarrow{c_\beta} E_x$ gilt, können parallele Kombinationen konstruiert werden, womit sich



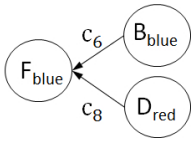
und



ergeben. Der generelle Erfolg eines Angriffs (F_x) wird primär auf Basis der zu erwartenden Waffenwirkung bemessen, was durch $B_x \xrightarrow{c_5} F_x$ respektive $B_x \xrightarrow{c_6} F_x$ abgebildet wird, wobei $\mathbf{F_{blue}}$ eine gute Erfolgschance blauer Teile aus roter Sicht, $\mathbf{F_{red}}$ gute Erfolgschancen roter Teile aus blauer Sicht beschreibt. Da der militärische Erfolg darüber hinaus stark von der Anzahl eingesetzter Kräfte abhängt, kann ferner noch $D_{red} \xrightarrow{c_7} F_{blue}$ und $D_{blue} \xrightarrow{c_8} F_{red}$ geschlossen werden. Ähnlich wie im Falle der Bewertung der Stellung lassen sich auch hier erneut parallele Kombinationen



und



konstruieren. Damit verarbeitet der Automat die ihm zur Verfügung stehenden Sensorinputs sowohl in der Bewertung der feindl. bzw. eigenen Stellung (E_x), als auch in der Bewertung möglicher guter Erfolgschancen beider Seiten (F_x).

Hierauf lassen sich nun seitens des Führungsautomaten Folgerungen respektive Elemente des eigenen Handelns überprüfen. Da die grundsätzliche Intention hinter dem Automaten die Bewertung und Ausführung des Angriffes vorsieht, muss nun in Folge noch das grundsätzliche Risiko[6] eines Angriffes in der gegenwärtigen Position bemessen werden. Da die Bewertung der feindlichen Position (Bewegung und Deckung) entscheidend für das resultierende Risiko im Angriff ist, ist ferner $E_{red} \xrightarrow{c_9} G_{red}$ bzw. $E_{blue} \xrightarrow{c_10} G_{blue}$ als serielle Kombination festzulegen. Dabei stehe $\mathbf{G_{red}}$ für ein Risiko der roten Seite während des Angriffes und $\mathbf{G_{blue}}$ für das korrespondierende Risiko der blauen Seite. Ebenfalls ist zweifelsfrei festzustellen, dass die Beurteilung des eigenen bzw. feindlichen Erfolges eine Auswirkung auf die Risikobewertung haben wird. Daher herrschen auch die Beziehungen $F_{red} \xrightarrow{c_{11}} G_{red}$ und $F_{blue} \xrightarrow{c_{12}} G_{blue}$ vor. Es lassen sich somit erneut – da $E_x \xrightarrow{c_\alpha} G_x$, als auch $F_x \xrightarrow{c_\beta} G_x$ gelten – die beiden parallelen Kombinationen

in Konklusion erstellen. Als letzten Schritt der Bewertung sind diese ermittelten Risikobewertungen in einen Vergleich zu stellen. Diese Gegenüberstellung diene zur Ermittlung, ob ein Angriff in der vorliegenden Lage aus blauer Sicht eine überlegene ($\mathbf{H_1}$), ausgeglichene ($\mathbf{H_2}$), oder unterlegene ($\mathbf{H_3}$) Situation führt. Formal kann dies als divergente Propagierungen $G_{red} \xrightarrow{c_{13}} H_1$, $G_{red} \xrightarrow{c_{14}} H_2$, $G_{red} \xrightarrow{c_{15}} H_3$ und $G_{blue} \xrightarrow{c_{16}} H_1$, $G_{blue} \xrightarrow{c_{17}} H_2$, $G_{blue} \xrightarrow{c_{18}} H_3$, d. h.

abgebildet werden. Auch hier ist bereits erkennbar, dass sich aus den divergenten Propagierungen $G_{red} \xrightarrow{c_\alpha} H_z$ und $G_{blue} \xrightarrow{c_\beta} H_z$ paarweise parallele Kombinationen bilden lassen. Hieraus folgt in Konklusion der Gesamtgraph in Abb. 3.2, welcher die vorliegenden Beziehungen innerhalb der deklarativen Wissensbasis qualitativ beschreibt.

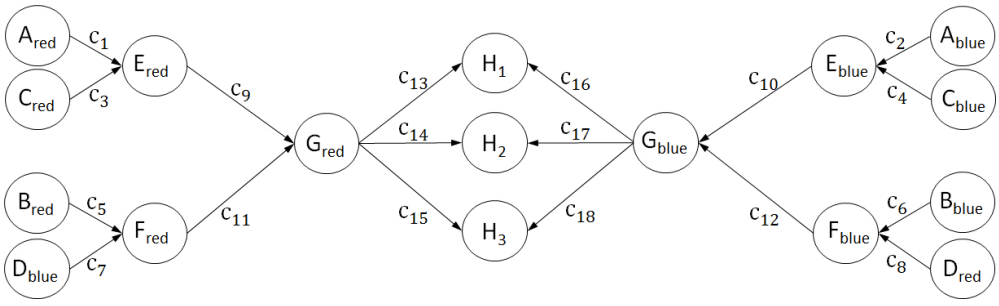


Abbildung 3.2: Gesamtgraph der extensionalen deklarativen Wissensbasis

6 Innerhalb des Führungsprozesses für Landstreitkräfte ist es geläufig, Einsatzwerte für Handlungsoptionen anhand dreier Kategorien – hoch, mittel und niedrig – zu vergeben. Das an dieser Stelle hier aufgeführte Risiko erfüllt de facto denselben Zweck, ist jedoch aus praktischen Gründen nur als einfacher Wert hinterlegt.

Zur Vervollständigung der deklarativen Wissensbasis ist nun noch notwendig, die konkreten Belegungen der Gewissheit $c_1, \ldots, c_n$ zu motivieren. Hier zeigt sich vorteilhaft, dass es sich im Falle der Sensorinputs um disjunkte Ereignisse und damit auch um ausschließlich getrennte Evidenzen handelt. Dies macht es möglich, dass die unterschiedlichen Sensorinputs A, B, C und D mit multiplen Werten für die Gewissheit belegt werden können[7]. Beispielhaft ist dies der folgenden Tab. 3.1 zu entnehmen.

	A_1	A_2	A_3	A_4
c_1	0.7	-0.5	-0.2	-0.1
c_2	-0.7	0.1	0.8	0.9

	B_1	B_2	B_3
c_5	-0.8	0.2	0.6
c_6	-0.8	0.2	0.6

	C_1	C_2	C_3
c_3	-0.5	0.3	0.5
c_4	-0.5	0.3	0.5

	D_1	D_2	D_3
c_7	-0.5	-0.2	0.5
c_8	-0.5	-0.2	0.5

$c_9 = cf_{E_1}(c_1, c_3)$ $\quad c_{10} = cf_{E_2}(c_2, c_4)$ $\quad c_{11} = -cf_{F_1}(c_5, c_7)$ $\quad c_{12} = -cf_{F_2}(c_6, c_8)$

$c_{13} = -cf_{G_1}(c_9, c_{11})$ $\quad c_{14}, c_{15} = cf_{G_1}(c_9, c_{11})$ $\quad c_{16}, c_{17} = cf_{G_2}(c_{10}, c_{12})$ $\quad c_{18} = -cf_{G_2}(c_{10}, c_{12})$

Tabelle 3.1: Zuweisung der Gewissheit

Da innerhalb der deklarativen Wissensbasis parallele Kombinationen abgebildet sind, werden zu deren Auflösung noch die jeweils zur Anwendung kommenden Gewissheitsfunktionen benötigt. Im Beispielsfall seien diese als einheitliche Funktion

$$cf_z(cf_x, cf_y) = \begin{cases} c_x + c_y - c_x c_y & c_x, c_y \geq 0 \\ c_x + c_y + c_x c_y & c_x, c_y < 0 \\ \frac{c_x + c_y}{1 - min(|c_x|, |c_y|)} & sonst \end{cases}$$

definiert. Die deklarative Wissensbasis ist damit vollständig, worauf der Führungsautomat nun aufgebaut wird. In diesem Beispiel sei der Automat als hybrider wissensbasierter Ansatz (vgl. Kapitel 2.3.3) realisiert, wobei der aufgezeigte Gesamtgraph in seiner Funktion als deklarative Wissensbasis hinterlegt wird. Das in Abb. 3.4 dargestellte Programm diene nun fortlaufend als Anhalt für die folgende Erläuterung der Funktionsweise.

	Initial	CF	Wert
A_{blue}	A_2	c_2	0.10
B_{blue}	B_3	c_5	0.60
C_{blue}	C_2	c_4	0.30
D_{blue}	D_3	c_7	0.50
A_{red}	A_1	c_1	0.70
B_{red}	B_3	c_6	0.60
C_{red}	C_3	c_3	0.50
D_{red}	D_1	c_8	−0.50

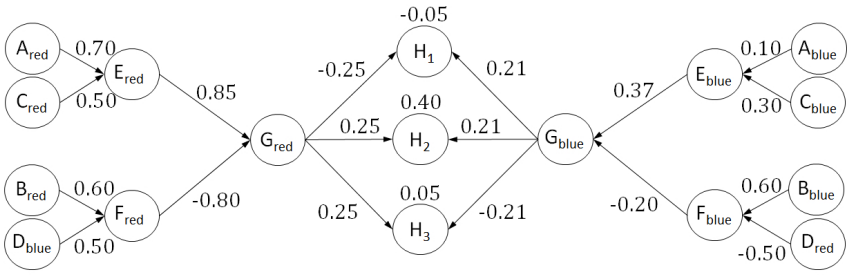


Abbildung 3.3: Beispielwerte in der extensionalen deklarativen Wissensbasis

Der Automat ermittelt anhand der deklarativen Wissensbasis die aktuelle Lage des blauen Jägerzugs und interpretiert parallel die rote Lage. Dies geschieht auf Basis der momentanen Wahrnehmung (*CPercepti-*

7 Ein Vorgehen, welches in ähnlicher Form Pearl (1988, 39ff) entnommen werden kann.

on), welche als Input für die Berechnung der Gewissheitsfaktoren mit Rückgriff auf die Beziehungen aus Abb. 3.3 dient. Die aufgrund der Wahrnehmung vorgefundenen Evidenzen werden an die deklarative Wissensbasis übergeben (*queryPerc*) und die hieraus resultierenden Gewissheitsfaktoren gem. der in Tab. 3.1 festgehaltenen Wertezuweisungen berechnet. Damit errechnet der Automat den aktuellen Systemzustand (*current*), der als Vektor von Zustands-Gewissheitsfaktor-Tupeln gespeichert wird. Die Auswertung (*outcome*) der gegenwärtigen Lage wird als outcome = max(H_1, H_2, H_3) gesetzt, womit derjenige Zustand H_x mit der höchsten gegenwärtigen Gewissheit die gefolgerte Beurteilung der Lage darstellt. So kommt der Automat in der gegebenen Beispiellage zur Bewertung, dass ein Angriff im momentanen Zustand am sichersten H_2 zuzuordnen ist - vgl. hierfür die Berechnungswerte aus Abb. 3.3. Dadurch wird dem Umstand Rechnung getragen, dass nach Einschätzung des Führungsautomaten ein Frontalangriff auf den in Stellung befindlichen Feind, bei eigener überlegener Stärke und teilgedecktem Angriffsweg, in etwa zu einem ausgeglichenen Gefecht führt. Dies stellt jedoch zweifellos nicht die beste Handlungsalternative dar, da die rote Infanteriegruppe durch die bezogene Verteidigungsposition in ihren Flanken sehr verwundbar ist.

Damit der Automat nun zu erkennen vermag, dass in bestimmten Ausgangssituationen ein Flanken- oder sogar Rückenangriff taktisch vorteilhaft ist, muss dieser als Teil der Entscheidungsfindung ggf. die Optionen des eigenen Handelns abwägen (*checkAlts*). Eine Möglichkeit stellt es in diesem Zusammenhang dar, die Auswirkung eines Positionswechsels auf den eigenen Einsatzwert zu beurteilen. Der korrespondierende Faktor $A_{blue} = \{A_1, A_2, A_3, A_4\}$ muss hierfür lediglich innerhalb des momentanen Zustands verändert, die Auswirkung dieser Veränderung auf die Bewertung mit Hilfe der deklarativen Datenbank automatisch überprüft und, im Falle einer positiven Veränderung des Einsatzwertvergleichs, innerhalb eines reformierten Zielzustand (*goal*) an den prozeduralen Anteil der Wissensbasis übergeben werden. Im Beispielfall bedeutet dies, dass der zu erwartende Effekt eines Wechsels von A_x zu A_{x+1} untersucht wird[8]. Somit wägt der Automat dessen gültige Optionen während des Angriffes ab. Auf Basis des resultierenden und ggf. erstrebenswerten Zielzustand plant der Führungsautomat im Anschluss den Angriff (*planAttack*). Der hieraus enstehende Plan (*seq*) wird dann schrittweise an die prozedurale Wissensbasis übergeben (*queryAct*), um die konkreten Handlungsanweisungen an den unterstellten Bereich zu ermitteln. Auf eine nähere Erläuterung dieses prozeduralen Anteils wird an dieser Stelle jedoch verzichtet, da dieser unabhängig vom aufgezeigten extensionalen Ansatz zu implementieren ist und daher für die weitere Diskussion in diesem Kapitel keinen Mehrwert liefert.

8 Dies begründet sich darauf, dass ein plötzlicher Wechsel aus dem Stillstand nur in Richtung der Front des Feindes, bzw. von der Bewegung auf die Front des Feindes nur in die Flanke des Feindes, etc. möglich ist. Die Indizes von A sind auf der Grundlage geplant worden, dass diese in aufsteigender Reihenfolge eine mögliche Option im nächsthöheren Indexwert erlauben. Weitere Sprünge von A_x zu A_{x+2} wären in der gegenwärtigen Lage aufgrund der Vorhersehbarkeit dieser Manöver nicht plausibel.

Programm 6 Extensionaler Führungsautomat

```
DataBase : KB;
Array of State: seq;
function C2(blue: Entity) returns Action
    State: current, goal;
    Action: next;
    current ← KB.queryPerc(blue.getCPerception());
    if seq == ∅ then
        if current.outcome == (H3 ∨ H2) then
            checkAlts(current);
            goal ← current;
        end if
        seq ← planAttack(blue, goal);
    end if
    next ← KB.queryAct(seq.pop());
    return next
end function
```

Suche nach Alternativen

```
procedure CHECKALTS(current: State)
    State: goal ← current;
    Knode: knode ← goal.get(A_blue);
    Integer: i ← knode.indexValue;
    Float CF1,CF2;
    if i < knode.maxIndex then
        goal.setKnodeVal(0, knode.indexValue++);
        CF1 ← goal.outcome.CF;
        CF2 ← current.outcome.CF;
        if (goal.outcome ==H1)∨ (CF1 > CF2) then
            current ← goal;
        end if
    end if
end procedure
```

Abbildung 3.4: Extensionaler Ansatz eines hybriden wissensbasierten Automaten

3.1.2 Schwächen des Ansatzes

Im Folgenden werden die konkreten Schwächen aufgezeigt, die mit der Nutzung extensionaler Ansätze einhergehen. Betrachtet werden im Schwerpunkt allgemeine Schwächen, wodurch diese nicht ausschließlich nur in der Anwendung innerhalb militärischer Führungsautomaten zu erwarten sind. Diese Untersuchung generell vorherrschender Nachteile bedingt es dabei, auf die Grundprinzipien der logischen Funktionsweise extensionaler Ansätze näher einzugehen, um von dort ausgehend funktionelle respektive mathematisch logische Diskrepanzen besser aufzeigen zu können.

Die offensichtlichste Stärke extensionaler Ansätze ist gleichzeitig auch deren größten Schwäche: Die klare Zugehörigkeit zur Klasse der deduktiven Systeme und damit die verbundene Erwartungskonformität in der Formulierung von Axiomen. Die Erweiterung der Aussagenlogik durch die Einführung externer Werte zur Bemessung der Gewissheit ist in ihrer Einfachheit jedoch trügerisch, da extensionale Ansätze leicht falsch interpretiert und auf Grundlage dieser falschen Interpretation auch zu falschen Rückschlüssen führen können. Ursache hierfür stellt das Prinzip der sog. **Modularität** dar, einer der Inferenzregeln klassischer Logik entliehenen, einheitlichen prozeduralen Festlegung zur Schlussbildung. Im einfachsten Fall ist dieses Prinzip am Beispiel eines implikativen Schlusses $A \Rightarrow B$ erläutert:

> „*If you* [das System, Anm. d. Verf.] *see A anywhere in the knowledge base, then regardless of what other things the knowledge base contains and regardless of how A was derived, you are given the license to assert B and add it to the database.*“ (Pearl, 1988, 5)

Die Modularität deduktiver Schlüsse zeichnet sich damit durch **Lokalität** (gleich welches Wissen die Wissensbasis sonst noch enthält) und der **Loslösung** vom Kontext (gleich wie A hergeleitet wurde) aus. Angehängte numerische Parameter, wie sie beispielsweise im Gewissheitsfaktorkalkül üblich sind, verändern dieses Prinzip grundsätzlich nicht (Pearl, 1988, 5). Tatsächlich aber führt die Erweiterung mit Gewissheitsfaktoren zu einer vollkommen anderen Interpretation. So kann die Modularität beispielsweise unter Nutzung von Gewissheitsfaktoren, d. h. im erweiterten Fall $A \xrightarrow{x} B$, folgendermaßen gedeutet werden:

> „*If you* [das System, Anm. d. Verf.] *see the certainty of A undergoing a change δ_A, then regardless of what other things the knowledge base contains and regardless of how δ_A was triggered, you are given an unqualified license to modify the current certainty of B by some amount δ_B, which may depend on x, on δ_A, and on the current certainty of B.*“ (Pearl, 1988, 5f)

Besonderes Augenmerk sei hierbei auf den Auslöser des Schlusses auf B gelegt, da dieser offensichtlich nicht von der Belegung, sondern von der Änderung in der Belegung des Symbols A abhängt. Dies stellt eine substanziell andere Aussage über das Ergebnis der Inferenz im Gegensatz zur herkömmlichen Aussagenlogik dar.

Um diesen Umstand besser zu verdeutlichen, ist es notwendig die grundsätzliche Definition extensionaler Schlüsse herzuleiten. Dies bedarf vorrangig einer Hilfsnotation[9] zur Darstellung der Schlüsse, welche folgend nun aufgestellt wird:

> Es beschreibe $H|\mathfrak{e}$ den Schluss auf einen hypothetischen Systemzustand $H \in \Omega_H$ unter vorliegender Evidenz $\mathfrak{e}$. Eine Funktion B weise dabei einem Systemzustand unter vorliegender Evidenz einen Wert für dessen Gewissheit zu, wobei folgenden Axiome gelten:
>
> (1) $0 \leq B(H|\mathfrak{e}) \leq 1$
>
> (2) $B(\Omega_H|\mathfrak{e}) = 1$
>
> (3) $B(H|\mathfrak{e}) + B(\neg H|\mathfrak{e}) = 1$ *Summenregel*
>
> (4) $B(H_1, H_2|\mathfrak{e}) = B(H_1|\mathfrak{e}) * B(H_2|H_1, \mathfrak{e})$ *Produktregel*
>
> Zusätzlich beschreibe $B(H|\mathfrak{E})$ den Wert der Gewissheit, sich in einen Systemzustand H unter einem a priori vorherrschenden Informationsstand $\mathfrak{E}$ zu befinden.

Auf diesem Gerüst lassen sich nun z. B. Gewissheitsfaktoren formal – in Folgerung aus der ursprünglichen Definition von Shortliffe und Buchanan (1984, 248ff) – definieren. Dabei stelle der Wert $MB(H|\mathfrak{e})$ den Wert

9 Sowohl die Notation, als auch die folgend aufgestellten Axiome sind den Ausführungen von Horvitz und Heckerman (1986, 140) entnommen und dem Kontext entsprechend angepasst worden.

der Gewissheit dar, sich in einen spezifischen Systemzustand H auf Basis einer vorliegenden Evidenz $\mathfrak{e}$ zu befinden und es gilt:

$$MB(H|\mathfrak{e}) \quad = \begin{cases} \dfrac{max(B(H|\mathfrak{e})\ ,\ B(H|\mathfrak{E})) - B(H|\mathfrak{E})}{1 - B(H|\mathfrak{E})} & B(H|\mathfrak{E}) \neq 1 \\ 1 & sonst \end{cases}$$

Konträr hierzu stelle der Wert $MD(H|\mathfrak{e})$ den Wert des Zweifels dar, sich in einen spezifischen Systemzustand H auf Basis einer vorliegenden Evidenz $\mathfrak{e}$ zu befinden, wobei

$$MD(H|\mathfrak{e}) \quad = \begin{cases} \dfrac{B(H|\mathfrak{E}) - min(B(H|\mathfrak{e})\ ,\ B(H|\mathfrak{E}))}{B(H|\mathfrak{E})} & B(H|\mathfrak{E}) \neq 0 \\ 1 & sonst \end{cases}$$

gilt. Ein Gewissheitsfaktor ist dann die Differenz der Werte der Funktionen (1) und (2):

$$CF(H|\mathfrak{e}) \quad = MB(H|\mathfrak{e}) - MD(H|\mathfrak{e}).$$

Besondere Aufmerksamkeit ist auf $B(H|\mathfrak{E})$ zu richten, da offensichtlich der a priori vorherrschende Wissenstand einen entscheidenden Einfluss auf Gewissheit, Zweifel und damit indirekt auch auf Gewissheitsfaktoren hat. Die Tragweite dieses Umstandes lässt sich jedoch erst bei einem Wechsel des Wissenstandes ermessen, welcher grundsätzlich bei Vorliegen einer neuen Evidenz eintritt. In diesem Zusammenhang sei in Erinnerung gerufen, wie der Begriff der Evidenz zu Beginn dieses Kapitels definiert wurde. So herrscht beispielsweise im Falle $A \xrightarrow{x} B$ erst dann Evidenz für A vor, wenn die Belegung von A keine logische Folgerung aus anderen Aussagen darstellt und A wortgemäß durch einen Sensorinput wahrgenommen wurde. Kausal ist dabei anzunehmen, dass zu Beginn jeder wahrzunehmende Faktor weder bestätigt noch abgelehnt werden kann, da erst durch tatsächliche Wahrnehmung oder Nichtwahrnehmung des jeweiligen Faktors eine sichere Aussage hinsichtlich der Gewissheit getroffen werden kann. Die stringente Umsetzung dieses Umstands bedeutet nun aber im Umkehrschluss, dass zur Abbildung dieser Indifferenz ($CF_A = 0$) a priori $B(A|\mathfrak{E}) = 1$ gelten muss, da nur so die korrespondierende Gewissheit $B(A|\mathfrak{e}) = 0$ systemintern abzubilden ist[10]. Simultan gilt dies dann aber auch für eine a priori vorherrschende Indifferenz hinsichtlich von B, womit auch dann $B(B|\mathfrak{E}) = 1$ – oder auch allgemein für alle $B(H|\mathfrak{E}) = 1$ – gelten muss, was nicht zuletzt eine logische Konsequenz aus der Produktregel darstellt.

Sinngemäß ist hieraus zu folgern, dass, obwohl ein Gewissheitsfaktor von 0 für eine Indifferenz im Belief State spricht, ein maximaler Gewissheitswert von 1 für den a priori Zustand zur Umsetzung von Input-gesteuerten Evidenzen vorherrschen muss. Dieser Umstand hat jedoch keinerlei tiefergehenden Auswirkungen auf die letztendlichen Schlüsse des Systems, da bei vorliegen von (neuen) Evidenzen lediglich die Abweichungen vom a priori vorherrschenden Informationsstand weitergegeben werden. Damit ignorieren extensionale Ansätze die a priori vorgegebenen Informationen und bemessen lediglich die Auswirkung des

10 Ein Hinweis dafür, dass dies gelten muss, liefert u. a. die allgemeine Formel zur Berechnung der seriellen Kombination: $cf_B(c_A) = c_B * max(0, cf_A)$ oder entsprechendes Einfügen des Wertes in die Formeln 1 und 2.

Wechsels im Belief State[11]. Dies wird jedoch wissentlich in Kauf genommen, wie durch Henrion (1987, 120) bspw. gefolgert wird:

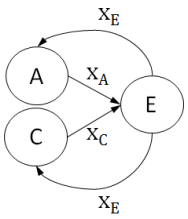
> *„Since Certainty Factors represent changes to priors, Mycin*[!]*-like systems could in principle incorporate priors, but typically applications of CFs have not. The rationale has been that prior probabilites are too hard to estimate, and it is better to avoid them. However for* [...] *Mycin*[!] *(and its derivatives), ignoring priors is functionally equivalent to assuming equal priors* [...]*. Occasionally such uniform priors may be appropriate, but more often it means ignoring important information about differing frequencies."*
> [Hervorhebungen aus dem Original übernommen]

Somit ist ein Großteil extensionaler Ansätze de facto gedächtnislos, was einerseits für deren Einfachheit sprechen mag, auf der anderen Seite jedoch die Mächtigkeit zur Abbildung von Unschärfen stark vermindert.

Das aufgezeigte Schlussverhalten deutet jedoch auf ein größeres Problem innerhalb extensionaler Ansätze hin. So beschreibt bspw. die Formel $A \xrightarrow{x_A} B$ rekapitulierend einen Schluss darauf, dass, wenn auf Basis einer Evidenz $\mathfrak{e}$ der hypothetische Systemzustand A angenommen wird, der Systemzustand B mit Gewissheit x_A zu folgern ist. Auf Grundlage der zuvor eingeführten Hilfsnotation ist dies durch die Produktregel abgedeckt, was im Gewissheitswert $B(A, C|\mathfrak{e}) = B(A|\mathfrak{e}) * B(C|A, \mathfrak{e})$ resultiert. Wie der bereits zu Beginn dieses Kapitels abgeleiteten Konvention für Sensorinput-basierten Evidenz entnommen werden kann, ist für $B(A|\mathfrak{e}) = 1$ in Folge einer gültigen Evidenz $\mathfrak{e}$ für A zu schließen. Gemäß dieser Konvention gilt dabei in Verbindung mit der Hilfsnotation zusätzlich $B(B|A, \mathfrak{e}) = x_A$, woraus $B(A, B|\mathfrak{e}) = x_A$ folgt und korrekterweise die seriellen Kombination widerspiegelt. Es ist in diesem Zusammenhang jedoch erwähnenswert, dass eine derart vorherrschende serielle Kombination einen abduktiven Schluss, d. h. $B \xrightarrow{x_B} A$ explizit ausschließt. Tatsächlich sind extensionale Ansätze grundsätzlich unfähig zur Abbildung von abduktiven Inferenzen (Pearl, 1988, 7). Der Grund hierfür wird nun in Folge auf Basis eines Beispiels aufgezeigt.

Beispiel Es sei – simultan zur Wissensbasis aus Kap. 3.1.1 – **A** ein Faktor zur Bemessung der Bewegung, **C** ein Faktor zur Bemessung der Deckung und **E** ein Faktor zur Bemessung der Stellungsgüte eines Feindes. Dabei gelte $A \xrightarrow{x_A} E$ bzw. $C \xrightarrow{x_C} E$ und damit auch $(A \wedge C) \xrightarrow{x_{A \wedge C}} E$. Besitzt der Automat kein vollständiges Lagebild, soll dieser nun dazu befähigt werden, durch Rückschluss der vorhandenen Evidenz auf den Belief State nicht wahrgenommener Zustände rückzuschließen. Hierfür werden zusätzliche abduktive Rückschlüsse mittels der Deduktionen $E \xrightarrow{x_E} A$ und $E \xrightarrow{x_E} C$ eingeführt, die bei vorliegender Evidenz $\mathfrak{e}$ für E die Gewissheit von A und C entsprechend verändern. Zur besseren Verdeutlichung sei hierzu auf die graphische Interpretation des somit entstehenden logischen Gefüges, d. h. auf

11 Dieser Tatsache ist es geschuldet, dass $B(H|\mathfrak{E}) = 1$ pauschal und hintergründig bereits in der Herleitung des Gewissheitsfaktorkalküls und damit für den Anwendungsfall der Führungsautomaten definiert wurde. Die Festlegung hierfür geschah jedoch nicht vollkommen willkürlich. Vielmehr konnten hierdurch gewisse logische Inkonsistenzen überbrückt werden. Für ein Beispiel derartiger Inkonsistenzen sei z. B. auf (Adams, 1976) verwiesen.



verwiesen. Es sei nun angenommen, dass eine Evidenz $\mathfrak{e}$ so vorliegt, dass $B(A|\mathfrak{e}) \geq 0$ ist. Aufgrund der vordefinierten Wechselwirkungen schließt der Automat folgerichtig $B(E|A, \mathfrak{e}) \geq 0$. Da der Automat jedoch dem Prinzip der Modularität unterliegt, folgt auf Grundlage der Lokalität und Loslösung mittels der initial festgelegten Regeln, dass hieraus durch Änderung der Gewissheit für C auch $B(C|A, E, \mathfrak{e}) \geq 0$ gilt, was wiederum $B(E|A, E, C, \mathfrak{e}) \geq 0$ auslöst, etc. Der Automat begibt sich somit in eine Endlosschleife, die eine permanente Neuberechnung aller involvierten Gewissheitsfaktoren initiiert.

Extensionale Ansätze sind ergo nicht in der Lage abduktive Schlüsse abzubilden, da die mutwillige Einbindung abduktiver Zusammenhänge – wie im Beispiel dargestellt – zu Endlosschleifen in der Berechnung führt. Sie können daher nur deduktive Zusammenhänge beschreiben. Es ist in diesem Zusammenhang jedoch festzuhalten, dass innerhalb des militärischen Führungsprozesses abduktive Schlüsse offensichtlich einen wertvollen Mechanismus darstellen, da diese mangelnde Lageinformationen kompensieren können. Zusätzlich schöpfen abduktive Inferenzen dadurch Mehwert, da diese einen Rückschluss von auslösenden Faktoren auf das mögliche Eintreten zukünftiger Phänomene und damit prädiktive Schlüsse ermöglichen. Dass extensionale Ansätze hierzu nicht in der Lage sind, schränkt die Leistungsfähigkeit für Führungsautomaten stark ein. Insbesondere mahnt dies aber auch zur Vorsicht in der Generierung von Regeln, da Berechnungskreise aufgrund der entstehenden Endlosschleifen zwingend vermieden werden sollten.

Zusammenfassend weisen extensionale Ansätze somit zwei primäre Schwächen für ihren Einsatz innerhalb von Führungsautomaten auf:

- **Ignorieren bzw. Gleichschalten von a priori Wissen**

  Extensionale Ansätze berechnen lediglich auf Grundlage der Abweichungen vom letzten Wert der Gewissheit den Wert der zukünftigen Gewissheit. Insbesondere bei durch Sensor-Inputs induzierten Evidenzen führt dies dazu, dass a priori Wissen initial gleichgesetzt und unabhängig vorheriger Kenntnis- und Berechnungsstände festgelegt werden muss.

- **Unfähigkeit zur Abbildung abduktiver und prädiktiver Schlüsse**

  Aufgrund des Modularitäts-Prinzips führen Zwei-Wege-Inferenzen innerhalb extensionaler Ansätze zu Endlosschleifen in der Berechnung, womit nur eine Berechnungsrichtung (und damit lediglich ein deduktiver, diagnostischer Schluss) umgesetzt werden kann.

3.2 Intensionale Ansätze

Als Gegenposition zu extensionalen Ansätzen stehen intensionale Ansätze ebenfalls für eine Menge unterschiedlicher Ausprägungen. Im Gegensatz ist hierbei jedoch festzustellen, dass intensionale Ansätze weniger konkrete Umsetzungen wie im Fall von MYCIN, sondern vielmehr formale Ansätze zur Bildung von Schlüssen, ähnlich dem prädikatenlogischen Kalkül darstellen. Da auch hier eine Diskussion unterschiedlicher Vertreter für die weitere Diskussion aufgrund der zu erwartenden Ähnlichkeit nicht zielführend ist, wird im Folgenden erneut wieder der Schwerpunkt auf einen Stellvertreter gelegt, von dessen fachlicher und funktioneller Beschreibung ausgehend ein intensionaler Führungsautomat als Beispiel gegeben werden kann.

Intensionale Ansätze (von lat. *intensio*: Anspannung, Spannung) ermitteln, anders als ihr extensionales Pendant, nicht die Auswirkungen einer durch Evidenzen ausgelösten Änderung in der Gewissheit des Zutreffens eines spezifischen Systemzustandes. Vielmehr untersuchen diese, mit welchen Maß eine vorliegende Evidenz einen von endlich vielen Systemzuständen abzudecken vermag. Eine im allgemeinen Sprachgebrauch hierfür sehr natürlich anmutende Interpretation stellen Wahrscheinlichkeiten dar. So lässt sich die Abdeckung einer Menge M von Systemzuständen durch eine Evidenz $\mathfrak{e} \subset M$ insofern deuten, dass diese die Menge zu x % abdeckt und damit im Verhältnis $\frac{x}{|M|}$ die in M enthaltenen Zustände erfasst bzw. in x von $|M|$ Fällen zutrifft. Ein solches Maß zur Abdeckung von Systemzuständen erfüllt dann die Voraussetzungen eines Wahrscheinlichkeitsfeldes, wie sich gemäß der Axiome von Kolmogoroff (1933, 2) verdeutlichen lässt:

- Sei E die Menge aller elementaren Zustandsräume $e_1, \ldots, e_n$ eines Systems und $\mathfrak{F}$ eine Menge von Teilmengen von E, die mit Hilfe der (Selbst-)Wahrnehmung eines Systems zu jeder Zeit ermittelt werden können. Dann ist ein Systemzustand ein Element aus $\mathfrak{F}$. In diesem Zusammenhang gilt für zwei unabhängige Systemzustände A und B, dass sowohl die Vereinigung $(A \cup B)$, der Durchschnitt $(A \cap B)$, als auch die Differenz $(A \backslash B)$ ein Element von $\mathfrak{F}$ darstellen. (1. Axiom)
- Da $\mathfrak{F}$ alle Teilmengen von E umfasst, muss auch $E \in \mathfrak{F}$ sein. (2. Axiom)
- Jeder Menge $A \in \mathfrak{F}$ kann eine nichtnegative reelle Zahl $P(A)$ zugeordnet werden, die aufgrund der Eintrittswahrscheinlichkeit von A bestimmt wird. (3. Axiom)
- $P(E) = 1$, da die Einnahme des gesamten Zustandsraumes ein sicheres Ereignis darstellt. (4. Axiom)
- Sind zwei vorliegende Systemzustände disjunkt, so gilt $P(A \cup B) = P(A) + P(B)$. (5. Axiom)

Dies wirft die Frage auf, inwieweit dieser Umstand zur Generierung logischer Wissensbasen beiträgt. In diesem Zusammenhang ist es erwähnenswert, dass sich formal gesehen die im Wahrscheinlichkeitsfeld als Mengen definierten Systemzustände auch als Aussagen beschreiben lassen. Ermöglicht wird dies, indem die durch das Feld beschriebenen Grundlagen innerhalb der Sätze eines mathematischen Modells interpretiert werden. Dies wird auch als probabilistisches Modell bezeichnet:

> *„A probabilistic model is an encoding of information that permits us to compute the probability of every well-formed sentence S in accordance with the axioms* [...] [Die Kolmogoroff-Axiome, Anm. d. Verf.]. *Starting with a set of atomic propositions A, B, C,..., the set of well-formed sentences consists of all*

> *Boolean formulas involving these propositions* [...]. *The traditional method of specifying probabilistic models employs a joint distribution function, which is a function that assigns nonnegative weights to every elementary event in the language (an elementary event being a conjunction in which every atomic proposition or its negation appears once) such that the sum of the weights adds up to 1.*“
> (Pearl, 2009, 6)

Die Wahrscheinlichkeit des Systems, sich innerhalb eines als Aussage A formalisierten Systemzustandes zu befinden, lässt sich daher als die Summe der Eintrittswahrscheinlichkeiten der korrespondierenden elementaren Aussagen $\phi \in \Phi$ darstellen: $P(A) = \sum_{\phi \in A} P(\phi)$ mit $P(\Phi) = \sum_{\phi \in \Phi} P(\phi) = 1$. Somit muss von der bereits bekannten Notation der Zustandsräume als aussagenlogische Formeln nicht abgewichen werden. Aus praktischen Gründen werden in diesem Zusammenhang Aussagen jedoch oftmals mit (Zufalls-)Variablen gleichgestellt, deren Ausprägung gleichbedeutend mit der Belegung elementarer Aussagen im aussagenlogischen Kalkül ist. Im Folgenden wird daher – zur Vermeidung möglicher Verwirrung im Gebrauch der Begriffe der Wahrscheinlichkeitstheorie – der Begriff der Variablen stets gleichbedeutend zu einer Aussage verwendet werden.

Intensionale Ansätze können somit zusammenfassend auf Basis wahrscheinlichkeitstheoretischer Paradigmen umgesetzt und mit Hilfe der Formalismen des aussagenlogischen Kalküls beschrieben werden. Der Vollständigkeit halber sei an dieser Stelle jedoch auch erwähnt, dass intensionale Ansätze nicht nur auf Basis der Wahrscheinlichkeitstheorie umgesetzt werden können. Andere Umsetzungsmöglichkeiten stellen die Fuzzy-Logik oder der Dempster-Shafer Kalkül dar (Russell und Norvig, 2010, 547). Innerhalb dieser Arbeit erfolgt jedoch lediglich eine Erläuterung intensionaler Ansätze auf Basis probabilistischer Umsetzungen, da diese nicht nur die älteste Realisierungsform darstellen, sondern auch in die anderen Möglichkeiten überführt werden können respektive gleichwertige Rückschlüsse hinsichtlich der Funktionsweise und Grenzen erlauben. Um mit letzterem beginnen zu können, ist jedoch eine vorangestellte Motivation der benötigten Begriffe der Wahrscheinlichkeitstheorie[12] notwendig, welche nun folgend aufgeführt wird.

Wie es bereits mehrfach in den vorhergehenden Kapiteln Erwähnung fand, bilden hauptsächlich logische Schlüsse unter Nutzung spezifischer Inferenzregeln die Grundlage einer Wissensbasis. Als ein wichtiger Mechanismus zur Einbindung derartiger Schlüsse innerhalb von Expertensystemen erweisen sich axiomatisch formulierte Regeln, welche u. a. mittels einfacher Wenn-Dann-Beziehungen das vorhandene Wissen abbilden. Primärer Träger dieser Beziehungen ist in wissensbasierten (Experten-) Systemen vorrangig die logische Implikation $A \Rightarrow B$, welche vordergründig einen deduktiven Schluss von der Ursache A auf die Wirkung B beschreibt. Wie es innerhalb der Ausführungen zu extensionalen Ansätzen zusätzlich beschrieben wurde, können derartige Schlüsse auch als diagnostischer Schluss von der Wahrnehmung einer konkreten Evidenz $\mathfrak{e}$ auf eine Variable A, d. h. $\mathfrak{e} \Rightarrow A$ beschrieben werden. Innerhalb der Wahrscheinlichkeitstheorie ist die der Implikation am nächsten stehende Interpretation dabei die bedingte Wahrscheinlichkeit $P(A|\mathfrak{e})$, die als Maß des Zutreffens von A unter gegebenem Wissen $\mathfrak{e}$ übersetzt werden kann. Damit weisen proba-

12 An dieser Stelle wird nur auf das für die weitere Diskussion benötigte Wissen eingegangen werden. Eine umfängliche Einführung der Notation der Wahrscheinlichkeitstheorie und die Beschreibung der inhärenten Zusammenhänge ist Kolmogoroff (1933) zu entnehmen. Die folgende Motivation ist zu großen Teilen aus (Korb und Nicholson, 2004; Pearl, 2009; Russell und Norvig, 2010; Neapolitan, 2012) gefolgert. In diesem Zusammenhang wird zur Vereinfachung der Notation auch primär davon ausgegangen werden, dass eine diskrete Verteilung der Wahrscheinlichkeiten vorliegt.

bilistische Umsetzungen zwar keine syntaktische, dafür aber semantische Umsetzung der Aussagenlogik auf. Innerhalb der Anwendung der Wahrscheinlichkeitstheorie sollte hierbei jedoch zwingend die Relation zwischen Wissen und Evidenz beachtet werden.

So heißt die Wahrscheinlichkeit $P(A|\mathfrak{e})$ einer Variable A bei einer vorliegenden Evidenz $\mathfrak{e}$ unbedingte oder **a priori Wahrscheinlichkeit**, wenn deren Wahrheitswert entweder unabhängig von $\mathfrak{e}$, oder die Menge möglicher Evidenzen $\Omega_{\mathfrak{e}}$ atomar ist. Hierbei gilt

$$P(A|\mathfrak{e}) \quad = P(A). \qquad \text{(\textit{a priori Wahrscheinlichkeit})}$$

Im gegensätzlichen Fall heißt die Wahrscheinlichkeit $P(A|\mathfrak{e})$ einer Variablen A bei vorliegender Evidenz $\mathfrak{e}$ bedingte oder **a posteriori Wahrscheinlichkeit**, wenn deren Wahrheitswert (stoch.) abhängig von $\mathfrak{e}$ ist. In diesem Fall gilt

$$P(A|\mathfrak{e}) \quad = \frac{P(A \wedge \mathfrak{e})}{P(\mathfrak{e})}. \qquad \text{(\textit{a posteriori Wahrscheinlichkeit})}$$

Problematisch erweist sich in diesem Fall, dass offensichtlich zur Berechnung der bedingten Wahrscheinlichkeit $P(A|\mathfrak{e})$ grundsätzlich auch $P(\mathfrak{e})$ bekannt sein muss. Die Herausforderung besteht hierbei darin, dass eine grundsätzliche Aussage a priori darüber zu tätigen ist, in welchem Maß eine auftretende Evidenz die Gesamtheit aller möglichen Evidenzen abdeckt. Eine Möglichkeit hierbei auftretende pauschale Festlegungen zu umgehen besteht in der Summierung aller Belegungen $a \in A$, die mit der Evidenz $\mathfrak{e}$ unbedingt in Verbindung stehen. Dies wird auch als **Marginalisierung** bezeichnet, aus der die sog. **Marginalwahrscheinlichkeit** folgt:

$$P(\mathfrak{e}) \quad = \sum_{a \in A} P(\mathfrak{e} \wedge a). \qquad \text{(\textit{Marginalwahrscheinlichkeit})}$$

Steht eine Evidenz $\mathfrak{e}$ jedoch in einer bedingten Verbindung zu Belegungen $a \in A$, so muss hier mit Hilfe der bedingten Wahrscheinlichkeiten aufsummiert werden, was auch als **Konditionierung** bezeichnet wird:

$$P(\mathfrak{e}) \quad = \sum_{a \in A} P(\mathfrak{e}|a)P(a). \qquad \text{(\textit{Konditionierung})}$$

Dass dies zur Vereinfachung der Berechnung einer bedingten Wahrscheinlichkeit $P(A|\mathfrak{e})$ sichtlich beiträgt, wird deutlich, wenn diese als

$$P(A|\mathfrak{e}) \quad = \alpha P(A \wedge \mathfrak{e}) \text{ mit } \alpha = (P(\mathfrak{e}))^{-1}$$

dargestellt wird. Der Faktor α fungiert hierbei als normalisierender Faktor und muss a priori nicht bekannt sein, da dieser auf Grundlage der Marginalisierung bzw. Konditionierung berechnet werden kann.

Hieraus lässt sich ein konkretes Schema für Inferenzen innerhalb probabilistischer Umsetzungen ableiten. Sei in diesem Zusammenhang A eine zu folgernde Variable bel. Belegung, $e \in \Omega_{\mathfrak{e}}$ eine aus einer endlichen Menge von möglichen Evidenzen wahrgenommene Teilmenge aller wahrnehmbaren Belegungen und $b \in B$ eine nicht durch e wahrgenommene – aber dennoch prinzipiell wahrnehmbare – Belegung weiterer Variablen B. Dann folgt als **Inferenzregel** für probabilistische Umsetzungen:

$$P(A|\mathfrak{e}) \quad = \alpha P(A \wedge \mathfrak{e}) = \alpha \sum_{b \in B} P(A \wedge \mathfrak{e} \wedge b). \qquad \textit{(Inferenzregel)}$$

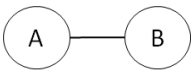
Damit sind die prinzipiellen Grundlagen für logische Schlüsse gegeben. Dies wirft wiederum die Frage auf, wie das für die Inferenz notwendige Wissen syntaktisch nun in einer Wissensbasis hinterlegt werden kann. Um dies näher zu beleuchten, diene im Folgenden als Anschauungsbeispiel nachfolgende Wahrscheinlichkeitsverteilung:

$$\begin{aligned} P(A \wedge B) &= 0.02 \qquad P(\neg A \wedge B) = 0.18 \\ P(A \wedge \neg B) &= 0.25 \qquad P(\neg A \wedge \neg B) = 0.55 \end{aligned}$$

Aus dieser lassen sich exemplarisch die bedingten Wahrscheinlichkeiten $P(A|B)$ und $P(B|A)$ ableiten:

$$\begin{aligned} P(A|B) = \alpha P(A \wedge B) \quad &= \Big(\sum_{a \in \{A, \neg A\}} P(B \wedge a)\Big)^{-1} P(A \wedge B) \\ &= \frac{0.02}{0.02 + 0.18} = 0.1 \\ P(B|A) = \alpha P(B \wedge A) \quad &= \Big(\sum_{b \in \{B, \neg B\}} P(A \wedge b)\Big)^{-1} P(B \wedge A) \\ &= \frac{0.02}{0.02 + 0.25} = 0.07 \end{aligned}$$

Dies verweist auf eine Abhängigkeitsbeziehung zwischen den Variablen A und B, die, gegeben einer vorherrschenden Evidenz $\mathfrak{e}$ für einen von vier sich gegenseitig ausschließenden Zuständen $\mathfrak{e} \in \{A, \neg A, B, \neg B\}$ in Verbindung mit der aufgeführten Verteilung, ausreichende Informationen für Schlüsse auf Basis der Inferenzregel bietet. Formal lassen sich derartige Abhängigkeitsbeziehung u. a. anhand eines (ungerichteten) Graphen[13] abbilden, woraus im gegebenen Fall bspw. die Abhängigkeit zwischen den Variablen A und B als



dargestellt werden kann. Dies weist einen Unterschied zu extensionalen Ansätzen auf, in welchen vorrangig gerichtete Graphen zur formalen Darstellung der Implikationsbeziehungen Verwendung fanden – vgl. hierzu Anfang Kapitel 3.1. Dieser Umstand ist großen Teilen der Loslösung von der klassischen Logik zuzuschreiben, wie es auch bspw. durch Neapolitan (2012, 126) entsprechend festgestellt wird:

> „*In a rule which represents a pure logical relationship, there is only one choice for the direction of implication in a rule. However, in a rule which represents uncertainty, the direction of implication can go in either direction.* [...] *The rules are not meant to represent a causal relationship between the assertions, but rather that the premise is evidence for the conclusion.* [...] *Thus, in rule-based systems which reason under uncertainty, we are free to choose the direction of implication in a rule.*“

Wie die bedingten Wahrscheinlichkeiten des Beispiels aufzeigen, herrschen auf Basis der Wahrscheinlichkeitsverteilung Implikationen von A auf B – und umgekehrt – vor. Diese Biimplikation kann dabei einerseits durch zwei gerichtete Kanten in entgegengesetzter Richtung oder mittels einer ungerichteten Kante dargestellt werden. Da sich durch die Nutzung ungerichteter Kanten somit die Graphen – ohne Einbußen in deren Aussagekraft – übersichtlicher darstellen lassen, wird nun bis auf Weiteres auf gerichtete Kanten verzichtet.

13 Eine umfassende fachliche Herleitung ist (Pearl, 1988, 82ff) zu entnehmen.

In Quintessenz bedeutet die durch eine Kante dargestellte Abhängigkeit, dass im Falle einer in der Wahrscheinlichkeitsverteilung nachweisbaren Verbindung zweier Variablen, diese Variablen als adjazente Ecken $v, v' \in V$ eines ungerichteten Graphen $G = (V, E)$ aufgefasst werden können. Aus graphentheoretischer Sicht bilden diese damit eine sog. Clique:

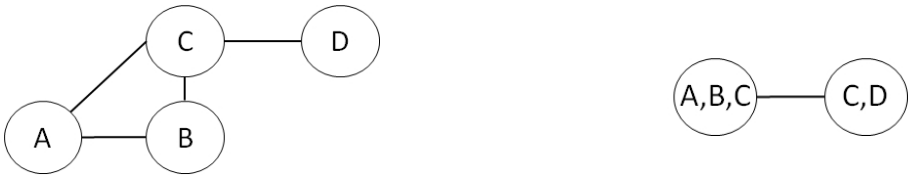
> „*Eine Eckenteilmenge $C \subset V$ heißt **Clique** (von G oder in G), wenn je zwei verschiedene Ecken $v, v' \in C$ adjazent sind, also $[v, v'] \in E$. Wir nennen eine Clique eine **maximale Clique**, wenn sie keine echte Teilmenge einer größeren Clique ist.*“(Krumke und Noltemeier, 2012, 56)
> [Hervorhebungen durch den Verfasser]

Zur vereinfachten Darstellung – und leichteren Berechnung – der Wahrscheinlichkeitsverteilungen probabilistischer Wissensbasen ist es dabei üblich vorauszusetzen, dass die innerhalb eines Graphen entstehenden Cliquen maximal aus drei adjazenten Kanten zusammengesetzt sind. Solche Graphen werden auch als triangulierte oder chordale Graphen bezeichnet:„*A graph G is triangulated if for every cycle $\mu = [p_1, \ldots, p_n, p_1]$ of length $l > 3$ there is an edge of G joining two noncosecutive vertices of μ; such edges are called chords of the cycle.*“(Berge, 1967, 158) Eine Folge hieraus ist, dass innerhalb eines chordalen Graphen $G = (V, E)$ zwei nicht adjazente Ecken $a, b \in E$ minimal durch eine Clique getrennt werden (Rose, 1970, 601). Dies erlaubt die Dekomposition der Wahrscheinlichkeitsverteilung eines probabilistischen Modells auf Basis der Cliquen des zugehörigen Graphs, wobei gem. Pearl (1988, 115) gilt:

$$\begin{aligned} P(x_1 \wedge x_2 \wedge \ldots \wedge x_n) &= \prod_i P(C_i | C_1 \wedge \ldots \wedge C_{i-1}) \\ &= \prod_i P(C_i | C_{j(i)}) \\ &= \prod_i P(C_i | C_i \cap C_{j(i)}) \\ &= \prod_i \frac{P(C_i)}{P(C_i \cap C_{j(i)})} \end{aligned}$$

Verbal ausgedrückt bedeutet dies, dass die Wahrscheinlichkeitsverteilung eines probabilistischen Modells P das Produkt der Verteilungen der Cliquen C_i des zugehörigen Graphen G, geteilt durch das Produkt der Verteilungen des Schnittes aller adjazent nachgeordneten Cliquen C_i und $C_{j(i)}$ ist[14]. Für beliebige Inferenzen innerhalb eines probabilistischen Modells kann daher einerseits die benötigte Wahrscheinlichkeitsverteilung auf Basis der Cliquen eines chordalen Graphen ermittelt, andererseits auf Grundlage einer gegebenen Wahrscheinlichkeitsverteilung der entsprechende triangulierte Graph gefolgert werden. Zur besseren Ersichtlichkeit dieses Umstands diene hierzu das Beispiel in Abb. 3.5.

14 Grundlage stellen hier sog. Verbundbäume (engl. Join Trees) dar. Pearl (1988) verwendet zur Bestimmung dieser Bäume eine Cliquenordnung, die von einer auf die Ecken des Graphen berechneten Ordnung abgeleitet werden kann. Letzteres wird in der Praxis u. a. mit dem LEX-BFS-Algorithmus [siehe (Krumke und Noltemeier, 2012, 163)] bzw. dem MCS-Algorithmus [siehe (Berry et al., 2002, 4)] ermittelt. Dies erlaubt es Cliquen hierarchisch gem. der höchsten ihnen zugeordneten Ordnungzahl einer Ecke zu anzuordnen. $C_i \cap C_{j(i)}$ bildet dann für alle i < j die Schnitte aller Cliquen eines Graphen.



(a) **(b)**

Abbildung 3.5: Beispielgraph (a) und dessen zugehöriger Cliquengraph (b)

Wie es hier ersichtlich ist, besitzt der Graph aufgrund der vorgegebenen Struktur zwei adjazente Cliquen $C_1 = \{A, B, C\}$ und $C_2 = \{C, D\}$, deren gemeinsamer Schnitt $C_1 \cap C_2 = \{C\}$ darstellt. Aufgrund der oben gegebenen Formel lässt sich hieraus die Wahrscheinlichkeitsverteilung

$$P(A \wedge B \wedge C \wedge D) \quad = \prod_i \frac{P(C_i)}{P(C_i \cap C_{j(i)})} = \frac{P(A \wedge B \wedge C)P(C \wedge D)}{P(C)}$$

ermitteln. Da sich $P(C)$ per Marginalisierung berechnen lässt, werden explizit die Wahrscheinlichkeitsverteilungen $P(A \wedge B \wedge C)$ und $P(C \wedge D)$ zur Berechnung des Gesamtgraphen benötigt. Diese seien dabei beispielhaft wie folgt gewählt:

$$\begin{array}{rrrr}
P(A \wedge B \wedge C) = 0.20 & P(A \wedge B \wedge \neg C) = 0.15 & P(A \wedge \neg B \wedge C) = 0.19 & P(A \wedge \neg B \wedge \neg C) = 0.30 \\
P(\neg A \wedge B \wedge C) = 0.01 & P(\neg A \wedge B \wedge \neg C) = 0.08 & P(\neg A \wedge \neg B \wedge C) = 0.02 & P(\neg A \wedge \neg B \wedge \neg C) = 0.05 \\
P(C \wedge D) = 0.35 & P(C \wedge \neg D) = 0.07 & P(\neg C \wedge D) = 0.08 & P(\neg C \wedge \neg D) = 0.50
\end{array}$$

Herrscht nun z. B. eine Evidenz für D vor, lässt sich auf Grundlage der Inferenzregel die Wahrscheinlichkeit für die von D indirekt abhängigen Variablen A und B wie folgt berechnen:

$$\begin{aligned}
P(A \wedge B|D) \quad &= \alpha \sum_c P(A \wedge B \wedge c \wedge D) \\
&= \alpha\left(\frac{P(A \wedge B \wedge C)P(C \wedge D)}{P(C)} + \frac{P(A \wedge B \wedge \neg C)P(\neg C \wedge D)}{P(\neg C)}\right) \\
&= \alpha\left(\frac{P(A \wedge B \wedge C)P(C \wedge D)}{\sum_d P(C, d)} + \frac{P(A \wedge B \wedge \neg C)P(\neg C \wedge D)}{\sum_d P(\neg C, d)}\right) \\
&= \alpha\left(\frac{0.07}{0.42} + \frac{0.01}{0.58}\right) = 0.43
\end{aligned}$$

Chordale ungerichtete Graphen stellen daher in Verbindung mit einer hintergründigen, korrespondierenden Wahrscheinlichkeitsverteilung einen Anhaltspunkt für beliebige Inferenzen über den beschriebenen Raum der Variablen dar. Rekapitulierend bietet dies wiederum die Grundvoraussetzungen für logische Schlüsse und damit auch eine geeignete Grundlage für Wissensbasen. Dieser Umstand wird nun als Ausgangspunkt des nächsten Kapitels angesehen, in welchen mit Hilfe eines probabilistischen Modells ein Beispiel für intensionale Führungsautomaten erarbeitet wird.

3.2.1 Beispiel eines Führungsautomaten

Zur besseren Vergleichbarkeit wird auf das bereits bekannte Lagebild aus Kapitel 2.3 zurückgegriffen, welches für die Betrachtung extensionaler Ansätze in Abb. 3.1 entsprechend mit Deckungsinformationen erweitert wurde. Als Operationsart des blauen Jägerzugs diene erneut der Angriff, die rote Infanteriegruppe

befindet sich in der Verteidigung. Zur Vereinfachung wird zum wiederholten Male nur das Verhalten des Jägerzuges beschrieben. Folgend wird der deklarative Anteil einer hybriden Wissensbasis aufgestellt. Der korrespondierende prozedurale Anteil findet keine Erwähnung, da sich dieser lediglich auf das Ergebnis des deklarativen Anteils – und damit auf den aufgezeigten intensionalen Ansatz – abstützt und daher keine Aussage zur Funktionsweise intensionaler Ansätze birgt. Zur Lagebeurteilung gelten erneut wieder die in Kap. 3.1.1 identifizierten Faktoren, die auf Basis der Deckung, der beobachteten (relativen) Bewegungen, des jeweiligen Kräfteansatzes und der erwarteten Waffenwirkung im Ziel getroffen wird. Abermals werden diese in vier logische Ausdrücke A, B, C und D unterteilt, die entsprechend als Variablen des probabilistischen Modells angesehen werden. Die jeweiligen Ausprägungen der Variablen seien dabei wie folgt:

- **[A]** Bewegung $\rightarrow$ [$\mathbf{a_1}$] Keine | [$\mathbf{a_2}$] Front | [$\mathbf{a_3}$] Flanke | [$\mathbf{a_4}$] Rücken
- **[B]** Waffenwirkung $\rightarrow$ [$\mathbf{b_1}$] Keine | [$\mathbf{b_2}$] Gering | [$\mathbf{b_3}$] Gut
- **[C]** Deckung $\rightarrow$ [$\mathbf{c_1}$] Ungedeckt | [$\mathbf{c_2}$] Teilgedeckt | [$\mathbf{c_3}$] Gedeckt
- **[D]** Kräfteansatz $\rightarrow$ [$\mathbf{d_1}$] Unterlegen | [$\mathbf{d_2}$] Ausgeglichen | [$\mathbf{d_3}$] Überlegen

Als erster Gegenstand zur Lagebeurteilung stelle erneut wieder die Bewertung der eigenen bzw. feindlichen Deckungsgüte dar. Diese wird wieder auf Grundlage der eigenen bzw. feindlichen Bewegung und der eigenen bzw. feindlichen Deckung abgeleitet. Hierbei gelte A als Variable für die feindlichen Bewegung, A' für die eigene Bewegung, C für die feindliche Deckung und C' für die eigene Deckung. Die entsprechende Beurteilung der Stellung sei für die rote Infanteriegruppe als Variable E, für den blauen Jägerzug als E' festgelegt. Graphisch werden diese gegenseitige Wechselwirkung von A, C und E bzw. von A', C' und E' zu

gefolgert. Die Variablen E und E′ besitzen dabei folgende Ausprägungen:

[E] Stellungsgüte $\rightarrow$ [$\mathbf{e_1}$] Vorteilhaft | [$\mathbf{e_2}$] Nicht vorteilhaft

Die generelle zu erwartende Erfolgsrate sei auf Grundlage der erwarteten Waffenwirkung im Ziel und des Kräfteansatzes im Vergleich bemessen. Die Variable B beschreibe hierbei die erwartete Waffenwirkung der blauen Teile, B' die erwartete Waffenwirkung der roten Teile, D das Ergebnis des Kräftevergleichs von blauer Seite und D' von roter Seite. Die Erfolgsrate sei dabei für den blauen Jägerzug mittels der Variablen F, für die rote Infateriegruppe mittels F' ausgedrückt. Die hieraus entstehenden biimplikativen Beziehungen von B, D und F respektive von B', D' und F' können in Graphenform durch

beschrieben werden. Folgende Ausprägungen seien dabei für F und F' festgelegt:

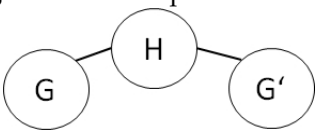
[F] Erfolgsrate $\rightarrow$ [$\mathbf{f_1}$] Hoch | [$\mathbf{f_2}$] Niedrig

Aus der Stellungsgüte des jeweiligen Gegenspielers und der Erfolgsrate in der gegenwärtigen Situation gehe wieder das Risiko der jeweiligen Partei hervor. Dabei stehe die Variable G für das resultierende Risiko des Jägerzuges und G' für das Risiko der roten Infanteriegruppe. Die hierbei beschriebenen Abhängigkeiten von E, F und G bzw. E', F' und G' können graphisch mittels

abgebildet werden. In diesem Zusammenhang erfolgt folgende Festlegung für die Ausprägung der Variablen G und G':

[G] Risiko $\rightarrow$ [$\mathbf{g_1}$] Niedrig | [$\mathbf{g_2}$] Mittel | [$\mathbf{g_3}$] Hoch

Die jeweilige Bewertung des Risikos fließe auch hier erneut wieder in eine Form des Einsatzwertvergleichs, wobei sowohl das eigene Risiko als auch das feindliche Risiko betrachtet wird. Dies findet konkrete Ausprägung in der Variable H, wobei die aufgeführte Abhängigkeitsbeziehung zwischen G und H bzw. zwischen G' und H innerhalb eines ungerichteten Graphen durch



zu folgern ist. Getreu der Ausführungen aus Kap. 3.1.1 besitze die Variable H dabei folgende Ausprägungen:

[H] Einsatzwert $\rightarrow$ [$\mathbf{h_1}$] Überlegen | [$\mathbf{h_2}$] Ausgeglichen | [$\mathbf{h_3}$] Unterlegen

In Summe ist damit festzuhalten, dass die beschriebenen Abhängigkeiten einen Variablenraum mit 15 Variablen beschreiben. Der Gesamtgraph der dargestellten Beziehungen kann hierzu Abb. 3.6 entnommen werden.

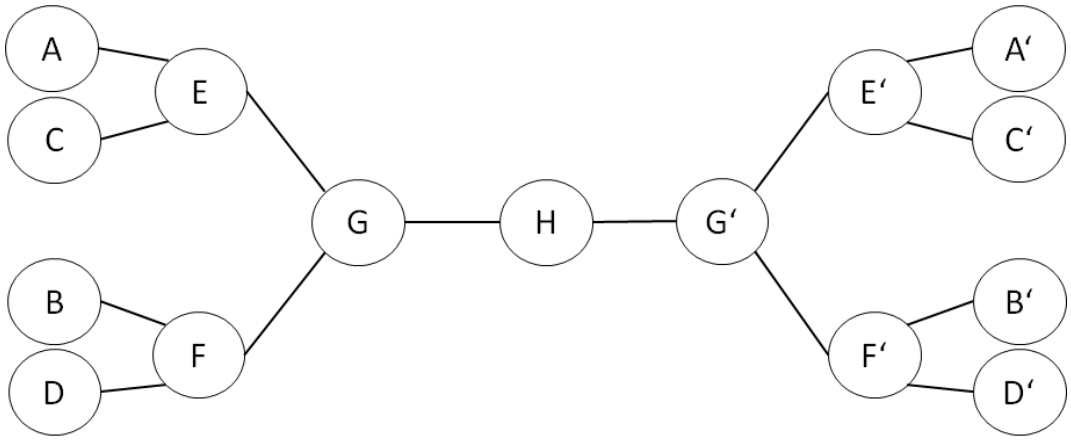


Abbildung 3.6: Gesamtgraph der intensionalen deklarativen Wissensbasis

Ohne zugeordnete Wahrscheinlichkeitsverteilung stellt dieser Graph jedoch vorläufig nur ein Skelett der intendierten intensionalen deklarativen Wissensbasis dar. Da eine direkte Ermittlung der Größe und Anzahl der vorherrschenden Abhängigkeiten nicht direkt ableitbar ist, ist es hierbei sinnvoll dem Graphen zuerst eine Eliminationsordnung zuzuweisen. Hierdurch kann schließlich die zur algorithmischen Ermittlung der Wahrscheinlichkeitsverteilung benötigte Cliquenordnung bestimmt werden. Zur Berechnung der Eliminationsordnung wird der MCS-Algorithmus (vgl. Anhang A.1) verwendet. Als erste gewählte Ecke im MCS-Algorithmus diene die Variable H, Tab. 3.2 ist dabei die hierdurch resultierende Eliminationsordnung zu entnehmen. Eine Herleitung dieses Ergebnisses ist in Anhang A.1 zu finden.

Var v	$\alpha(v)$
H	15
G	14
E	13
A	12
C	11
F	10
B	9
D	8

Var v	$\alpha(v)$
G′	7
E′	6
A′	5
C′	4
F′	3
B′	2
D′	1

Clique c	$\alpha(c)$
(G,H)	14
(G′,H)	13
(E,G)	12
(F,G)	11
(A,E)	10
(C,E)	9
(B,F)	8

Clique c	$\alpha(c)$
(D,F)	7
(E′,G′)	6
(F′,G′)	5
(A′,E′)	4
(C′,E′)	3
(B′,F′)	2
(D′,F′)	1

Tabelle 3.2: Eliminations- und Cliquenordnung des Beispielgraphen

Zusätzlich zeigt Tab. 3.2 die Zuordnung der Ecken in die jeweiligen Cliquen, was in Verbindung mit der durch die Eliminationsordnung bestimmten Kardinalitätszahlen zur dort aufgeführten Ordnung der Cliquen führt. Wie es im letzten Unterkapitel beschrieben wurde, lässt sich hierauf aufbauend und mit Hilfe des Algorithmus nach (Pearl, 1988, 115) die Wahrscheinlichkeitsverteilung des Graphen ermitteln:

$$P(A \wedge \ldots \wedge G \wedge H \wedge G' \wedge \ldots \wedge A') = \frac{P(A \wedge E)P(C \wedge E)P(B \wedge F)P(D \wedge F)P(E \wedge G)P(F \wedge G)}{P(E)P(E)P(F)P(F)P(G)} \cdot \frac{P(F' \wedge G')P(E' \wedge G')P(D' \wedge F')P(B' \wedge F')P(C' \wedge E')P(A' \wedge E')}{P(G')P(F')P(F')P(E')P(E')} \cdot \frac{P(G \wedge H)P(G' \wedge H)}{P(G)P(H)P(G')}$$

Den Ausführungen des letzten Kapitels folgend, wird für nun folgende Berechnungen noch eine Aufstellung der Wahrscheinlichkeiten der jeweiligen Ausprägungen benötigt. Hierzu ist beispielhaft Tab. 3.3 eine subjektive Festlegung zu entnehmen.

$\wedge$	a_1	a_2	a_3	a_4
e_1	0.30	0.025	0.075	0.10
e_2	0.00	0.25	0.15	0.10

$\wedge$	c_1	c_2	c_3
e_1	0.05	0.15	0.30
e_2	0.30	0.15	0.05

$\wedge$	b_1	b_2	b_3
f_1	0.00	0.25	0.35
f_2	0.30	0.10	0

$\wedge$	a'_1	a'_2	a'_3	a'_4
e'_1	0.00	0.08	0.20	0.20
e'_2	0.20	0.30	0.02	0.00

$\wedge$	c'_1	c'_2	c'_3
e'_1	0.00	0.18	0.30
e'_2	0.25	0.15	0.12

$\wedge$	b'_1	b'_2	b'_3
f'_1	0.02	0.07	0.46
f'_2	0.30	0.14	0.01

$\wedge$	d_1	d_2	d_3
f_1	0.05	0.20	0.35
f_2	0.25	0.10	0.05

$\wedge$	g_1	g_2	g_3
e_1	0.20	0.25	0.05
e_2	0.01	0.10	0.39

$\wedge$	g_1	g_2	g_3
f_1	0.21	0.30	0.09
f_2	0.00	0.05	0.35

$\wedge$	d'_1	d'_2	d'_3
f'_1	0.01	0.24	0.30
f'_2	0.20	0.20	0.05

$\wedge$	g'_1	g'_2	g'_3
e'_1	0.06	0.18	0.24
e'_2	0.38	0.10	0.04

$\wedge$	g'_1	g'_2	g'_3
f'_1	0.40	0.14	0.01
f'_2	0.04	0.14	0.27

$\wedge$	g_1	g_2	g_3
h_1	0.15	0.04	0.01
h_2	0.06	0.25	0.20
h_3	0.00	0.06	0.23

$\wedge$	g'_1	g'_2	g'_3
h_1	0.00	0.05	0.15
h_2	0.20	0.19	0.12
h_3	0.24	0.04	0.01

Tabelle 3.3: Wahrscheinlichkeitsverteilungen des intensionalen Führungsautomaten

Damit ist die deklarative Wissensbasis des intensionalen Führungsautomaten funktionsfähig. Wird beispielsweise simultan zum Beispiel aus Kapitel 3.1.1, Abb. 3.3 eine vorherrschende Evidenz $\mathfrak{e} = (a_1 \wedge b_3 \wedge c_3 \wedge d_3 \wedge a'_2 \wedge b'_3 \wedge c'_2 \wedge d'_1)$ angenommen, so lassen sich mit Hilfe der Inferenzregel die Wahrscheinlichkeiten

$$\begin{aligned} P(h_1|\mathfrak{e}) &= \alpha P(h_1 \wedge \mathfrak{e}) = 0.27, \\ P(h_2|\mathfrak{e}) &= \alpha P(h_2 \wedge \mathfrak{e}) = \mathbf{0.57}, \\ P(h_3|\mathfrak{e}) &= \alpha P(h_3 \wedge \mathfrak{e}) = 0.15 \end{aligned}$$

berechnen. Hierdurch kommt die intensionale deklarative Wissensbasis unter Vorhandensein derselben Evidenz auch zum selben Schluss wie im Beispielfall der extensionalen deklarativen Wissensbasis: In der gegenwärtig dargestellten Situation stehen sich der blaue Jägerzug und die rote Infanteriegruppe am ehesten ausgeglichen gegenüber. Dieser manuell durchgeführte Schluss dient nun als Basis für die funktionale Beschreibung des Führungsautomaten in Abb. 3.7, dessen deklarative Wissensbasis befähigt ist, derartige Berechnungen durchzuführen. In einem ersten Schritt ermittelt der Automat zur Bestimmung beliebiger a posteriori Wahrscheinlichkeiten $P(h_x|\mathfrak{e})$ die a priori Wahrscheinlichkeiten $P(h_x \wedge \mathfrak{e})$ und $P(\mathfrak{e})$. Die Ermittlung der a priori Wahrscheinlichkeiten erfolgt hierbei mit Hilfe der Marginalisierung (*marginal*) über der gegebenen Wahrscheinlichkeitsverteilung des Graphen. Sowohl Verteilung (*distr*), als auch konkrete Wahrscheinlichkeitswerte (*map*) befinden sich in der Wissensbasis (*KB*) des Automaten, wobei eine Abfrage (*queryPerc*) auf Grundlage der momentan vorliegenden Wahrnehmung (*CPerc*) denjenigen Zustand $h \in \{h_1, h_2, h_3\}$ ausgibt, der die höchste Wahrscheinlichkeit zum Zeitpunkt der Abfrage besitzt.

Wie es auch bereits in Kapitel 3.1.1 zuletzt aufgeführt wurde, kann zu diesem Zeitpunkt zusätzlich eine Abfrage (*checkAlts*) eingebaut werden, die dem Automaten einen potenziell günstigeren Zielzustand vorgibt. Diese Abfrage vergleicht dabei, ob die Veränderung der blauen Angriffsrichtung – welche de facto durch Erhöhung des Index der hierarchisch organisierten Variable A' vorgegeben wird – zu einer signifikanten Lageänderung, d. h. einem höherwertigen Einsatzwert führt. Die Veränderung von A' wird hierbei in Form einer neuen Evidenz $\mathfrak{e}'$ einem potenziellen Zielzustand (*goal*) hinzugefügt und das Auskommen dieser Änderung mit dem des bisherigen Zustands (*state*) verglichen. Im gegebenen Beispielfall hieße dies, dass die vorliegende Evidenz $\mathfrak{e}$ zu $\mathfrak{e}' = (a_1 \wedge b_3 \wedge c_3 \wedge d_3 \wedge \mathbf{a'_3} \wedge b'_3 \wedge c'_2 \wedge d'_1)$ verändert wird. Dies führt zu den Wahrscheinlichkeiten

$$\begin{aligned} P(h_1|\mathfrak{e}') &= \alpha P(h_1 \wedge \mathfrak{e}') = \mathbf{0.50}, \\ P(h_2|\mathfrak{e}') &= \alpha P(h_2 \wedge \mathfrak{e}') = 0.44, \\ P(h_3|\mathfrak{e}') &= \alpha P(h_3 \wedge \mathfrak{e}') = 0.05 \end{aligned}$$

und im Sinne der Definition zu einer signifikanten Lageänderung[15]. Da diese Änderung zu einem potenziell günstigeren Ausgangszustand führt, wird der hierfür ermittelte Zustand in Folge als Zielzustand festgesetzt. Damit ist der Automat gleichermaßen befähigt, sowohl die eigene Lage aufgrund des momentanen Zustands, als auch für ihn möglicherweise günstigere Zielzustände zu ermitteln.

15 Hier „Überlegen" im Vergleich zu „Ausgeglichen". Dies ist taktisch gesehen nicht verwunderlich, da ein Angriff auf die Flanke des Feindes dessen überlegene Position weitestgehend kompensiert.

Programm 7 Intensionaler Führungsautomat

```
KnowledgeBase : KB;
Array of State: seq;
function C2(blue: Entity) returns Action
    State: current, goal;
    Action: next;
    current ← KB.queryPerc(blue.getCPerception());
    if seq == ∅ then
        if current.outcome == (H3 || H2) then
            checkAlts(KB, current);
            goal ← current;
        end if
        seq ← planAttack(blue, goal);
    end if
    next ← KB.queryAct(seq.pop());
    return next
end function
```

Berechne Zustand

```
Formula: distr;
HashMap[Record, Float]: map;
Array of Variable: h;
function QUERYPERC(a: Array of Variable) returns State
    State: state;
    Float: x,y;
    for Integer: i = 0 to a.length do
        state.evidence.append(a[i]);
    end for
    for Integer: j = 0 to h.length do
        y ← marginal(distr,map,a∪h[i]) / marginal(distr,map,a);
        if (x == null || x < y) then
            x ← y;
            state.outcome = h[j];
        end if
    end for
    return state
end function
```

Suche nach Alternativen

```
procedure CHECKALTS(state: State)
    State: goal ← state;
    Variable: cur ← goal.evidence.fetch(„a'“);
    Integer: i ← cur.index;
    Float: x,y;
    if i < 4 then
        goal.evidence.remove(„a'“+ parseString(i));
        goal.evidence.append(„a'“+ parseString(i+1));
        goal ← KB.queryPerc(goal.evidence);
        if goal.outcome == h1 || c(goal.outcome == h2 && state.outcome == h3) then
            state ← goal;
        end if
    end if
end procedure
```

Abbildung 3.7: Intensionaler Ansatz eines hybriden wissensbasierten Automaten

3.2.2 Schwächen des Ansatzes

Da intensionale Ansätze Schlüsse primär auf der Grundlage numerischer Werte bilden, ist es sinnvoll im Vorlauf der Erläuterungen der Schwächen die Herkunft eben jener Werte darzustellen. Absicht ist es hier aufzuzeigen, dass intensionale Ansätze im Gegensatz zu ihrem extensionalen Pendant – aufgrund des numerischen Unterbaus – in ihren konkreten Ausprägungen zu kontroversen Diskussionen führen können. Dies wird als Einleitung verwendet, um im weiteren Verlauf darzustellen, wie die ausschließliche Verwendung schwierig zu evaluierender Zahlen hinderlich für die Bemessung der tatsächlich benötigten Größe der Wissensbasis ist und inwiefern dies einen negativen Einfluss auf Zeit- und Speicherkomplexität für Führungsautomaten hat.

Eine im Zusammenhang mit numerischen Ansätzen stets berechtigte Frage stellt es dar, woher die verwendeten numerischen Werte stammen und inwiefern diese sich im Vergleich zueinander unterscheiden lassen. In der einführenden Darstellung intensionaler Ansätze wurde hierzu bereits eine entsprechende Interpretation aufgeführt: Die relative Häufigkeit des Aufkommens spezifischer Ereignisse. Dies ergibt jedoch nicht immer Sinn, da beispielsweise deren Deutung im Sinne eines Laplace'schen Experiments zumindest hintergründig voraussetzt, dass alle beobachtbaren Ereignisse im Sinne der gedeuteten Häufigkeit gleichverteilt sind[16]. In der Realität ist dies jedoch nicht ohne Weiteres auf alle beobachtbaren Systeme übertragbar, da zumindest aufgrund physikalischer Eigenschaften bestimmte Ereignisse im Vergleich zueinander nachweisbar häufiger bzw. weniger häufiger auftreten können. Aus diesem Grund weichen die Herkunftspositionen numerischer Werte innerhalb intensionaler Ansätze tendenziell von der Sicht eines Laplace'schen Experiments ab, wobei generell zwischen **frequentistischen** und **subjektivisitschen** Positionen unterschieden wird. Da die Positionen zu Diskussionen führen können und insbesondere die Wahl der Position eine entsprechende Relevanz für Anteile dieser Arbeit besitzt, werden diese nun im folgenden näher beschrieben. Hierzu ist es jedoch notwendig, eine Hilfsnotation – frei gefolgert aus den Ausführungen von Neapolitan (2012, 32) – voran zu stellen:

> Es sei Ω eine Menge von experimentell ermittelbarer Systemzustände, $\mathfrak{F}$ eine Menge von Ereignissen, die relativ zu Ω auftreten und P eine Funktion, die jedem Ereignis $E \in \mathfrak{F}$ eine reellwertige Zahl zuweist. Gleichzeitig wird angenommen, dass P die folgenden Axiome erfüllt:
>
> (1) $\forall E \in \mathfrak{F} : 0 \leq P(E) \leq 1$
>
> (2) $P(\Omega) = 1$
>
> (3) Wenn $P(E_1 \cap E_2) = \varnothing$ und $E_1, E_2 \in \mathfrak{F}$, dann ist $P(E_1 \vee E_2) = P(E_1) + P(E_2)$

Im Vorlauf der Darstellungen der beiden Positionen sei ferner noch festgehalten, dass die folgenden Erläuterungen zu weiten Teilen der Grundsatzliteratur probabilistischer Ansätze entnommen wurden. Die aufgeführten Argumente, inklusive der nachfolgenden Deutung zur Herkunft numerischer Werte, ist hierbei

16 Dies wird auch als das **Indifferenzprinzip** bezeichnet, ein Begriff der auf die Ausführungen von Keynes (1948, 44) zurückzuführen ist: *„The Principle of Indifference asserts that if there is no known reason for predicating of our subject one rather than another of several alternatives, then relatively to such knowledge the assertions of each of these alternatives have an equal probability."*

zwar von der wahrscheinlichkeitstheoretischen Perspektive aus formuliert, lässt sich jedoch nahtlos auf andere intensionale Ansätze, d. h. der Fuzzy-Logik oder dem Dempster-Shafer-Kalkül übertragen.

Die **frequentistische Position** (von lat. *frequentis*, Häufigkeit) deutet Zahlen auf der Grundlage gleichartiger und unabhängiger Beobachtungen, die in einem vielfältig wiederholbaren Vorgang nachgewiesen werden können. Ein in diesem Zusammenhang wichtiger Begriff stellen sog. Kollektive dar, wie sie durch R. von Mises (1972, 13) ursprünglich definiert wurden: „*Wir wollen eine Gesamtheit von Vorgängen oder Erscheinungen, die sich im einzelnen durch irgendein beobachtbares Merkmal* [...] *unterscheiden, kurz ein Kollektiv nennen.*“ Per Definition lassen sich damit numerische Werte auf Basis ihrer relativen Häufigkeit innerhalb einer Menge von Ereignissen mit denselben Merkmalen ableiten. Die Größe eines Kollektivs definiert sich hierbei auf Basis einer objektiven Betrachtungsweise des Aufkommens spezifischer Merkmale, wobei die Beobachtung im frequentistischen Sinne unbegrenzt durchführbar sein muss. Im Sinne der Hilfsnotation bedeutet dies also, dass der einem Ereignis E zugewiesene numerische Wert als

$$P(E) \quad = \lim_{n \to \infty} \frac{S^n(E)}{n}$$

zu deuten ist, wobei $S^n(E)$ die ermittelte Anzahl eines beobachtbaren Merkmals darstellt. Aus dieser Deutung folgt, dass aus frequentistischer Sicht heraus nur dann einem Ereignis ein numerischer Wert mit Sicherheit zugewiesen werden kann, wenn dieses einem entsprechend eindeutig abgrenzbaren Kollektiv zugeordnet ist. Aufgrund dieser Tatsache sind numerische Werte an die sorgfältige Definition des hintergründigen Kollektivs gebunden:

> „*Bei vielen Anwendungen des Wahrscheinlichkeitsbegriffes im gewöhnlichen Leben, wenn sie berechtigt sind, kann man das zugehörige Kollektiv konstruieren. Wo dies nicht möglich ist, hat der Gebrauch des Wortes mit der Wahrscheinlichkeitsrechnung nichts zu tun und eine* ***zahlenmäßige Bestimmung*** *der Wahrscheinlichkeit ist* ***nicht möglich****. Andererseits gibt es sicher auch Fälle, in denen die Abgrenzung des* ***Kollektivs in verschiedenster Weise*** *vorgenommen werden kann, und das sind diejenigen, in denen sich über die* ***Größe der betreffenden Wahrscheinlichkeiten streiten*** *läßt.*“(von Mises, 1972, 23f)
> [Hervorhebungen durch den Verfasser]

Aus dem Blickwinkel der **subjektivistischen Position** heraus, stellt die Definition eines Kollektivs jedoch nur ein Hilfsmittel zur Einordnung bestimmter Wertezuweisungen dar. Sie vertritt den Standpunkt, dass die letztendliche Zuweisung von Werten immer mit der Erwartung (eines Experten) einhergeht und damit grundsätzlich nicht objektiv erfolgt. Dies wird bspw. durch die Ausführungen von de Finetti (1981, 106) bestätigt:

> „*Das P stellt die Ansichten eines Individuums angesichts ungewisser Tatsachen dar, wobei jeder Zufallsgröße X die Bewertung entspricht, die er selbst P(X), der Erwartung von X, gibt, deren Bedeutung operativ auf die des (gleichwertigen) Preises von X, als Gewinn ausgedrückt, zurückzuführen ist und die im besonderen die Wahrscheinlichkeit enthält (denn dies ist der spezifischere Name, den man der Erwartung gibt, wenn X ein Ereignis ist).*“

Bestimmende Forderung ist hierbei, dass die gewählte Zuweisung des numerischen Wertes kohärent ist. Kohärenz bedeutet in diesem Sinne, dass es, wenn eine entsprechende Zuweisung P gewählt wird, keine andere

Zuweisung P' gibt, die in jedweder Situation zu besseren Ergebnissen führt. Somit kann streng genommen auch der subjektivistischen Position ein frequentistisches Fundament entnommen werden, obgleich dies sich lediglich auf die objektiv wahrzunehmende Kohärenz bezieht (Neapolitan, 2012, 58). Dass der Nachweis der Kohärenz von P jedoch – ebenso wie die Definition eines Kollektivs – in Führungsautomaten problematisch ist, sei mittels des nachfolgenden Beispiels verdeutlicht.

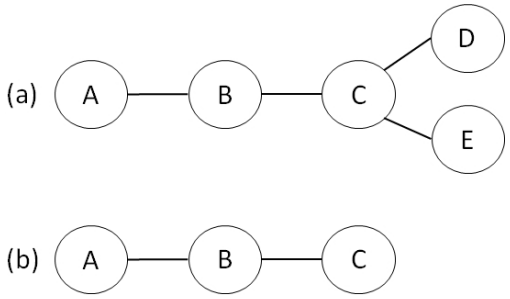
Beispiel Die deklarative Wissensbasis eines Führungsautomaten umfasse u. a. die Bewertung, ob sich ein Infanteriezug auf Basis seines Deckungszustandes in einer für einen Angriff günstigen Ausgangsposition befinde. Hierfür soll gemäß der drei typischen Deckungszustände „*Gedeckt*“, „*Teilgedeckt*“ und „*Ungedeckt*“ eine entsprechende a posteriori Wahrscheinlichkeit für weitere Schlüsse bestimmt werden.

Aus frequentistischer Sicht heraus entbehrt die Bestimmung der zugehörigen Wahrscheinlichkeiten im Beispielfall weiterer Details, da mangels eindeutiger Merkmale für den Deckungszustand kein Kollektiv festgelegt werden kann. Die entsprechende Folgerung hieraus wäre nun, die Bewertung des Zustands durch die Festlegung weiterer Merkmale einzugrenzen. Die Festlegung spezifischerer Merkmale, bspw. dass es sich um einen Gebirgsjägerzug handeln möge, der Feind hinter einer Mauer steht, etc. stellt jedoch insgesamt keine ausreichend objektive Bemessungsgrundlage für die Einteilung in eine homogene Ereignismenge dar. Dies lässt sich dadurch begründen, dass das Ereignis $E =$ „befindet sich in einer günstigen Ausgangsposition“ durch den vorhandenen Informationsstand, z. B. $I_1 =$ „Gebirgsjägerzug“ $\wedge$ „Gedeckt“ singulär und daher eine von ähnlichen Ereignissen disjunkte Teilmenge ist. Somit lässt sich kein eindeutiges Kollektiv bilden, womit aus frequentistischer Position entsprechend eine auf Basis der relativen Häufigkeit dieser vorkommenden Situationen ermittelte Wahrscheinlichkeit anfechtbar ist.

Eine ähnliche Situation ergibt sich aus Sicht der subjektivistischen Position. Die Tatsache, dass es durchaus geländeabhängig ist, ob ein Infanteriezug sich in einer günstigeren Ausgangsposition befindet, führt dazu, dass eine situationsspezifische Kohärenz vorherrscht. Erklärt ist dieser Umstand dadurch, dass etwaige Zuordnungen P nur dann kohärent sind, wenn diese auf einen jeweiligen Infanteriezug und dessen Ausrüstung bzw. Ausbildung zugeschnitten werden. So wäre ein Gebirgsjägerzug im Hochgebirge selbst in teilgedeckter Position gegenüber einem Panzergrenadierzug in jedweder Deckung in einer günstigen Ausgangsposition. Somit ist selbst eine subjektive Festlegung der Wahrscheinlichkeit an dieser Stelle argumentativ anfechtbar, da rein praktisch keine eindeutige und kohärente Eingrenzung der zutreffende Fälle ermittelt werden kann. Dieses Problem lässt sich auf beliebig viele weitere Fälle erweitern - tatsächlich trifft dieses in einem Großteil der im vorherigen Kapitel durch den beispielhaft aufgestellten Führungsautomaten verwendeten Wahrscheinlichkeiten zu. Hierdurch wird das sog. **Bezugsklassenproblem** (engl. *reference class problem*) beschrieben, ein Begriff, der auf die Interpretation dieses Häufigkeitsproblems mit Hilfe von Klassen durch Reichenbach (1994, 382) zurückgeht.

In der Nutzung intensionaler Ansätze ist es anzustreben, möglichst wenige Präzedenzfälle zu generieren, die dem Bezugsklassenproblem unterliegen. Eine hierfür geeignete Methodik wurde im Sinne der frequentistischen Position bereits aufgezeigt: Die Wahl der engsten Bezugsklasse, d. h. die Eingrenzung des Kollektivs durch spezifischere Merkmale. Dies birgt jedoch gewisse Nachteile, namentlich dass bei steigender

Anzahl möglicher Variablen eigentlich unerhebliche Wahrscheinlichkeiten erhoben und rein auf Grundlage der argumentativen Absicherung eingebettet werden. Relevante und irrelevante Informationen decken hierbei denselben Ereignisraum ab, lassen sich jedoch unterschiedlichen Klassen zuordnen. Mathematisch gesehen stellt dies den Schnitt beider Klassen dar[17] und für Ereignisse E_1 und E_2, die jeweils einer der Klassen zugeordnet werden können, gilt rein formal, dass $P(E_1 \cap E_2) \neq \varnothing$ und $P(E_1) = P(E_2)$. Die Abhängigkeiten von Variablen – und daher auch indirekt von Klassen – wurden in den vorangestellten Kapiteln durch Graphen abgebildet, auf die nun zur besseren Ersichtlichkeit dieses Umstandes zurückgegriffen wird. Wie schnell unnötige Merkmale dabei den tatsächlichen Informationsbedarf überdecken können, sei auf Grundlage von Abb. 3.8 nachfolgend erläutert.



$\wedge$	a_1	a_2
b_1	0.5	0
b_2	0.1	0.4

$\wedge$	b_1	b_2
c_1	0.3	0.4
c_2	0.2	0.1

$\wedge$	c_1	c_2
d_1	0.5	0.1
d_2	0.2	0.2

$\wedge$	c_1	c_2
e_1	0.6	0.1
e_2	0.1	0.3

Abbildung 3.8: Beispiel zur Verdeutlichung des Problems der engsten Bezugsklasse

Die jeweiligen Wahrscheinlichkeitsverteilungen können hieraus auf Basis der gegebenen Informationen zu

(a) $$P(A \wedge B \wedge C \wedge D \wedge E) = \frac{P(A \wedge B)P(B \wedge C)P(C \wedge D)P(C \wedge E)}{P(B)P(C)P(C)} \text{ und}$$

(b) $$P(A \wedge B \wedge C) = \frac{P(A \wedge B)P(B \wedge C)}{P(B)}$$

gefolgert werden. Besonderes Augenmerk sei dabei auf die Variablen D und E gerichtet, welche zu einer sichtlichen Steigerung der Komplexität der Verteilung führen. Kausal gesehen verfeinern sowohl D, als auch E die Merkmalsausprägung von C, da diese in einer bedingten Abhängigkeit stehen. Somit müsste erwartungsgemäß auch die Inferenz von C mit jedwedem Informationsstand durch den Kenntnisstand hinsichtlich D und E beeinflusst werden. Es lässt sich hierbei nun berechnen, dass bspw. die bedingte Wahrscheinlichkeit $P(c_1|b_1)$ hiervon jedoch gänzlich unberührt bleibt:

(a) $$P(c_1|b_1) = \alpha \sum_{a,d,e} P(A \wedge b_1 \wedge c_1 \wedge D \wedge E) = 0.6,$$

(b) $$P(c_1|b_1) = \alpha \sum_{a} P(A \wedge b_1 \wedge c_1) = 0.6.$$

Damit sind die Variablen D und E in Relation zur Abfrage irrelevant, was aus den durch den Graphen beschriebenen Abhängigkeiten bzw. der Verteilung nicht direkt abzuleiten ist. Dieses Minimalbeispiel zeigt daher schon ein Problem großer intensionaler Wissensbasen: Es ist nicht ohne tiefere Kenntnis der Mechanismen bzw. der hintergründigen Bezugsklassen ersichtlich, welche Merkmalsausprägungen für die intendierte

17 *„Haben wir es mit zwei Klassen zu tun, die sich überschneiden, so wählen wir den Durchschnitt."*(Reichenbach, 1994, 383)

Schlussbildung relevant sind. Damit kann als Schwäche intensionaler Ansätze festgehalten werden, dass die Erhöhung der Plausibilität die Größe der Wissensbasen gleichermaßen unnötig erhöhen kann.

Dass dies in der Tat ein weitreichendes Problem ist, wird nun als Abschluss der Betrachtungen der Schwächen intensionaler Ansätze näher erläutert. Tatsächlich lässt sich nachweisen, dass mit steigender Größe der Wissensbasis auch der zu tätigende Aufwand zur Berechnung einzelner Schlüsse unverhältnismäßig stark anwächst. Hierfür sei die häufig in den vergangenen zwei Unterkapiteln aufgeführte Inferenzregel als Beispiel aufgeführt. Denn wie durch Ausklammern einzelner Summenblöcke leichter ersichtlich wird, stellt die Berechnung eines Schlusses in intensionalen Ansätzen lediglich die Aufsummierung aller Ausprägungen unbekannter, d. h. freier Variablen innerhalb der gegebenen Wahrscheinlichkeitsverteilung einer Wissensbasis dar. So ließe sich beispielsweise für eine Wahrscheinlichkeit $P(A \wedge B \wedge C \wedge D)$ zur Ermittlung von $P(A|B)$ wie folgt ausklammern:

$$P(A|B) \quad = \frac{\sum\limits_{c,d} P(A \wedge B \wedge c \wedge d)}{\sum\limits_{a,c,d} P(a \wedge B \wedge c \wedge d)} = \frac{(\sum\limits_{c}(\sum\limits_{d} P(A \wedge B \wedge c \wedge d)))}{\sum\limits_{a}(\sum\limits_{c}(\sum\limits_{d} P(a \wedge B \wedge c \wedge d)))))}.$$

Diese Umformung zeigt, dass zur Inferenz alle möglichen Permutationen der Ausprägung freier Variablen aufsummiert werden. Dies ist im vorliegenden Fall nicht sehr aufwändig und auch generell lassen sich diese Teilschritte innerhalb eines Algorithmus mit adäquater Laufzeit und geringem Speicherbedarf umsetzen. Trotzdem steigt mit höherer Anzahl freier Variablen der Aufwand zur Berechnung, wobei es sich nachweisen lässt, dass Laufzeit und Speicherbedarf hierbei exponentiell steigen - vgl. hierzu Anhang A.2. Dieser Umstand ist dabei unabhängig vom Reifegrad des gewählten Algorithmus, wie bspw. auch durch Cooper (1990, 403) festgestellt wird: „*All of them* [die Algorithmen zur Umsetzung der Inferenzregel, Anm. d. V.] *have a time complexity that, in the worst case, is exponential as a function of the number of unistantiated variables in the network.*“ Heckerman und Breese (1996, 5) verdeutlichen dies an einem anschaulichen Beispiel:

> „*When modeling the real world, we often encounter situations in which an effect has many potential causes. In these situations, probability specification and inference can be impractical if not impossible. For example, suppose we have n binary* [...] *causes* $c_1, \ldots, c_n$ *bearing on a single binary effect e* [...]. *According to the definition* [...], *we must specify the probability distribution of e conditional on every state of its parents. Thus, in this example, we must specify* 2^n *probability distributions* [...]. *Furthermore, probabilistic inference (e. g., the computation of* $P(c_1|e)$*) has time complexity* $\mathcal{O}(2^n)$.“

Daher besitzen intesionale deklarative Wissensbasen ein für Führungsautomaten nicht vorteilhaftes Laufzeit- und Speicherverhalten, sofern diese eine (unnötig) hohe Anzahl freier Variablen vorweisen. Dies kann im schlimmsten Fall dazu führen, dass ressourcenbedingt keine Schlüsse mittels Inferenz berechnet werden können.

Zusammenfassend weisen intensionale Ansätze somit drei primäre Schwächen für ihren Einsatz in Führungsautomaten auf:

- **Erklärungszwang in der Auswahl der numerischen Werte**

  Innerhalb intensionaler Ansätze lassen sich aufgrund des vorherrschenden Bezugsklassenproblems häufig nur subjektive Wertezuweisungen finden. Dies ist einerseits eine aus frequentistischer Position nicht haltbare Argumentationsbasis, andererseits führt diese im Vergleich zu anderen subjektiven Wertezuweisung stets zum Zwang, die Kohärenz im Anwendungsfall zweifelsfrei nachzuweisen.

- **Unnötig große Wissensbasen**

  Zur Vermeidung diskussionswürdiger Wertezuweisungen verbleibt es häufig nur, die Bezugsklasse durch zusätzliche Merkmale einzugrenzen. Dies führt auf lange Sicht zu einem Anwachsen der Wissensbasis, die jedoch rein mathematisch gesehen hierfür keinen Bedarf besitzt. Die letztendliche Relevanz eines Merkmals ist hierbei nicht zwingend aus der graphischen Repräsentation ersichtlich, wodurch Wissensbasen aus guten Gründen unnötig vergrößert werden.

- **Laufzeit- und Speicherprobleme in der Berechnung**

  Eine hohe Anzahl freier, d. h. für bestimmte Inferenzen nicht beobachtete oder benötigte Variablen, führt in intensionalen Ansätzen zu einem ungünstigen Laufzeit- und Speicherverhalten. Wie hier nachweisbar ist, wächst sowohl die Laufzeit, als auch der Speicherbedarf einer Inferenz mit steigender Anzahl der freien Variablen exponentiell an. Dies führt ggf. dazu, dass Abfragen über die Wissensbasis ressourcenbedingt zu keinem Ergebnis führen.

3.3 Konklusion

Wie die letzten Unterkapitel aufzeigten, sind extensionale und intensionale Ansätze zur Abbildung einer unscharfen Lagefeststellung und damit zur Realisierung deklarativer Wissensbasen für Führungsautomaten geeignet. Die für die Ansätze jeweilig aufgeführten Schwächen lassen sich zwar nicht relativieren, nichtsdestotrotz sei in diesem Zusammenhang betont, dass die Schwächen klassischer, rein-logischer Ansätze aus den zu Beginn dieses Kapitels aufgeführten Gründen dennoch stark überwiegen. Wie dargestellt wurde, kommen sowohl extensionale als auch intensionale Ansätze prinzipiell zu denselben Schlüssen. Dennoch stellt sich grundsätzlich die Frage, welcher Ansatz methodisch gesehen eine solidere Ausgangsbasis für Führungsautomaten bildet. Dieses Unterkapitel dient nun dazu, die im Vergleich mit ihrem extensionalen Pendant aussagekräftigeren intensionalen Ansätze hervorzuheben. Dafür werden die aus Sicht eines Führungsautomaten ungünstigen Eigenschaften extensionaler Ansätze herausgestellt.

Der argumentative Grundstein wird dadurch gelegt, inwiefern die aufgeführten Ansätze ein unvollständiges Lagebild einzuschätzen vermögen. Hierfür sei abermals auf die bereits bekannte Beispiellage aus Kapitel 2.3 zurückgegriffen. In diesem Zusammenhang ist es erwähnenswert, dass die in Kapitel 3.1.1 und 3.2.1 aufgeführten Umsetzungen für Führungsautomaten auf Basis eines vollständigen Lagebildes operierten. Somit konnten alle wahrnehmbaren Faktoren tatsächlich auch durch eine vorliegende Evidenz in den Automaten eingespeist werden. Dies ist jedoch nicht sonderlich realistisch, vor allem da unter anderem Aspekte wie die konkrete Feindlage und ein hieraus zu ziehender Kräfte-/Wirkungsvergleich taktisch nicht pauschal vorauszusetzende Informationsstände darstellen. Die nachfolgende Abbildung 3.9 trägt diesem Umstand

Rechnung und zeichnet ein entschieden realistischeres Lagebild. Wesentliche Unterschiede stellen hierbei dar, dass Truppengattung, Position und Deckungszustand der roten Gruppe durch den blauen Infanteriezug nicht einsehbar und damit auch nicht evident sind.

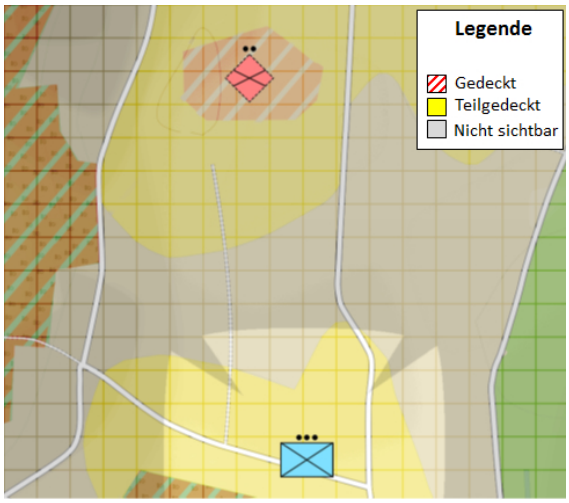


Abbildung 3.9: Unscharfe Ausgangslage

Wie sich nun zeigt, führt dies innerhalb der Ansätze zu durchaus unterschiedlichen Lagebeurteilungen. So lässt sich in der gegebenen Lage mit der vorliegenden Evidenz $\mathfrak{e} = (C_{blue} = C_2 \wedge D_{red} = D_1 \wedge D_{blue} = D_3)$ feststellen, dass dies im extensionalen Beispielautomaten aus Kapitel 3.1.1 zu den Gewissheitsfaktoren

$$c_{H_1} = \mathbf{0.64}; \quad c_{H_2} = -0.30; \quad c_{H_3} = -0.64$$

führt. Damit folgert der Automat auf Basis der vorliegenden Evidenz, dass sich der blaue Jägerzug am ehesten in einer vorteilhaften Ausgangsposition befindet. Analog dazu folgert der intensionale Beispielautomat aus Kapitel 3.2.1 für die vorliegende Evidenz $\mathfrak{e} = (D = d_3 \wedge C' = c_2' \wedge D' = d_1')$ die Wahrscheinlichkeiten

$$P(h_1|\mathfrak{e}) = 0.296; \quad P(h_2|\mathfrak{e}) = \mathbf{0.516}; \quad P(h_3|\mathfrak{e}) = 0.188$$

und kommt damit zu einem anderen Ergebnis. Wie ersichtlich ist, schließt der intensionale Führungsautomat trotz mangelnder Informationen darauf, dass sich die blaue Jägergruppe in einer ausgeglichen Ausgangsposition befindet. Damit weicht der Automat von seiner grundsätzlichen Lageeinschätzung im vollständigen Lagebild nicht ab. Dies wirft die Frage auf, worin dies nun begründet liegt und wieso beide Ansätze unter Unschärfen zu unterschiedlichen Lageeinschätzungen kommen.

Tatsächlich lassen sich hier die Gründe in einer abweichenden Interpretation des unscharfen Informationsstands finden. So ignorieren extensionale Ansätze offenbar das fehlende Wissen und schließen rein auf der Grundlage dessen, was in der jeweiligen Situation evident vorliegt. Bedingt ist dies durch den unterliegenden regelbasierten Mechanismus, wodurch der Automat immer nur dann auf konkrete Zustände schließen kann, wenn die Prämissen der zugehörigen Regeln auf Basis der Lageinformationen bestätigt

wurden. Intensionale Ansätze hingegen sind im Kontrast hierzu in der Lage, sowohl durch vorliegende Evidenzen bestätigtes, als auch unbestätigtes Wissen in die Lageeinschätzung einfließen zu lassen. Dies befähigt den Automaten eine etwaig vorliegende lückenhafte Wahrnehmung zu kompensieren und daher auch auf Zustände zu schließen, welche gemäß des unbestätigten Wissens zutreffen könnten. Dadurch bestätigen intensionale Ansätze somit nicht etwa bestimmte Zustände, sondern geben lediglich ein Maß für die Eintrittswahrscheinlichkeit, dass sich bestimmte Zustände manifestieren.

Dies erklärt die dargestellten Unterschiede im gegebenen Beispielfall. Aus Sicht des extensionalen Automaten gibt es keine bestätigenden Gründe, welche dafür sprechen, dass keine vorteilhafte Position vorliegt. Konträr hierzu gibt es aus dem Blickwinkel des intensionalen Ansatz heraus entschieden mehr mögliche Systemzustände, in denen sich der blaue Jägerzug gegenwärtig in einer ausgeglichene Position befindet. Ergo unterscheiden sich extensionale und intensionale Ansätze zusammenfassend dadurch, dass diese den unscharfen Wissenstand entweder mittels Bestätigung, oder als Maß des Zutreffens (kurz: Wahrscheinlichkeit) für die Manifestierung eines Zielzustandes auswerten[18]. Extensionale Ansätze leiden damit, aufgrund ihrer erwähnten Eigenart zur Bestätigung bestimmter Fälle, unter denselben Limitierungen wie herkömmliche logikbasierte Ansätze. Sie schöpfen daher im Vergleich zu intensionalen Ansätzen in der Verarbeitung unscharfer Lagebilder weniger Mehrwert. Zwar decken die extern hinzugefügten Gewissheitswerte einen weitaus breiteren Ereignisraum ab, nichtsdestotrotz verbleibt es fragwürdig, ob dieser zusätzliche Aufwand tatsächlich deren Einsatz innerhalb von Führungsautomaten rechtfertigen mag. Im Vergleich hierzu bieten intensionale Ansätze eine für den militärischen Kontext adäquatere Verarbeitung unscharfer Lagebilder, da diese auch Eventualfälle mit einbeziehen. Zusätzlich sind Wahrscheinlichkeiten o. ä. einfacher und argumentativ in ihrer Schätzung besser vertretbar, da Werte zur Bemessung der Gewissheit nur schwer über Häufigkeiten zu ermitteln oder rechtfertigen sind.

Gleichwohl bleibt die im Vergleich kontextadäquatere Verarbeitung von Unschärfen im Falle der intensionalen Ansätze nicht ohne Preis, wie insbesondere die Ausführungen hinsichtlich der benötigten Laufzeit bzw. des benötigten Speicherbedarfs aufzeigen. Diese Auffassung teilt auch Pearl (1988, 3): „*Extensional systems are computationally convenient but semantically sloppy, while intensional systems are semantically clear but computationally clumsy.*“ Nichtsdestotrotz ist der Mehrgewinn in puncto Aussagekraft gegenüber extensionalen Ansätzen erheblich, wie u. a. deren Unfähigkeit zur Abbildung abduktiver Schlüsse aufzeigt. Die Vorteile hieraus sind jedoch nicht sofort ersichtlich, wie bspw. auch die folgenden abduktiven Zusammenhänge aufgrund ihrer niedrigen Eintrittswahrscheinlichkeiten aufzeigen:

$$P(\mathfrak{e}|h_1) = 0.081; \quad P(\mathfrak{e}|h_2) = 0.029; \quad P(\mathfrak{e}|h_3) = 0.008.$$

Damit sind Schlussfolgerungen auf Basis abduktiver Zusammenhänge im Gegensatz zu deduktiven Schlüssen nicht sonderlich wertschöpfend. Attraktiver ist hierbei u. a. deren Einsatz zur Berechnung der nur schwer schätz- und ermittelbaren Verbundwahrscheinlichkeiten $P(H_x \wedge \mathfrak{e})$, welche die Basis für die bisher aufgezeigten intensionalen Ansätze darstellen. Unter Kenntnis der jeweiligen a priori Wahrscheinlichkeit $P(h_x)$ lassen sich diese so z. B. durch Umformung der Formel für die Berechnung einer bedingten Wahrscheinlichkeit

18 Vergleiche als argumentative Stütze hierzu auch Harré (1970, 167): „*To say of a proposition that it is probable, is to say that it will or is likely to turn out true, to say that it is confirmed is to say that is has turned out to be true.*“

$$P(\mathfrak{e}|h_1)P(h_1) = 0.016; \quad P(\mathfrak{e}|h_2)P(h_2) = 0.015; \quad P(\mathfrak{e}|h_3)P(h_3) = 0.002$$

einsetzen. Bedeutsamer ist jedoch der Nutzen abduktiver Wahrscheinlichkeiten für die Berechnung von deduktiven Schlüssen. So lassen sich die in der Praxis häufig nur schwer mit numerischen Werten zu hinterlegende deduktive Schlüsse, d. h. Wirkungs-Ursache-Beziehungen ($P(H|\mathfrak{e})$) mit Hilfe der abduktiven Ursache-Wirkungs-Beziehung ($P(\mathfrak{e}|H)$) leichter ermitteln (Pearl, 1988). Hilfreich ist dabei die durch Umformung und Gleichsetzen der Formeln der a posteriori Wahrscheinlichkeiten $P(H|\mathfrak{e})$ und $P(\mathfrak{e}|H)$ zu gewinnende Tautologie

$$P(H|\mathfrak{e}) \quad = \frac{P(\mathfrak{e}|H)P(H)}{P(\mathfrak{e})},$$

welche auch als **Satz von Bayes** bekannt ist. Logisch interpretiert stellt dieser Satz den Zusammenhang zwischen diagnostischen (deduktiven) und prädiktiven (abduktiven) Schluss dar, womit beliebig vom Wert des einen, auf den Wert des anderen Schlusses gefolgert werden kann. Dieser Umstand birgt einen entscheidenden Vorteil zur Gewinnung des hintergründig benötigten Wissens. So ist es tatsächlich besser durch einen Experten beurteilbar, ob ein Infanteriezug sich gegenüber einer roten Infanteriegruppen in einer vorteilhaften Position befinden möge, indem von der Prämisse „Befindet sich in einer vorteilhaften Position“ abduktiv auf „In gedeckter Position“ geschlussfolgert wird. Eine Gegenrichtung wäre in diesem Falle nicht kausal herleitbar, insbesondere weil von der Prämisse „In gedeckter Position“ nicht auf „Befindet sich in einer vorteilhaften Position“ geschlossen werden kann, ohne Kenntnis über weitere Details zur Lage vorauszusetzen. Das folgende Kapitel wird im Detail hierzu aufführen, wie dieser Umstand gewinnbringend dazu eingesetzt werden kann, um die Nachteile intensionaler Ansätze hinsichtlich Laufzeit und Speicherbedarf entscheidend zu reduzieren.

Es ist zusammenfassend festzuhalten, dass intensionale Ansätze für Führungsautomaten eine insgesamt bessere Ausgangsbasis darstellen und eine für den Gebrauch im militärischen Kontext adäquatere Verarbeitung von Unschärfen aufweisen. Sie bieten darüber hinaus die Möglichkeit zur Abbildung abduktiver Schlüsse, was sich positiv für die Ermittlung des benötigen hintergründigen Wissens auswirkt. Im weiteren Verlauf dieser Arbeit wird daher nun der Schwerpunkt auf intensionale Ansätze gelegt werden. Gemäß dieser Festlegung wird folgend eine spezielle Ausprägung stellvertretend näher betrachtet : kausale Netzwerke. Ziel wird es dabei sein zu zeigen, dass sich mit Hilfe von kausalen Netzwerken die Komplexität der benötigten Berechnungen einerseits verringern, dabei jedoch auch andererseits die Funktionsweise deklarativer Wissensbasen für Führungsautomaten anschaulicher darstellen lässt.

4 Kausale Führungsautomaten

Wie es in Kapitel 3.2 bereits Erwähnung fand, können innerhalb intensionaler Ansätze Abhängigkeitsbeziehungen eines probabilistischen Modells mithilfe eines ungerichteten Graphen repräsentiert werden. Hierbei wurde bewusst zur Vereinfachung auf eine konkrete Richtung der Kanten des Graphen verzichtet, da dies ohne Einbußen die letztendliche Aussagekraft der dargestellten Abhängigkeit abzubilden vermag. Dennoch haben gerichtete Kanten zur Darstellung eben jener Abhängigkeiten durchaus ihre Vorteile, die sich neben der besseren Aussagekraft auch positiv auf die Berechnung von Inferenzen und der Erhebung des hierfür benötigten Wissens auswirken. Ziel dieses Kapitels ist es aufzuzeigen, dass mithilfe eines gerichteten azyklischen Graphen kausale Zusammenhänge abgebildet werden können und inwieweit sich dies zugunsten von Führungsautomaten einsetzen lässt. Hierzu wird zuerst das theoretische Fundament bayes'scher Inferenz näher beleuchtet, welcher im folgenden Unterkapitel die Beschreibung eines Inferenzalgorithmus folgt. Dieser Algorithmus dient als Basis der Funktionsweise eines kausalen Führungsautomaten, dessen beispielhafte Umsetzung zum Abschluss dieses Kapitels beschrieben wird und im weiteren Verlauf den Schwerpunkt der Untersuchungen darstellt.

Gemäß der bereits getätigten Ausführungen ist es innerhalb der graphischen Repräsentation von Abhängigkeitsbeziehungen zweier Variablen A und B freigestellt, in welche Richtung diese Abhängigkeit zeigen mag. Ungerichtete Kanten stellen in diesem Zusammenhang eine biimplikative Abhängigkeitsrelation dar, welche bei Veränderung des Wertes der einen Variablen automatisch den Wert der anderen Variable mit beeinflusst. Diese zur graphischen Darstellung eindeutige Abhängigkeitsbeziehung zweier Variablen weist jedoch bei komplexeren Abhängigkeitsbeziehungen gewisse Diskrepanzen auf, wie es sich im Falle mehrerer parallel vorherrschenden Abhängigkeiten aufzeigen lässt.

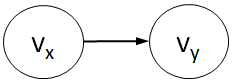
Beispiel Gegeben seien drei Variablen A, B und C. Es herrsche eine direkte Abhängigkeit zwischen A und C vor, die – hier aus Gründen der Vereinfachung innerhalb der Zeile dargestellt – graphisch zu **A—C** gefolgert werden kann. Darüber hinaus herrsche eine Abhängigkeit zwischen B und C vor, welche graphisch in **C—B** resultiert. Die Variablen A und B seien darüber hinaus stochastisch unabhängig, d. h. dass insofern $P(A|B) = P(A)$ und $P(B|A) = P(B)$ gilt und innerhalb des Graphen daher keine direkte Verbindung zwischen beiden Variablen vorherrscht. Folgerichtig trennt C somit innerhalb des Graphen A und B, was graphisch durch **A—C—B** repräsentiert wird. Es liege nun eine Evidenz $\mathfrak{e} = (A \wedge C)$ vor. Die Evidenz hat nun jedoch insbesondere $P(B|\mathfrak{e}) = P(B|A \wedge C) \neq P(B|C)$ zur Folge, da A aufgrund der gegebenen Abhängigkeit zu C einen Einfluss auf dessen Ausprägung besitzt und C wiederum, wegen vorherrschender Abhängigkeit, eine Auswirkung auf die Ausprägung von B vorweist. Die stochastische Unabhängigkeit der Variablen A und B bleibt hierdurch zwar weiterhin gültig, jedoch ist, im Vorhandensein des Wissens um die konkrete Ausprägung von C, das zu erwartende Ergebnis der Ausprägung von B offenbar hypothetisch auch von A abhängig. Somit müsste also der ungerichtete Graph auch **A—B** zur korrekten Darstellung aller möglichen Abhängigkeitsbeziehungen beinhalten, obgleich die Variablen rein stochastisch keine (direkte) Abhängigkeit aufweisen.

Wie das vorhergehende Beispiel aufzeigt, ist es zur korrekten Darstellung einer vorherrschenden Abhängigkeitsbeziehung mithilfe eines ungerichteten Graphen notwendig, die aus den direkten Abhängigkeiten ableitbaren, hypothetischen und indirekten Abhängigkeiten ebenfalls mithilfe einer Kante darzustellen[1]. Ungerichtete Graphen weisen daher eine Schwäche zur graphischen Visualisierung von wirklich vorherrschenden Abhängigkeiten auf. Viel mehr noch vermögen diese offensichtlich nur durch eingehende Analyse des abzubildenden probabilistischen Modells, dieses auch graphisch korrekt abzubilden. Dies ist gleichwohl nachteilig zur Repräsentation tatsächlich vorherrschender Unabhängigkeiten, wie durch Pearl (1988, 116) entsprechend konstatiert wird:

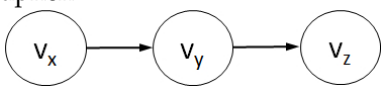
> „*The main weakness of* [...] [these] *networks is their inability to represent induced and non-transitive dependencies; two independent variables will be directly connected by an edge merely because some other variable depends on both. As a result, many useful independencies go unrepresented in the network.*"

Ungerichtete Graphen sind daher ungeeignet, um die innerhalb eines probabilistischen Modells vorherrschenden Abhängigkeitsbeziehungen unmissverständlich abzubilden. Tatsächlich zeigen sich hier gerichtete Graphen den ungerichteten Graphen überlegen, da diese eindeutige Abhängigkeitsbeziehungen durch die definierte Richtung der Kante klarer herausstellen und somit nicht auf Kantenzüge angewiesen sind, die hypothetische Abhängigkeiten darstellen. Dies näher zu erläutern bedarf einer näheren Betrachtung gerichteter Graphen in diesem Kontext, welche nun folgend vorgenommen wird.

Parallel zur Definition aus Kapitel 2.3.2, sei erneut ein gerichteter Graph ein Quadrupel $G_E := (V, E, \alpha, \omega)$, wobei V eine endliche Menge von Ecken, E eine endliche Menge von Kanten und $\alpha, \omega : E \to V$ Abbildungsfunktionen für die Ausgangs- (α) bzw. Zielecke (ω) einer Kante sei. Eine Ecke v_x heiße Vater einer Ecke v_y, wenn eine Kante $e := (v_x, v_y)$ existiert, mit $\alpha(e) = v_x$ und $\omega(e) = v_y$. Parallel dazu heiße dann v_y auch Kind der Ecke v_x. Eine Ecke ohne Kind werde ferner als Blatt, eine Ecke ohne Vater als Wurzel bezeichnet. Eine gerichtete Kante gehe dabei immer vom Vater zum Kind, was im beschriebenen Fall in



resultiert. Die Ecke v_x stellt dann in diesem Fall eine Wurzel und die Ecke v_y ein Blatt dar. Es beschreibe $\nu : V \to M_\alpha$ eine Funktion, die die Menge $M_y \in M_\alpha$ aller Vatereckeneiner Ecke v_y bestimmt. In der Gegenrichtung bestimme die Funktion $k : V \to M_\omega$ die Menge $M_x \in M_\omega$ aller Kindecken einer Ecke v_x. So ließe sich beispielhaft anhand des Graphen



1 Dies mag nun die berechtigte Frage aufwerfen, ob der im letzten Kapitel aufgeführte ungerichtete Graph des intensionalen Führungsautomaten unvollständig ist, da dieser eben jene Kanten vermissen lässt und somit keine korrekte Abbildung des hintergründigen probabilistischen Modells darstellt. Aus diesem Aspekt heraus gesehen mag dies einerseits richtig sein, obgleich festzuhalten ist, dass der formal korrekte Beispielgraph hinsichtlich seines Aufbaus nur wenig Vergleichbarkeit zu dessen extensionalem Pendant zugelassen hätte. Andererseits ist aber auch zweifellos festzuhalten, dass selbst der betrachtete, unvollständige Graph dennoch im Sinne des durch Pearl (1988, 96) begründeten Abbildungsmechanismus eine korrekte Abbildung darstellt: „*Having failed to provide isomorphic graphical representations* [...] *we settle for the following compromise: instead of complete graph isomorphism, we will consider only* [...] *graphs that faithfully display every dependency. However, acknowledging that some independencies will escape representation, we shall insist that their number be kept a minimum* [...]."

ermitteln, dass $\nu(v_z) = \{v_y\}$ und $k(v_y) = \{v_z\}$ ist. Es existiert demnach eine Kante von v_x nach v_y, wenn $v_y \in k(v_x)$ bzw. $v_x \in \nu(v_y)$ vorliegt. In Erweiterung hierzu separiere eine Ecke v_y zwei adjazente Ecken v_x und v_z, wenn gilt:

(1) $(v_y \in k(v_x), v_z \in k(v_y)) \Rightarrow (v_z \notin k(v_x))$,

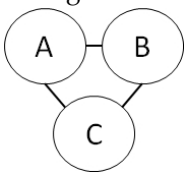
(2) $(v_x, v_z \in k(v_y), v_z \notin k(v_x), v_x \notin k(v_z)) \Rightarrow (v_y \in \nu(v_z), v_y \in \nu(v_x))$, bzw. wenn gilt

(3) $(v_y \in k(v_x), v_y \in k(v_z), v_x \neq v_z) \Rightarrow (v_x \notin k(v_z), v_z \notin k(v_x))$.

Zur Verdeutlichung lassen sich diese drei Fälle auch durch $\mathbf{v_x} \rightarrow \mathbf{v_y} \rightarrow \mathbf{v_z}$ (1), $\mathbf{v_x} \leftarrow \mathbf{v_y} \rightarrow \mathbf{v_z}$ (2) und $\mathbf{v_x} \rightarrow \mathbf{v_y} \leftarrow \mathbf{v_z}$ (3) darstellen[2]. Ein Pfad heiße dabei **aktiv**, oder durch v_y aktiviert, wenn dieser Fall (1) erfüllt und **geblockt**, oder durch v_y blockiert, wenn dieser Fall (2) oder Fall (3) erfüllt.

Aktive Pfade stellen kausale Zusammenhänge zwischen zwei oder mehreren Variablen dar. So kann im Beispielgraphen $v_x \rightarrow v_y \rightarrow v_z$ darauf geschlossen werden, dass die mittels der jeweiligen Ecken repräsentierten Variablen v_y von v_x und v_z von v_y (stochastisch) abhängig sind. Dass dieser Umstand nicht nur zur eindeutigen Abbildung von Abhängigkeiten beiträgt, sondern auch gleichzeitig für die Berechnung von Inferenzen in keinster Weise Nachteile gegenüber ungerichteten Graphen besitzt, sei durch das nachfolgende Beispiel näher erläutert.

Beispiel Erneut seien drei Variablen A, B und C gegeben, wobei A und C, sowie B und C eine direkte Abhängigkeit vorweisen. Die Variablen A und B seien abermals unabhängig voneinander. Wie bereits aufgeführt wurde, wird hierdurch der ungerichtete Graph

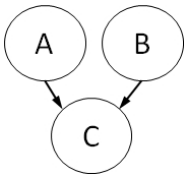


beschrieben, wobei insbesondere der Teilgraph **A—B** eine hypothetische Abhängigkeit aufgrund der gemeinsamen Zusammenhangsbeziehung mit der Variable C darstellt. Gemäß der im letzten Kapitel aufgeführten Konvention, kann aufgrund der graphischen Repräsentation gefolgert werden, dass für etwaige Inferenzen die Gesamtwahrscheinlichkeitsverteilung $P(A \wedge B \wedge C)$ benötigt wird. Diese sei mit $A = \{a_1, a_2\}$, $B = \{b_1, b_2\}$ und $C = \{c_1, c_2\}$ wie folgt gegeben:

$$P(a_1, b_1, c_1) = 0.27 \quad P(a_1, b_2, c_1) = 0.02 \quad P(a_2, b_1, c_1) = 0.19 \quad P(a_2, b_2, c_1) = 0.04$$
$$P(a_1, b_1, c_2) = 0.22 \quad P(a_1, b_2, c_2) = 0.10 \quad P(a_2, b_1, c_2) = 0.12 \quad P(a_2, b_2, c_2) = 0.04$$

Es soll nun gezeigt werden, dass ein gerichteter Graph $\mathbf{A} \rightarrow \mathbf{C} \leftarrow \mathbf{B}$, d. h.

2 Dies stellt eine eigene (und auf eine Ecke begrenzte) Herleitung der sog. **d-Separation** dar, wie sie bspw. in ihrer ursprünglich Form in (Geiger und Pearl, 1990, 6) gefunden werden kann. Auf Basis der d-Separation lässt sich beweisen, dass sich Abhängigkeiten innerhalb eines gerichteten Graphen durch die Kanten zwischen ein- oder mehreren Ecken darstellen lassen. Folgerichtig führt dabei das Entfernen einzelner oder mehreren Ecken zur Spaltung des Graphen in zwei disjunkte Teilgraphen, wodurch keine Abhängigkeitsbeziehung zwischen den somit separierten Ecken vorherrscht. Wie auch Pearl (1988, 116ff) hierzu entnommen werden kann, stellt dies die Grundlage dar, die (Un-)Abhängigkeiten eines probabilistischen Modells mithilfe eines gerichteten Graphen vollständig darzustellen.



nicht nur äquivalent hierzu ist, sondern auch gleichzeitig für Inferenzen auf die explizit abgebildete Unabhängigkeit der Variablen A und B zurückgreift. Wie es bereits Erwähnung fand, bildet die Richtung der Kanten eine explizite Abhängigkeit innerhalb des Graphen ab, welche, im vorherrschenden Fall der gleichzeitigen Abhängigkeit der Variable C von den Variablen A und B, auch als bedingte Wahrscheinlichkeit $P(C|A \wedge B)$ interpretiert werden kann. Mittels Konditionierung lässt sich dabei eine Verbindung der bereits aufgeführten Wahrscheinlichkeitsverteilung $P(A \wedge B \wedge C)$ und $P(C|A \wedge B)$ herstellen:

$$P(A \wedge B \wedge C) \quad = P(C|B \wedge A)P(B|A)P(A).$$

Da insbesondere innerhalb des Graphen die Variable C die Variablen A und B blockiert (Fall (3)), existiert keine direkte Abhängigkeit zwischen den Variablen, wodurch diese stochastisch unabhängig sind. Hieraus folgt, dass im vorliegenden Modell $P(B|A) = P(B)$ gelten muss. Zusammen mit der Kommutativität der Konjunktion folgt hieraus:

$$P(A \wedge B \wedge C) \quad = P(C|A \wedge B)P(B)P(A).$$

So ließe sich nun bspw. mittels Umformung die bedingte Wahrscheinlichkeit $P(C = c_1|A = a1 \wedge B = b1)$ ermitteln:

$$\begin{aligned} P(c_1|a_1 \wedge b_1) \quad &= \frac{P(a_1 \wedge b_1 \wedge c_1)}{P(b_1)P(a_1)} \\ &= \frac{0.27}{0.8 * 0.61} = \mathbf{0.55} \end{aligned}$$

Vergleichend führt die Anwendung der Inferenzregel über der Verteilung des ungerichteten Graphen zu demselben Ergebnis:

$$\begin{aligned} P(c_1|a_1 \wedge b_1) \quad &= \alpha P(a_1 \wedge b_1 \wedge c_1) \\ &= \frac{P(a_1 \wedge b_1 \wedge c_1)}{\sum\limits_{C=\{c_1,c_2\}} P(a_1 \wedge b_1 \wedge C)} \\ &= \frac{0.27}{0.27 + 0.22} = \mathbf{0.55} \end{aligned}$$

Damit weist der gerichtete Graph eine explizite Abbildung der vorherrschenden Unabhängigkeitsbeziehung vor, ist jedoch für die Berechnung von Inferenzen äquivalent zum ungerichteten Graphen, der eben jene Unabhängigkeitsbeziehung mittels einer Kante darstellt. Die Variablen A und B weisen dabei im gerichteten Graphen nicht nur keine Kante zueinander vor, sie werden darüber hinaus aufgrund der durch die gerichteten Kanten dargestellten Abhängigkeitsbeziehung durch die Variable C getrennt. Hierdurch ist der gerichtete Graph zur Berechnung beliebiger Inferenzen insbesondere auf die im probabilistischen Modell vorherrschenden Unabhängigkeit zwischen A und B angewiesen.

Wie das Beispiel zeigt, können innerhalb eines gerichteten Graphen Inferenzen auf Basis der Richtung und der hieraus entstehenden (stochastischen) Abhängigkeiten berechnet werden. Viel mehr noch lässt sich die Wahrscheinlichkeitsverteilung eines probabilistischen Modells mithilfe der Konditionierung ermitteln. Dies stellt die Grundlage einer durch die Richtung der jeweiligen Kanten eines Graphen ableitbaren Formel für die Wahrscheinlichkeitsverteilung eines probabilistischen Modells – in Folgerung der Formel von Pearl (2009, 14) bzw. Neapolitan (2012, 166ff) – dar:

$$\begin{aligned} P(x_1 \wedge x_2 \wedge \ldots \wedge x_n) &= P(x_n|x_{n-1} \wedge \ldots \wedge x_1)P(x_{n-1}|x_{n-2} \wedge \ldots \wedge x_1)\ldots P(x_2|x_1)P(x_1) \\ &= \prod_i P(x_i|\nu(x_i)) \end{aligned}$$

Hieraus wird ersichtlich, dass sich die Wahrscheinlichkeitsverteilung eines probabilistischen Modells als Produkt aller enthaltenen, direkt einander bedingenden Wahrscheinlichkeiten interpretieren lässt. Dies erweist sich aus zweierlei Hinsicht als vorteilhaft. So können die einerseits schwer zu evaluierenden Wahrscheinlichkeitsverteilungen von Cliquen, die eine Konjunktion aller in einer Abhängigkeitsbeziehung stehenden Variablen umfassen, durch die durch Experten verlässlicher zu schätzenden, bedingten Wahrscheinlichkeiten ersetzt werden. Andererseits erlaubt dies innerhalb des Graphen, aufgrund der explizit aufgeführten Unabhängigkeitsbeziehungen, Pfade von Informationsflüssen zu bestimmen, die, sofern sich evidente Variablen auf diesen Pfaden befinden, für Inferenzen verwendet werden können. Um Rückkopplungen und hierdurch ausgelöste Endloßschleifen innerhalb dieser Inferenzen zu vermeiden, ist es notwendig, dass die repräsentativen Graphen kreisfrei sind. Gerichtete kreisfreie Graphen mit zugeordnetem probabilistischen Modell werden auch als Bayes'sche Netzwerke bezeichnet.[3] Die innerhalb des Netzwerks explizit aufgeführten Abhängigkeitsbeziehungen können dazu genutzt werden, um kausale Zusammenhänge zwischen einer oder mehreren Variablen abzubilden.

Hierbei ist es wichtig anzumerken, dass die durch das probabilistische Modell ausgedrückten Abhängigkeitsbeziehungen nicht pauschal für die Beschreibung kausaler Zusammenhänge stehen. Dies begründet sich auf der Tatsache, dass in probabilistischen Modellen die Schlussrichtung frei gewählt und Abhängigkeiten deshalb auch entgegen der eigentlichen kausalen Zusammenhangsbeziehung definiert werden können. Die Definition von Abhängigkeitsbeziehungen entgegen der kausalen Schlussrichtung stellt jedoch eine Seltenheit dar, da sich die benötigten bedingten Wahrscheinlichkeiten nicht auf Grundlage beobachtbarer Evidenzen aufbauen lassen. Damit bilden Bayes'sche Netzwerke in der Praxis primär kausale Zusammenhänge ab, was durch Pearl (2009, 21) entsprechend gestützt wird:

> „*However, the ubiquity of DAG* [Gerichtete kreisfreie Graphen, Anm. d. Verf.] *models in statistical and AI applications stems (often unwillingly) primarily from their causal interpretation - that is, as a system of processes, one per family, that could account for the generation of the observed data. It is this causal interpretation that explains why DAG models are rarely used in any variable ordering other than those which resprect the direction of time and causation.* "

3 Vergleiche hierzu die ursprünglich durch Pearl (1985, 330) vorgenommene Definition: „*Bayes Networks are directed acyclic graphs in which the nodes represent propositions (or variables), the arcs signify the existence of direct causal influences between the linked propositions, and the strengths of these influences are quantified by conditional probabilities [...].*"

Bayes'sche Netzwerke beschreiben damit die Obermenge aller gerichteten kreisfreien Graphen, welchen ein probabilistisches Modell zugeordnet wird. Da die Anordung der Variablen innerhalb der Netzwerke auch rein probabilistische Zusammenhänge abzubilden vermögen, ist es hier üblich eine Unterklassifzierung für kausale Zusammenhänge vorzunehmen. Bayes'sche Netzwerke, die rein kausale Zusammenhänge abbilden, werden dabei auch als kausale Bayes'sche Netzwerke oder kurz als kausale Netzwerke bezeichnet (Pearl, 2009, 21f; Neapolitan, 2012, 158f). Innerhalb dieser Arbeit liegt der Fokus auf der Abbildung kausaler Zusammenhänge, da Schlüsse der hierauf aufbauenden Führungsautomaten sich auf beobacht- und messbaren Phänomenen abstützen sollen. Kausale Netzwerke weisen dabei Funktionalitäten auf, welche u. a. die Bestimmung und Validierung kausaler Abhängigkeitsbeziehungen erlauben. Innerhalb dieser Arbeit werden diese Mechanismen nicht im Schwerpunkt betrachtet, da vielmehr das Augenmerk auf den grundsätzliche Einsatz der Netzwerke innerhalb von wissensbasierten Automaten liegt. Für Weiteres zu kausalen Modellen sei daher auf Pearl (2009) bzw. Pearl et al. (2016) verwiesen.

Zusammenfassend sind gerichtete Graphen gegenüber ungerichteten Graphen in ihrer Aussagekraft zur Darstellung von Abhängigkeiten nicht nur gleichwertig, sie weisen auch eine eindrücklichere Abbildung der vorherrschenden Unabhängigkeiten eines zugeordneten probabilistischen Modells auf. Die konkrete Abbildung von (Un-)Abhängigkeitsbeziehungen ermöglicht es dabei kausale Zusammenhänge abzubilden. Gerichtete kreisfreie Graphen, welchen ein entsprechendes probabilistisches Modell zugeordnet wird, werden dabei auch als kausale Netzwerke bezeichnet, wenn die Anordnung der Variablen rein kausalen Zusammenhängen folgt. Die explizite Adressierung dieser Abhängigkeitsbeziehungen mithilfe gerichteter Kanten, hat dabei positive Auswirkungen auf die Ausdrucksfähigkeit des Modells. Im nachfolgenden Kapitel 4.1 wird nun in Folge dargestellt, wie mithilfe der gesteigerten Ausdrucksfähigkeit die Berechnung von Inferenzen positiv beeinflusst werden kann. Dieser Umstand wird anschließend in Kapitel 4.2 dazu verwendet, eine konkrete Umsetzung kausaler Netzwerke innerhalb von Führungsautomaten zu entwickeln.

4.1 Propagierung des Belief State

Ziel dieses Kapitels ist die Erläuterung eines Algorithmus zur Propagierung des Belief State, welcher im weiteren Verlauf dieser Arbeit zur Berechnung von Inferenzen in gerichteten azyklischen Graphen eingesetzt wird. Begonnen wird dabei mit der grundsätzlichen Idee zur Propagierung von Daten innerhalb eines Graphen. Anschließend wird mit dem **Belief Propagation Algorithmus** ein Mechanismus näher betrachtet, welcher Inferenzen innerhalb kausaler Netzwerke ausdrucksstark und schnell zu berechnen vermag. Die Grundlagen dieses Algorithmus wurden von Pearl (1988, 175ff) erarbeitet. Die folgende Erläuterung folgt hierbei überwiegend dessen Ausführungen. Auf dieser Grundlage ist es im Folgekapitel anschließend möglich, das Funktionsprinzip des hierauf aufbauenden kausalen Führungsautomaten zu beschreiben.

Wie bereits in Kapitel 3.2.2 dargestellt wird, besitzen intensionale Ansätze grundsätzlich die negative Eigenschaft, dass die Berechnung exakter Inferenzen mit steigender Anzahl freier Variablen eine hohe Zeitkomplexität aufweist. Wie entsprechend auch in Anhang A.2 aufgezeigt wird, rührt diese hohe Zeit-

komplexität insbesondere von der Tatsache, dass die exakte Inferenz einer Marginalisierung über alle ungebundenen Variablen der Verteilung bedarf. Wie der Einführung zu diesem Kapitel zu entnehmen ist, lässt sich dieser Umstand durch die Etablierung expliziter (Un-)Abhängigkeitsbeziehungen mindern, wodurch nur noch bedingte Wahrscheinlichkeiten für die Berechnung benötigt werden. Dies führt einerseits dazu, dass sich die Komplexität der Wahrscheinlichkeitsverteilung eines probabilistischen Modells stark verringert. Die Abbildung des probabilistischen Modells in Form gerichteter Graphen ermöglicht andererseits den Einsatz entsprechender Algorithmen, welche einen Nutzen aus den gerichteten Verbindungen des Graphen ziehen.

Entsprechende Algorithmen fanden hierbei ihren Ursprung in den Arbeiten von Pearl (1982) bzw. Kim und Pearl (1983), welche erstmals Inferenzen innerhalb baumartiger Strukturen auf Basis eines Propagierungsalgorithmus untersuchten. In Folge wurde dieser Algorithmus auf Polybäume übertragen (Rebane und Pearl, 1987; Pearl, 1988). Lauritzen und Spiegelhalter (1988) erweiterten diesen Gedanken und setzten schließlich einen Inferenzalgorithmus für allgemeine Netze um. Trotz der augenscheinlichen Gemeinsamkeit zur Berechnung von Inferenzen innerhalb von Bayes'schen Netzwerken, lassen sich diese Algorithmen jedoch eindeutig unterscheiden. Darwiche (2009, 340) fasst dies entsprechend zusammen: „*There are two versions of belief propagation, one computing joint marginals and the other computing conditional marginals.*" So stützt sich der Algorithmus von Pearl (1988) auf die Berechnung von bedingten Wahrscheinlichkeiten ab, während der Algorithmus von Lauritzen und Spiegelhalter (1988) sich auf die Berechnung von Verbundwahrscheinlichkeiten konzentriert.[4] Für die Umsetzung innerhalb der Gefechtssimulation bieten diese Algorithmen eine solide Ausgangsposition. Da Führungsautomaten insbesondere in den unteren taktischen Ebenen in kurzen Zeitintervallen einer wesentlichen Lageänderung unterliegen, können jedoch die hierfür benötigten komplexen Berechnungsschritte mitunter die mögliche Anzahl der zeitgleich simulierten Entitäten stark verringern. Damit unterliegt die Auswahl des Algorithmus dem Zwang, möglichst Zeit- und Recheneffizient zu sein, um auch größere Truppenkörper hochauflösen abbilden zu können.

Baumstrukturen erweisen sich hier im Allgemeinen als sehr geeignet, da rekursive Algorithmen auf diesen ein vorwiegend gutes Konvergenzverhalten besitzen. Zusätzlich ist anzumerken, dass der Algorithmus von Lauritzen und Spiegelhalter (1988) zwar auf gerichteten Graphen aufbaut, diese jedoch in einem weiteren Schritt zu Cliquengraphen[5] übersetzt. Dieser Algorithmus verspricht damit unter Berücksichtigung von Kapitel 3.2.2 keine wesentliche Komplexitätsminderung in der Berechnung von Inferenzen. Eine Reduktion der Graphen auf Baumstrukturen wirkt sich im Vergleich hier insbesondere positiv aus. Den Vorarbeiten von Rebane und Pearl (1987) folgend, muss der Graph hierfür einem Polybaum entsprechen. Diese Arbeit stützt sich simultan hierzu auf sog. Chow-Liu-Bäumen[6] ab. Damit in Folge der Propagation Algorithmus nun in seiner Funktionsweise aufgezeigt werden kann, ist es jedoch notwendig zuerst die Grundlagen der Propagierung von Informationen innerhalb eines Graphen darzustellen.

4 Wie Pearl und Mackenzie (2018, 348) auch aufführen, erfolgte die Entwicklung des Algorithmus durch Pearl (1986) ursprünglich für den Einsatz innerhalb von Expertensystemen. Lauritzen und Spiegelhalter (1988) bauten auf diesem Grundgedanken auf, und ermöglichten die Berechenbarkeit von rekursiven Inferenzen (sog. loopy networks). Für Inferenzen in Polybäumen ist der Algorithmus von Pearl (1988) nach wie vor die Referenz - vgl. (Darwiche, 2009, 340), (Pearl, 2009, 20) und (Pearl und Mackenzie, 2018, 348). Der Algorithmus von Lauritzen und Spiegelhalter (1988) wiederfuhr durch Shafer und Shenoy (1990, 17ff) eine eigene Interpretation, welche auch entsprechend in (Nielsen und Jensen, 2007, 127ff) und (Kruse et al., 2011, 403ff) als Referenz für Inferenzen in allgemeinen Graphen aufgeführt wird.

5 Siehe Kapitel 3.2 für näheres zu Cliquengraphen als Repräsentation von probabilistischen Modellen.

6 Vgl. hierzu (Chow und Liu, 1968). Aus den durch Kim und Pearl (1983, 191) geäußerten Gründen hat diese Festlegung entscheidende Vorteile für die Gewinnung und Validierung von benötigten Expertenwissen.

Die Grundidee des Belief Propagation Algorithmus ist es, dass jeder Ecke des Graphen ein separates Wissensvermittlungssystem angehangen wird, welches die Parameter ihrer zugeordneten Variable überwacht und die Kommunikation mit den benachbarten, in Abhängigkeitsbeziehung stehenden Variablen verwaltet (Pearl, 1988, 145). Erfüllt die Änderung der Variablenwerte in einem oder mehreren benachbarten Ecken hierbei bestimmte Voraussetzungen, so notiert das Wissensvermittlungssystem dies und ändert entsprechend den Parameterwert der eigenen Variablen. Diese Änderung wird durch das System daraufhin an die benachbarten Ecken weitergegeben, welche ihrerseits die eingegangene Änderung prüfen und ggf. eigene Schritte einleiten. In der Theorie ensteht hieraus eine Form der Wissenspropagierung, die, unter Vorlage einer beliebigen Evidenz, die Auswirkungen auf die einzelnen Ecken so lange schrittweise an benachbarte Ecken weiterleitet, bis sich innerhalb des Netzes keine Änderung an den Parametern der Variablen mehr ergeben. Dies ähnelt stark klassischen Propagierungsschemata der logischen Inferenz, welche bspw. im Falle vorherrschender Aussagen $A \Rightarrow B$ und $B \Rightarrow C$, mithilfe des Modus Ponens einen transiviten Schluss von A über B auf C erlauben, dafür jedoch lediglich eine Evidenz für A benötigen:

$$\dfrac{\dfrac{A \Rightarrow B, \quad A}{B,}\,(MP) \qquad B \Rightarrow C}{\therefore C}\,(MP)$$

Somit führt die Erfüllung der ursprüngliche Bedingung (hier: Eine Evidenz für A) in transitiven Schlüssen dazu, dass alle folgenden Bedingungen (hier: Eine Evidenz für B auf der Grundlage von A) ebenfalls erfüllt werden. Einzelnen Bedingungen werden demnach, in anderen Worten, von einer Aussage zur nächsten propagiert und bilden hieraus den Mechanismus der sog. **Bedingungspropagierung** (engl. *Constraint Propagation*).

Mit Rückblick auf Kapitel 3.1.2 unterliegen Schlüsse auf Basis der klassisch logischen Inferenzregeln insbesondere dem Prinzip der Modularität. Wie anhand der extensionalen Ansätze dort aufgezeigt wurde, unterliegen auch Abkömmlinge dieser klassischen Inferenzmechanismen diesem Prinzip. Obgleich das Modularitätsprinzip gem. der in Kapitel 3.1.2 aufgeführten Argumentation negative Eigenschaften für extensionale Ansätze aufwirft, birgt dieses jedoch auch entsprechende Vorteile für die Berechnung von Schlüssen:

> „[...] [A]*n inference rule of the form "If premise A, then action B" constitutes a very attractive unit of computation for three reasons:*
>
> 1. *The triggering mechanism is local, i. e., it grants a license to initiate the action whenever the premise A is true in the knowledge base K, regardless of other information that K contains.*
> 2. *The action is computationally simple and normally involves onoly a minor adjustment on the knowledge base.*
> 3. *Both the triggering mechanism and the action are conceptually meaningful and are therefore easy to program, modify and explain.*"(Pearl, 1988, 147)

Im entsprechendem Maße gelten diese Vorteile auch für die Bedingungspropagierung, wie durch Pearl (1988, 148) folgend konstatiert wird:

> *„Almost identical features hold in the constraint propagation formalism. Each processor* [Wissensvermittlungssystem, Anm. d. Verf.] *receives a permanent license to act whenever a certain condition develops among its neighbors. The action is simple, since it involves only a local pertubation of the processor's parameters. Both the activation and the action are meaningful because they engage semantically related propositions.*"

Für die folgende Betrachtungen werde nun ein gerichteter Graph als ein Inferenznetz nach der ursprünglichen Definition von Duda et al. (1976, 1075ff) aufgebaut. Jede Ecke $v \in V$ stelle dabei ein Knoten des Netzes dar, welcher wiederum eine beliebige Variable eines zugeordneten probabilistischen Modells repräsentiert. Eine gerichtete Kante $e \in E,\ e = (v_x, v_y)$ mit $\alpha(e) = v_x$ und $\omega(e) = v_y$ stelle in Einklang mit dem letzten Kapitel die Repräsentation der Abhängigkeitsbeziehung zwischen den Variablen X und Y dar, wobei der Wert der Variable Y als Kind direkt von der Belegung des Vaterknotens X abhängt. In diesem Sinne gilt daher im vorliegenden Fall $P(Y) = \sum_{x_i} P(Y|X)$, wobei x_i für eine beliebige Belegung der Variablen X stehe. Im Weiteren wird sich auf das folgende Fragment eines Baumes in Abb. 4.1 abgestützt werden, dem ein nicht näher definiertes probabilistisches Modell zugeordnet wurde.

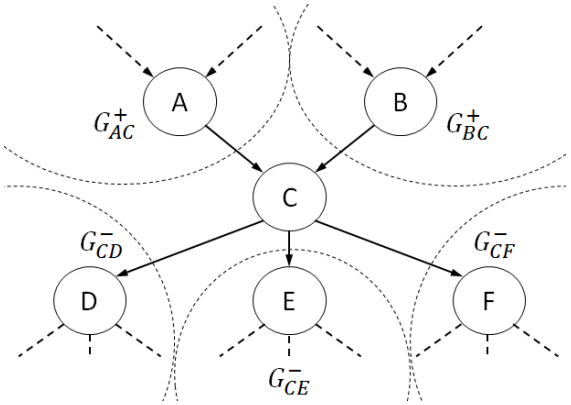


Abbildung 4.1: Beispielhaftes Fragment eines Chow-Liu Baumes

Gemäß der geltenden Einschränkungen für die Struktur der verwendeten Bäume gilt, dass immer nur ein eindeutiger Pfad zwischen zwei Knoten existieren kann. Da der Graph zusätzlich kreisfrei ist, teilt somit jede beliebige Abhängigkeit $X \rightarrow Y$ den Graph in zwei Teile, die einen oberen (G^+_{XY}) und unteren Teilgraphen (G^-_{XY}) beschreiben. Wie Kim und Pearl (1983, 192f) hierzu entnommen werden kann, bilden diese zwei Teilgraphen eine hierarchische Repräsentation für alle Mengen von Daten, welche entsprechend auch durch D^+_{XY} und D^-_{XY} bezeichnet werden. Diese Daten stellen hierbei die jeweiligen Beobachtungen und a priori Wissensstände dar, welche am Rand des Netzes entnommen werden können (Kim und Pearl, 1983). Mit Blick auf den Beispielgraphen aus Abb. 4.1 bedeutet dies demnach, dass dieser bspw. ausgehend vom Knoten C in fünf Teilgraphen unterteilt werden kann: Zwei obere Teilgraphen G^+_{AC}, G^+_{BC} und drei untere Teilgraphen G^-_{CD}, G^-_{CE}, G^-_{CF}. Da immer nur ein eindeutiger Pfad zwischen einer Variablen und einem adjanzenten Teilgraphen besteht, ist hierbei auch zu folgern, dass auch nur die direkt mit dem Teilgraphen verbundenen Variablen durch die im jeweiligen Vorgängerknoten vorherrschenden Daten beeinflusst werden. Im gegebenen Beispiel bedeutet dies, dass, ausgehend vom Knoten D, die Daten D^+_{CD} die Wahrscheinlichkeit

zur Manifestation einer Ausprägung d_i der Variablen D nur durch die konkrete Ausprägung c_j von C beeinflussen, wodurch

$$P(d_i|c_j \wedge D^+_{CD}) \quad = P(d_i|c_j)$$

gilt. Nicht immer ist hierbei jedoch davon auszugehen, dass eine konkrete Ausprägung des jeweiligen Vaterknotens vorliegt. Wie bspw. aus Abb. 4.1 ersichtlich wird, stellt C keinen Randknoten des Netzes dar und kann somit entsprechend als Folgerung der Ausführungen von Kim und Pearl (1983) a priori auch nicht festgelegt sein. In Unkenntnis der konkreten Ausprägung ist es zur Berechnung der a posteriori Wahrscheinlichkeit für die Manifestation von d_i notwendig, diese in Abhängigkeit von C mittels Konditionierung zu ermitteln, da zwar die Variable D nicht direkt von den vorliegenden Daten D^+_{CD} abhängen mag, wohl aber die Ausprägungen von C durch diese direkt beeinflusst werden können:

$$P(d_i|D^+_{CD}) \quad = \sum_j P(d_i|c_j)P(c_j|D^+_{CD}).$$

Damit lässt sich der Einfluss eines Knotens auf einen anderen Knoten vollständig durch die Aufsummierung der möglichen Werte aller Knoten und Ausprägungen entlang des Pfades zwischen beiden Knoten ermitteln (Kim und Pearl, 1983, 192). Dies beschreibt die Weitergabe von Informationen eines Knotens an einen einzelnen Nachfahren, womit noch zur Vollständigkeit die Weitergabe der Informationen eines Knotens an alle untergeordneten Abkömmlinge aussteht. So stellt im Beispiel die durch einen Knoten repräsentierte Variable C gleichzeitig Vaterknoten der Variablen D, E und F dar. Da besagte Variablen jedoch nur die gemeinsame Ursache C teilen, weisen diese keine weitere Verbindung zueinander auf und sind infolge dessen stochastisch unabhängig zueinander. Hieraus folgt, dass der Einfluss der Daten aus den oberen Teilgraphen G^+_{CD}, G^+_{CE} und G^+_{CF} auf die jeweiligen Variablen ebenfalls nur anhand der vorherrschenden Ausprägung der Variable C bestimmt wird und damit

$$P(d_i \wedge e_j \wedge f_k|D^+_C) \quad = \sum_l P(d_i|c_l)P(e_j|c_l)P(f_k|c_l)P(c_l|D^+_C)$$

zu folgern ist.

Gleichzeitig können auch Daten innerhalb der unteren Teilgraphen Rückschlüsse auf die Wahrscheinlichkeit zur Manifestation der Ausprägung einer spezifischen Variablen eines (Vater-)Knotens zulassen. Dies bedingt einen Mechanismus, welcher die Auswirkung von Informationen der unteren Teilgraphen und damit auch die indirekte Auswirkung von Daten der Kindknoten auf ihre Vorfahren bestimmt. Hierzu kann das Wissen um den Umstand verwendet werden, dass alle Wurzeln und Pfade voneinander stochastisch unabhängig sind. Kombiniert mit der Voraussetzung, dass Daten nur am Rand des Netzes entnommen werden, bedeutet dies, dass ein vorliegender Informationsstand auch vorrangig nur eine direkte Auswirkung auf die Ausprägung der Blätter hat und diese Auswirkung kaskadierend an alle Vorfahren zu übermitteln ist. Simultan zum bereits bekannten Vorgehen innerhalb der oberen Teilgraphen, erweist es sich erneut als notwendig, die entsprechende Schlüsse durch Konditionierung zu berechnen, da auch im unteren Teilgraphen das konkrete Wissen um die Ausprägung einer Variablen nur am Rand des Netzes vorliegen kann. So herrscht z. B. in dem in Abbildung 4.1 dargestellten Graphen für C nur eine indirekte Abhängigkeitsbeziehung zu D vor, womit für die bedingte Wahrscheinlichkeit

$$P(D^-_{CD}|c_i) \quad = \sum_j P(D^-_{CD} \wedge d_j|c_i)P(D^-_{CD}|d_j)$$

und insbesondere aufgrund der Unabhängigkeit der Variable C von den tiefer liegenden Daten innerhalb des unteren Teilgraphen G^-_{CD}

$$P(D^-_{CD}|c_i) \quad = \sum_j P(d_j|c_i)P(D^-_{CD}|d_j)$$

zu folgern ist. Simultan zur Weitergabe der Daten aus dem oberen Teilgraphen, kann im aufgezeigten Beispiel von den Ursachen D, E und F lediglich auf die gemeinsame Ursache C zurückgeschlossen werden. Da auch in diesem Fall parallele Pfade unabhängig voneinander sind, kann der Einfluss des Informationsstandes aus den unteren Teilgraphen G^-_{CD}, G^-_{CE} und G^-_{CF} auf die Wahrscheinlichkeit zur Manifestation einer Ausprägung c_i der Variablen C mittels

$$P(D^-_{CD} \wedge D^-_{CE} \wedge D^-_{CF}|c_i) \quad = P(D^-_{CD}|c_i)P(D^-_{CE}|c_i)P(D^-_{CF}|c_i)$$

beschrieben werden.

Werden die dargestellten Mechanismen zur Weitergabe von Daten aus den oberen und unteren Teilgraphen nun miteinander verbunden, lässt sich die hieraus folgende Auswirkung auf eine beliebige Variable innerhalb des Graphen ableiten. So kann z. B. – in direkter Folgerung aus (Kim und Pearl, 1983, 192; Pearl, 1988, 177f) und unter mehrfacher Anwendung des Satzes von Bayes – die Wahrscheinlichkeit zur Manifestation einer beliebigen Repräsentation c_i von C als

$$\begin{aligned} P(c_i|D^+_{AC} \wedge D^+_{BC} \wedge D^-_{CD} \wedge D^-_{CE} \wedge D^-_{CF}) \quad &= \alpha P(c_i|D^+_{AC} \wedge D^+_{BC})P(D^-_{CD} \wedge D^-_{CE} \wedge D^-_{CF}|c_i) \\ &= \alpha[\sum_{j,k} P(c_i|a_j \wedge b_k)P(a_i|D^+_{AC})P(b_k|D^+_{BC})] \\ &\quad P(D^-_{CD}|c_i)P(D^-_{CE}|c_i)P(D^-_{CF}|c_i) \end{aligned}$$

ermittelt werden. Vereinfacht wird diese Formel, indem die Einflüsse der oberen und unteren Teilgraphen durch Funktionen substituiert werden. So lässt sich der Einfluss des oberen Teilgraphen mittels der Funktion

$$\pi(c_i) \quad = \sum_{j,k} P(c_i|a_j \wedge b_k)P(a_j|D^+_{AC})P(b_k|D^+_{BC})$$

und der Einfluss des unteren Teilgraphen durch

$$\lambda(c_i) \quad = P(D^-_{CD}|c_i)P(D^-_{CE}|c_i)P(D^-_{CF}|c_i)$$

ersetzen, was zur allgemeinen substituierten Form

$$P(c_i|D^+_{AC} \wedge D^+_{BC} \wedge D^-_{CD} \wedge D^-_{CE} \wedge D^-_{CF}) \quad = \alpha\pi(c_i)\lambda(c_i)$$

führt. Dies suggeriert bereits, welche Daten folglich innerhalb des Wissensvermittlungssystems eines Knotens X zur Berechnung der Wahrscheinlichkeit für die Manifestation einer beliebigen Repräsentation x_k vorzuhalten sind:

1.) Die bedingte Wahrscheinlichkeit $P(x_k|\nu_1 \wedge \ldots \wedge \nu_n)$,

2.) Die evidente Stärke des Einflusses aus dem oberen Teilgraphen $\pi(x_k)$, und

3.) Die evidente Stärke des Einflusses aus dem unteren Teilgraphen $\lambda(x_k)$.

Beschreibe hierzu $\mathcal{V}_l \in \nu(X)$ einen beliebigen Vaterknoten von X mit einer beliebigen Anzahl Repräsentationen ν_i. Es sei $K_j \in k(X)$ ein beliebiger Kindknoten von X und $\mathfrak{e} = \mathfrak{e}^+ \wedge \mathfrak{e}^-$ mit $\mathfrak{e}^+ = (D^+_{\mathcal{V}_1 X} \wedge \ldots \wedge D^+_{\mathcal{V}_n X})$ bzw. $\mathfrak{e}^- = (D^-_{XK_1} \wedge \ldots \wedge D^-_{XK_m})$ die jeweils in den oberen und unteren Teilgraphen vorhandenen Daten. Dann kann frei nach Pearl (1988, 177f) bzw. Neapolitan (2012, 288ff) allgemein die Wahrscheinlichkeit für die Manifestation von x_k mittels der Formel

$$\begin{aligned} Bel(x_k) = P(x_k|\mathfrak{e}) \quad &= P(x_k|\mathfrak{e}^+ \wedge \mathfrak{e}^-) \\ &= \alpha\pi(x_k)\lambda(x_k) \\ &= \alpha P(x_k|\mathfrak{e}^+)P(\mathfrak{e}^-|x_k) \\ &= \alpha[\sum_{\nu_i} P(x_k|\nu_1 \wedge \ldots \wedge \nu_n) \prod_{i=1}^{n} P(\nu_i|D^+_{X_i})][\prod_{j=1}^{m} P(D^-_{X_j}|x_k)] \\ &= \alpha[\sum_{\nu_i} P(x_k|\nu_1 \wedge \ldots \wedge \nu_n) \prod_{i=1}^{n} \pi_X(\nu_i)][\prod_{j=1}^{m} \lambda_{K_j}(x_k)] \end{aligned}$$

berechnet werden. Die offensichtliche Komplexität dieser Formel ist jedoch trügerisch, da diese durch lokale Berechnungen innerhalb der Wissenvermittlungssysteme einzelner Knoten geteilt und der hieraus entstehende Belief State $Bel(X)$ mithilfe von Nachrichten durch das Inferenznetz schrittweise propagiert werden kann. Wie Kim und Pearl (1983, 193) bzw. Pearl (1988, 178) hierzu zu entnehmen ist, sind derartige Nachrichten aufgrund der jeweiligen Propagierungsrichtungen vorhandener Daten aufzuteilen:

- Der Propagierung aus dem oberen Teilgraphen in den unteren Teilgraphen (*Top-down propagation*), in welcher ein Knoten X eine sog. π**-Nachricht** an dessen Kinder $K_n \in k(X)$ sendet. Diese wird in Abhängigkeit der jeweiligen Ausprägung k_i von K_n mittels $\pi_X(k_i)$ beschrieben.
- Der Propagierung aus den unteren Teilgraphen in die oberen Teilgraphen (*Bottom-up propagation*), in welcher ein Knoten X eine sog. λ**-Nachricht** an dessen Väter $\mathcal{V}_j \in \nu(X)$ sendet. Diese wird in Abhängigkeit der jeweiligen Ausprägung x_k von X mittels $\lambda_{\mathcal{V}_j}(x_k)$ beschrieben.

Zur Verdeutlichung sei hierzu auf Abb. 4.2 verwiesen, die alle im bisher betrachteten Beispielfall möglichen Nachrichten und deren jeweilige Propagierungsrichtung aufführt. Eine Nachricht wird genau dann initiiert, wenn eine Änderung der Parameter in der direkten Nachbarschaft eines Wissensvermittlungssystems stattgefunden hat. Hierbei ist zwischen vier grundsätzlichen Fällen zu unterscheiden:

1.) Der Initialisierung der Parameter,

2.) Der Instantiierung einer Variablen,

3.) Der Änderung des Belief State in einem oder mehreren Vaterknoten, oder

4.) Der Änderung des Belief State in einem oder mehreren Kindknoten.

Hierdurch werden entsprechend die Werte der Funktionen $\pi(X)$ und $\lambda(X)$ durch die korrespondierenden π- bzw. λ-Nachrichten im Falle einer auftretenden Parameteränderung lokal berechnet und die Wahrscheinlichkeit der jeweiligen, mit dem Wissenvermittlungssystem verknüpften, Variable somit sukzessive aktualisiert. Dies beendet die grundsätzliche Darstellung der Funktionsweise des Belief Propagation Algorithmus. Nachfolgend wird dieser nun als Interpretation der Ausführungen von Neapolitan (2012, 238ff) bzw. Korb und Nicholson (2004, 59) aufgeführt.

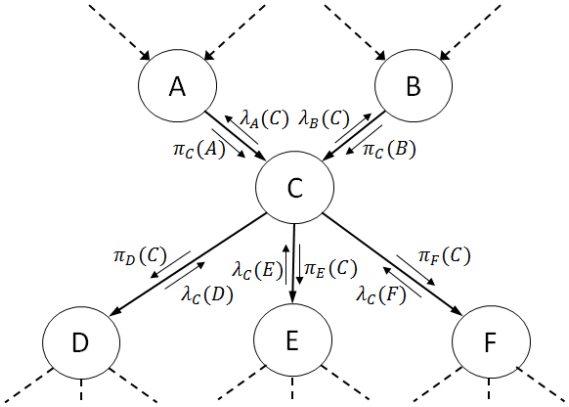


Abbildung 4.2: Nachrichtenfluss im Beispielnetz

Zur besseren Übersichtlichkeit ist hierbei eine vorangestellte Aufführung der im Algorithmus verwendeten Formeln von Pearl (1988, 183) sinnvoll. Folgender Tab. 4.1 sind diese zu entnehmen, wobei im Laufe der nachgestellten Aufführung des Algorithmus die jeweiligen Formeln über die rechts stehenden Ziffern referenziert werden. Der Algorithmus selbst ist Abb. 4.3 zu entnehmen. Auf eine explizite Aufführung der Funktionsweise wird an dieser Stelle jedoch verzichtet, da eine entsprechende Implementierung dem nachfolgenden Kapitel zu entnehmen ist. Beispiele fernab der Anwendung in Führungsautomaten können Neapolitan (2012, 241ff), Neapolitan et al. (2004, 144ff) bzw. Korb und Nicholson (2004, 62f) entnommen werden. Das nachfolgende Kapitel skizziert nun besagte Beispielimplementierung. Dort wird u. a. gezeigt werden, wie mithilfe des Algorithmus eine militärische Lagefeststellung und Entscheidungsfindung umgesetzt werden kann.

$\pi(x_k)$	$= \beta \sum_{\nu_1,\ldots,\nu_n} P(x_k \mid \nu_1 \wedge \ldots \wedge \nu_n) \prod_i \pi_X(\nu_i)$	(4.1)
$\lambda(x_k)$	$= \prod_{K \in k(X)} \lambda_K(x_k)$	(4.2)
$Bel(x_k)$	$= \alpha \pi(x_k) \lambda(x_k)$	(4.3)
$\lambda_X(\nu_k)$	$= \sum_j \lambda(x_j) \sum_{\nu_i : i \neq k} P(x_j \mid \nu_1 \wedge \ldots \wedge \nu_n) \prod_{i \neq k} \pi_X(\nu_i)$	(4.4)
$\pi_Y(x_k)$	$= \frac{Bel(x_k)}{\lambda_Y(x_k)}$	(4.5)

Tabelle 4.1: Formeln des Belief Propagation Algorithmus

Belief Propagation Algorithmus

Initialisierung der Parameter

1. Setze alle λ-Werte, λ-Nachrichten und π-Nachrichten auf 1.
2. Für alle Wurzelknoten $\mathcal{V}$, für alle Ausprägungen $\nu_1, \ldots, \nu_n$, setze für $1 \leq j \leq m$:

$$\pi(\nu_j) = P(\nu_j)$$

3. Für alle Wurzelknoten $\mathcal{V}$, berechne neuen Belief State **[Formel (4.3)]**
4. Für alle Wurzelknoten $\mathcal{V}$, sende eine π-Nachricht an alle Kinder K **[Formel (4.5)]**

Immer wenn eine Variable instantiiert, oder eine λ- respektive π-Nachricht empfangen wird, dann führe die entsprechende Aktualisierungs-Prozedur durch.

Instatiierung einer Variablen

Sei x_j diejenige Ausprägung von X, welche als evident gesetzt wurde. Dann:

1. Setze $Bel(x_j) = 1$ und für alle $i \neq j$ setze $Bel(x_i) = 0$
2. Setze $\lambda(x_j) = 1$ und für alle $i \neq j$ setze $\lambda(x_i) = 0$
3. Sende λ-Nachrichten an alle Väter $\mathcal{V}$ von X **[Formel (4.4)]**

Empfangen einer λ-Nachricht

1. Berechne neue λ-Werte **[Formel (4.2)]**
2. Berechne neuen Belief State **[Formel (4.3)]**
3. Sende λ-Nachrichten an alle Vaterknoten $\mathcal{V}$ **[Formel (4.4)]**
4. Sende π-Nachrichten an alle anderen Kindknoten K **[Formel (4.5)]**

Empfangen einer π-Nachricht

1. Berechne neue π-Werte **[Formel (4.1)]**
2. Berechne neuen Belief State **[Formel (4.3)]**
3. Sende *pi*-Nachrichten an alle Kindknoten K **[Formel (4.5)]**

Wenn Knoten ein Blattknoten ist und instantiiert wurde, dann zusätzlich:

4. Sende λ-Nachrichten an alle anderen Vaterknoten $\mathcal{V}$ **[Formel (4.4)]**

Abbildung 4.3: Kim und Pearls Belief Propagation Algorithmus

4.2 Anwendungsbeispiel

Inhalt dieses Kapitels stellt die konkrete Umsetzung eines Führungsautomaten auf Basis unscharfer Wissensverarbeitung mithilfe eines kausalen Netzwerks dar. Hierfür wird sowohl die Grundidee der Verarbeitung von Sensorinputs, als auch die letztendliche Entscheidungsfindung dieses neuartigen Automaten anhand eines Minimalbeispiels aufgezeigt. Das hierdurch dargestellte Funktionsprinzip wird im nachfolgenden

Kapitel erneut aufgegriffen werden, wo dessen Implementierung innerhalb eines Experimentalsystems näher beschrieben wird.

In Anlehnung an die bereits aufgeführten Beispiele innerhalb dieser Arbeit, wird erneut die bekannte Angriffslage eines Jägerzuges auf eine in Stellung befindliche Infanteriegruppe betrachtet werden. Dabei komme das unscharfe Lagebild gem. Abb. 3.9 zum Tragen. Der Fokus wird hierbei entgegen der bisherigen Beispiele nicht rein auf die Abbildung der unscharfen Lagefeststellung mithilfe einer deklarativen Wissensbasis gesetzt werden, sondern auch Anteile des Entscheidungsfindungsprozesses umfassen. Dem ungeachtet findet erneut keine nähere Beschreibung einer prozeduralen Wissensbasis Erwähnung. Als mögliche durch den Jägerzug wahrnehmbare Zustände seien sowohl das Erreichen eines designierten Zielpunktes, als auch ein etwaiger Feindkontakt verfügbar. Als Zielpunkt sei die Anfangs noch vermutete Position der roten Infanteriegruppe gesetzt. Ein Feindkontakt herrsche in diesem Zusammenhang immer dann vor, wenn sich die Infanteriegruppe innerhalb des sichtbaren Bereichs des Jägerzugs befindet. Diese Zustände seien mittels der Variablen A für das Erreichen der Zielposition und B für die eigene Wahrnehmung eines Feindes stellvertretend bezeichnet. Als Ausprägung der jeweiligen Variablen gelte $A := \{erreicht; \neg erreicht\}$ und $B := \{Feindkontakt; \neg Feindkontakt\}$, wobei die Ausprägung der jeweiligen Variable X mittels x_i für die Indexposition i gekennzeichnet wird. Als stellvertretender Zustand für die konkrete Lagebeurteilung gelte die imminente Gefahr C für den Jägerzug, für dessen Ausprägungen $C := \{Gefahr; \neg Gefahr\}$ gelte. Zwischen der Gefahr und dem Erreichen der Zielposition bzw. des Vorliegens eines Feindkontakts herrsche ein kausaler Zusammenhang insofern vor, dass sowohl das Erreichen der Zielposition, als auch das Vorliegen eines Feindkontakts von der unmittelbaren Gefahr für den Jägerzug abhängt. Logisch gesehen herrschen somit Abhängigkeiten $\mathbf{C} \rightarrow \mathbf{A}$ und $\mathbf{C} \rightarrow \mathbf{B}$ vor, welche zusammenfassend zu $\mathbf{A} \leftarrow \mathbf{C} \rightarrow \mathbf{B}$ vereinigt werden können.

Aus dieser festgelegten Abhängigkeitbeziehung lassen sich nun die für weitere Berechnungen benötigten bedingten Wahrscheinlichkeiten $P(A|C)$, $P(B|C)$, bzw. die Wahrscheinlichkeitsverteilung für $P(C)$ folgern. Diese seien wie folgt festgelegt:

$$\begin{array}{lllll} P(a_1|c_1) = 0.1 & P(a_2|c_1) = 0.9 & P(b_1|c_1) = 0.999 & P(b_2|c_1) = 0.001 & P(c_1) = 0.5 \\ P(a_1|c_2) = 0.7 & P(a_2|c_2) = 0.3 & P(b_1|c_2) = 0.1 & P(b_2|c_2) = 0.9 & P(c_2) = 0.5 \end{array}$$

Die hier dargestellten Wahrscheinlichkeiten wurden subjektiv derart gewählt, dass diese zwei Phänomene konkret widerzuspiegeln vermögen:

1. Die Wahrscheinlichkeit eines Feindkontaktes ist sehr hoch, wenn sich der Jägerzug in Gefahr befindet. In derselben Situation ist die Wahrscheinlichkeit die Zielposition nicht erreicht zu haben ebenfalls hoch.

2. Die Wahrscheinlichkeit eines Feindkontaktes ist gering, wenn sich der Jägerzug nicht in Gefahr befindet. Hier ist die Wahrscheinlichkeit die Zielposition erreicht zu haben eher hoch.

Da sowohl die eigene Position, als auch die Wahrnehmung eines Feindkontakts primär mittels eines Sichtsensors abgedeckt werden können, ist stets auch immer eine konkrete Ausprägung der Variablen A und B evident. In der in Abb. 3.9 dargestellten Ausgangslage bedeutet dies konkret, dass bspw. $A = a_2$ und $B = b_2$ gilt.

Zur Berechnung des letztendlich hieraus resultierenden Belief State von C, wird nun der zum Abschluss des letzten Kapitels aufgeführte Belief Propagation Algorithmus aus Abb. 4.3 exemplarisch verwendet werden. Die gegebene Abhängigkeitsbeziehung der Variablen A, B und C wird dafür nun in ein kausales Netzwerk überführt, welches der folgenden Abb. 4.4 entnommen werden kann.

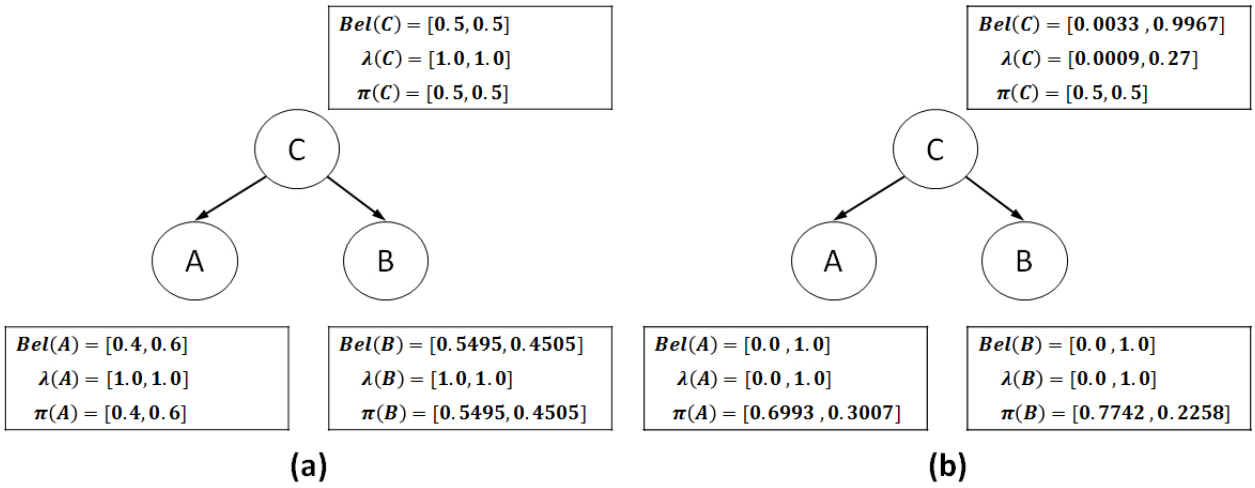


Abbildung 4.4: Initialisiertes (a) und instantiiertes (b) Beispielnetz

Dem Algorithmus folgend wird das Netzwerk, mithilfe der entsprechenden Prozeduren, in einem ersten Schritt initialisiert. Die hierdurch resultierende Belegung der Parameter ist Abb. 4.4 (a) zu entnehmen. Da sowohl konkrete Ausprägungen für A, als auch für B zum Startzeitpunkt evident sind, werden diese anschließend im initialisierten Netz für deren zugehörigen Variablen instantiiert. Die entstehende Belegung der Parameter zeigt für diesen Fall Abb. 4.4 (b). Auf Details der Berechnungen zu Initialisierung und Instantiierung wird an dieser Stelle zugunsten der besseren Übersichtlichkeit verzichtet. Zur Verdeutlichung des Algorithmus – und aus Gründen der Vollständigkeit – können jedoch die einzelnen Berechnungsschritte Anhang A.3 entnommen werden.

Wie den Belief States aus Abb. 4.4 zu entnehmen ist, schließt das kausale Netzwerk im Rahmen der Ausgangslage darauf, dass sich der blaue Jägerzug mit einer an Sicherheit grenzender Wahrscheinlichkeit in keiner imminenten Gefahr befindet. In diesem Fall soll der Automat zur Konklusion kommen, dass ein Marsch des Jägerzugs in Richtung des designierten Zielpunkts die erfolgversprechendste Handlungsoption darstellt. Als ein geeigneter Mechanismus erweist sich hierfür die Verwendung von Nutzwerten, welche optionsbedingt mit den Eintrittswahrscheinlichkeiten ein oder mehrerer Ausprägungen der Variablen verbunden werden. In Kombination ergeben Nutzwert und Eintrittswahrscheinlichkeit den erwarteten Nutzen einer Handlungsoption, der, aufsummiert über alle möglichen Kombinationen der Repräsentationen erwähnter Variablen, den Nutzen einer Handlungsoption in der gegenwärtig wahrgenommenen Lage bemisst. Diejenige Option mit dem höchsten Nutzen stellt dann die für den Automaten adäquateste Entscheidung dar. Dieses Vorgehen basiert auf der Auswahl auf Grundlage des maximal erwarteten Nutzwerts (engl. Maximum Expected Utility, kurz: MEU) und stellt ein häufig für die Ermittlung von Handlungsoptionen

unter Unsicherheit referenziertes Vorgehen dar – vgl. hierzu bspw. (Pearl, 1988, 289ff; Neapolitan, 2012, 371; Korb und Nicholson, 2004, 15; Nielsen und Jensen, 2007, 283ff).

Es beschreibe nun entsprechend $U(a_x|r_k)$ den Nutzwert einer Handlungsoption a_x für eine Formel r_k von Variablenbelegungen der Wurzelknoten und es sei

$$EU(a_x|\mathfrak{e}) \quad = \sum_{r_k} P(r_k|\mathfrak{e})U(a_x|r_k)$$

der erwartete Nutzen der Option gegeben einer Evidenz $\mathfrak{e}$. Dann heißt eine Option a die erfolgversprechendste Handlungsoption, wenn gilt

$$a \quad = max_{a_x}(EU(a_x|\mathfrak{e})).$$

Da im Beispiel nur ein Wurzelknoten definiert ist, können lediglich nur atomare Formeln $r_1 = c_1$ und $r_2 = c_2$ gebildet werden. Erste mögliche Handlungsoption stelle der im vorhergehenden Paragraphen bereits aufgeführte Marsch zur designierten Zielposition dar, welche als Option $a_1 :=$ „*Marsch zur Zielposition*“ hinterlegt wird. Als Alternativoption sei dem Automaten zusätzlich das Bekämpfen des Feindes, d. h. $a_2 :=$ „*Bekämpfe erkannten Feind*“ hinterlegt. Unter keiner vorherrschenden imminenten Gefahr – d. h. für $C := c_2$ – wird ein hoher Nutzwert für die Option a_1 gesetzt. Dem entgegen herrsche zur selben Zeit ein niedriger Nutzwert für a_2 vor. Subjektiv seien diese Festlegungen über die konkreten Nutzwerte $U(a_1|r_2) = 0.8$ und $U(a_2|r_2) = 0.2$ wiedergegeben, womit im beschriebenen Fall für den Automaten ein Marsch zur Zielposition am vielversprechendsten erscheint, jedoch auch in Unkenntnis der Feindposition das Bekämpfen dieses Feindes plausibel sein kann. Im Falle einer imminenten Gefahr – d. h. für $C := c_1$ – sei dies genau umgekehrt, d. h. $U(a_1|r_1) = 0.2$ und $U(a_2|r_1) = 0.8$. Die Berechnung des erwarteten Nutzens ergibt hierbei gemäß des durch den Automaten erfassten Lagebildes aus Abb. 4.4 (b):

$$\begin{aligned}
EU(a_1|\mathfrak{e}) \quad &= \sum_{r_k} P(r_k|\mathfrak{e})U(a_x|r_k) \\
&= P(c_1|\mathfrak{e}) * U(a_1|c_1) + P(c_2|\mathfrak{e}) * U(a_1|c_2) \\
&= 0.0033 * 0.2 + 0.9967 * 0.8 = 0.798 \\
EU(a_2|\mathfrak{e}) \quad &= \sum_{r_k} P(r_k|\mathfrak{e})U(a_x|r_k) \\
&= P(c_1|\mathfrak{e}) * U(a_2|c_1) + P(c_2|\mathfrak{e}) * U(a_2|c_2) \\
&= 0.0033 * 0.8 + 0.9967 * 0.2 = 0202
\end{aligned}$$

Somit ist $a = max_{a_x}(EU(a_x|\mathfrak{e})) = a_1$, womit der Automat wie beabsichtigt als passendste Handlungsoption den Marsch zur Zielposition bestimmt.

Dass dieses Vorgehen zur Ermittlung der passendsten Option auch im Falle einer Lageänderung zielführend ist, wird nun noch nachfolgend aufgezeigt. Wird hierzu bspw. der blaue Jägerzug frontal und in gegenseitiger Sichtweite zur roten Infanteriegruppe gestellt, so sorgt dies dafür, dass der evidente Zustand von $B = b_2$ auf $B = b_1$ wechselt. Diese veränderte Lage ist entsprechend Abb. 4.5 zu entnehmen. Durch den Wechsel der Belegung kommt es zur erneuten Instantiierung der Variable B, was entsprechend des Belief Propagation Algorithmus den Belief State von C verändert. Das Ergebnis auf die Parameter des kausalen

Netzwerks ist hierfür gesammelt Abb. 4.5 zu entnehmen. Wie auch im Falle der vorhergehenden Berechnungen wird an dieser Stelle auf eine detaillierte Aufführung der Zwischenergebnisse und -schritte verzichtet. Diese können jedoch abermals in Anhang A.3 konsultiert werden. Das Ergebnis zeigt hierbei, dass der Wechsel in der evidenten Repräsentation von B eine (de facto) Invertierung der Eintrittswahrscheinlichkeiten von C im Vergleich zur Ausgangslage bewirkt. Die Auswirkungen auf die erwarteten Nutzen der jeweiligen Handlungsoptionen ist daher erwartungskonform:

$$\begin{aligned} EU(a_1|\mathfrak{e}) &= \sum_{r_k} P(r_k|\mathfrak{e})U(a_x|r_k) \\ &= P(c_1|\mathfrak{e}) * U(a_1|c_1) + P(c_2|\mathfrak{e}) * U(a_1|c_2) \\ &= 0.9677 * 0.2 + 0.0323 * 0.8 = 0.2192 \\ EU(a_2|\mathfrak{e}) &= \sum_{r_k} P(r_k|\mathfrak{e})U(a_x|r_k) \\ &= P(c_1|\mathfrak{e}) * U(a_2|c_1) + P(c_2|\mathfrak{e}) * U(a_2|c_2) \\ &= 0.9677 * 0.8 + 0.0323 * 0.2 = 0.7806 \end{aligned}$$

Die passendste Handlungsoption $a = max_{a_x}(EU(a_x|\mathfrak{e})) = a_2$ legt es somit aus Sicht des Automaten offen, den erkannten Feind zu bekämpfen.

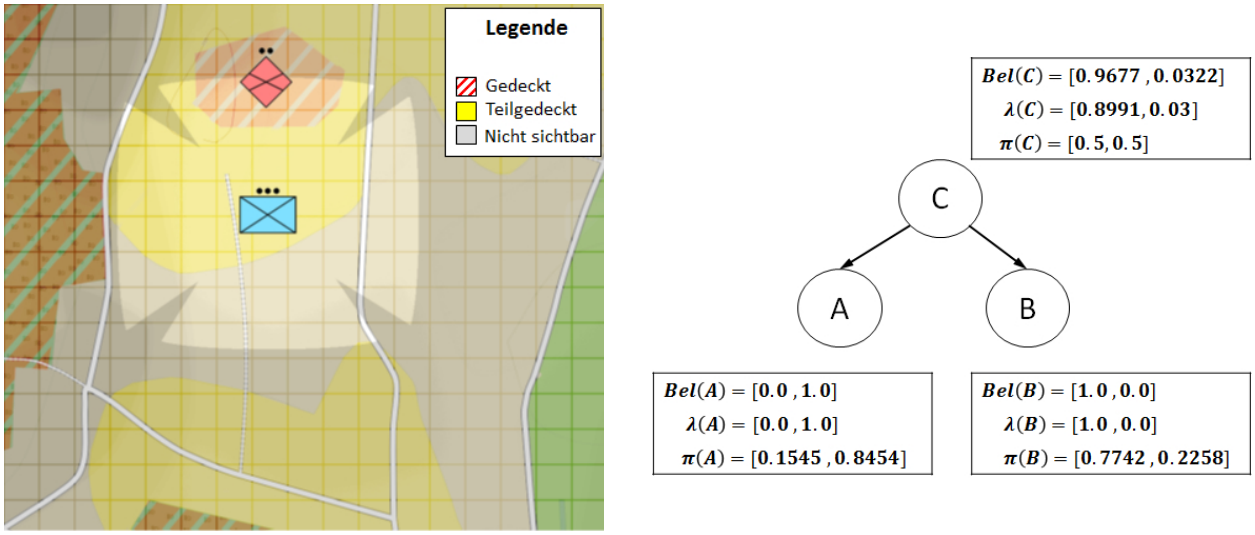


Abbildung 4.5: Änderung der Lage und die resultierende Parameterbelegung

Abschließend sei auf der Basis der bisher dargestellten Funktionsweise die gesamtheitliche funktionale Beschreibung des Führungsautomaten dargestellt. Wie Abb. 4.6 hierfür zu entnehmen ist, weist ein kausaler Führungsautomat weiterhin den grundsätzlichen Aufbau eines intensionalen, hybriden wissensbasierten Ansatzes auf. In Abweichung zum allgemeinen Schema benötigt der kausale Führungsautomat jedoch im Vorhinein erzeugte Instanzen aller möglichen Handlungsoptionen (hier als Array von Optionen) und Nutzwerte (hier als Array von Nutzwert-Tupeln). Im weiteren Verlauf wird nun davon ausgegangen, dass das innerhalb der Wissensbasis eingebettete kausale Netz zum Zeitpunkt des Aufrufs des Führungsautomaten bereits initialisiert wurde. Der Führungsautomat ermittelt dann in einem ersten Schritt den aktuellen Zustand der untergebenen Einheiten (getCPerception) und weist das Ergebnis dieser Ermittlung an die

deklarative Wissensbasis zur Auswertung (queryPerc) weiter. Der aktuell messbare Zustand sei hierbei als Array ganzzahliger Werte definiert, welche stellvertretend für die jeweilig wahrgenommene Ausprägung der Blattknoten stehen. Mittels einer Abfrage über die momentan gesetzten Ausprägungen innerhalb der Blattknoten ermittelt der Automat in Folge, ob sich eine Änderung ergeben hat. Ist dies der Fall, so wird für das kausale Netzwerk die entsprechende Evidenz initialisiert. Die Belief States der Wurzelknoten werden in Folge in ein Array geschrieben, welches wiederum als resultierender Zustand der unterstellten Entitäten gesetzt wird. Da dem Automaten hierdurch die konkreten Eintrittswahrscheinlichkeiten bekannt sind, berechnet dieser mithilfe der hinterlegten Nutzwerte den erwarteten Nutzen der jeweiligen Optionen (computeUtils). Diejenige Handlungsoption mit dem höchsten erwarteten Nutzen wird anschließend ausgewählt und entsprechend dem in der Wissensbasis hinterlegten Schema in eine Folge von Aktionen überführt (planAction).

Dies beschließt die funktionale Beschreibung des Führungsautomaten. Die Umsetzung dieses Führungsautomaten ist in kurzer Beschreibung dem folgenden Kapitel zu entnehmen. In Kapitel 6 wird dieser Ansatz dann entsprechend in einen Vergleich mit einem regelbasierten Automaten gestellt werden.

Programm 8 Kausaler Führungsautomat

```
KnowledgeBase: KB;
Array of Action: seq;
Array of Options: options;
Array of UseValRec: vals;
function C2(blue: Entity) returns Action
    State: current;
    Action: next;
    Array of Double: utils;
    current ← KB.queryPerc(blue.getCPerception());
    if seq == ∅ then
        utils ← computeUtils(current, vals);
        seq ← planAction(blue, options[getMaxIndex(utils)]);
    end if
    next ← KB.queryAct(seq.pop());
    return next;
end function
```

Aktualisierung der Zustände

```
CausalNetwork: net;
function QUERYPERC(a: Array of Integer) returns State
    State: state;
    for Integer: i = 1 to net.leaves.length do
        if (a[i] == null) || ¬(net.leaves[i].rep == a[i]) then
            instantiate(net, i, a[i]);
        end if
    end for
    for Integer: j = 1 to net.roots.length do
        state[j] = net.roots[i].belief;
    end for
    return state;
end function
```

Abbildung 4.6: Beispiel eines kausalen Führungsautomaten

5 Experimentalsystem

Dieses Kapitel umfasst die Beschreibung des verwendeten Experimentalsystems, welches zum Vergleich von Führungsautomaten für diese Arbeit entwickelt wurde. Im Detail werden Aufbau und Funktionsweise der Kernkomponenten des Experimentalsystems in einem Ausmaß erläutert, damit Einblicke hinter den formalen Aufbau der Experimente des nachfolgenden Kapitels möglich werden.

Das Experimentalsystem ist als statisches objektorientiertes Prüfsystem für die Ermittlung der Entscheidungen eines Führungsautomaten in spezifischen Situationen entworfen worden. Es besitzt keinen dezidierten Zeitablaufmechanismus, der selbstständig und fortlaufend die Simulationszeit kontinuierlich oder diskret voranschreiten lässt. Der Nutzer des Systems ist jedoch jederzeit selbst in der Lage, Änderungen an den Parametern der platzierten Simulationsobjekte vorzunehmen. Somit ist es möglich Objekte in fest vorgegebene Situationen zu versetzen, um die resultierende Entscheidung eines Führungsautomaten in eben jener Situation zu ermitteln. Dadurch weist das Experimentalsystem im Kern ein rein deterministisches Verhalten auf, welches durch einen Nutzer mittels einer entsprechenden Eingabe initiiert wird. Alle diese Eingaben werden mittels Mausklick getätigt. Die Möglichkeiten der Beeinflussung sind hierbei:

- Verschieben der Entitäten mittels Drag & Drop (Linke Maustaste)
- Verändern der Sichtrichtung mittels Drag & Drop (Rechte Maustaste)
- Aufruf des Kontextmenüs (Mittlere Maustaste)

Das Kontextmenü bietet dabei diverse Funktionalitäten zur Beeinflussung des Experimentaufbaus und der Parameter ausgewählter Simulationsobjekte. Darüber hinaus bietet dieses die Möglichkeit die Regelbasis eines Führungsautomaten abzurufen, die Belegung der Variablen einzusehen respektive die in der momentanen Situation ermittelte Handlungsoption auszugeben.

Im verwendeten Entwicklungsstand unterstützt das Experimentsystem eine feste Zeichenfläche von 800x600 Bildpunkten, welche beliebig mit einem entsprechenden nicht-georeferenzierbaren Geländeraster belegt werden kann. Eine dynamische Anpassung an größere oder kleinere Raster erfolgt nicht. Die Zeichenfläche stellt zeitgleich das Simulationsfeld dar, auf welchem Objekte in einem kartesischen Koordinatensystem gesetzt werden können. Die Skalierung des Koordinatensystems erlaubt es hierbei Koordinaten mit 64-Bit Fließkommazahlen abzubilden. Die Aktualisierung der auf den Zeichenfeld dargestellten Objekte erfolgt zu festen Zeitpunkten: Bei Veränderung der Position – und damit der Koordinate – eines Objekts, oder allgemein bei Änderung eines internen Objektparameters. Unterschieden wird dabei zwischen statischen und dynamischen Simulationsobjekten, d. h. Objekten, welche entweder hintergründig und ohne Interaktionsmöglichkeit auf dem Simulationsfeld gesetzt wurden und Objekten, die durch direkte Interaktion beeinflusst bzw. deren Position verändert werden kann. Statische Objekte stellen in diesem Zusammenhang Bewuchs bzw. Bebauung und dynamische Objekte militärische Akteure dar. Beide Objekttypen werden mittels einer

graphischen Repräsentation auf dem Simulationsfeld für den Nutzer sichtbar dargestellt. Details sind hierzu den folgenden Unterkapiteln zu entnehmen, wo nun folgend die einzelnen Komponenten des Systems näher erläutert werden. Begonnen wird hierzu mit der Geländeabbildung in Kap. 5.1. Diesem Abschnitt folgen Erläuterungen zur Abbildung der militärischen Akteure in Kap. 5.2, der Umsetzung der Sensorik in Kap. 5.3 und der Beschreibung der logischen Schlussmechanismen in Kap. 5.4.

5.1 Gelände, Bebauung und Bewuchs

Der folgende Abschnitt umfasst die konkrete Umsetzung in der Abbildung von Gelände und statischen Simulationsobjekten im Experimentalsystem. Die einzelnen implementierten Geländetypen und deren Auswirkung auf dynamische Simulationsobjekte werden erläutert. Abschluss stellt die kurze Aufführung des Sonderstatus statischer Simulationsobjekte dar.

Innerhalb des Experimentalsystems erfolgt die Abbildung konkreter Geländetypen auf Grundlage frei definier- und stapelbarer, nicht sichtbarer Polygone. Diese bilden vier grundsätzliche Typen von Gelände ab: Grasland, Ackerland, Waldland und urbanes Gelände. Standardmäßig umfasst das komplette Simulationsfeld den Geländetyp Grasland, womit der auf dem Bildschirm sichtbare Bereich grundsätzlich mindestens durch ein Polygon ausgefüllt wird. Weitere Geländetypen werden durch Hinzufügen neuer Polygone implementiert. Liegen dabei zwei Polygone übereinander, so wird vom größeren Polygon ausgehend mit dem kleineren Polygon die Differenzmenge gebildet, wodurch an einem bestimmten Ort immer nur genau ein Geländetyp vorherrschen kann. Das Bilden der Differenzmenge erfolgt hierbei umgekehrt zur Implementierungsreihenfolge, womit im einfachsten Fall durch das Hinzufügen eines zweiten Geländetyps lediglich die Differenzmenge zwischen dem Standard-Grasland-Polygon und dem neuen Geländepolygon gebildet wird.

Die Definition eines neuen typisierten Geländeabschnitts ist entsprechend im Quellcode der Zeichenfläche hinterlegt. Grundsätzlich bietet keiner der definierten Geländetypen eine Einschränkung für das Setzen eines übergeordneten Objekts. Eine Abbildung von Höheninformation erfolgt nicht. Nichtsdestotrotz weisen Geländetypen einzelnen statischen Objekten einen Deckungswert im Bereich zwischen 0 und 2 zu. Hierbei steht der Wert 0 stellvertretend für „Ungedeckt“, 1 für „Teilgedeckt“ und 2 für „Gedeckt“[1]. Die Werte wurden aufgrund der zu erwartenden Eignung des Geländes für die Operations- und Auftragsdurchführung in Abhängigkeit des militärischen Einheitentyps subjektiv vergeben und sind Tab. 5.1 zu entnehmen. Eine direkte Umsetzung in der Darstellung von Deckungen durch Mauern, Gräben oder ähnliches erfolgt nicht, wodurch der gesetzte Geländetyp allein für die situativ vorliegende Deckung verantwortlich ist.

1 In diesem Zusammenhang sei auf die Bedeutung der Deckungsbegriffe hingewiesen: Ungedeckt bedeutet, dass eine Einheit ohne Einschränkungen wirken und bewirkt werden kann. Teilgedeckt bedeutet, dass eine Einheit eingeschränkt wirken kann, jedoch auch nur eingeschränkt bewirkt werden kann. Gedeckt ist eine Einheit genau dann, wenn diese nicht wirken und gleichzeitig auch nicht bewirkt werden kann.

Geländetyp	Leichte Infanterie	Schwere Infanterie	Leicht Gepanzert	Schwer Gepanzert
Grasland	0	0	0	0
Ackerland	1	1	0	0
Waldland	1	1	1	2
Urbanes Gelände	1	1	1	0

Tabelle 5.1: Deckungswerte je Gelände- und Einheitentyp

Im Bedarfsfall kann eine explizite Abbildung von Bebauung und Bewuchs erfolgen, welche anhand statischer Simulationsobjekte dargestellt werden. Wie Geländeabschnitte werden diese mittels frei definierbarer, jedoch sichtbarer Polygone realisiert und auf die gleiche Art hart-kodiert innerhalb des Quellcodes der Zeichenfläche definiert. Sie werden auf dem Simulationsfeld grundsätzlich als rot umrandete, grau gefüllte Polygone dargestellt.

Statische Objekte behindern grundsätzlich das Sichtfeld eines dynamischen Objekts, was entsprechend in Kap. 5.3 näher erläutert wird. Statische Objekte nehmen somit einen Sonderstatus in der Abbildung von Gelände ein, da diese nicht nur explizit auf dem Simulationsfeld abgebildet und sichtbar sind, sondern auch aktiv die Parameter der dynamischen Objekte beeinflussen.

5.2 Auflösung und Repräsentation militärischer Akteure

Nachfolgend wird die konkrete Umsetzung in der Realisierung dynamischer Simulationsobjekte des Experimentalsystems näher erläutert. Hierfür werden sowohl die gewählte Auflösung der Abbildung militärischer Akteure, als auch die während der Simulation für den Führungsprozess verwendeten Parameter beschrieben.

Militärische Akteure sind als Agenten umgesetzt. Dies erfolgte wissentlich unter der Prämisse, dass Agenten – wie der Definition aus Kapitel 2.1 zu entnehmen ist – für die Umsetzung von Individualverhalten und damit auch zur Repräsentation von Individuen an sich geeignet sind. Im Experimentalsystem dienen diese als repräsentative, dynamische Objekte zur Abbildung von Teileinheiten und Teileinheitsführern. Jeder Agent bildet daher entweder Züge, Gruppen oder Trupps respektive deren äquivalentes Führungselement ab. Somit existieren im Experimentalsystem zwei Typen von Agenten: Teileinheits- und Führungsagenten. Teileinheitsagenten stellen den primären Akteur innerhalb der Simulation dar, deren Aufgabe es ist, feindliche (Teileinheits-)Agenten aufzuklären und das Aufklärungsergebnis, zusammen mit dem eigenen Zustand, an ihre übergeordneten Führer zu melden. Hierfür besitzen diese interne Parameter, welche insbesondere die Ermittlung des eigenen Zustands ermöglichen:

- **Zugewiesener militärischer Einheitentyp**

  Der zugewiesene Einheitentyp ist ein Klassifikator, der den Agenten entweder als leichte Infanterie, schwere Infanterie, leicht gepanzerte oder schwer gepanzerte motorisierte Kräfte ausweist. Dies dient vorrangig zur Bestimmung des Deckungswertes auf dem gegenwärtigen Geländeabschnitt.

- **Zugehörige Fraktion**

  Das System unterstützt drei Fraktionen: freundlich (Blau), feindlich (Rot) und neutral (Grün). Die Zuweisung einer Fraktion ermöglicht eine Filterung der Aufklärungsergebnisse.

- **Zugewiesenes taktisches Symbol**

  Das zugewiesene taktische Symbol stellt die graphische Repräsentation des Simulationsobjektes dar. Es bestimmt Position und räumliche Ausdehnung des Agenten auf dem Simulationsfeld. Zusätzlich dient es als Interaktionsfläche für die Änderungen der Parameter mithilfe des Kontextmenüs.

- **Initial gesetzte Stärke**

  Repräsentativer Wert der Teileinheitsstärke ohne Ausfälle. Die initiale Stärke wird zu Beginn der Simulation gesetzt und während eines aktiven Simulationslaufs nicht verändert.

- **Aktuell gesetzte Stärke**

  Repräsentativer Wert für die augenblickliche Teileinheitsstärke. Die aktuelle Stärke wird zu Beginn der Simulation auf den Initialwert gesetzt und kann mithilfe des Kontextmenüs ad hoc reduziert werden.

- **Liste vorhandener Sensoren**

  Dem Agenten werden zu Beginn der Simulation Sensoren für die Wahrnehmung seiner Umwelt zugewiesen. Im Experimentalsystem ist hier standardmäßig ein optischer Sensor gesetzt, womit eine modellhafte Wahrnehmung aufgrund von Sicht möglich ist.

- **Liste gesichteter Feinde**

  Die mithilfe der zugewiesenen Sensoren wahrgenommenen Agenten werden innerhalb einer Liste gespeichert. Standardmäßig gilt für die Listen, dass diese für Agenten der blauen Fraktion alle lokal wahrgenommenen Agenten der roten Fraktion und umgekehrt speichern.

Die Weitergabe der Parameterwerte erfolgt ohne weiteres Zutun der jeweiligen Teileinheitsagenten. Ihnen übergeordnet ist jeweils ein Führungsagent, welcher die Parameterwerte der ihm untergebenen Teileinheiten auf Anfrage des Nutzers aggregiert und den Führungsprozess durch einen Führungsautomaten initiiert. Die dabei durch den jeweiligen Führungsautomaten benötigten Informationen werden durch den zugehörigen Führungsagenten in Form von internen Parametern gehalten:

- **Liste der Untergebenen**

  Der Agent führt alle auf dem Simulationsfeld gesetzten Elemente einer vordefinierten Fraktion. Die hierfür zutreffenden Teileinheiten werden zu Beginn der Simulation und bei Hinzufügen eines Elements durch den Nutzer automatisch aktualisiert.

- **Liste der Vorgesetzten**

  Eine generelle Forderung zur Abbildung militärischer Führungsprozesse sind kaskadierende Führungsstrukturen, d. h. dass ein Auftrag vom höheren Vorgesetzten an den ihn unterstellten Bereich verfeinert weitergegeben wird. Dies wird im momentanen Entwicklungsstand des Experimentalsystems jedoch nicht unterstützt, wodurch die Liste der Vorgesetzten standardmäßig als leer initialisiert wird.

- **Liste der möglichen Zustände**

  Zur Feststellung der eigenen Lage ist es unverzichtbar, dem Führungsautomaten eine Liste designierter Zustände zuzuweisen, welche eine entsprechende Ableitung aus den aggregierten Parametern des untergeordneten Bereichs erlaubt. Dies bildet den Grundstock für weitere Schlussfolgerungen mithilfe der Wissensbasis.

- **Zugeordnete Wissensbasis**

  Die zugeordnete Wissensbasis umfasst axiomatisch formuliertes Wissen für die Ableitung konkreter Handlungsoptionen auf Basis der momentan erfassten Lage.

- **Designierte Zielposition**

  Mit Hilfe einer zugeordneten Zielposition, ist der Agent in der Lage auftragsgerecht zu entscheiden, ob ein positionsspezifisches Zwischenziel erreicht wurde. Diese wird dem Führungsagenten zu Beginn der Simulation in Form von Koordinaten festgesetzt und während der Simulation, aufgrund des statischen Versuchsaufbaus, nicht verändert.

- **Momentane Option**

  Die momentan durch Absolvieren des Führungsprozess ermittelte Handlungsoption wird in einer entsprechenden Variable zwischengespeichert. Hierdurch kann auf diese leichter zugegriffen bzw. ein Vergleich der Handlungsoptionen zweier unterschiedlicher Führungsagenten durchgeführt werden.

Führungsagenten unterscheiden sich untereinander anhand der hinterlegten Wissensbasis. Das Experimentalsystem unterscheidet hierbei zwischen regelbasierten und kausalen Führungsagenten, welche jedoch – mit Ausnahme der zugeordneten Wissensbasis – einen identischen Aufbau und Funktionsumfang besitzen. Die Zuordnung einer Wissensbasis findet dabei zur Initialisierung eines Führungsagenten statt, was – simultan zu den Teileinheitsagenten – hart-kodiert innerhalb des Quellcodes des Simulationsfelds definiert ist. Anders als Teileinheiten werden Führer jedoch nicht explizit auf das Simulationsfeld mittels eines repräsentativen Symbols gesetzt. Folglich können diese fernab einer Abfrage ihrer Parameter durch den Nutzer zur Laufzeit der Simulation nicht beeinflusst werden.

5.3 Sensorik

Der nachfolgende Abschnitt befasst sich mit der Umsetzung von Sensoren im Experimentalsystem. Hierbei wird deren Aufbau kurz erläutert, die Funktionsweise näher dargestellt und aufgeführt, wie Sensoren auf dem Simulationsfeld andere Simulationsobjekte zu detektieren vermögen bzw. die Bestimmung der relativen Position des Sensorträgers zu einem anderen Agenten ermöglichen.

Sensoren stellen im Experimentalsystem modulare Bausteine eines Agenten dar, die zur Initialisierung einer Teileinheit zugeordnet werden. Während der eigentliche Sensor an sich graphisch nicht abgebildet wird, kann der durch den Sensor abgedeckte Bereich optional jedoch sichtbar geschaltet werden. Möglich wird dies dadurch, dass jeder Sensor den durch ihn abgedeckten Bereich mittels eines Polygons zwischenspeichert und dieses auf dem Simulationsfeld als halb-durchsichtige gelbe Fläche darstellt. Größe und Form dieser Fläche werden dabei sowohl gleichermaßen durch den jeweiligen Sensortyp, als auch durch die eingestellte maximale Sensorreichweite bestimmt. Das Experimentalsystem unterstützt im momentanen Entwicklungsstadium lediglich optische Sensoren, welche mittels Sichtkegel abstrahiert werden. Da das System keine Höheninformationen abbildet, wird dieser Sichtkegel zusätzlich zu einem zweidimensionalen Kegelsegment vereinfacht. Jedes gespeicherte Abdeckungspolygon stellt auf dieser Grundlage ein Kegelsegment dar, dessen Länge durch die Sensorreichweite vorgegeben wird. Die Spitze des Kegels wird dabei innerhalb des Mittelpunktes der graphischen Repräsentation des Sensorträgers, d. h. des jeweiligen Agenten festgesetzt.

Da optische Sensoren per se gerichtete Sensoren darstellen, muss diesen eine konkrete Richtung vorgegeben werden. Zum Initialisierungszeitpunkt stellt dies der Mittelpunkt des Simulationsfeldes dar. Dies kann jedoch – wie bereits in der Einleitung dieses Kapitels beschrieben wurde – mithilfe eines rechten Mausklicks beliebig verändert werden. Das System berechnet dabei im Falle jedes Wechsels der Parameterbelegung des Zielpunktes, d. h. initial zum Programmstart und mit jeder manuellen Änderung des Parameters, das zutreffende Abdeckunspolygon. Grundsätzlich wird dies auf Basis einfacher Vektoroperationen bewerkstelligt, welche drei wesentliche Schritte umfassen:

1.) Berechne Vektor v' und die hieraus resultierende Steigung vom Mittelpunkt des Sensorträgers zum Zielpunkt :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x_Z - x_M \\ y_Z - y_M \end{pmatrix}, m = \frac{y'}{x'}$$

2.) Stauche bzw. Strecke die Länge des Vektors v' auf die maximale Sensorreichweite r:

$$x_k = \pm\sqrt{\frac{r^2}{1+m^2}}$$
$$y_k = m * x_k$$

3.) Berechne alle Eckpunkte E_Θ des Abdeckungspolygons durch Drehen des Vektors um den Winkel $\Theta \in [-45°, 45°]$ in Schritten zu je einem Grad:

$$\begin{pmatrix} x_{E_\Theta} \\ y_{E_\Theta} \end{pmatrix} = \begin{pmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} x_M \\ y_M \end{pmatrix}$$

Ein beispielhaftes Resultat dieser Schritte ist zur besseren Ersichtlichkeit Abb. 5.1 (a) zu entnehmen. Wie es in Kapitel 5.1 jedoch Erwähnung fand, kann die Abdeckung eines Sensors durch statische Simulationsobjekte eingeschränkt werden. In ähnlicher Form wie nicht-durchsichtige Gegenstände in der Realität aufgrund deren Lichtundurchlässigkeit den dahinterliegenden Bereich abdecken, endet auch innerhalb des Experimentalsystems der Abdeckungsbereich eines Sensors an der Außenfläche[2] eines statischen Simulationsobjektes. Die Einschränkung der Abdeckung entspricht hierbei gemäß des realen Vorbilds einer Verschiebung der äußeren Eckpunkte des Abdeckungspolygons auf die Randfläche des Polygons des statischen Simulationsobjektes. Hierdurch ist der theoretisch noch hinter dem statischen Objekt verbleibende Abdeckungsbereich nicht mehr Teil des Polygons des Sensors, wodurch ein blinder – d.h. ein nicht-einsehbarer Bereich – für den Sensorträger entsteht.

Umgesetzt wird dies durch die Überprüfung, ob eine ununterbrochene Verbindung zwischen Mittelpunkt des Sensorträgers und den in Teilschritt 3 berechneten Eckpunkten des Abdeckungspolygons besteht. Schneidet eine derartige Verbindung dabei das Polygon eines statischen Objektes, so wird der Eckpunkt auf den Schnittpunkt zwischen der Verbindung und der Polygonrandfläche des Objektes gesetzt. Die Umrisse statischer Objekte werden auf diese Weise mit jedem Schritt abgetastet, bis erneut eine ununterbrochene Verbindung vorliegt. Der im Hintergrund des Objektes gelegene Bereich ist hierdurch nicht mehr Teil des Abdeckungspolygons und damit de facto durch den Sensor auch nicht mehr einsehbar. Dies wird durch Abbildung 5.1 (b) verdeutlicht. In diesem Zusammenhang gilt es noch zu klären, wie Sensoren mithilfe des Abdeckungspolygons Teileinheiten im Experimentalsystem zu detektieren vermögen.

Eine Teileinheit gilt in diesem Sinne als wahrgenommen, wenn diese innerhalb des Abdeckungsbereichs des Sensors liegt. Dies bedeutet im Speziellen, dass für eine Detektion grundsätzlich die auf dem Simulationsfeld sichtbare graphische Repräsentation zu großen Teilen oder vollständig innerhalb des Abdeckungspolygons liegen muss. Hierbei wird pauschal davon ausgegangen, dass die räumliche Ausdehnung der Teileinheit durch die Größe des die graphische Repräsentation umgebenden Rechtecks hinreichend genau approximiert wird und die teilweise oder vollständige Abdeckung des Rechtecks somit eine eindeutige und abgrenzbare Zuordnung der vorliegenden Truppengattung und -stärke zulässt. Eine teilweise Abdeckung bedeutet in diesem Zusammenhang, dass die äußere Kante eines Abdeckungspolygons eine minimale Entfernung[3] zum Mittelpunkt der graphischen Repräsentation besitzt. Somit können Teile einer graphischen Repräsentation im Abdeckungsbereich eines Sensors enthalten sein, führen jedoch nicht automatisch zu einer Detektion der zugeordneten Teileinheit.

2 Halbdurchsichtige oder gar durchsichtige statische Simulationsobjekte sind nicht implementiert. Es wird daher grundsätzlich davon ausgegangen, dass ein statisches Objekt das Sichtfeld eines optischen Sensors im jeweilig auftreffenden Winkel einschränkt.

3 Die momentan implementierte Größe einer graphischen Repräsentation umfasst grob 30x25 Pixel. Die minimal nötige Entfernung des Mittelpunkts zum Abdeckungspolygon wurde auf 10 Pixel festgesetzt.

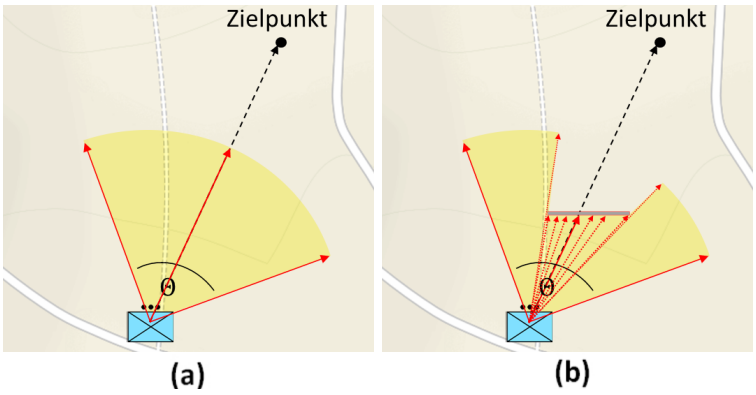


Abbildung 5.1: Bilden des Sichtfeldes (a) ohne und (b) mit Einschränkung

Über die Zwecke der Detektion hinausgehend, ermöglicht die Berechnung des Abdeckungsbereichs zusätzlich noch die Bestimmung der relativen Position des Sensorträgers zu einem anderen Agenten. Hierfür wird ausgenutzt, dass das durch die aufgeführten Berechnungsschritte aufgespannte Abdeckungspolygon den vierten Teil eines Kreises umfasst. Erweitert man dieses um drei gleiche Anteile, so erhält ein Agent hierdurch vier Zonen, welche zur Bestimmung der relativen Position eines Sensorträgers zu besagtem Agenten verwendet werden kann. Die durch das Abdeckungspolygon des Sensors aufgespannte Fläche markiert hierbei das frontal zum Agenten gelegene Gebiet, welches innerhalb des Kreises einen Kreiswinkel $\Theta_{Front} \in [-45°, 45°]$ vor dem Sensorträger aufspannt. Beide hierbei angrenzenden Kreissegmente mit Kreiswinkel $\Theta_{LFlank} \in [-135°, -45°[$ und $\Theta_{RFlank} \in]45°, 135°]$ stehen daher für das jeweils in der linken oder rechten Flanke , das übrige Segment mit Kreiswinkel $\Theta_{Back} \in]135°, 225°[$ bzw. $\Theta_{Back} \in]-225°, -135°[$ für das im Rücken gelegene Gebiet. Dies wird in Abb. 5.2 entsprechend verdeutlicht. Ein Agent befindet sich frontal zum, oder in der Flanke respektive im Rücken des Sensorträgers, wenn dessen Front zu einem der designierten Gebiete hin gerichtet ist. Folgerichtig ist ein Agent in der Flanke oder im Rücken des Sensorträgers für diesen nicht sichtbar.

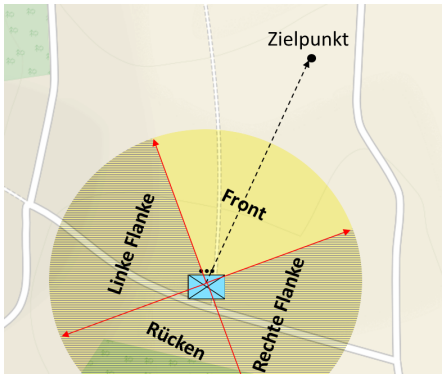


Abbildung 5.2: Relative Positionen eines Agenten

5.4 Logikkomponente und Wissensbasen

Der folgende Abschnitt befasst sich mit der konkreten Umsetzung von logischen Schlüssen und der Implementierung der Führungsautomaten im Experimentalsystem. Das verwendete logische Kalkül, als auch die in diesem Sinne entworfenen Komponenten werden hierfür erläutert. Zwei Unterkapitel erläutern abschließend die grundlegenden logischen Bausteine und generelle Funktionsweise der beiden implementierten Führungsautomaten, wobei hier insbesondere der Fokus auf den Wissensbasen liegt.

Die Logikkomponente des Experimentalsystems dient für Führungsagenten zum Bilden logischer Schlüsse auf Basis der durch die jeweiligen Teileinheiten wahrgenommenen Lageinformationen. Wie auch ein Großteil der bisher in dieser Arbeit aufgezeigten Umsetzungen für Führungsautomaten, basiert die Komponente hierbei auf dem aussagenlogischen Kalkül. Eine Aussage A des Systems hat dabei die Form $A \equiv \varphi R \alpha \varrho$, mit $\varphi, \varrho, \alpha \in \mathbb{R}$ und $R \in \{\leq, <, =, >, \geq\}$. Aussagen müssen vor der Laufzeit der Simulation initialisiert werden, jedoch lassen sich deren Werte φ und ϱ zu jeder Zeit dynamisch verändern. Letzteres geschieht mithilfe von Zuständen, welchen jeweils mindestens eine Aussage zugeordnet ist. Zustände dienen dabei zum Halten der für den Führungsprozess benötigten Lageinformationen. Sie bestehen hierzu aus vier Komponenten:

- **Bezeichner**

  Kurze textuelle Bezeichnung der Situation, welche durch den Zustand abgebildet wird. Dies kann möglicherweise mit anderen Zuständen übereinstimmen, bspw. in der Beschreibung des Deckungszustandes für mehrere Teileinheiten.

- **Typ**

  Der Zustandstyp beschreibt ein Element aus einer endlichen Menge von Zuständen, welche zur einheitlichen Aktualisierung der zugeordneten Aussagen verwendet wird. Der zugewiesene Typ ist dabei dafür verantwortlich, inwiefern die mit dem Zustand verknüpften Aussagen im Falle einer Aktualisierung belegt werden. Verfügbare Zustandstypen im Experimentalsystem sind dabei:

  - *strengthState* (Stärke im Vergleich zum Feind)
  - *ownStrengthState* (Eigene momentane Stärke)
  - *enemStrengthState* (Feindliche Stärke)
  - *distanceState* (Entfernung zum Feind)
  - *coverState* (Eigene Deckung)
  - *ownVisibilityState* (Sichtbarkeit eigener Kräfte)
  - *enemVisibilityState* (Sichtbarkeit feindlicher Kräfte)
  - *positionState* (Eigene Position zum designierten Zielpunkt)
  - *angleState* (Relative Position zum Feind)

- **Liste der untergeordneten Zustände**

  Ein Zustand kann als Container für Unterzustände verwendet werden. Unterzustände stellen hierbei Klassifikatoren für bestimmte Situationen dar. So kann bspw. das Vorliegen eines Feindkontaktes ein Unterzustand für den übergeordneten Fall der generellen Sichtbarkeit eines Feindes sein.

- **Aussagendatenbank**

  Zur Überprüfung des Eintreffens einer spezifischen Situation müssen Zuständen ein oder mehrere Aussagen zugeordnet werden, welche in Folge aufgrund der Belegung der enthaltenen Variablen ein Wahrheitswert zugewiesen wird. Dabei wird entweder jedem Klassifikator – d. h. jedem Unterzustand – bzw. dem Zustand an sich eine endliche Menge von Aussagen innerhalb eines indexbasierten Speichers hinterlegt.

Zustände werden automatisch nach jeder Beeinflussung der Parameter eines beliebigen Simulationsobjektes aktualisiert. Hierdurch wird gewährleistet, dass das auf dem Simulationsfeld gesetzte Lagebild identisch mit den durch die Zustände gehaltenen Lageinformationen ist.

An dieser Stelle muss es jedoch Erwähnung finden, dass Zustände zusätzlich entsprechend der gewählten Umsetzung eines Führungsautomaten unterschieden werden. Das Experimentalsystem umfasst hierbei zwei unterschiedliche Formen von implementierten Führungsautomaten: regelbasierte und kausale Automaten. In Folgerung wird daher im System auch zwischen den für regelbasierten Automaten typischen, auf binärer Logik aufbauenden Zuständen und den für kausale Schlüsse notwendigen probabilistischen Zuständen unterschieden. Da sich diese Zustände, wie auch die von dem jeweiligen Führungsautomaten verwendete Wissensbasis, generell in ihrer weiteren Verwendung und im Aufbau unterscheiden, befassen sich die folgenden Unterabschnitte nun gesondert mit der tiefer gehenden Umsetzung von Zuständen und Wissensbasen im Experimentalsystem. Grundsätzlich ist hierbei jedoch zu betonen, dass die folgenden Beschreibungen den bereits erfolgten Ausführungen zur Funktionsweise von Führungsautomaten aus den jeweiligen Kapiteln dieser Arbeit folgt.

5.4.1 Binärlogische Zustände und regelbasierte Wissensbasis

Als binärlogische Zustände werden alle Zustände des Experimentalsystems bezeichnet, deren Evidenz auf Grundlage der Belegung der verknüpften Aussagen mit wahr oder falsch bemessen wird. Gleichwohl diese Zustände das grundsätzliche Lagebild widerspiegeln, werden diese jedoch außerhalb ihrer Funktion zur Aktualisierung von Aussagen und in der Ermittlung einer validen Handlungsoption nicht weitergehend verwendet. Somit erfolgt durch den Automaten keine zusätzliche Interpretation der Zustände, womit dieser ergo auch keine deklarative Wissensbasis besitzt. Dieser Umstand ist damit zu rechtfertigen, dass der Führungsautomat auf diese Art und Weise die momentan in der Gefechtssimulation gebräuchlichen, regelbasierten Führungs- und Steuerungsautomaten stellvertretend darzustellen vermag. Als Folge hieraus besitzt der regelbasierte Führungsautomat lediglich eine prozedurale Wissensbasis. Kern dieser Wissensbasis stellt eine Menge disjunkter binärlogischer Formeln zustandsorientierter Aussagen dar. Diese sind wie folgt definiert:

1. Ein Zustand X sei eine atomare Aussage A.
2. Jede atomare Aussage A ist zugleich eine Formel f.
3. Die Zuordnung eines Wahrheitswertes einer Formel $f \in S$ erfolgt auf Basis der Abbildungsfunktion $B : S \to \{\bot, \top\}$ mit $B(f) = \bigwedge B(A)$, wobei $A \in \mathcal{A}_f$ all diejenigen Aussagen darstellen, welche den Zuständen einer Formel zugewiesen wurden.
4. Das Negat $\neg f$ einer Formel f ist auch eine Formel und es gilt $B(\neg f) = \neg B(f)$.
5. Sind f_1 und f_2 zustandsbasierte Formeln, dann ist $(f_1 \vee f_2)$ ebenfalls eine Formel und es gilt $B(f_1 \vee f_2) = B(f_1) \vee B(f_2)$.
6. Sind f_1 und f_2 zustandsbasierte Formeln, dann ist $(f_1 \wedge f_2)$ ebenfalls eine Formel und es gilt $B(f_1 \wedge f_2) = B(f_1) \wedge B(f_2)$.

Innerhalb der Wissensbasis wird jeder Formel f eine konkreten Handlungsoption a_x zugewiesen, welche im Falle des Eintretens der Situation eine plausible Entscheidung darstellt. Dies bildet den formalen Grundstock der Regelbasis, womit eine Regel demnach wie folgt aufgebaut ist:

$$R_n : \textbf{IF} \quad f \quad \textbf{THEN} \quad a_x$$

Die Wissensbasis speichert hierbei alle hinterlegten Regeln in einem indexbasierten Speicher. Der Führungsagent iteriert zum Zeitpunkt des Abrufs der momentan validen Handlungsoption durch alle Indexbereiche dieses Speichers. Eine Handlungsoption a_x gilt in diesem Zusammenhang als valide, wenn das Antezedenz einer oder mehrerer betroffener Regeln wahr ist. Im Falle mehrfach zutreffender Handlungsoptionen wird auf Grundlage des kleinsten Index[4] der betroffenen Regeln gewählt. Die Regelbasis und die momentan gültige Handlungsoption können mithilfe des Kontextmenüs abgerufen werden. Aussagen, binärlogische Zustände und Wissensbasis sind vor dem Start der Simulation zwingend hart-kodiert zu hinterlegen und werden nach dem Setzen der Simulationsobjekte initialisiert und aktualisiert.

5.4.2 Probabilistische Zustände und kausale Wissensbasis

Probabilistische Zustände stellen ein Derivat binärlogischer Zustände dar, d. h. die Evidenz eines Zustands wird auch hier auf Grundlage der Belegung der verknüpften Aussagen mit wahr oder falsch bemessen. Im Kontrast zu ihrem binärlogischen Pendant ist die Vergabe eines Wahrheitswertes jedoch nicht zwingend, weswegen auch unbelegte Aussagen – z. B. bei Ungewissheit – existieren können. Des Weiteren hat die erfolgte oder nicht-erfolgte Zuweisung von Wahrheitswerten konkrete Auswirkungen auf die Ausprägungen von Variablen innerhalb eines kausalen Netzwerks. Jedem probabilistischen Zustand ist hierfür zusätzlich ein Blattknoten jenes Netzwerkes zugewiesen, wodurch entsprechend der Belegung der verknüpften Aussagen die Ausprägung der repräsentierten Variablen gesetzt wird. Das Netzwerk agiert hierbei als deklarative Wissensbasis, deren Belief States der Wurzelknoten eine Einschätzung der momentanen Lage aus Sicht

4 Hierdurch wird dem Modularitätsprinzip entsprechend Rechnung getragen. Eine anderweitige Reihung, z. B. auf Basis eines zufälligen Ziehens, würde die Ergebnisse an dieser Stelle bereits verfälschen, da diese die Modularität durch Einführen einer zusätzlichen Regel verstecken würde.

des Führungsagenten wiedergeben. Als Folgerung hieraus werden im kausalen Führungsautomaten zur Bestimmung einer validen Handlungsoption nicht die binärlogisch bemessenen Zustände oder deren verknüpfte Aussagen, sondern die Variablen der Wurzelknoten verwendet. Ausprägungen einer oder mehrer Variablen können hierbei innerhalb einer Formel, ähnlich der bereits aufgezeigten aussagenlogischen Formeln, mithilfe logischer Operatoren miteinander verknüpft werden. Eine derartige probabilistische Formel r ist wie folgt definiert:

1. Eine Variablenbelegung $X = x_i$ der Variablen X mit der Ausprägung x_i sei eine atomare Aussage A.
2. Jede atomare Aussage A ist zugleich eine Formel r.
3. Die Zuordnung der Eintrittswahrscheinlichkeit einer Formel $r \in S$ erfolgt auf Basis der Abbildungsfunktion $B : S \rightarrow [0,1]$ mit $B := Bel(r)$.
4. Das Negat einer Formel r ist auch eine Formel, und es gilt $Bel(\neg r) = 1 - Bel(r)$.
5. Sind r_1 und r_2 probabilistische Formeln, dann ist $(r_1 \wedge r_2)$ ebenfalls eine Formel und es gilt $Bel(r_1 \wedge r_2) = Bel(r_1) * Bel(r_2)$.
6. Sind r_1 und r_2 probabilistische Formeln, dann ist $(r_1 \vee r_2)$ ebenfalls eine Formel und es gilt $Bel(r_1 \vee r_2) = max(r_1, r_2)$.

Jeder probabilistischen Formel werden zusätzlich ein oder mehrere Nutzwerte $U(a_x|r_n)$ für Handlungsoptionen a_x zugewiesen. Die kausale Wissensbasis hält diese probabilistische Formeln innerhalb eines indexbasierten Speichers und alle möglichen Handlungsoptionen innerhalb eines Arrays vor. Eine Handlungsoption a_x ist in diesem Zusammenhang valide, wenn der entsprechende Nutzwert größer als Null ist[5]. Die korrespondierenden Nutzwerte werden durch die Wissensbasis mittels schrittweiser Iteration durch die Indexbereiche des Formelspeichers über alle gültigen Handlungsoptionen bestimmt. Im Falle mehrerer wertgleicher Nutzwerte entscheidet das Los. Das kausale Netzwerk, die dort gegenwärtig vorherrschenden a posteriori Belief States, probabilistische Zustände, Formeln und die momentan valide Handlungsoption können über das Kontextmenü abgerufen werden. Aussagen, probabilistische Zustände, kausales Netzwerk, Formeln und Wissensbasis sind vor dem Start der Simulation hart-kodiert zu hinterlegen und werden zusammen mit ihrem regelbasierten Pendant initialisiert und aktualisiert.

Dies schließt die Beschreibung des formalen Aufbaus des Experimentalsystems ab. Das nachfolgende Kapitel beschreibt nun den Einsatz des beschriebenen Systems zum Vergleich der implementierten Führungsautomaten innerhalb eines vorgegebenen militärischen Szenars.

5 Näheres kann hierfür Kap. 4.2 entnommen werden.

6 Experiment

Dieses Kapitel umfasst die mit dem Experimentalsystem durchgeführten Versuchsreihen, welche die Entscheidungsfindung eines regelbasierten und kausalen Führungsautomaten in konkreten Situationen miteinander vergleichen. Die Zielstellung ist es hierbei, ganz im Sinne der Forschungsfrage, die Robustheit beider Automaten auf Basis starr gesetzter Wissensbasen zu überprüfen. Es wird in diesem Zusammenhang gezeigt werden, dass kausale Führungsautomaten generell keine Defizite zu regelbasierten Automaten hinsichtlich ihrer Entscheidungsfähigkeit unter Gewissheit aufweisen. Darüber hinaus wird abgeleitet, dass kausale Führungsautomaten bei Entscheidungen in nicht explizit durch die Wissensbasis abgedeckten Situationen eine erhöhte Robustheit vorweisen. Das Vorgehen orientiert sich hierzu am generellen Vorgehensschema militärischer Simulationsstudien. Den Einstieg bereitet die Darstellung der Ausgangslage und aller abgebildeten Entitäten, was zur besseren Einordnung und Nachvollziehbarkeit der durch die Führungsautomaten gezogenen Schlüsse dient. Gefolgt wird dies von der Aufschlüsselung der jeweiligen Wissensbasen, wo im Detail die verwendeten aussagenlogischen und probabilistischen Formeln, Regelsätze und sonstige einprogrammierte Daten aufgeführt sind. Hierauf folgt das eigentliche Experiment, welches in einzelne Versuchsreihen unterteilt dargestellt ist. Abschluss stellt die objektive Auswertung der Experimentergebnisse anhand der gewonnen Datenlage dar, für welche im Speziellen die Robustheit auf Basis der Befragung eines Expertengremiums bewertet wird.

6.1 Beschreibung des Szenars

Grundlage des Experiments stellt eine fortgeschrittene Gefechtslage innerhalb eines fiktiven Einsatzlandes dar. Hierbei wird demnach davon ausgegangen, dass bereits im Vorhinein Kampfhandlungen zwischen den Konfliktparteien stattgefunden haben und aufgrund dieser Tatsache nur ein verminderter Kräfteansatz auf beiden Seiten betrachtet wird. Als Szenar findet dabei die bereits in mehreren Beispielen innerhalb dieser Arbeit verwendete Lage eines angreifenden blauen Jägerzuges und einer sich in Verteidigung befindenden roten Infanteriegruppe Betrachtung. Dieses Szenar wird entsprechend nun um operative Details erweitert, da nur so die mittels eines Führungsautomaten gewählten Handlungsoptionen auf Plausbilität und Zweckmäßigkeit in der gegenwärtigen Situation bemessen werden können.

Ausgangspunkt dieser Verfeinerung stellt dabei das Lagebild in Abb. 6.1 dar. Wie dort aufgezeigt wird, werden aus Sicht der blauen Fraktion rote Freindkräfte in maximal Gruppenstärke innerhalb der Region vermutet. Anzunehmende Absicht des Feindes ist innerhalb dieses Szenars die Einrichtung einer Verteidigungslinie entlang der Linie C-Dorf - D-Dorf, wobei der Feind versucht, ein Vorrücken blauer Kräfte nachhaltig zu verzögern. Auftrag des Jägerzuges ist es, den Raum um das Gehöft aufzuklären, das Gehöft schnellstmöglich zu nehmen, starke Feinde zu binden und schwache Feinde zu werfen. Hierdurch soll der Angriff auf B-Dorf vorbereitet werden. Der Zug wird hierzu in drei Spähtrupps zu je Gruppenstärke aufgegliedert.

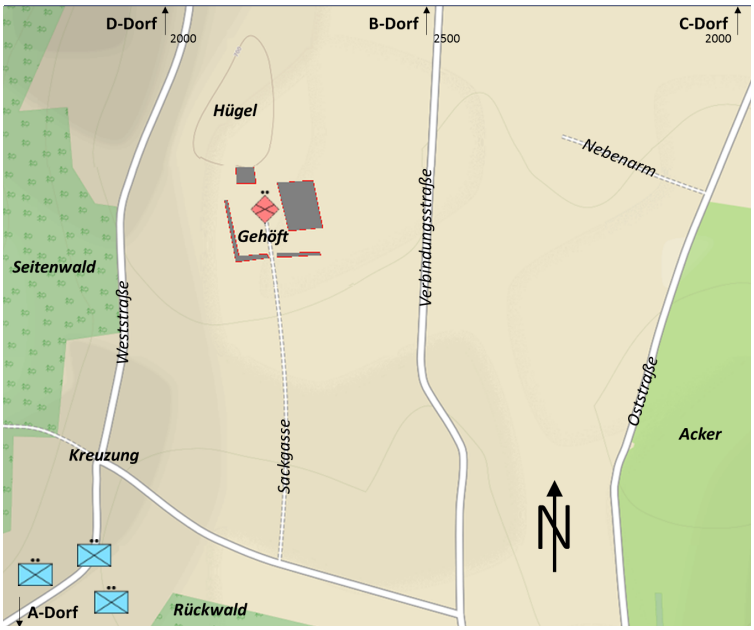


Abbildung 6.1: Gesamtlagebild des Experiments

Innerhalb des simulierten Abschnittes ist durch den Führungsautomaten die taktische Steuerung der jeweiligen Spähtrupps zu übernehmen. Aufgrund der unbekannten Feindposition wird die Wissensbasis gemäß Vorbefehl derart ausgelegt, dass bei plötzlich auftretenden Feind bis einschließlich Truppstärke das Begegnungsgefecht gesucht wird. Bei stärkerem Feind haben die blauen Kräfte sich zu lösen, an einem undefinierten Punkt zu sammeln und anschließend den Feind vereint zu werfen. Als Vereinfachung wird dabei pauschal davon ausgegangen, dass rote Kräfte keine eigenen taktischen Manöver vollführen, lediglich als statische Akteure ohne Verhaltensautomat auf dem Simulationsfeld gesetzt werden und sich stets in verteidigender und befestigter Stellung befinden. Die Lage wird hierzu mittels der Entitätstypen aus Tab. 6.1 im Experimentalsystem abgebildet, zivile oder neutrale Kräfte werden nicht betrachtet. Die Umsetzung dieser operativen Vorgaben innerhalb der Wissensbasen ist den folgenden Unterkapiteln zu entnehmen.

Entität	**Parameter**
Jägergruppe	• **Zugewiesener militärischer Einheitentyp**: Leichte Infanterie • **Zugehörige Fraktion**: Blau • **Initial gesetzte Stärke**: 8 • **Zugewiesenes taktisches Symbol**: • **Aktuell gesetzte Stärke**: 8 • **Liste vorhandener Sensoren**: Optischer Sensor (200m) • **Liste gesichteter Feinde**: nil

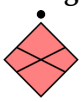
Infanteriegruppe	• **Zugewiesener militärischer Einheitentyp**: Leichte Infanterie • **Zugehörige Fraktion**: Rot • **Initial gesetzte Stärke**: 8 • **Zugewiesenes taktisches Symbol**: • **Aktuell gesetzte Stärke**: 8 • **Liste vorhandener Sensoren**: Optischer Sensor (200m) • **Liste gesichteter Feinde**: nil
Infanterietrupp	• **Zugewiesener militärischer Einheitentyp**: Leichte Infanterie • **Zugehörige Fraktion**: Rot • **Initial gesetzte Stärke**: 2 • **Zugewiesenes taktisches Symbol**:  • **Aktuell gesetzte Stärke**: 2 • **Liste vorhandener Sensoren**: Optischer Sensor (200m) • **Liste gesichteter Feinde**: nil

Tabelle 6.1: Entitäten des Experiments

6.2 Inhalt der verwendeten Wissensbasen

Dieses Unterkapitel beschreibt die im Experiment zur Abbildung von Führungsentscheidungen zugrundeliegenden Zustände und aussagenlogischen Verknüpfungen. Wie bereits im vorherigen Kapitel beschrieben wurde, erfolgt hierbei eine Aufteilung in binärlogische und probabilistische Anteile, welche den Inhalt der Wissensbasis des korrespondierenden regelbasierten bzw. kausalen Führungsautomaten darstellen.

Zustände werden im Folgenden tabellarisch mit ihrer entsprechenden Parametrisierung aufgeführt. Zugeordnete Aussagen sind in der bereits bekannten Form $\varphi R \alpha \varrho$ beschrieben, wobei eine in der Beschreibung bereits durchgeführte Belegung der Symbole φ, ϱ und α stets als Belegung mit einem konstanten Wert anzusehen ist. In diesem Sinne repräsentiert die Aufführung eines Symbols einen dynamisch während der Simulation vergebenen Wert. Das Symbol φ wird in diesem Zusammenhang grundsätzlich immer nur mit Parametern der dem Führungsautomaten untergeordneten Truppenteile belegt. Das Symbol ϱ wird immer nur mit Parametern belegt, welche an bzw. von feindlichen Kräften detektiert werden konnten. Alternativ wird ϱ immer dann belegt, wenn ein Vergleich zu φ gezogen werden soll.

6.2.1 Regelbasierter Führungsautomat

Dieser Abschnitt beschreibt die für das Experiment verwendeten binärlogischen Zustände und die hieraus abgeleiteten Regeln. Da der regelbasierte Führungsautomat keine weitergehenden Schlüsse abseits der Ermittlung einer validen Handlungsoption tätigt, wird nachfolgend der Aufbau der prozeduralen Wissensbasis als alleiniger Träger der Entscheidungsfindung aufgeführt.

Die prozedurale Wissensbasis des regelbasierten Führungsautomaten umfasst sechs binärlogische Zustände. Die Zustände wurden so gewählt, dass diese drei grundlegende Entscheidungsgrundlagen seitens des blauen Führers innerhalb der Simulation abzubilden vermögen:

- Der Feststellung ob und in welchen Umfang Feindkontakt vorherrscht,
- die Bemessung, in welcher gegenwärtigen taktischen Ausgangsposition sich die Gruppe befindet und
- die Bemessung, ob der Gesamtauftrag im gegenwärtigen Zustand der Gruppe noch erfüllbar ist.

Erstere Grundlage dient hierbei dazu, den formulierten Auftrag zum kämpfenden Aufklären abzubilden. Die zweite Grundlage dient in diesem Sinne dazu, die erfolgversprechendste Handlungsoption im Falle eines vorherrschenden Feindkontakts zu ermitteln. Die Bemessung der Erfüllbarkeit des Gesamtauftrags dient als Rückfallposition des Führungsagenten: Sollte eine unvorhergesehene Lage eintreten, bspw. durch Eintreten starker Verluste oder dem Aufkommen eines zahlenmäßig respektive taktisch stark überlegenen Feindes, so lässt sich ein Leerlauf innerhalb des Automaten grundsätzlich durch Erkennen dieser Situationen vermeiden. Die in diesem Sinne formulierten Zustände sind der folgenden Tab. 6.2 zu entnehmen.

Zustandsakronym	**Parameter**	**Kommentar**
ZPos	• **Bezeichner**: Erreichen der Zielposition • **Typ**: *positionState* • **Liste der untergeordneten Zustände**: – A_1: „Zielposition erreicht“ – A_2: „Zielposition nicht erreicht“ • **Aussagendatenbank**: [0] $A_1 \rightarrow (\varphi \leq 10.0)$ [1] $A_2 \rightarrow (\varphi > 10.0)$	Zielposition ist grundsätzlich immer in der Mitte des Gehöfts. Die Zielposition ist erreicht, wenn der Mittelpunkt der graphischen Repräsentation innerhalb eines 10m Umkreises um die gesetzte Position liegt.
OwnStr	• **Bezeichner**: Eigene Stärke • **Typ**: *ownStrengthState* • **Liste der untergeordneten Zustände**: – B_1: „Zerschlagen“ – B_2: „Aufgerieben“ – B_3: „Verringert“ – B_4: „Volle Stärke“	Eigene momentane Stärke (φ) wird in Relation zur Ausgangsstärke (ϱ) bemessen. Hierbei gilt, dass eine Stärke unter 100% als verringert, zwischen 45% und 55% als aufgerieben, und unter 45% als zerschlagen anzusehen ist.

	• **Aussagendatenbank**: [0] $B_1 \rightarrow (\varphi < 0.45\varrho)$ [1] $B_2 \rightarrow (\varphi \geq 0.45\varrho), (\varphi < 0.55\varrho)$ [2] $B_3 \rightarrow (\varphi \geq 0.55\varrho), (\varphi < \varrho)$ [3] $B_4 \rightarrow (\varphi = \varrho)$	
OwnVis	• **Bezeichner**: Sichtbarkeit eines Feindes • **Typ**: *ownVisibilityState* • **Liste der untergeordneten Zustände**: – C_1: „Kein Feindkontakt“ – C_2: „Feindkontakt“ • **Aussagendatenbank**: [0] $C_1 \rightarrow (\varphi = 0)$ [1] $C_2 \rightarrow (\varphi \geq 1)$	Inhalt der Liste detektierter Feinde wird auf Anzahl der enthaltenen Elemente (φ) untersucht.
EnemVis	• **Bezeichner**: Sichtbarkeit für den Feind • **Typ**: *enemVisibilityState* • **Liste der untergeordneten Zustände**: – D_1: „Nicht sichtbar“ – D_2: „Sichtbar“ • **Aussagendatenbank**: [0] $D_1 \rightarrow (\varphi = 0)$ [1] $D_2 \rightarrow (\varphi \geq 1)$	Anzahl der sichtbaren feindlichen Kräfte (φ), deren Front zu einem Element der eigenen Fraktion gerichtet ist.
RelPos	• **Bezeichner**: Relative Position zum Feind • **Typ**: *angleState* • **Liste der untergeordneten Zustände**: – E_1: „Frontal zum Feind“ – E_2: „In der Flanke des Feindes“ – E_3: „Im Rücken des Feindes“ • **Aussagendatenbank**: [0] $E_1 \rightarrow (\varphi \geq 1)$ [1] $E_2 \rightarrow (\varphi \geq 1)$ [2] $E_3 \rightarrow (\varphi \geq 1)$	Anzahl der Feinde (φ), in deren Front, Flanke oder Rücken sich Elemente der eigenen Fraktion befinden. Unterzustände schließen sich nicht gegenseitig aus.
Strength	• **Bezeichner**: Kräftevergleich • **Typ**: *strengthState* • **Liste der untergeordneten Zustände**: – H_1: „Unterlegen“ – H_2: „Ebenbürtig“ – H_3: „Überlegen“ • **Aussagendatenbank**: [0] $H_1 \rightarrow (\varphi < 3\varrho)$	Eigene momentane Stärke (φ) wird mit der feindlichen Stärke (ϱ) verglichen. Da eigene Kräfte grundsätzlich den Angreifer darstellen, muss für ein überlegenes Kräfteverhältnis mehr als die dreifache Stärke, für ein ausgeglichenes Verhältnis die dreifache Stärke und für ein unterle-

	[1] $H_2 \rightarrow (\varphi = 3\varrho)$ [2] $H_3 \rightarrow (\varphi > 3\varrho)$	genes Verhältnis weniger als die dreifache Stärke vorhanden sein.

Tabelle 6.2: Binärlogische Zustände des Experiments

Dem regelbasierten Führungsautomaten werden fünf Handlungsoptionen für die Umsetzung des Auftrages zugeteilt. Da ermittelte Handlungsoptionen aufgrund des fehlenden Zeitfortschrittmechanismus innerhalb der Experimentalumgebung nicht weiter durch die zu steuernde Führungsebene in Einzelaufträge aufgeteilt werden müssen, stellen diese die im weiteren Sinne atomare taktische Entscheidungen dar. Diese gestalten sich wie folgt:

- a_0: Vorrücken zum Zielpunkt
- a_1: Bekämpfe sichtbaren Feind
- a_2: Feind niederhalten und umfassen
- a_3: Lösen vom Feind
- a_4: Ausweichen

Die zur Auswahl der geeigneten Handlungsoption zuständige prozedurale Wissensbasis enthält hierbei sieben Regeln. Diese wurden in wortwörtlicher Auslegung des Auftrages entworfen, d. h. ein Feind wird den Regeln folgend immer dann bekämpft, wenn dieser nicht überlegen ist. Ist der Feind jedoch überlegen, so lässt der Führungsautomat wenn möglich ausweichen. In allen weiteren Situationen bemisst der Automat die Erfüllbarkeit des Auftrages. Ist die jeweilige Gruppe hinsichtlich der Stärke nicht mehr kampffähig, so lässt der Automat ausweichen. Ansonsten wird das Gehöft als Priorität immer genommen. Für einen besseren Überblick können die beschriebenen Regeln zusammenfassend in Tab. 6.3 eingesehen werden.

$\mathbf{R_0}$:	**IF** ZPos = „*Zielposition erreicht*“ ∧ OwnVis = „*Kein Feindkontakt*“ ∧ ¬(OwnStr = „*Zerschlagen*“ ∨ OwnStr = „*Aufgerieben*“) **THEN** a_0
$\mathbf{R_1}$:	**IF** OwnVis = „*Feindkontakt*“ ∧ EnemVis = „*Nicht Sichtbar*“ ∧ ¬(OwnStr = „*Zerschlagen*“ ∨ OwnStr = „*Aufgerieben*“) **THEN** a_1
$\mathbf{R_2}$:	**IF** OwnVis = „*Feindkontakt*“ ∧ EnemVis = „*Sichtbar*“ ∧ RelPos = „*Frontal zum Feind*“ ∧ (Strength = „*Ebenbürtig*“ ∨ Strength = „*Überlegen*“) **THEN** a_2
$\mathbf{R_3}$:	**IF** OwnVis = „*Feindkontakt*“ ∧ EnemVis = „*Sichtbar*“ ∧ (Strength = „*Ebenbürtig*“ ∨ Strength = „*Überlegen*“) ∧ (RelPos = „*In der Flanke des Feindes*“ ∨ RelPos = „*Im Rücken des Feindes*“) **THEN** a_1
$\mathbf{R_4}$:	**IF** OwnVis = „*Feindkontakt*“ ∧ Strength = „*Unterlegen*“ **THEN** a_4
$\mathbf{R_5}$:	**IF** OwnVis = „*Feindkontakt*“ ∧ EnemVis = „*Sichtbar*“ ∧ (RelPos = „*In der Flanke des Feindes*“ ∨ RelPos = „*Im Rücken des Feindes*“) **THEN** a_3
$\mathbf{R_6}$:	**IF** Strength = „*Zerschlagen*“ ∨ Strength = „*Aufgerieben*“ **THEN** a_4

Tabelle 6.3: Prozedurale Wissensbasis des regelbasierten Führungsautomaten

6.2.2 Kausaler Führungsautomat

Die vom kausalen Führungsautomaten verwendeten probabilistischen Zustände sind im weitestgehenden Sinne an den binärlogischen Zuständen orientiert, um eine generelle Vergleichbarkeit in der Lagebeurteilung zuzulassen. Hauptsächlicher Unterschied stellt hierbei eine Aufteilung der Bewertung der relativen Position zum Feind dar, da Knoten innerhalb des kausalen Netzwerks – anders als mit Wahrheitswerten belegte binärlogische Zustände – kein Zutreffen mehrerer Ausprägungen in den zugeordneten Variablen erlauben. Da in der Bewertung somit zwischen Front, Flanke und Rücken getrennt werden muss, erhöht sich die Anzahl der verwendeten Zustände auf Acht. Diese können in Tab. 6.4 eingesehen werden. Hierbei ist zu beachten, dass probabilistischen Zuständen zusätzlich noch Knoten eines kausalen Netwerkes zugewiesen sind, dessen näherer Aufbau dem nachfolgenden Unterabschnitt zur deklarativen Wissensbasis zu entnehmen ist.

Zustandsakronym	**Parameter**	**Kommentar**
ZPos	• **Bezeichner**: Erreichen der Zielposition • **Typ**: *positionState* • **Liste der untergeordneten Zustände**: – a_1: „Zielposition erreicht“ – a_2: „Zielposition nicht erreicht“ • **Aussagendatenbank**: [0] $a_1 \rightarrow (\varphi \leq 10.0)$ [1] $a_2 \rightarrow (\varphi > 10.0)$ • **Zugewiesener Blattknoten**: A	Zielposition ist grundsätzlich immer in der Mitte des Gehöfts. Die Zielposition ist erreicht, wenn der Mittelpunkt der graphischen Repräsentation innerhalb eines 10m Umkreises um die gesetzte Position liegt.
OwnStr	• **Bezeichner**: Eigene Stärke • **Typ**: *ownStrengthState* • **Liste der untergeordneten Zustände**: – b_1: „Zerschlagen“ – b_2: „Aufgerieben“ – b_3: „Verringert“ – b_4: „Volle Stärke“ • **Aussagendatenbank**: [0] $b_1 \rightarrow (\varphi < 0.45\varrho)$ [1] $b_2 \rightarrow (\varphi \geq 0.45\varrho), (\varphi < 0.55\varrho)$ [2] $b_3 \rightarrow (\varphi \geq 0.55\varrho), (\varphi < \varrho)$ [3] $b_4 \rightarrow (\varphi = \varrho)$ • **Zugewiesener Blattknoten**: B	Eigene momentane Stärke (φ) wird in Relation zur Ausgangsstärke (ϱ) bemessen. Hierbei gilt, dass eine Stärke unter 100% als verringert, zwischen 45 und 55% als aufgerieben, und unter 45% als zerschlagen anzusehen ist.
OwnVis	• **Bezeichner**: Sichtbarkeit eines Feindes • **Typ**: *ownVisibilityState* • **Liste der untergeordneten Zustände**: – c_1: „Feindkontakt“ – c_2: „Kein Feindkontakt“	Inhalt der Liste detektierter Feinde wird auf Anzahl der enthaltenen Elemente (φ) untersucht.

	• **Aussagendatenbank**: [0] $c_1 \rightarrow (\varphi \geq 1)$ [1] $c_2 \rightarrow (\varphi = 0)$ • **Zugewiesener Blattknoten**: C	
EnemVis	• **Bezeichner**: Sichtbarkeit für den Feind • **Typ**: *enemVisibilityState* • **Liste der untergeordneten Zustände**: – d_1: „Sichtbar“ – d_2: „Nicht sichtbar“ • **Aussagendatenbank**: [0] $d_1 \rightarrow (\varphi \geq 1)$ [1] $d_2 \rightarrow (\varphi = 0)$ • **Zugewiesener Blattknoten**: D	Anzahl der sichtbaren feindlichen Kräfte (φ), deren Front zu einem Element der eigenen Fraktion gerichtet ist.
FrontPos	• **Bezeichner**: Relative Position zum Feind • **Typ**: *angleState* • **Liste der untergeordneten Zustände**: – e_1: „Frontal zum Feind“ – e_2: „Nicht frontal zum Feind“ • **Aussagendatenbank**: [0] $e_1 \rightarrow (\varphi \geq 1)$ [1] $e_2 \rightarrow (\varphi < 1)$ • **Zugewiesener Blattknoten**: E	Anzahl der Feinde (φ), in deren Front sich Elemente der eigenen Fraktion befinden.
FlankPos	• **Bezeichner**: Relative Position zum Feind • **Typ**: *angleState* • **Liste der untergeordneten Zustände**: – f_1: „In der Flanke des Feindes“ – f_2: „Nicht in der Flanke des Feindes“ • **Aussagendatenbank**: [0] $f_1 \rightarrow (\varphi \geq 1)$ [1] $f_2 \rightarrow (\varphi < 1)$ • **Zugewiesener Blattknoten**: F	Anzahl der Feinde (φ), in deren Flanke sich Elemente der eigenen Fraktion befinden.
BackPos	• **Bezeichner**: Relative Position zum Feind • **Typ**: *angleState* • **Liste der untergeordneten Zustände**: – g_1: „Im Rücken des Feindes“ – g_2: „Nicht im Rücken des Feindes“ • **Aussagendatenbank**: [0] $g_1 \rightarrow (\varphi \geq 1)$ [1] $g_2 \rightarrow (\varphi < 1)$ • **Zugewiesener Blattknoten**: G	Anzahl der Feinde (φ), in deren Rücken sich Elemente der eigenen Fraktion befinden.

Strength	• **Bezeichner**: Kräftevergleich • **Typ**: *strengthState* • **Liste der untergeordneten Zustände**: – h_1: „Unterlegen“ – h_2: „Ebenbürtig“ – h_3: „Überlegen“ • **Aussagendatenbank**: [0] $h_1 \rightarrow (\varphi < 3\varrho)$ [1] $h_2 \rightarrow (\varphi = 3\varrho)$ [2] $h_3 \rightarrow (\varphi > 3\varrho)$ • **Zugewiesener Blattknoten**: H	Eigene momentane Stärke (φ) wird mit der feindlichen Stärke (ϱ) verglichen. Da eigene Kräfte grundsätzlich den Angreifer darstellen, muss für ein überlegenes Kräfteverhältnis mehr als die dreifache Stärke, für ein ausgeglichenes Verhältnis die dreifache Stärke und für ein unterlegenes Verhältnis weniger als dreifache Stärke vorhanden sein.

Tabelle 6.4: Probabilistische Zustände des Experiments

6.2.2.1 Deklarative Wissensbasis

Da die deklarative Wissensbasis im Kern durch ein kausales Netzwerk abgebildet wird, ist es in Anlehnung des Anwendungsbeispiels aus Kapitel 4.2 notwendig, sowohl die enthaltenen Knoten, als auch deren Abhängigkeitsbeziehungen untereinander näher zu erläutern. Ausgangspunkt dieser Erläuterung stellen die in Tab. 6.4 aufgeführten probabilistischen Zustände dar, welche entsprechend zugewiesene Blattknoten des Netzwerks besitzen. Die in diesem Sinne zugewiesenen Blattknoten stellen in Gänze alle für weitere Schlüsse verwendbaren Blattknoten des Netzwerkes dar. Zur Lagebeurteilung seien fünf mit den Blattknoten verknüpfte Vaterknoten definiert:

- **Gefahr** (Danger)

  Die Gefahr I stellt einen stellvertretenden Zustand für die in der momentanen Situation vorliegende imminente Gefahr dar. Als Ausprägungen gelte $I := \{Ungefährdet, Gefährdet\}$.

- **Risiko** (Risk)

  Das Risiko J bemisst das in der momentanen Situation vorherrschende Risiko für die Auftragserfüllung. Es seien als Ausprägungen $J := \{Niedrig, Hoch\}$ definiert.

- **Feuer** (Fire)

  Feuer K wertet aus, ob die geführte Jägergruppe in der momentanen Situation unter Feuer steht. Hierfür seien Ausprägungen $K := \{Unter\ Feuer, Nicht\ unter\ Feuer\}$ gesetzt.

- **Relative Position zum Feind** (RelPos)

  Die relative Position zum Feind L wertet aus, ob sich die geführte Jägergruppe in der gegenwärtigen Situation in einer vorteilhaften Position gegenüber eines (potenziell) wahrgenommenen Feindes befinden.

- **Kampfwert** (SitRep)

  Der Kampfwert M beschreibt den zu erwartenden Erfolg der Jägergruppe im Falle eines situativ vorliegenden Gefechts. Je höher der Kampfwert, desto eher ist mit einem Erfolg in der Auftragserfüllung zu rechnen. Ausprägungen seien dabei mit $M := \{Niedrig, Ausgeglichen, Hoch\}$ vorgegeben.

Erster ausschlaggebender Faktor für die Gefahr der geführten Jägergruppe ist die Nähe zum Zielpunkt (RelPos, Knoten A), da durch Einnehmen des Gehöfts grundsätzlich davon auszugehen ist, dass sämtliche Feindkräfte vernichtet wurden. Ist dies nicht der Fall, ist für die Jäger immer mit Feindkontakt und damit mit Gefahr für Leib und Leben zu rechnen. Zusätzlich hat die eigene Stärke (OwnStr, Knoten B) erheblichen Einfluss auf die imminente Gefahr, da durch einen eigenen verringerten Kräfteansatz die Chancen steigen, durch den Feind im Falle des Aufeinandertreffens zahlenmäßig überwältigt zu werden. Zugleich hängt die Gefahr auch vom tatsächlichen Vorliegen eines Feindkontaktes (OwnVis, Knoten C) ab. Diese kausalen Zusammenhänge beschreiben zusammenfassend Abhängigkeiten $I \rightarrow A$, $I \rightarrow B$ und $I \rightarrow C$. Innerhalb des kausalen Netzwerks ist der Knoten I somit als Vaterknoten der Knoten A, B und C zu setzen.

Gemäß der festgelegten Operationsform, wird das Risiko zur Erfüllung des Auftrags in direktem Maße davon beeinflusst, ob die geführte Jägergruppe in der Lage ist, den Feind im Falle eines Aufeinandertreffens zu überwältigen. In diesem Sinne besitzt die eigene Stärke (OwnStr, Knoten B) eine Auswirkung auf das zu erwartende Risiko. Auch die Bewertung des besagten Risikos hängt in direktem Maße davon ab, ob ein Feindkontakt vorliegt (OwnVis, Knoten C). Zugleich bleibt es in dieser Situation abzuwägen, ob ein Feind die eigene Position bereits aufgeklärt hat, da es diesem hierdurch erlaubt wird, auf eigene taktische Entscheidungen bereits im Vorhinein zu reagieren. Damit herrscht in Folgerung auch eine Beeinflussung der Auftragserfüllung durch die Sichtbarkeit eigener Kräfte (EnemVis, Knoten D) vor. In Summe resultieren hieraus die Abhängigkeiten $J \rightarrow B$, $J \rightarrow C$ und $J \rightarrow D$. Der Knoten J stellt hierdurch Vaterknoten von B, C und D dar.

Die Einschätzung, ob mit Feuer in der gegenwärtigen Situation zu rechnen ist und ob ggf. sogar eigene Truppenteile bereits unter Feuer stehen, wird primär davon beeinflusst, ob ein Feindkontakt (OwnVis, Knoten C) vorliegt. Dies stellt jedoch ein nur hinreichendes Kriterium für die Bemessung dieses Zustandes dar, da der Feind unter Umständen auch aus dem Hinterhalt respektive generell aus unbekannter Position heraus den Feuerkampf eröffnen kann. Gleichermaßen ist es somit ausschlaggebend, ob eigene Kräfte für den Feind sichtbar sind (EnemVis, Knoten D), da nur im Falle einer direkten Sichtlinie zwischen Feind und eigener Truppe ein Beschuss des Feindes mit Flachfeuerwaffen stattfinden kann[1]. Hierdurch herrschen Abhängigkeiten $K \rightarrow C$ bzw. $K \rightarrow D$ vor, womit K innerhalb des Netzwerkes einen Vaterknoten von C und D darstellt.

Zur Bemessung der relativen Position zum Feind sei der Grundsatz verwendet, dass das Einräumen der Möglichkeit zum Beschuss eigener Kräfte einen negativen Einfluss auf den eigenen Vorteil in eben jener Position besitzt. Somit beeinflusst insbesondere die Einnahme einer frontalen Position zum Feind (FrontPos, Knoten E) den zu erwartenden Vorteil negativ. Dem hingegen erweist sich eine flankierende Position

1 Die Möglichkeiten eines Einsatzes von Steilfeuerwaffen und Ausfälle durch Eigenbeschuss sei im Rahmen des vorliegenden Szenaraufbaus vernachlässigt.

(FlankPos, Knoten F) vorteilhafter, welche das Potenzial zum Wirken auf eigene Kräfte deutlich verringert. Am vorteilhaftesten erweist sich in diesem Zusammenhang eine Position im Rücken des Feindes (BackPos, Knoten G), da hierbei das Feuer auf feindliche Kräfte ohne imminente Gegenwehr eröffnet werden kann. Zusammenfassend ist hieraus zu schließen, dass Abhängigkeiten $L \rightarrow E$, $L \rightarrow F$ und $L \rightarrow G$ vorherrschen. In diesem Zusammenhang ist daher auch L Vaterknoten der Knoten E, F und G.

Die zusammenfassende Beurteilung des Kampfwertes stellt im taktischen Führungsprozess eine entscheidende Bewertung einzelner Handlungsoptionen dar. Gemäß einer vergleichenden Betrachtung wird hier untersucht, ob gegeben des aktuellen Lagebildes und der eigenen Möglichkeiten ein Vorteil gegenüber dem Feind vorherrscht. Gleichermaßen hängt dies dabei davon ab, ob der Feind bereits eigene Truppenteile bewirkt (Fire, Knoten K) und in welcher relativen Position zum Feind sich eigene Kräfte befinden (RelPos, Knoten L). Dabei gilt, dass ein bereits vorliegender Beschuss und eine taktisch unvorteilhafte Position generell den eigenen Kampfwert reduzieren. Dies ist jedoch kein zwingendes Kriterium, da bestimmte Operationsformen grundsätzlich die Bewegung unter Feuer und in unvorteilhafter Position erfordern. Hier ist es insbesondere noch ausschlaggebend, wie es um das eigene Kräfteverhältnis zum Feind (Strength, Knoten H) bestellt ist, da eine zahlenmäßige Überlegenheit unvorteilhafte Situationen aufzuwiegen vermag. In diesem Sinne lassen sich folglich zur Beurteilung des Kampfwertes Abhängigkeiten $M \rightarrow K$, $M \rightarrow L$ und $M \rightarrow H$ bestimmen. Der Knoten M stellt damit ferner Vaterknoten von K, L und H dar.

Auf Grundlage der vorhergehenden Absätze lässt sich nun die Struktur des kausalen Netzwerkes der deklarativen Wissensbasis ableiten, welche entsprechend Abb. 6.2 zu entnehmen ist. Gleichermaßen gilt auch hier, wie in den bereits erfolgten Betrachtungen intensionaler Ansätze, dass zur Funktionsfähigkeit des Netzes ferner noch die vorherrschenden Abhängigkeiten mit entsprechenden Wahrscheinlichkeitsverteilungen zu hinterlegen sind. Die Wahl fiel hierbei auf eine subjektive Festlegung geeigneter

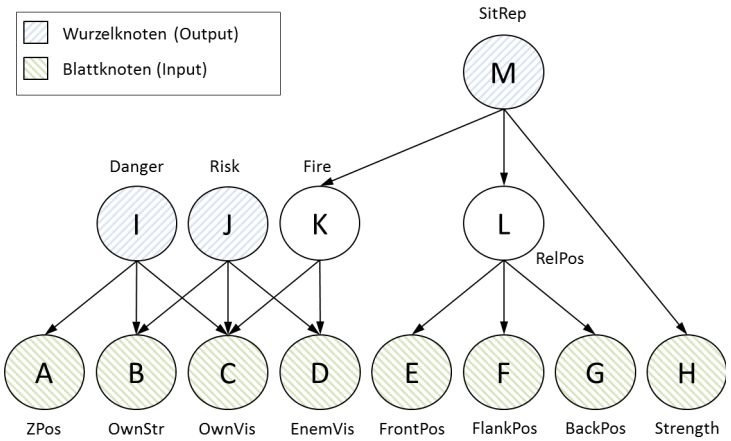


Abbildung 6.2: Kausales Netzwerk des Experiments

Wahrscheinlichkeiten[2], welche anhand der militärischen Ausbildung des Autors und unter zu Rate ziehen eines Infanterismusexperten so gewählt wurden, dass diese kohärent im Sinne des gegebenen Auftrages sind. Konform mit den bisher in dieser Arbeit verwendeten Wahrscheinlichkeitsverteilungen, bilden diese diskrete Wahrscheinlichkeiten ab und können in der folgenden Tab. 6.5 eingesehen werden, wo diese mithilfe bedingter Wahrscheinlichkeiten die aufgeführten Abhängigkeitsbeziehungen des kausalen Netzwerkes abbilden. Der hieraus ableitbare a priori Zustand des kausalen Netzwerkes ist zur Vollständigkeit Abb. 6.3 zu entnehmen.

$A\|I$	a_1	a_2
i_1	0.9	0.1
i_2	0.3	0.7

$F\|L$	f_1	f_2
l_1	0.03	0.97
l_2	0.8	0.2
l_3	0.7	0.3

$E\|L$	e_1	e_2
l_1	0.99	0.01
l_2	0.2	0.8
l_3	0.001	0.999

$G\|L$	g_1	g_2
l_1	0.001	0.999
l_2	0.2	0.8
l_3	0.9	0.1

M	m_1	m_2	m_3
	0.1	0.5	0.4

$B\|I,J$	b_1	b_2	b_3	b_4
$i_1\ j_1$	0.01	0.19	0.3	0.5
$i_1\ j_2$	0.2	0.3	0.4	0.1
$i_2\ j_1$	0.1	0.39	0.5	0.01
$i_2\ j_2$	0.2	0.49	0.3	0.01

$L\|M$	l_1	l_2	l_3
m_1	0.8	0.19	0.01
m_2	0.3	0.6	0.1
m_3	0.01	0.19	0.8

$H\|M$	h_1	h_2	h_3
m_1	0.98	0.019	0.001
m_2	0.001	0.98	0.019
m_3	0.001	0.019	0.98

$K\|M$	k_1	k_2
m_1	0.8	0.2
m_2	0.3	0.7
m_3	0.01	0.99

$C\|I,K,J$	c_1	c_2
$i_1\ k_1\ j_1$	0.8	0.2
$i_1\ k_1\ j_2$	0.9	0.1
$i_1\ k_2\ j_1$	0.01	0.99
$i_1\ k_2\ j_2$	0.5	0.5
$i_2\ k_1\ j_1$	0.99	0.01
$i_2\ k_1\ j_2$	0.999	0.001
$i_2\ k_2\ j_1$	0.7	0.3
$i_2\ k_2\ j_2$	0.6	0.4

$D\|K,J$	d_1	d_2
$k_1\ j_1$	0.2	0.8
$k_1\ j_2$	0.8	0.2
$k_1\ j_2$	0.7	0.3

J	j_1	j_2
	0.3	0.7

I	i_1	i_2
	0.5	0.5

Tabelle 6.5: Wahrscheinlichkeitsverteilungen der deklarativen Wissensbasis

2 Entsprechend wird hiermit die subjektivistische Position (vgl. Kapitel 3.2.2) vertreten. Eine Sensitivitätsanalyse der vergebenen Parameter wurde nicht durchgeführt. Gleichwohl ist festzuhalten, dass die Beobachtungen des Autors bei der Wahl der Wahrscheinlichkeiten eine hohe Robustheit in der Klassifikation erkennen lassen. Dieser gewonnene Eindruck folgt den Beobachtungen von Ben-Bassat et al. (1980), obgleich eine genauere Untersuchung der Sensitivität subjektiv vergebener Wahrscheinlichkeiten innerhalb militärischer Führungsmodelle zukünftigen Arbeiten vorbehalten bleibt.

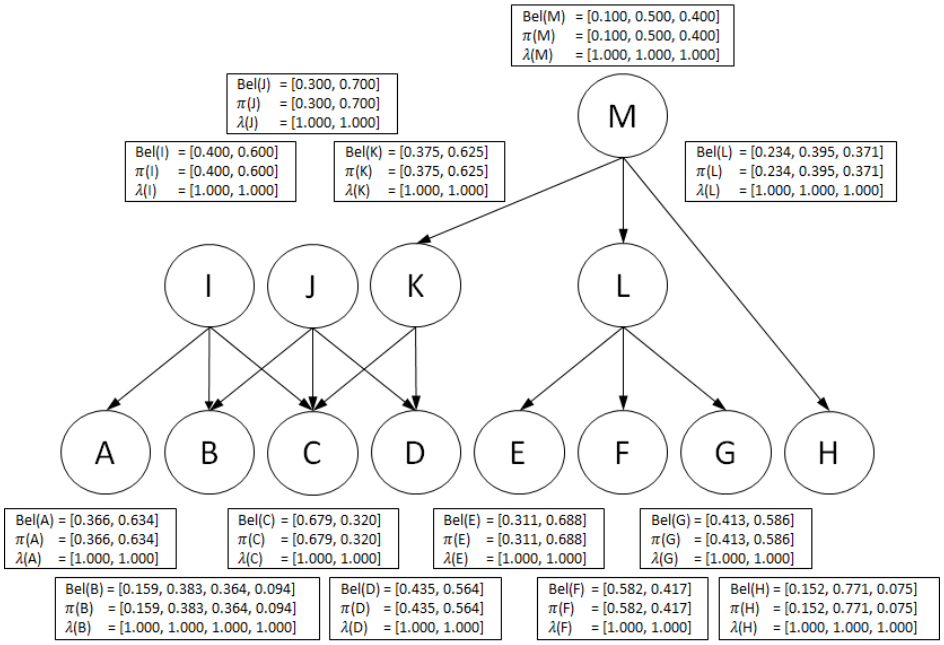


Abbildung 6.3: Initialisiertes kausales Netzwerk des Experiments

6.2.2.2 Prozedurale Wissensbasis

Die prozedurale Wissensbasis des kausalen Führungsautomaten adressiert exakt dieselben fünf Handlungsoptionen, wie deren regelbasiertes Pendant. Gemäß des in Kap. 5.4.2 aufgeführten Schemas, werden hierbei probabilistische Formeln aus den durch die Wurzelknoten des Netzwerks repräsentierten Variablen verwendet, welchen zusätzlich für jede Handlungsoption ein entsprechender Nutzwert zugewiesen wird. Wie Abb. 6.2 zu entnehmen ist, bedeutet dies, dass alle folgend aufzustellenden probabilistische Formeln auf Grundlage der Knoten I, J und M zu bilden sind. Die Wahl der jeweils hierdurch verknüpften Variablen und Nutzwerte erfolgte dabei subjektiv, gleichwohl nicht vollkommen willkürlich. Vielmehr sind die für das Experiment gewählten Formeln und Nutzwerte das Ergebnis einer längeren Kalibrierung, welche die Entscheidungen des kausalen Führungsautomaten so weit wie möglich den Ergebnissen des regelbasierten Pendant anglich. Damit soll gewährleistet werden, dass der kausale Führungsautomat bei Entscheidungen gemäß des intendierten Szenaraufbaus keine oder nur geringe Abweichungen aufweist. Die prozedurale Wissensbasis des kausalen Automaten umfasst dafür zwölf Formeln, welche inklusive der den jeweiligen Formeln zugewiesenen Nutzwerte der folgenden Tab. 6.6 zu entnehmen sind. Hierbei ist zu beachten, dass nicht aufgeführte Nutzwerte pauschal als Null gesetzt werden.

Probabilistische Formel		Erwartete Nutzwerte
r_0 :	Danger = „*Ungefährdet*“ ∧ Risk = „*Niedrig*“ ∧ SitRep = „*Ausgeglichen*“	$U(a_0) = 1.0,\ U(a_1) = 0.5$
r_1 :	Danger = „*Gefährdet*“ ∧ Risk = „*Niedrig*“ ∧ SitRep = „*Hoch*“	$U(a_2) = 0.5,\ U(a_3) = -0.2$
r_2 :	Danger = „*Gefährdet*“ ∧ Risk = „*Hoch*“ ∧ SitRep = „*Hoch*“	$U(a_3) = 0.5$
r_3 :	Danger = „*Gefährdet*“ ∧ Risk = „*Niedrig*“ ∧ SitRep = „*Ausgeglichen*“	$U(a_0) = 0.3,\ U(a_2) = 0.5,\ U(a_4) = 0.1$
r_4 :	Danger = „*Gefährdet*“ ∧ Risk = „*Hoch*“ ∧ SitRep = „*Ausgeglichen*“	$U(a_0) = 0.3,\ U(a_3) = 0.4,\ U(a_4) = 0.2$
r_5 :	Danger = „*Gefährdet*“ ∧ Risk = „*Hoch*“ ∧ SitRep = „*Niedrig*“	$U(a_0) = -0.15,\ U(a_3) = 0.6,\ U(a_4) = 0.5$
r_6 :	Danger = „*Gefährdet*“ ∧ Risk = „*Niedrig*“ ∧ SitRep = „*Niedrig*“	$U(a_4) = 1.0$
r_7 :	Danger = „*Ungefährdet*“ ∧ Risk = „*Niedrig*“ ∧ SitRep = „*Hoch*“	$U(a_1) = 0.5$
r_8 :	Danger = „*Ungefährdet*“ ∧ Risk = „*Niedrig*“ ∧ SitRep = „*Niedrig*“	$U(a_0) = -5.0,\ U(a_3) = 1.0,\ U(a_4) = 1.0$
r_9 :	Danger = „*Ungefährdet*“ ∧ Risk = „*Hoch*“ ∧ SitRep = „*Ausgeglichen*“	$U(a_1) = 0.8,\ U(a_2) = 0.8,\ U(a_4) = 0.1$
r_{10} :	Danger = „*Ungefährdet*“ ∧ Risk = „*Hoch*“ ∧ SitRep = „*Niedrig*“	$U(a_3) = 1.0,\ U(a_4) = -1.0$
r_{11} :	Danger = „*Ungefährdet*“ ∧ Risk = „*Hoch*“ ∧ SitRep = „*Hoch*“	$U(a_2) = 1.0,\ U(a_3) = 1.0,\ U(a_4) = 0.7$

Tabelle 6.6: Prozedurale Wissensbasis des kausalen Führungsautomaten

6.3 Experimentaufbau

Der folgende Abschnitt befasst sich mit dem konkreten Aufbau des Experiments und den einzelnen durchgeführten Versuchen. Hierbei werden weitere zur Auswertung relevante Daten und eine kurze – dank der Einbindung eines Experten fundierte – taktische Bewertung der durch die Führungsautomaten getroffenen Entscheidungen aufgeführt. Die Plausibilität und Zweckmäßigkeit der hierbei getroffenen Entscheidungen wird in einen Vergleich mit dem Inhalt der deklarativen Wissensbasis des kausalen Führungsautomaten gestellt werden. Hierdurch wird aufgezeigt, dass sich die getroffenen Entscheidungen mit Hilfe der innerhalb des kausalen Netzwerks befindlichen Daten nachvollziehbar begründen lässt.

Das durchgeführte Experiment setzt sich aus fünf Versuchsreihen zusammen. Die einzelnen Versuchsreihen umfassen dabei drei bis vier Versuche, welche die gewählten Handlungsoptionen der jeweiligen

Führungsautomaten gegenüber stellen. Um eine einfachere Interpretation der Wahl der Handlungsoptionen zu erlauben, wird zusätzlich die Belegung der Wissensbasen dargestellt. Dies stellt in Verbindung mit den ermittelten Optionen und dem den Führungsautomaten zugrundeliegenden Auftrag die Basis der taktischen Interpretation dar, welche in Folge zur Bemessung der Plausibilität der getroffenen Entscheidungen dient. Für die Ermittlung der Versuchsergebnisse der Reihen Eins bis Drei wurde ein Versuchsansatz gewählt, welcher die Handlungsoptionen jeweils auf Grundlage einer eigenständigen Software-Instanz ohne weitere vorherige Beeinflussungen untersucht. Dies geschah vorrangig mit Blick auf den kausalen Führungsautomaten, da etwaige vorangestellte a posteriori Belegungen des kausalen Netzwerkes einen Einfluss auf die berechneten Nutzwerte in fortschreitenden oder unbekannten Situation besitzen[3].

Die ersten drei Versuchsreihen dienen unter dieser Prämisse für einen möglichst neutralen Vergleich der ermittelten Handlungsoptionen von regelbasierten und kausalen Automaten im intendierten Einsatzraum. Ziel ist es hierbei aufzuzeigen, dass kausale Führungsautomaten hier keine wesentliche Unterschiede zu ihrem regelbasierten Pendant aufweisen. Innerhalb der Versuchsreihen Vier und Fünf wird die Lage bewusst mit Blick auf den intendierten Einsatzraum derart verändert, sodass die hierbei beschriebenen Situationen nicht mehr explizit durch die Wissensbasen abgedeckt werden. Dies dient dem Zweck zur Überprüfung des Verhaltens in nicht vorgesehenen Einsatzräumen und damit direkt zum Test der Robustheit der implementierten Führungsautomaten.

6.3.1 Versuchsreihe 1: Verhalten bei voller Stärke

Die erste Versuchsreihe besteht aus vier Einzelversuchen. Untersucht wird hierbei die durch die Führungsautomaten ermittelte Handlungsoption unter optimalen Ausgangsbedingungen, d. h. dass keinerlei Änderungen an den internen Parametern der blauen Jägergruppe vorgenommen wurden. Die Jägergruppe befindet sich dabei in unterschiedlichen Situationen, welche in Folge die einzelnen Versuche darstellen: Kein Feindkontakt, plötzlich frontal auftauchender Feind, Auftreffen auf einen Feind in dessen Flanke und Auftreffen auf einen Feind in dessen Rücken. Dieser Versuchsaufbau ist schematisch Abb. 6.4 zu entnehmen.

3 Diese Eigenheit des kausalen Führungsautomaten wird insbesondere in Versuchsreihe Fünf näher untersucht werden. Näheres zu diesem Phänomen, inklusive einer Erklärung des Zustandekommens, ist Kapitel 7 zu entnehmen.

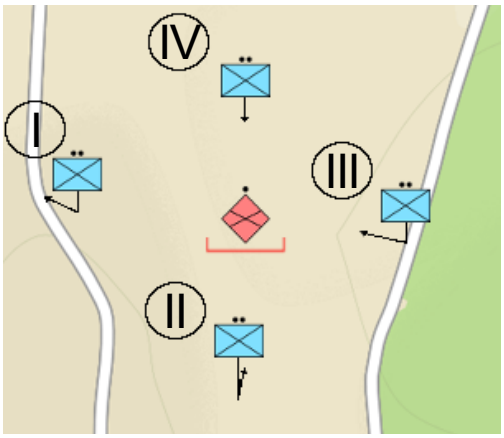


Abbildung 6.4: Aufbau und Ablauf der ersten Versuchsreihe

Kein Feindkontakt

Die Jägergruppe wird an einen beliebigen Ort ohne direkte Sicht auf den Feind und abseits der Zielposition vom Gehöft gesetzt. Im regelbasierten Führungsautomaten führt dies unter Einbezug der vollen Stärke dazu, dass eine Belegung der Zustände A_2, B_4, C_1 und D_1 mit $B(A_2) = \top, B(B_4) = \top, B(C_1) = \top$ bzw. $B(D_1) = \top$ erfolgt. Da weder ein Feind in Sicht ist, noch ein weitergehender Vergleich mit diesem gezogen werden kann, schließt der Automat für alle anderen Zustände $Z \in \mathfrak{Z} \backslash \{A_1, B_4, C_1, D_1\}$, dass $B(Z) = \bot$ ist. Gemäß prozeduraler Regelbasis des regelbasierten Führungsautomaten folgt hieraus, dass das Antezedenz der Regel R_0 erfüllt ist, wodurch als ermittelte Handlungsoption a_0 folgt.

Regelbasiert		**Kausal**			
R_0 :	**true**	r_0 :	0.4584	$EU(a_0\|\mathfrak{e}) =$	**0.4124**
R_1 :	false	r_1 :	0.0191	$EU(a_1\|\mathfrak{e}) =$	0.1689
R_2 :	false	r_2 :	0.0073	$EU(a_2\|\mathfrak{e}) =$	0.2390
R_3 :	false	r_3 :	0.1588	$EU(a_3\|\mathfrak{e}) =$	0.0783
R_4 :	false	r_4 :	0.0612	$EU(a_4\|\mathfrak{e}) =$	0.0621
R_5 :	false	r_5 :	0.0029		
R_6 :	false	r_6 :	0.0077		
		r_7 :	0.0551		
		r_8 :	0.0223		
		r_9 :	0.1767		
		r_{10} :	0.0212		
		r_{11} :	0.0086		

Tabelle 6.7: Ergebnis Versuchsreihe 1, Versuch I

Innerhalb der deklarativen Wissensbasis des kausalen Führungsautomaten führt die beschriebene Situation

dazu, dass die Variablen A, B und C durch die Ausprägungen $A = a_2, B = b_4$ respektive $C = c_2$ belegt werden. Da der Automat über alle weiteren Zustände mangels Information keine Aussage treffen kann, werden diese nicht belegt und verbleiben ohne Ausprägung. Auf dieser Grundlage wird das kausale Netzwerk aktualisiert, was entsprechend zur Berechnung der Belief States $Bel(I) = [0.7426, 0.2573], Bel(J) = [0.7217, 0.2782]$ und $Bel(M) = [0.0416, 0.8553, 0.1029]$ führt. Innerhalb der prozeduralen Wissensbasis folgt, dass durch die entsprechende Neuberechnung der probabilistischen Formeln die Handlungsoption mit dem größten Nutzen $a = argmax_{a_x}(EU(a_x|\mathfrak{e})) = a_0$ ist.

Beide Führungsautomaten kommen somit zu demselben Entschluss. Die Auswahl der Handlungsoption a_0 ist in der gewählten Situation am plausibelsten, da als Hauptauftrag das Gehöft einzunehmen ist und kein Feind bisher die eigene Bewegung hinderte. Die Bewertung der momentanen Lage durch die prozedurale Wissensbasis des kausalen Führungsautomaten offenbart, dass dieser grundsätzlich $I = i_1, J = j_1$ und $M = m_2$ die höchsten Eintrittswahrscheinlichkeiten zuweist. Somit sieht sich der Führungsautomat in der gegenwärtigen Situation keiner Gefahr ausgesetzt, bewertet das Risiko für die eigene Truppe als niedrig und die Erfolgschancen zur Auftragserfüllung als ausgeglichen. Die durch die deklarative Wissensbasis durchgeführte Lageeinschätzung läuft konform zur oben aufgeführten Gesamtbewertung. Eine Aufführung des gegenwärtigen Zustands der prozeduralen Wissensbasen von regelbasierten und kausalen Führungsautomaten ist zur Vollständigkeit Tab. 6.7 zu entnehmen.

Plötzlich frontal auftauchender Feind

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland, die blaue Jägergruppe mit Blickrichtung Nord frontal zum Trupp gesetzt. Dies führt innerhalb der prozeduralen Wissensbasis des regelbasierten Führungsautomaten zur Belegung der Zustände A_2, B_4, C_2, D_2, E_1 und H_3 mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_2) = \top, B(E_1) = \top$ bzw. $B(H_3) = \top$. Zeitgleich führt dieser Umstand dazu, dass die Belegung aller weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_2, E_1, H_3\}$ mit $B(Z) = \bot$ erfolgt. Wie Tab. 6.8 zu entnehmen ist, resultiert dies darin, dass das Antezedenz der Regel 2 erfüllt ist. Der regelbasierte Führungsautomat kommt demnach zum Entschluss, dass in der gegenwärtigen Situation die Handlungsoption a_2 zu wählen ist.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0910	$EU(a_0) =$	0.1863
R_1 :	false	r_1 :	0.0123	$EU(a_1) =$	0.2840
R_2 :	**true**	r_2 :	0.0471	$EU(a_2) =$	**0.3412**
R_3 :	false	r_3 :	0.0827	$EU(a_3) =$	0.2287
R_4 :	false	r_4 :	0.3147	$EU(a_4) =$	0.0812
R_5 :	false	r_5 :	0.0149		
R_6 :	false	r_6 :	0.0039		
		r_7 :	0.0136		
		r_8 :	0.0043		
		r_9 :	0.3465		
		r_{10} :	0.0519		
		r_{11} :	0.0165		

Tabelle 6.8: Ergebnis Versuchsreihe 1, Versuch II

Simultan hierzu führt die gegebene Situation innerhalb der deklarativen Wissensbasis des kausalen Führungsautomaten zur evidenten Ausprägung der Variablen $A = a_2, B = b_4, C = c_1, D = d_1, E = e_1, F = f_2, G = g_2$ und $H = h_3$. Die Aktualisierung des kausalen Netzwerks resultiert dabei in den Belief States $Bel(I) = [0.524, 0.4759], Bel(J) = [0.2080, 0.7919]$ und $Bel(M) = [0.0397, 0.8350, 0.1251]$, auf deren Basis als nützlichste Handlungsoption $a = argmax_{a_x}(EU(a_x|\mathfrak{e})) = a_2$ ermittelt wird.

Die durch den regelbasierten Führungsautomaten ermittelte Handlungsoption deckt sich somit mit der des kausalen Führungsautomaten. Die adäquateste Reaktion auf einen plötzlich auftauchenden Feind stellt in der gegebenen Reaktion das Gewinnen der Feuerüberlegenheit und die Herbeiführung der Entscheidung durch dessen Vernichtung dar. Wie bzw. auf welche Art dies geschieht liegt dabei im Ermessen des taktischen Führers, gleichwohl die äußeren Umstände, bspw. die gegenwärtig ungünstige Position, durchaus in die Entscheidung mit einbezogen werden sollten. Beide Automaten kommen daher durch die Entscheidung zum Niederhalten und Umfassen (a_2) zu einer validen Handlungsoption, da diese die taktische Ausgangsposition für den Angriff der eigenen Gruppe tendenziell verbessert und den Feind durch Feuer und Bewegung wirft. Der Grund für diese Entscheidung lässt sich in direktem Maße der deklarativen Wissensbasis des kausalen Führungsautomaten entnehmen. So ist zu erkennen, dass den Ausprägungen $I = i_1, J = j_2$ und $M = m_2$ die höchsten Eintrittswahrscheinlichkeiten zugewiesen sind, wobei zwischen $I = i_1$ und $I = i_2$ nur ein marginaler Unterschied festzustellen ist. Damit ist der Automat grundsätzlich indifferent, ob in der gegenwärtigen Situation tatsächlich eine Gefahr für die unterstellte Gruppe vorherrschen mag, gleichwohl das Risiko aufgrund des Feindes im Vorfeld als hoch bemessen wird. Zeitgleich wiegt die eigene zahlenmäßige Überlegenheit in dieser Situation durchaus schwer, da diese die für den Angriff taktisch durchaus nachteilige Position in der Front des Feindes aufwiegt. Somit ist das Verlassen der momentanen Position und der in Richtung des Feindes geführte, dynamische Feuerkampf erfolgsversprechend.

Auftreffen auf Feind in dessen Flanke

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland, die blaue Jägergruppe mit Blickrichtung West und direkter Sichtlinie zum roten Trupp in dessen Flanke gesetzt. Der regelbasierte Führungsautomat folgert hieraus die Belegung der Zustände A_2, B_4, C_2, D_1, E_2 und H_3 mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_1) = \top, B(E_2) = \top, B(H_3) = \top$. Zeitgleich schließt dieser hieraus auf die Belegung aller weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_1, E_2, H_3\}$, dass $B(Z) = \bot$ ist. Dies führt innerhalb der prozeduralen Wissensbasis zur Erfüllung des Antezedenz der Regel R_1 (vgl. Tab. 6.9), womit der regelbasierte Führungsautomat a_1 als adäquate Handlungsoption ermittelt.

Regelbasiert		**Kausal**			
$R_0:$	false	$r_0:$	0.1358	$EU(a_0) =$	0.1798
$R_1:$	**true**	$r_1:$	0.2597	$EU(a_1) =$	0.1691
$R_2:$	false	$r_2:$	0.0626	$EU(a_2) =$	**0.2178**
$R_3:$	false	$r_3:$	0.1233	$EU(a_3) =$	0.0607
$R_4:$	false	$r_4:$	0.0297	$EU(a_4) =$	-0.0466
$R_5:$	false	$r_5:$	0.0000		
$R_6:$	false	$r_6:$	0.0003		
		$r_7:$	0.2859		
		$r_8:$	0.0003		
		$r_9:$	0.0327		
		$r_{10}:$	0.0689		
		$r_{11}:$	0.0000		

Tabelle 6.9: Ergebnis Versuchsreihe 1, Versuch III

In Anlehnung daran nimmt der kausale Führungsautomat eine Evidenz $\mathfrak{e} = \{(A = a_2) \wedge (B = b_4) \wedge (C = c_1) \wedge (D = d_2) \wedge (e = e_2) \wedge (F = f_1) \wedge (G = g_2) \wedge (H = h_3)\}$ wahr. Diese führt zur Aktualisierung des kausalen Netzwerks, woraus die neuen Belief States $Bel(I) = [0.5240, 0.4759]$, $Bel(J) = [0.8056, 0.1943]$ und $Bel(M) = [0.0009, 0.3217, 0.6773]$ folgen. Hierauf aufbauend wird durch den Automaten als nützlichste Handlungsoption $a = argmax_{a_x}(EU(a_x|\mathfrak{e})) = a_2$ errechnet.

Entgegen der ersten beiden Versuche weichen die ermittelten Handlungsoptionen von regelbasiertem und kausalen Führungsautomaten nun voneinander ab. Innerhalb der gegebenen Situation herrscht für die blaue Seite ein klarer Vorteil vor, da dieser sowohl an Stärke, als auch in puncto Position überlegen ist. Adäquate Reaktion stellt in diesem Fall die Bekämpfung des Feindes dar. Eine direkte Bekämpfung (a_1) ist auf der einen Seite in diesem Falle durchaus naheliegend, da diese in der gegebenen Situation wohl zum gewünschten Ergebnis, d. h. der Vernichtung des Feindes führt. Auf der anderen Seite ist durch die direkte Bekämpfung des Feindes mit einer Neuausrichtung des Feindes zur Hauptkampfrichtung hin zu rechnen, weswegen eine vorrangige Bekämpfung inklusive Umfassen (a_2) als Handlungsalternative einem potenziell kämpfend ausweichenden Feind aktiv vorbeugt. Von Interesse ist hierbei, wie der kausale Führungsautomat diesen Schluss zu führen vermag. Wie auch im vorhergehenden Versuch bemisst dieser den Ausprägungen

der Variablen I ähnliche Eintrittswahrscheinlichkeiten zu, was auch hier erneut für eine Indifferenz in der Bewertung sprechen mag, ob eine tatsächliche Gefahr für die eigene Truppe in der gegenwärtigen Situation vorherrscht. Eindeutig ist jedoch die Zuweisung der Eintrittswahrscheinlichkeiten der Ausprägungen $J = j_1$ und $M = m_3$. Die hier durch den Automaten errechneten Werte offenbaren, dass dieser sich grundsätzlich keinem großen Risiko ausgesetzt sieht und den Kampfwert als hoch bewertet. Der kausale Führungsautomat identifiziert hiermit einen Handlungsspielraum, welcher fernab der direkten Aufnahme des Feuerkampfes (a_1) gegeben ist.

Auftreffen auf Feind in dessen Rücken

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland, die blaue Jägergruppe mit Blickrichtung Süd und direkter Sichtlinie zum roten Trupp in dessen Rücken gesetzt. In der geschilderten Situation folgert der regelbasierte Führungsautomat die Belegung der Zustände A_2, B_4, C_2, D_1, E_3 und H_3 mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_1) = \top, B(E_3) = \top, B(H_3) = \top$. Für alle weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_1, E_3, H_3\}$ resultiert hieraus die Belegung $B(Z) = \bot$. Wie Tab. 6.10 zu entnehmen ist, folgt dem entsprechend innerhalb der prozeduralen Wissensbasis die Erfüllung des Antezedenz der Regel R_1. Der regelbasierte Führungsautomat kommt damit zur Entscheidung, dass Handlungsoption a_1 gegenwärtig zu wählen ist.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0194	$EU(a_0) =$	0.0258
R_1 :	**true**	r_1 :	0.3658	$EU(a_1) =$	**0.2051**
R_2 :	false	r_2 :	0.0882	$EU(a_2) =$	0.1954
R_3 :	false	r_3 :	0.0176	$EU(a_3) =$	0.0698
R_4 :	false	r_4 :	0.0042	$EU(a_4) =$	-0.940
R_5 :	false	r_5 :	0.0000		
R_6 :	false	r_6 :	0.0000		
		r_7 :	0.4027		
		r_8 :	0.0000		
		r_9 :	0.0046		
		r_{10} :	0.0971		
		r_{11} :	0.0000		

Tabelle 6.10: Ergebnis Versuchsreihe 1, Versuch IV

Situationsbedingt kommt es zeitgleich innerhalb der deklarativen Wissensbasis des kausalen Führungsautomaten zur Belegung der Variablen A, B, C, D, E, F, G und H mit $A = a_2, B = b_4, C = c_1, D = d_2, E = e_2, F = f_2, G = g_1$ bzw. $H = h_3$. Die hierauf aufbauende Aktualisierung des kausalen Netzwerks führt zur Festsetzung der Belief States $Bel(I) = [0.5240, 0.4759]$, $Bel(J) = [0.8056, 0.1943]$ und $Bel(M) = [0.0000, 0.0459, 0.9539]$. Als nützlichste Handlungsoption ermittelt der kausale Führungsautomat hieraus $a = argmax_{a_x}(EU(a_x|\mathfrak{e})) = a_1$.

Die ermittelten Handlungsoptionen des regelbasierten und kausalen Führungsautomaten sind somit dekkungsgleich. Die im Versuch beschriebene Situation ergibt für die blaue Jägergruppe einen entscheidenden Vorteil, welcher letztlich nur noch auszunutzen ist. Die Auswirkungen eines Angriffes in den Rücken des Feindes sind für diesen dabei potenziell verheerend, weswegen die Aufnahme des Feuerkampfes (a_1) eine aussichtsreiche und durchaus plausible Handlungsoption für die Jägergruppe darstellt. Erneut sei auch hier zur Verdeutlichung der Beweggründe darauf verwiesen, wie der kausale Führungsautomat zu diesem Schluss kam. Auch hier lohnt ein vergleichender Blick auf den vorhergehenden Versuch, da die Interpretation der momentan vorherrschenden Gefahr sich durch die Veränderung der Position von der Flanke in den Rücken grundsätzlich nicht geändert hat. Dies gilt ebenfalls für die berechneten Eintrittswahrscheinlichkeiten der Ausprägungen der beiden anderen Variablen J und M, da auch hier $J = j_1$ und $M = m_3$ die höchsten Werte vorweisen. Oberflächlich herrscht daher kein Unterschied innerhalb der deklarativen Wissensbasis im Vergleich zur flankierenden Position vor. Jedoch kommt es nun trotzdem seitens des kausalen Führungsautomaten zu einer anderen Entscheidung. Grund hierfür sind in diesem Fall nicht die Ausprägungen an sich, sondern die den jeweiligen Ausprägungen zugewiesenen Eintrittswahrscheinlichkeiten. Im Speziellen betrifft dies die Wahrscheinlichkeit für das Eintreffen eines hohen Kampfwerts, welche mit $Bel(M = m_3) = 0.9539$ fast als gewiss zu interpretieren ist. Der kausale Automat kommt daher zum Schluss, dass dieser die eigene Ausgangssituation nicht ändern muss, um den Erfolg der Auftragserfüllung zu gewährleisten.

6.3.2 Versuchsreihe 2: Verhalten bei reduzierter Stärke

Die zweite Versuchsreihe besteht – simultan zur ersten Versuchsreihe – aus vier Einzelversuchen. In ihr wird untersucht, welche Handlungsoptionen durch die Führungsautomaten unter veränderten Ausgangsbedingungen ermittelt werden. Hierzu wird zu Beginn jedes Versuchs die aktuelle Stärke der blauen Jägergruppe um Eins reduziert. Dies hat zur Folge, dass die Gruppe sich grundsätzlich im Zustand „reduzierte Stärke“ befindet. Die einzelnen Versuche orientieren sich hierbei am Versuchsaufbau der ersten Versuchsreihe. Dadurch lässt sich der Versuchsaufbau der zweiten Versuchsreihe gemäß der folgenden Abb. 6.5 subsummieren.

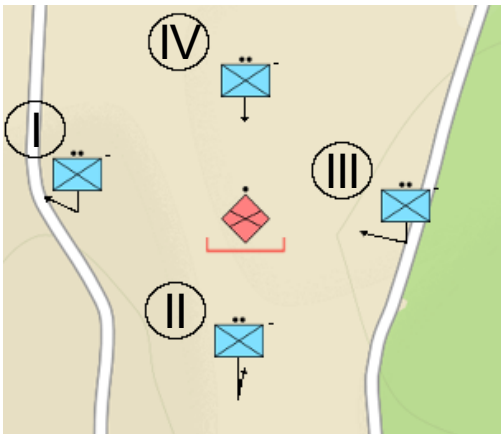


Abbildung 6.5: Aufbau und Ablauf der zweiten Versuchsreihe

Kein Feindkontakt

Die Jägergruppe wird an einen beliebigen Ort ohne direkte Sicht auf den Feind und abseits der Zielposition vom Gehöft gesetzt. Die Stärke der Gruppe wird auf Sieben reduziert. Der regelbasierte Führungsautomat ändert unter dieser Prämisse die Belegungen der Zustände A_2, B_3, C_2 und D_1 auf $B(A_2) = \top, B(B_3) = \top, B(C_2) = \top, B(D_1) = \top$ respektive die Belegungen aller weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_3, C_2, D_1\}$ auf $B(Z) = \bot$. Dies erfüllt das Antezedenz der Regel R_0 der prozeduralen Wissensbasis, vgl. hierfür Tab. 6.11. Der regelbasierte Führungsautomat ermittelt somit die Handlungsoption a_0.

Regelbasiert		**Kausal**			
R_0 :	**true**	r_0 :	0.0445	$EU(a_0) =$	**0.2553**
R_1 :	false	r_1 :	0.0379	$EU(a_1) =$	0.0518
R_2 :	false	r_2 :	0.0522	$EU(a_2) =$	0.2286
R_3 :	false	r_3 :	0.3151	$EU(a_3) =$	0.2175
R_4 :	false	r_4 :	0.4343	$EU(a_4) =$	0.1473
R_5 :	false	r_5 :	0.0211		
R_6 :	false	r_6 :	0.0153		
		r_7 :	0.0053		
		r_8 :	0.0021		
		r_9 :	0.0613		
		r_{10} :	0.0073		
		r_{11} :	0.0029		

Tabelle 6.11: Ergebnis Versuchsreihe 2, Versuch I

Innerhalb der deklarativen Wissensbasis des kausalen Führungsautomaten führt die geänderte Ausgangslage zur Manipulation der Variablen A, B und C, welche mit den Ausprägungen $A = a_2, B = b_3$ bzw.

$C = c_2$ belegt werden. Alle weiteren Variablen verbleiben hierbei mangels Information ohne Belegung. Die Aktualisierung des kausalen Netzwerkes resultiert hierauf aufbauend in den neuen Belief States der Wurzelknotenvariablen I, J und M, was konkret in der Berechnung von $Bel(I) = [0.1238, 0.8761]$, $Bel(J) = [0.4204, 0.5795]$, $Bel(M) = [0.0416, 0.8553, 0.1029]$ gipfelt. Daraus hervorgehend stellt die Handlungsoption mit dem höchsten zu erwartenden Nutzen, unter Einbezug der prozeduralen Wissensbasis, für den kausalen Führungsautomaten $a = argmax_a(EU(a_x|\mathfrak{e}) = a_0$ dar.

Trotz geänderter Ausgangslage kommen beide Führungsautomaten somit erneut zu demselben Entschluss. Der Ausfall eines Mannes hindert die Jägergruppe in erster Linie nicht an der Durchführung des Hauptauftrags, womit dieser – wohl unter dem Einbezug der Tatsache, dass kein Feindkontakt vorherrscht – auch weiterhin bedenkenlos durch die Gruppe durchgeführt werden kann. Die Wahl zum Vorrücken in Richtung des Zielpunktes (a_0) ist demnach auch plausibel. Im Hintergrund zeigt sich jedoch, dass der kausale Führungsautomat in der Einschätzung der Lage weitaus weniger bedenkenlos ist. Wie nämlich innerhalb der deklarativen Wissensbasis zu erkennen ist, weist der Automat den Ausprägungen $I = i_2$, $J = j_2$ und $M = m_2$ die höchsten Eintrittswahrscheinlichkeiten zu. So sorgt der plötzliche Wegfall eines Mannes dafür, dass der Automat grundsätzlich die Annahme tätigt, dass dieser sich in Gefahr befindet und einem Risiko ausgesetzt ist – wohl aber auch immer noch einen ausgeglichenen Kampfwert sieht. Diese Einschätzung ist durchaus nachvollziehbar, da der Wegfall eines Mannes kausal nur durch Feindfeuer eines noch nicht aufgeklärten Feindes zu begründen ist. Nichtsdestotrotz hindert dieses den Automaten – zumindest vorübergehend – nicht an dessen Auftragserfüllung, was folgerichtig auch so durch die deklarative Wissensbasis festgestellt wird.

Plötzlich frontal auftauchender Feind

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland gesetzt. Die Stärke der blauen Jägergruppe wird auf Sieben reduziert, die Position der Gruppe so verändert, dass diese sich mit Blickrichtung Nord frontal zum Trupp befindet. Innerhalb der prozeduralen Wissensbasis des regelbasierten Führungsautomaten führt dies zur Belegung der Zustände A_2, B_3, C_2, D_2, E_1 und H_3 mit $B(A_2) = \top, B(B_3) = \top, B(C_2) = \top, B(D_2) = \top, B(E_1) = \top$ bzw. $B(H_3) = \top$. Hieraus folgt für alle weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_3, C_2, D_2, E_1, H_3\}$, dass $B(Z) = \bot$ gilt. Dies erfüllt das Antezedenz der Regel R_2 – vgl. Tab. 6.12. Der regelbasierte Führungsautomat folgert demnach, dass in der vorliegenden Situation die Handlungsoption a_2 zu wählen ist.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0027	$EU(a_0) =$	0.2350
R_1 :	false	r_1 :	0.0080	$EU(a_1) =$	0.0346
R_2 :	**true**	r_2 :	0.1109	$EU(a_2) =$	0.0631
R_3 :	false	r_3 :	0.0536	$EU(a_3) =$	**0.3789**
R_4 :	false	r_4 :	0.7405	$EU(a_4) =$	0.1731
R_5 :	false	r_5 :	0.0352		
R_6 :	false	r_6 :	0.0025		
		r_7 :	0.0004		
		r_8 :	0.0001		
		r_9 :	0.0380		
		r_{10} :	0.0057		
		r_{11} :	0.0018		

Tabelle 6.12: Ergebnis Versuchsreihe 2, Versuch II

Die gegebene Situation führt in der deklarativen Wissensbasis des kausalen Führungsautomaten zur Belegung der Variablen A, B, C, D, E, F, G und H mit den entsprechenden Ausprägungen $A = a_2, B = b_3, C = c_1, D = d_1, E = e_1, F = f_2, G = g_2$ bzw. $H = h_3$. In Folge der Aktualisierung des kausalen Netzwerkes führt dieser Umstand zur Berechnung der Belief States $Bel(I) = [0.0489, 0.9510], Bel(J) = [0.0675, 0.9324]$ und $Bel(M) = [0.0397, 0.8350, 0.1251]$. Deren Aufschlüsselung in den probabilistischen Formeln lassen mit Hilfe der prozeduralen Wissensbasis in Folge $a = argmax_a(EU(a_x|\mathfrak{e}) = a_3$ als Handlungsoption mit dem höchsten zu erwartenden Nutzen erkennen.

Die durch die Führungsautomaten gewählten Handlungsoptionen weichen demnach voneinander ab. Die Verringerung der Stärke der Jägergruppe hat offensichtlich einen entscheidenden Einfluss auf die Wahl des taktischen Manövers seitens des kausalen Führungsautomaten. Während dieser im Ausgangsfall ohne Verringerung den Feind durch Umfassen zu werfen versuchte (a_2), führen die ihm im Vorhinein zugefügten Verluste dazu, dass dieser gegenwärtig deutlich risikoaverser vorgeht. Die Wahl zum Lösen vom Feind (a_3) ist insofern im Vergleich mit der unbeirrten Entscheidung des regelbasierten Automaten zum Umfassen (a_2) des Feindes umso deutlicher – wenngleich nicht weniger plausibel. Das Zustandekommen von Verlusten muss einem klaren Grund unterliegen. Aus taktischer Sicht bedeutet dies, dass beim plötzlichen Auftreffen auf den Feindtrupp dieser entweder mit weiteren Trupps aus dem Hinterhalt agiert oder durch gezieltes Feuer mit hoher Reichweite unterstützt wird. Gleichgültig des Grundes für den aufgetretenen Verlust, scheint sich die Jägergruppe offensichtlich in akuter Gefahr zu befinden. Diesen Umstand trägt die Bewertung der deklarativen Wissensbasis durchaus Rechnung, da für $Bel(I = i_2)$ und $Bel(J = j_2)$ nahezu an Bestimmtheit grenzende Eintrittswahrscheinlichkeiten errechnet wurden. Somit mag die taktische Ausgangsposition zum Feindtrupp aus Sicht kausalen Führungsautomaten durchaus einen ausgeglichenen Kampfwert ($M = m_2$) bieten, kann jedoch nicht darüber hinwegtäuschen, dass der Automat die momentane Situation als gefährlich und mit hohem Risiko für die unterstellte Gruppe bewertet. Dadurch ist ein Ausweichen unter Feuer aus taktischer Sicht heraus nachvollziehbar.

Auftreffen auf Feind in dessen Flanke

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland gesetzt. Die momentane Stärke der blauen Jägergruppe wird auf Sieben reduziert, die Position mit Blickrichtung West in die Flanke und in Reichweite zum roten Trupp hin verschoben. Als Resultat folgert der regelbasierte Führungsautomat die Belegung der Zustände A_2, B_3, C_2, D_1, E_2 bzw. H_3 mit $B(A_2) = \top, B(B_3) = \top, B(C_2) = \top, B(D_1) = \top, B(E_2) = \top$ bzw. $B(H_3) = \top$ und die Belegung aller weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_3, C_2, D_1, E_2, H_3\}$ mit $B(Z) = \top$. Innerhalb der prozeduralen Wissensbasis des Automaten führt dies zur Erfüllung des Antezedenz der Regel R_1 – vgl. Tab. 6.13 – womit als Handlungsoption a_1 ermittelt wird.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0083	$EU(a_0) =$	0.1000
R_1 :	**true**	r_1 :	0.3437	$EU(a_1) =$	0.0147
R_2 :	false	r_2 :	0.3004	$EU(a_2) =$	**0.2593**
R_3 :	false	r_3 :	0.1632	$EU(a_3) =$	0.1543
R_4 :	false	r_4 :	0.1427	$EU(a_4) =$	0.0308
R_5 :	false	r_5 :	0.0004		
R_6 :	false	r_6 :	0.0004		
		r_7 :	0.0176		
		r_8 :	0.0000		
		r_9 :	0.0078		
		r_{10} :	0.0154		
		r_{11} :	0.0000		

Tabelle 6.13: Ergebnis Versuchsreihe 2, Versuch III

Im Falle des kausalen Führungsautomaten führt die Situation zur Belegung der Variablen A, B, C, D, E, F, G und H mit den Ausprägungen $A = a_2, B = b_3, C = c_1, D = d_2, E = e_2, F = f_1, G = g_2$ und $H = h_3$. Die hierauf erfolgte Aktualisierung des kausalen Netzwerkes führt zur Berechnung der Belief States $Bel(I) = [0.0489, 0.9510], Bel(J) = [0.5535, 0.4664]$ und $Bel(M) = [0.0009, 0.3217, 0.6673]$. Der Automat ermittelt in Folge auf Basis der in der prozeduralen Wissensbasis hinterlegten Formeln, dass Handlungsoption a_2 den höchsten zu erwartenden Nutzen besitzt.

Der Versucht offenbart, dass der geänderte Versuchsaufbau im Ergebnis vorläufig keine Änderung zur Ausgangslage mit voller Stärke vorweist. Dies ist in erster Linie plausibel, da der Verlust eines Soldaten den taktischen Vorteil der blauen Jägergruppe gegenwärtig nicht zu mindern vermag. Nichtsdestotrotz ist der Umstand des Verlusts aus taktischer Sicht in der gegebenen Situation fragwürdig, da der Feind nur bei groben taktischen Fehlentscheidungen die eigene Flanke entblößen würde. Es liegt daher nahe, dass die Verluste durch einen weiteren, noch nicht aufgeklärten Feind entstanden sind. Mit Blick auf die deklarative Wissensbasis zeigt sich, dass der kausale Führungsautomat sich dieses Umstandes durchaus bewusst ist. So betrachtet dieser – trotz vorteilhafter Ausgangsposition – die unterstellte Gruppe als in Gefahr befindlich ($I = i_2$). Jedoch kann der Automat in Ermangelung weiterer Aufklärungsergebnisse nicht

zwingend darauf schließen, ob ein potenzielles Risiko für die Auftragserfüllung vorliegt. Diese Indifferenz ist es letztlich, welche ausschlaggebend für das Festhalten am bisherigen Vorgehen ist, da die flankierende Position nichtsdestotrotz eine hohe Erfolgschance ($M = m_3$) vorweist.

Auftreffen auf Feind in dessen Rücken

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland gesetzt und die momentane Stärke der blauen Jägergruppe auf Sieben reduziert. Anschließend wird die Gruppe in den Rücken des Trupps mit Blickrichtung Süd positioniert. Dies führt im regelbasierten Führungsautomaten zur Belegung der Variablen A_2, B_3, C_2, D_1, E_3 bzw. H_3 mit $B(A_2) = \top, B(B_3) = \top, B(C_2) = \top, B(D_1) = \top, B(E_3) = \top, B(H_3) = \top$ und aller übrigen Variablen $Z \in \mathfrak{Z} \backslash \{A_2, B_3, C_2, D_1, E_3, H_3\}$ mit $B(Z) = \bot$.Wie Tab. 6.14 zu entnehmen ist, führen diese Belegungen zur Erfüllung des Antezedenz der Regel R_1. Der Automat folgert demnach, dass a_1 die passendste Handlungsoption darstellt.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0012	$EU(a_0) =$	0.0143
R_1 :	**true**	r_1 :	0.4840	$EU(a_1) =$	0.0132
R_2 :	false	r_2 :	0.4232	$EU(a_2) =$	**0.2545**
R_3 :	false	r_3 :	0.0233	$EU(a_3) =$	0.1447
R_4 :	false	r_4 :	0.0203	$EU(a_4) =$	-0.0152
R_5 :	false	r_5 :	0.0000		
R_6 :	false	r_6 :	0.0000		
		r_7 :	0.0248		
		r_8 :	0.0000		
		r_9 :	0.0010		
		r_{10} :	0.0217		
		r_{11} :	0.0000		

Tabelle 6.14: Ergebnis Versuchsreihe 2, Versuch IV

Auf Seite des kausalen Führungsautomaten führt die gegebene Situation zur Belegung der Variablen A, B, C, D, E, F, G und H mit den Ausprägungen $A = a_2, B = b_3, C = c_1, D = d_2, E = e_2, F = f_2, G = g_1$ bzw. $H = h_3$. Mittels Aktualisierung des kausalen Netzwerks errechnet dieser im Anschluss die Belief States $Bel(I) = [0.0489, 0.9510], Bel(J) = [0.5335, 0.4664]$ bzw. $Bel(M) = [0.0000, 0.0459, 0.9539]$ für die Wurzelknotenvariablen I, J und M. Auf Basis der mittels der Ausprägungen gegebenen Evidenz, folgert dieser abschließend die Werte der probabilistischen Formeln, welche wiederum zur Berechnung der Handlungsoption $a = argmax_a(EU(a_x|\mathfrak{e}) = a_2$ mit dem höchsten erwarteten Nutzen führt.

Im Vergleich mit dem vorherigen Versuch zeichnet sich erneut dasselbe Ergebnis ab: Während der regelbasierte Automat den direkten Angriff (a_1) sucht, wählt der kausale Führungsautomat mit dem Umfassen des Feindes (a_2) eine risikoaverse Handlungsalternative. Bestimmend ist auch in diesem Fall erneut wieder

die Einschätzung der Situation durch den kausalen Steuerungsmechanismus, welcher auf Grundlage der vorgenommenen Reduzierung der Ausgangsstärke abermals von einer hohen Gefahr für die Jägergruppe ($I = i_2$) ausgeht. Die in diesem Fall vorliegende, beinahe an Gewissheit grenzende Eintrittswahrscheinlichkeite eines hohen Kampfwerts ($M = m_3$) kann dabei nicht darüber hinwegtäuschen, dass durch den Automaten keine eindeutige Einschätzung des vorherrschenden Risikos vorgenommen werden kann. Die Wahl des Umfassens ist daher durchaus plausibel, da dieser den Schwerpunkt der eigenen Kräfte verlagert und somit weniger Angriffsfläche für eine mögliche unerwartete feindliche Intervention bietet, ohne dabei das Ziel zur Vernichtung des erkannten Feindes zu vernachlässigen.

6.3.3 Versuchsreihe 3: Verhalten bei Aufreiben bzw. Zerschlagen

Grundlage der dritten Versuchsreihe stellen abermals vier Einzelversuche dar. Die aus der Versuchsreihe 2 vorgenommene Lageänderung wird hierbei verschärft. Zu Beginn jedes Versuchs wird die aktuelle Stärke der blauen Jägergruppe um Vier reduziert, wodurch sich diese grundsätzlich im Zustand „Aufgerieben“ befindet. Der bekannte Aufbau der beiden vorherigen Versuchsreihen kommt auch hier zum tragen, was inklusive der beschriebenen Änderung der Lage in Abb. 6.6 zusammengefasst wird. Aus militärischer Sicht ist hierzu anzumerken, dass die vorgenommene Reduzierung der Stärke auf 50% einen starken Einschnitt für die Jägergruppe in deren Fähigkeit zu Handeln und Wirken darstellt. Tatsächlich ist den Führungsautomaten dabei jedoch neben dem Zustand „Aufgerieben“ zusätzlich der Zustand „Zerschlagen“ bekannt, welcher bei einer Stärke von unter 45% eintritt. Innerhalb weitergehender Versuche konnte kein wesentlicher Unterschied in der Entscheidung der Führungsautomaten zwischen den genannten Zuständen festgestellt werden – ein Phänomen, welches sich rational aus dem bereits erwähnten Einschnitt in die Fähigkeit zur Handlung und Wirkung ableiten lässt. Es wird daher auf eine tiefergehende Betrachtung des Verhaltens bei Zerschlagung verzichtet, wodurch im Folgenden das aufgeführte Verhalten bei Aufreiben sinngemäß zu übertragen ist.

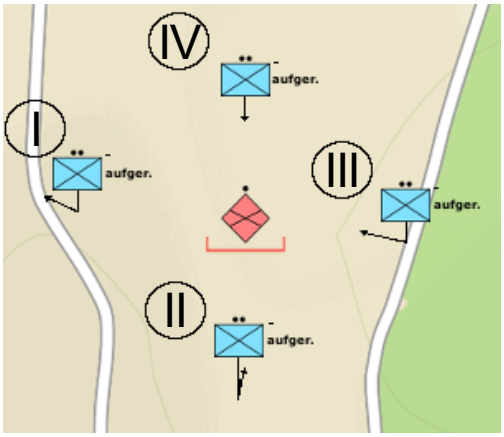


Abbildung 6.6: Aufbau und Ablauf der dritten Versuchsreihe

Kein Feindkontakt

Die Jägergruppe wird an einen beliebigen Ort ohne direkte Sicht auf den Feind und abseits der Zielposition vom Gehöft gesetzt. Die Stärke der Gruppe wird auf Vier reduziert. Als Folge ändert der regelbasierte Führungsautomat die Belegungen der Zustände A_2, B_2, C_2 und D_1 auf $B(A_2) = \top, B(B_2) = \top, B(C_2) = \top$ bzw. $B(D_1) = \top$ und für alle weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_2, C_2, D_1\}$ auf $B(Z) = \bot$. Dies resultiert im Falle des regelbasierten Automaten in der Erfüllung des Antezedenz der Regel R_6 – vgl. hierfür Tab. 6.15. Damit ermittelt der Führungsautomat a_4 als passendste Handlungsoption.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0193	$EU(a_0) =$	0.2482
R_1 :	false	r_1 :	0.0291	$EU(a_1) =$	0.0363
R_2 :	false	r_2 :	0.0662	$EU(a_2) =$	0.1728
R_3 :	false	r_3 :	0.2419	$EU(a_3) =$	**0.2718**
R_4 :	false	r_4 :	0.5501	$EU(a_4) =$	0.1604
R_5 :	false	r_5 :	0.0268		
R_6 :	**true**	r_6 :	0.3117		
		r_7 :	0.0023		
		r_8 :	0.0009		
		r_9 :	0.0439		
		r_{10} :	0.0052		
		r_{11} :	0.0021		

Tabelle 6.15: Ergebnis Versuchsreihe 3, Versuch I

Die deklarative Wissensbasis des kausalen Führungsautomaten wird auf Grundlage der geänderten Ausgangslage aktualisiert, was zuerst in der Belegung der Variablen A, B und C mit den Ausprägungen $A = a_2, B = b_2$ und $C = c_2$ resultiert. Dies hat zur Folge, dass die Belief States der Wurzelknoten I, J und M des kausalen Netzwerks auf Stand gebracht werden, was zu $Bel(I) = [0.0739, 0.9260], Bel(J) = [0.3054, 0.6945]$ bzw. $Bel(M) = [0.0416, 0.8553, 0.1029]$ führt. Die hierauf aufbauende Neuberechnung der probabilistischen Formeln führt in Konklusion dazu, dass der kausale Automat $a = argmax_a(EU(a_x|\mathfrak{e}) = a_3$ als Handlungsoption mit dem höchsten zu erwartenden Nutzen folgert.

Im Gegensatz zu den vergleichbaren Versuchen der beiden bisherigen Versuchsreihen, weichen regelbasierter und kausaler Führungsautomat somit in ihrer Wahl der passendsten Handlungsoption voneinander ab. Wie es bereits in der Einleitung der Versuchsreihe Erwähnung fand, stellt die vorliegende Reduzierung der Ausgangsstärke einen drastischen Einschnitt in das Handlungsvermögen der Jägergruppe dar. Der Wegfall der Hälfte des verfügbaren Truppenkörpers ist taktisch nicht zu kompensieren, wodurch eine weitere Erfüllung des Auftrages aus eigener Kraft de facto außer Frage steht. Somit stellt das Ausweichen (a_4) in dieser Situation eine durchaus valide Handlungsoption dar, da die eigentliche Operationsführung aufgrund der vorherrschenden Verluste nicht mehr fortgesetzt werden kann. Da kein offensichtlicher Feindkontakt gemäß Lagekarte vorherrscht, mag nun jedoch die abweichende Wahl des kausalen Führungsautomaten im

ersten Moment als kurios anmuten.

Mit vertieftem Blick auf den Inhalt des kausalen Netzwerkes offenbart sich jedoch, dass aus Sicht des Automaten diese Wahl durchaus begründet ist. So folgert dieser aus dem Wegfall der eigenen Kräfte, dass sich die Jägergruppe am ehesten innerhalb einer gefährlichen Situation ($I = i_2$) mit hohem Risiko für die weitere Erfüllung des Auftrages ($J = j_2$) befindet. Im übertragenen Sinne sieht sich der Automat damit einer akut negativen Situation ausgesetzt, wobei dieser mangels weiterer Informationen zu Position und Stärke des die Ausfälle verursachenden Feindes lediglich davon ausgehen kann, gegenwärtig einen ausgeglichenen Kampfwert zu besitzen. Dem initial gesetzten Auftrag folgend, steht es der Jägergruppe hierbei frei auszuweichen und den Zusammenschluss mit einer verbündeten Gruppe zu suchen. Hierzu erscheint ohne explizit erkannten Feind das Ausweichen unter Feuer als absurd, nichtsdestotrotz ist das Verlassen der gegenwärtigen Stellung unter unterdrückendem Feuer (a_3) auf die vermutete Feindposition eine plausible Handlungsalternative.

Plötzlich frontal auftauchender Feind

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Gelände gesetzt. Die Stärke der blauen Jägergruppe wird auf Vier reduziert. Die Position der Gruppe wird so verändert, dass diese sich mit Blickrichtung Nord frontal zum Trupp befindet. Der regelbasierte Führungsautomat ändert aufgrund dieser veränderten Umstände die Belegung der Zustände A_2, B_2, C_2, D_2 und H_1 in dessen Wissensbais, was entsprechend zu $B(A_2) = \top, B(B_2) = \top, B(C_2) = \top, B(D_2) = \top, B(H_1) = \top$ und für alle weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_2, C_2, D_2, H_1\}$ zu $B(Z) = \bot$ führt. Diese Belegung hat zur Folge, dass das Antezedenz der Regeln R_4 und R_6 erfüllt ist. Gemäß Konvention zählt dabei der niedrigste Index aller erfüllten Regeln, wodurch der Automat somit auf Basis der Regel R_4 die Handlungsoption a_4 auswählt.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0000	$EU(a_0) =$	-0.1451
R_1 :	false	r_1 :	0.0000	$EU(a_1) =$	0.0000
R_2 :	false	r_2 :	0.0000	$EU(a_2) =$	0.0270
R_3 :	false	r_3 :	0.0000	$EU(a_3) =$	**0.5865**
R_4 :	**true**	r_4 :	0.0010	$EU(a_4) =$	0.5260
R_5 :	false	r_5 :	0.9297		
R_6 :	true	r_6 :	0.0408		
		r_7 :	0.0000		
		r_8 :	0.0011		
		r_9 :	0.0000		
		r_{10} :	0.0000		
		r_{11} :	0.0270		

Tabelle 6.16: Ergebnis Versuchsreihe 3, Versuch II

Dem Ergebnis seines regelbasierten Pendants gegenübergestellt, folgert der kausale Führungsautomat auf Basis der erfolgten Lageänderung innerhalb der deklarativer Wissensbasis eine Belegung der Variablen A, B, C, D, E, F, G und H. Diese werden dabei durch die evidenten Ausprägungen $A = a_2, B = b_2, C = c_1, D = d_1, E = e_1, F = f_2, G = g_2$ und $H = h_1$ belegt. Die in Folge durchgeführte Aktualisierung des kausalen Netzwerkes führt zur Berechnung der Belief States in den Randknoten I, J und M, welche konkret in $Bel(I) = [0.0282, 0.9717], Bel(J) = [0.0420, 0.9579]$ bzw. $Bel(M) = [0.9988, 0.0011, 0.0000]$ resultiert. Als Reaktion darauf kommt es zur Neuberechnung der Wertbelegung in den probabilistischen Formeln der prozeduralen Wissensbasis, wodurch der kausale Führungsautomat als Handlungsoption a_3 mit dem höchsten zu erwartenden Nutzen klassifiziert.

Wie auch im vorhergehenden Versuch weichen die die ermittelten Handlungsoptionen von regelbasierten und kausalen Führungsautomaten voneinander ab. Taktisch gesehen ist das Ausweichen (a_4) in der gegebenen Situation durchaus plausibel, da aufgrund der starken Verminderung der eigenen Stärke grundsätzlich zu vermuten ist, dass ein unternommener Versuch zum Werfen des erkannten Feindes zum Totalausfall aller unterstellten Kräfte führt. Mit einem Erfolg in der Auftragserfüllung ist aus Sicht der zerschlagenen Jägergruppe daher nicht zu rechnen, womit die ermittelte Handlungsoption durchaus plausibel, unter dem potenziell vorherrschenden Feindfeuer jedoch nicht zweckmäßig ist. Erneut sei auch hier wieder das Augenmerk auf die deklarative Wissensbasis des kausalen Automat gerichtet, da diese die vorliegende Situation in ähnlicher Form beurteilt. So berechnet der Automat die Ausprägungen $I = i_2, J = j_2$ und $M = m_1$ als Belegungen der Variablen I, J bzw. M mit den höchsten Eintrittswahrscheinlichkeiten, was somit eine einseitig negativ zu bewertende Situation für die Jägergruppe widerspiegelt. Aus Sicht des kausalen Führungsautomaten befindet sich daher die Gruppe in hoher Gefahr ($I = i_2$), welche verbunden mit einem hohen Risiko für die letztendliche Erfolgschance zur Auftragserfüllung ($J = j_2$) das Lösen vom Feind (a_3) rechtfertigt. Zusätzlich wird die taktische Ausgangsposition ebenfalls als negativ bewertet, wodurch ein niedriger Kampfwert ($M = m_1$) als bestärkender Faktor wirkt.

Auftreffen auf Feind in dessen Flanke

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland gesetzt. Die momentane Stärke der blauen Jägergruppe wird auf Vier reduziert und die Position der Gruppe mit Blickrichtung West in die Flanke des roten Trupps verschoben. Der regelbasierte Automat folgert hieraus die Belegung der Zustände A_2, B_2, C_2, D_1, E_2 und H_1 mit $B(A_2) = \top, B(B_2) = \top, B(C_2) = \top, B(D_1) = \top, B(E_2) = \top$ bzw. $B(H_1) = \top$. Zusätzlich belegt der Automat alle weiteren Variablen $Z \in \mathfrak{Z} \backslash \{A_2, B_2, C_2, D_1, E_2, H_1\}$ mit $B(Z) = \bot$. Die Verarbeitung der damit vorliegenden Evidenz in der prozedurale Wissensbasis des Automaten führt zur Erfüllung der Bedingungen der Regeln R_4 und R_6 – vgl. hierzu Tab. 6.17. Der entstehende Konflikt in der Erfüllung zweier Regeln wird durch die Konvention zur Wahl des niedrigsten Regelindex aufgelöst, wodurch der regelbasierte Automat somit die Handlungsoption a_4 auf Basis der Regel R_4 folgert.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0002	$EU(a_0) =$	-0.1347
R_1 :	false	r_1 :	0.0029	$EU(a_1) =$	0.0002
R_2 :	false	r_2 :	0.0042	$EU(a_2) =$	0.0215
R_3 :	false	r_3 :	0.0072	$EU(a_3) =$	0.3688
R_4 :	**true**	r_4 :	0.0104	$EU(a_4) =$	**0.6926**
R_5 :	false	r_5 :	0.5591		
R_6 :	true	r_6 :	0.3876		
		r_7 :	0.0000		
		r_8 :	0.0112		
		r_9 :	0.0003		
		r_{10} :	0.0001		
		r_{11} :	0.0162		

Tabelle 6.17: Ergebnis Versuchsreihe 3, Versuch III

Der kausale Führungsautomat folgert auf Basis der evidenten Zustandsinformationen eine Neubelegung der Variablen A, B, C, D, E, F, G und H, welche entsprechend mit den Ausprägungen $A = a_2, B = b_2, C = c_1, D = d_2, E = e_2, F = f_1, G = g_2$ bzw. $H = h_1$ belegt werden. Die hierauf aufbauend ausgeführte Aktualisierung der Belief States der Wurzelknotenvariablen I, J und M führt darauf hin zur Berechnung von $Bel(I) = [0.0282, 0.9717], Bel(J) = [0.4094, 0.5905]$ bzw. $Bel(M) = [0.9743, 0.0182, 0.0074]$. Wie entsprechend auch Tabelle 6.17 zu entnehmen ist, resultiert die Kombinationen dieser Belief States innerhalb der probabilistischen Formeln der prozeduralen Wissensbasis – in Verbindung mit den verknüpften Nutzwerten der jeweiligen Handlungsoptionen – darin, dass Option a_4 den gegenwärtig höchsten zu erwartenden Nutzwert aus Sicht des Führungsautomaten besitzt.

Wie dem Ergebnis des Versuches zu entnehmen ist, kommen regelbasierter und kausaler Führungsautomat zu derselben Entscheidung. Der durch die starken Verluste in der gegenwärtigen Situation vorherrschende Nachteil wird durch die vorteilhafte Position in der Flanke des roten Trupps keineswegs aufgewogen, weswegen das Ausweichen (a_4) eine plausible Entscheidung darstellt. Dies läuft konform zur Bewertung des kausalen Führungsautomaten, welcher innerhalb seiner deklarativen Wissensbasis die Ausprägungen $I = i_2$ und $M = m_1$ mit den höchsten Eintrittswahrscheinlichkeiten versieht. Wie in den vergangenen Versuchen dieser Versuchsreihe schließt der Automat aus den Verlusten darauf, dass dieser einer grundsätzlich hohen Gefahr ausgesetzt ist. Unter diesen Voraussetzungen stellt das plötzliche Auftauchen des Feindes einen taktischen Nachteil dar, worauf auf Basis des reinen Stärkevergleichs im Angriff ein niedriger Kampfwert zu folgern ist. Die Einschätzung des vorherrschenden Risikos zur Auftragserfüllung weist dabei mit Tendenz zu $J = j_2$ ein prinzipiell hohes Risiko auf, was in Anbetracht der gegebenen Umstände ebenfalls eine plausible Ableitung darstellt.

Auftreffen auf Feind in dessen Rücken

Ein roter Infanterietrupp wird mit Blickrichtung Süd auf freies Grasland gesetzt. Die momentane Stärke der blauen Jägergruppe wird auf Vier reduziert, die Position der Gruppe in den Rücken des Trupps und die Blickrichtung auf einen Referenzpunkt in südlicher Verlängerung hierzu gesetzt. Hieraus leitet der regelbasierte Führungsautomat die Belegung der Zustände A_2, B_2, C_2, D_1, E_3 bzw. H_1 mit $B(A_2) = \top, B(B_2) = \top, B(C_2) = \top, B(D_1) = \top, B(E_3) = \top, B(H_1) = \top$ und die Belegung aller weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_2, C_2, D_1, E_3, H_1\}$ mit $B(Z) = \bot$ ab. Dies führt zur Erfüllung des Antezedenz der Regeln R_4 und R_6 – vgl. Tab. 6.18. Der regelbasierte Automat schließt gemäß gültiger Konvention auf die Handlungsoption der Regel mit dem geringsten Index, wodurch Handlungsoption a_4 auf Basis der gewählten Regel R_4 ausgewählt wird.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0003	$EU(a_0) =$	-0.1166
R_1 :	false	r_1 :	0.0426	$EU(a_1) =$	0.0009
R_2 :	false	r_2 :	0.0614	$EU(a_2) =$	0.0414
R_3 :	false	r_3 :	0.0105	$EU(a_3) =$	0.3528
R_4 :	**true**	r_4 :	0.0152	$EU(a_4) =$	**0.6157**
R_5 :	false	r_5 :	0.4971		
R_6 :	true	r_6 :	0.3446		
		r_7 :	0.0012		
		r_8 :	0.0100		
		r_9 :	0.0004		
		r_{10} :	0.0017		
		r_{11} :	0.0144		

Tabelle 6.18: Ergebnis Versuchsreihe 3, Versuch IV

In der gegebenen Lage folgert der kausale Führungsautomat die entsprechende Belegung der Variablen A, B, C, D, E, F, G und $H4$ mit den Ausprägungen $A = a_2, B = b_2, C = c_2, D = d_1, E = e_2, F = f_2, G = g_1$ bzw. $H = h_1$. Die Aktualisierung des kausalen Netzwerkes führt in diesem Fall zur Berechnung der Belief States $Bel(I) = [0.0282, 0.9717], Bel(J) = [0.4094, 0.5905]$ und $Bel(M) = [0.8662, 0.0266, 0.1071]$. Die Verknüpfung der Werte innerhalb der probabilistischen Formeln führen den kausalen Automaten dabei zum Schluss, wie Tab. 6.18 entsprechend aufzeigt, dass der Handlungsoption $a = argmax_a(EU(a_x|\mathfrak{e}) = a_4$ der gegenwärtig höchste zu erwartende Nutzen zuzuordnen ist.

Simultan zum Vorversuch kommen kausaler und regelbasierter Führungsautomat zu demselben Ergebnis. Erneut ist auch hier zu erkennen, dass die vorteilhafte Position der blauen Jägergruppe die im Vorhinein erfahrenen Ausfälle nicht aufzuwiegen vermag. Da selbst die Position im Rücken des Feindes unter dieser Prämisse keine gute Ausgangsbasis für einen möglichen Angriffserfolg darstellt, ist das Ausweichen (a_4) in der gegebenen Situation tatsächlich auch die adäquateste Handlungsoption. Die auf Basis der evidenten Zustände aktualisierte deklarative Wissensbasis des kausalen Führungsautomaten bestätigt dieses Bild.

Wie dort zu erkennen ist, herrschen für $I = i_1$ und $M = m_1$ die höchsten Eintrittswahrscheinlichkeiten vor, wodurch – ähnlich der Bewertung im Falle der flankierenden Position – der Automat von einer hohen Gefahr bei niedrigem Kampfwert ausgeht. Die Bewertung des Risikos verbleibt dabei mit einer leichten Tendenz zu $J = j_2$ und damit eher in Richtung eines hohen Risikos zur Auftragserfüllung weiterhin bestehen.

6.3.4 Versuchsreihe 4: Verhalten bei ebenbürtigem Feind

Als erste von zwei Versuchsreihen zur Untersuchung des Verhaltens der Führungsautomaten bei geänderter Ausgangslage, umfasst die folgende Versuchsreihe die Entscheidungsfindung bei Auffinden eines zahlenmäßig ebenbürtigen Feindes. Die Parameter der blauen Jägergruppe werden hierzu im weiteren Verlauf – ähnlich der ersten Reihe – im Ausgangszustand belassen. Die Jägergruppe weist somit eine volle Ausgangsstärke vor, wodurch wiederum zum Vorfinden eines zahlenmäßig ebenbürtigen Feindes eine Erhöhung der roten Ausgangsstärke vonnöten ist. Zu diesem Zweck wird der bisher auf dem Simulationsfeld gesetzte feindliche Infanterietrupp durch eine Infanteriegruppe ersetzt. Der Versuchsaufbau wird aus Gründen der Redundanz zur ersten Versuchsreihe auf drei Einzelversuche reduziert, welche die bereits bekannten Fälle eines plötzlich frontal angetroffenen Feindes, das plötzliche Antreffen des Feindes in dessen Flanke und das plötzliche Antreffen des Feindes in dessen Rücken umfassen. Dies ist zur Vollständigkeit der nachfolgenden Abb. 6.7 bildlich nochmals zu entnehmen.

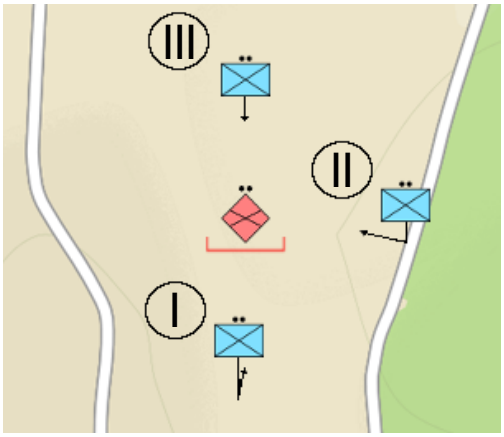


Abbildung 6.7: Aufbau und Ablauf der vierten Versuchsreihe

Plötzlich frontal auftauchender Feind

Eine rote Infanteriegruppe wird auf freies Grasland mit Blickrichtung Süd gesetzt. Die blaue Jägergruppe wird frontal zur feindlichen Gruppe mit Blickrichtung Nord und in Sichtweite des Feindes positioniert. Der regelbasierte Automat schließt hieraus eine Neubelegung der Zustände A_2, B_4, C_2, D_2, E_1 bzw. H_1, welche entsprechend mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_2) = \top, B(E_1) = \top, B(H_1) = \top$ und als direkte Folge alle weiteren Zustände $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_2, E_1, H_1\}$ mit $B(Z) = \bot$ belegt werden. Der hierdurch

beschriebene Systemzustand führt innerhalb der prozeduralen Wissensbasis zur Erfüllung des Antezedenz der Regel R_4 – vgl. hierfür Tab. 6.19. Der regelbasierte Automat schließt somit auf a_4 als gegenwärtig passendste Handlungsoption.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0001	$EU(a_0) =$	-0.6008
R_1 :	false	r_1 :	0.0000	$EU(a_1) =$	0.0003
R_2 :	false	r_2 :	0.0000	$EU(a_2) =$	0.4149
R_3 :	false	r_3 :	0.0001	$EU(a_3) =$	**0.7494**
R_4 :	**true**	r_4 :	0.0004	$EU(a_4) =$	0.6863
R_5 :	false	r_5 :	0.3765		
R_6 :	false	r_6 :	0.0989		
		r_7 :	0.0000		
		r_8 :	0.1089		
		r_9 :	0.0004		
		r_{10} :	0.0000		
		r_{11} :	0.4144		

Tabelle 6.19: Ergebnis Versuchsreihe 4, Versuch I

Simultan hierzu folgert der kausale Führungsautomat in der gegebenen Situation eine vorliegende Evidenz $\mathfrak{e} = \{A = a_2, B = b_4, C = c_1, D = d_1, E = e_1, F = f_2, G = g_2, H = h_1\}$, was entsprechend zur Aktualisierung der deklarativen Wissensbasis und damit zur Neuberechnung der Belief States $Bel(I) = [0.5240, 0.4759]$, $Bel(J) = [0.2080, 0.7919]$ und $Bel(M) = [0.9988, 0.0011, 0.0000]$ führt. Die hierauf aufbauende neue Wertstellung der probabilistischen Formeln gipfelt in der Berechnung der zu erwartenden Nutzwerte der einzelnen Handlungsoptionen, wobei der Automat Option a_3 den gegenwärtig höchsten zu erwartenden Nutzen zuweist.

In Summe kommen regelbasierter und kausaler Führungsautomat damit zu unterschiedlichen Ergebnissen. Die gegenwärtig wenig erfolgversprechende Situation ergibt sich primär aus der frontalen Position zum Feind, ein Nachteil der insbesondere dadurch verschärft wird, dass der Feind mit gleichem Kräfteansatz verteidigt. Aus taktischer Sicht ist diese Begegnung durch die blaue Jägergruppe nur unter hohen eigenen Verlusten im Angriff zu bewältigen, wodurch Ausweichen (a_4) eine plausible Handlungsoption darstellt. Dieser Umstand ist insbesondere unter dem Gesichtspunkt nachvollziehbar, dass gemäß Auftrag die Jägergruppe bei stärkerem Feind auszuweichen hat. Warum die Entscheidung zum Ausweichen in der gegebenen Situation taktisch plausibel, jedoch nicht zwingend zweckmäßig ist, lässt sich dabei anhand der deklarativen Wissensbasis des kausalen Führungsautomaten herleiten. Wie hier auf Grundlage der Belief States der Wurzelknoten zu erkennen ist, ist der Automat annähernd indifferent ob der momentan vorherrschenden Gefahr für die unterstellten Kräfte. Dies ist insofern auch nicht verwunderlich, da trotz des Feindkontaktes noch keine Verluste aufgetreten sind. Nichtsdestotrotz bewertet der Führungsautomat die vorherrschende Situation mit einem hohen Risiko für den zu erwartenden Erfolg in der Auftragserfüllung ($J = j_2$), ein Fakt der durch den abgeleiteten niedrigen Kampfwert ($M = m_2$) zusätzlich noch

unterstrichen wird. Aus taktischer Bewertung heraus folgt damit, dass bei eigener Position in der Front und Reichweite des Feindes das Feindfeuer bereits auf die unterstellten Kräfte eröffnet wurde ggf. in naher Zukunft damit zu rechnen ist. Dadurch ist ein Ausweichen unter Feuer und damit das Lösen vom Feind (a_3) zweckmäßiger.

Auftreffen auf Feind in dessen Flanke

Eine rote Infanteriegruppe wird mit Blickrichtung Süd auf freies Grasland gesetzt. Die blaue Jägergruppe wird mit Blickrichtung West und Sicht auf die feindliche Gruppe in deren Flanke verschoben. Innerhalb der prozeduralen Wissensbasis des regelbasierten Führungsautomaten führt dieser Umstand zur Belegung der Variablen A_2, B_4, C_2, D_2, E_2 bzw. H_1 mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_2) = \top, B(E_2) = \top$ respektive $B(H_1) = \top$ und zur Belegung aller weiteren Variablen $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_2, E_2, H_1\}$ mit $B(Z) = \bot$. Dies erfüllt das Antezedenz der Regeln R_1 und R_4, wobei aufgrund der geltenden Konvention die Regel mit der niedrigeren Indexzahl und damit R_1 gewählt wird – vgl. hierzu Tab. 6.20. Damit kommt der regelbasierte Automat zur Entscheidung, dass a_1 die momentan erfolgversprechendste Handlungsoption darstellt.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0076	$EU(a_0) =$	-2.0600
R_1 :	**true**	r_1 :	0.0028	$EU(a_1) =$	0.0030
R_2 :	false	r_2 :	0.0006	$EU(a_2) =$	0.1056
R_3 :	false	r_3 :	0.0069	$EU(a_3) =$	0.5658
R_4 :	true	r_4 :	0.0016	$EU(a_4) =$	**0.8999**
R_5 :	false	r_5 :	0.0901		
R_6 :	false	r_6 :	0.3736		
		r_7 :	0.0031		
		r_8 :	0.4113		
		r_9 :	0.0018		
		r_{10} :	0.0007		
		r_{11} :	0.0992		

Tabelle 6.20: Ergebnis Versuchsreihe 4, Versuch II

Die zu Anfang beschriebene Situation führt in der deklarativen Wissensbasis des kausalen Führungsautomaten zur Belegung der Variablen A, B, C, D, E, F, G bzw. H mit den Ausprägungen $A = a_2, B = b_4, C = c_1, D = d_2, E = e_2, F = f_1, G = g_2$ und $H = h_1$. Dies stößt den Aktualisierungsprozess des kausalen Netzwerkes an, was wiederum zur Berechnung der Belief States $Bel(I) = [0.5240, 0.4759]$, $Bel(J) = [0.8056, 0.1943]$ und $Bel(M) = [0.9743, 0.0182, 0.0074]$ führt. Hierauf aufbauend berechnet der Automat die zugehörigen Wahrscheinlichkeiten aller probabilistischen Formeln, was wiederum in der Ermittlung der mit den Formeln verknüpften Nutzwerte bzw. der Handlungsoption mit dem höchsten zu erwartenden Nutzen führt. Der

kausale Führungsautomat kommt hierbei zum Ergebnis, dass Option a_4 gegenwärtig die adäquateste Entscheidung darstellt.

Somit weicht die getroffene Auswahl der passendsten Handlungsoption von regelbasierten und kausalen Führungsautomaten voneinander ab. Aus taktischer Sicht besitzt die blaue Jägergruppe in der gegebenen Situation eine vorteilhafte Ausgangsposition, da aus der Flanke heraus der Feind überrascht werden kann und dieser somit ein geringeres Wirkvermögen zur Verteidigung besitzt. Nichtsdestotrotz vermag dies jedoch nicht vom Fakt abzulenken, dass aufgrund der ebenbürtigen Feindstärke selbst dieses verringerte Wirkvermögen eine erhebliche Gegenwehr zum eigenen taktischen Wirken darstellen und somit ein Angriff hohe Verluste zur Folge haben wird. Da letzteres die Möglichkeiten zur Erfüllung des Gesamtauftrages durchaus gefährdet, ist das Ausweichen (a_4) eine durchaus plausible taktische Entscheidung, gleichwohl die Initiierung des Angriffes (a_1) zwar grundsätzlich auch plausibel, aber dennoch nicht zweckmäßig, weil wenig erfolgsversprechend ist. Diese Kausalkette lässt sich erneut auf Basis des Inhaltes der deklarativen Wissensbasis des kausalen Automaten gut nachvollziehen. Die vorherrschende und vermeintlich günstige Ausgangsposition wird durch die potenziell indifferente Eintrittswahrscheinlichkeit der möglichen Gefahr für eigene Kräfte und eine starke Präferenz hin zum niedrigen Risiko für die Erfüllung des Gesamtauftrages ($J = j_1$) korrekt eingeschätzt. Entscheidend ist in diesem Falle jedoch nicht die günstige Ausgangsposition, sondern vielmehr die taktische Bewertung eines Angriffes auf den erkannten Feind aus eben jener Position. Da der Feind aufgrund seiner Stärke in verteidigender Stellung einen entscheidenden taktischen Vorteil besitzt, ist ein niedriger Kampfwert ($M = m_1$) für den Angriff in eben jener Situation zu erwarten. Somit ist dieses Vorgehen just nicht erfolgsversprechend, weswegen das Ausweichen die zweckmäßigere Handlungsoption darstellt.

Auftreffen auf Feind in dessen Rücken

Eine rote Infanteriegruppe wird mit Blickrichtung Süd auf freies Grasland gesetzt. Die Position der blauen Jägergruppe wird mit Blickrichtung Süd in den Rücken der feindlichen Gruppe mit Blick auf selbige verschoben. Der regelbasierte Führungsautomat folgert auf dieser Grundlage eine neue Belegungen der Variablen A_2, B_4, C_2, D_1, E_3 bzw. H_1 mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_1) = \top, B(E_3) = \top$ und $B(H_1) = \top$, sowie für alle weiteren Variablen $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_1, E_3, H_1\}$ mit $B(Z) = \bot$. In dessen prozeduralen Wissensbasis führt dies zur Erfüllung der Bedingungen der Regeln R_1 und R_4, wobei aufgrund der geltenden Konvention die Regel mit dem niedrigsten Index und damit R_1 ausgewählt wird – vgl. hierfür Tab. 6.21. Damit entscheidet sich der regelbasierte Automat in der gegenwärtigen Situation für Handlungsoption a_1.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0076	$EU(a_0) =$	-1.8257
R_1 :	**true**	r_1 :	0.0028	$EU(a_1) =$	0.0247
R_2 :	false	r_2 :	0.0006	$EU(a_2) =$	0.1160
R_3 :	false	r_3 :	0.0069	$EU(a_3) =$	0.5106
R_4 :	true	r_4 :	0.0016	$EU(a_4) =$	**0.7906**
R_5 :	false	r_5 :	0.0901		
R_6 :	false	r_6 :	0.3736		
		r_7 :	0.0031		
		r_8 :	0.4113		
		r_9 :	0.0018		
		r_{10} :	0.0007		
		r_{11} :	0.0992		

Tabelle 6.21: Ergebnis Versuchsreihe 4, Versuch III

Innerhalb der deklarativen Wissensbasis des kausalen Führungsautomaten führt die gegenwärtig wahrgenommene Situation zur Belegung der Variablen A, B, C, D, E, F, G und H mit den entsprechend Ausprägungen $A = a_2, B = b_4, C = c_1, D = d_2, E = e_2, F = f_2, G = g_1$ bzw. $H = h_1$. Die aufgrund der Änderung der evidenten Variablenbelegungen initiierte Aktualisierung des kausalen Netzwerks führt hierbei zur Berechnung der Belief States $Bel(I) = [0.5240, 0.4759], Bel(J) = [0.8056, 0.1943]$ und $Bel(M) = [0.8662, 0.0266, 0.1071]$. Die auf dieser Grundlage ebenfalls neu berechneten probabilistischen Formeln bilden in der prozeduralen Wissensbasis den Ausgangspunkt zur Ermittlung von $a = argmax_a(EU(a_x|\mathfrak{e}) = a_4$ als Handlungsoption mit dem gegenwärtig höchsten zu erwartenden Nutzen.

Simultan zum vorgehenden Versuch weichen somit die ermittelten Handlungsoptionen von regelbasierten und kausalen Führungsautomaten voneinander ab. Erneut erweist sich ein Angriff auf den Feind (a_1) in der gegenwärtigen Ausgangslage als plausibel, ist jedoch im Vergleich zur flankierenden Position des Vorversuchs zugleich auch zweckmäßig. Die stark überlegene Ausgangsposition verspricht dabei dem Feind bereits nach Aufnahme des Feuergefechts hohe Verluste zuzufügen und hierdurch ein entscheidendes Moment zum Werfen des Feindes aufzunehmen. Die Entscheidung zum Angriff steht jedoch gleichermaßen auch im Gegensatz zur ursprünglichen Befehlsgebung, welche vorsah, dass die blaue Jägergruppe sich bei überlegenem Feind von diesem zu lösen hat. Die Auswahl des kausalen Führungsautomaten zum Ausweichen (a_4) adressiert diesen Umstand, stellt in der gegebenen Situation jedoch keine plausible, weil unzweckmäßige Entscheidung dar. Der Grund für diese Entscheidung des kausalen Führungsautomaten ist erneut wieder in dessen deklarativer Wissensbasis zu finden, wo dieser den unterstellten Kräften im Angriff gegenwärtig einen niedrigen Kampfwert ($M = m_1$) zuspricht, obgleich hierbei die Eintrittswahrscheinlichkeit im Vergleich zum vorhergehenden Versuch niedriger ist. Diese Einschätzung genügt aus taktischer Bewertung heraus zum Verwerfen der ursprünglichen Intention des Angriffes, wodurch die Entscheidung zum Ausweichen grundsätzlich nachvollziehbar ist.

6.3.5 Versuchsreihe 5: Verhalten bei fortschreitender Lageentwicklung

Als letzte Versuchreihe dieser Arbeit stellt der folgende Unterabschnitt das Verhalten der implementierten Führungsautomaten bei fortschreitender Lageentwicklung dar. Intention ist es hierbei die ermittelten Handlungsoptionen unter einer nicht explizit durch die Wissensbasen erfassten Situation zu vergleichen. Die blaue Jägergruppe wird hierfür in eine vermeintlich sichere Ausgangsposition in den Rücken eines feindlichen Infanterietrupps gebracht, welcher jedoch durch einen weiteren Trupp durch ein Angriffsmanöver über die tiefe linken Flanke in den Rücken der blauen Gruppe Deckungsfeuer erhält. In Folge dessen wird die blaue Stärke aufgrund von Ausfällen kontinuierlich verringert, ohne dass die Jägergruppe den Grund und damit den feindlichen Trupp im Rücken aufzuklären vermag. Der Versuchsaufbau unterteilt sich hierzu in vier Einzelversuche, welche jeweils dem in Abb. 6.8 dargestellten Ablauf folgen.

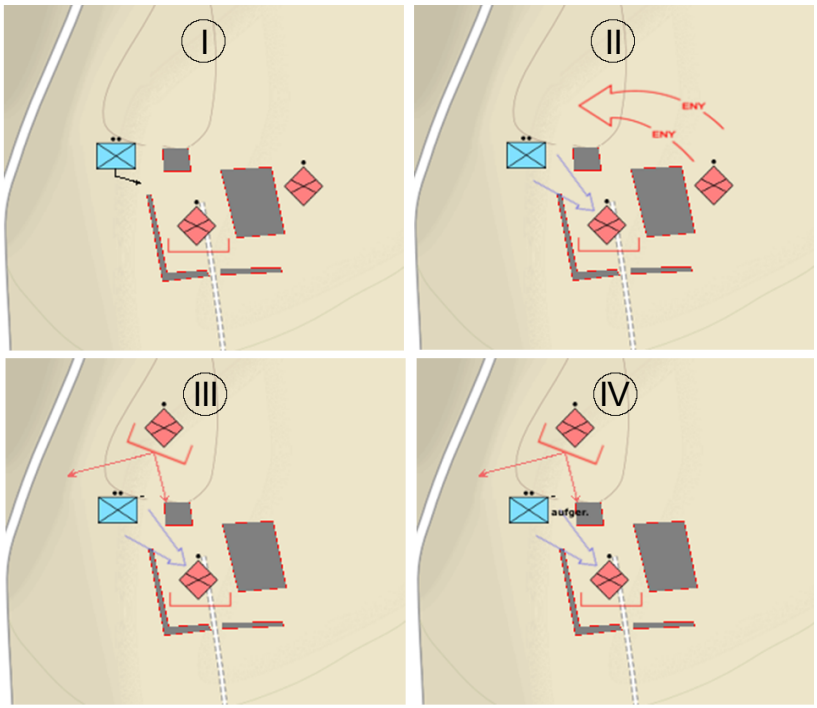


Abbildung 6.8: Aufbau und Ablauf der fünften Versuchsreihe

Keine Verluste

Ein roter Infanterietrupp wird mit Blickrichtung Südost nahe des Zentrums im Gehöft, ein weiterer Trupp weiter östlich hinter dem zentralen Hauptgebäude des Gehöfts positioniert. Eine blaue Jägergruppe wird nordwestlich des Gehöfts mit Blickrichtung Südost in den Rücken des sich im Gehöft befindlichen Trupps mit direkten Blick auf den selbigen gesetzt. Die beschriebene Situation führt im Falle des regelbasierten Führungsautomaten zur Belegung der Variablen A_2, B_4, C_2, D_1, E_3 bzw. H_3 mit $B(A_2) = \top, B(B_4) = \top, B(C_2) = \top, B(D_1) = \top, B(E_3) = \top$ respektive $B(H_3) = \top$ und zur Belegung aller weiteren Variablen $Z \in \mathfrak{Z} \backslash \{A_2, B_4, C_2, D_1, E_3, H_3\}$ mit $B(Z) = \bot$. Innerhalb der prozeduralen Wissensbasis des Automaten

erfüllt dies die Bedingung der Regel R_1 – vgl. Tab. 6.22. Dadurch folgert dieser a_1 als passendste Handlungsoption.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0194	$EU(a_0) =$	0.0258
R_1 :	**true**	r_1 :	0.3658	$EU(a_1) =$	**0.2051**
R_2 :	false	r_2 :	0.00882	$EU(a_2) =$	0.1954
R_3 :	false	r_3 :	0.0176	$EU(a_3) =$	0.0698
R_4 :	false	r_4 :	0.0042	$EU(a_4) =$	-0.0940
R_5 :	false	r_5 :	0.0000		
R_6 :	false	r_6 :	0.0000		
		r_7 :	0.4027		
		r_8 :	0.0000		
		r_9 :	0.0046		
		r_{10} :	0.0971		
		r_{11} :	0.0000		

Tabelle 6.22: Ergebnis Versuchsreihe 5, Versuch I

Die vorgegebene Ausgangssituation führt im kausalen Führungsautomaten zur Belegung der Variablen A, B, C, D, E, F, G und H mit den Ausprägungen $A = a_2, B = b_4, C = c_1, D = d_2, E = e_2, F = f_2, G = g_1$ bzw. $H = h_3$, was wiederum in Folge den Aktualisierungsprozess des kausalen Netzwerkes initiiert. Dies resultiert in der Berechnung der Belief States der Wurzelknoten I, J und M, was konkret in $Bel(I) = [0.5240, 0.4759], Bel(J) = [0.8056, 0.1943]$ bzw. $Bel(M) = [0.0000, 0.0459, 0.9539]$ mündet. Die Neuberechnung der probabilistischen Formeln führt hierauf aufbauend dazu, dass der Automat auf $a = argmax_a(EU(a_x|\mathfrak{e}) = a_1$ als Handlungsoption mit dem gegenwärtig höchsten zu erwartenden Nutzen schließt.

Somit kommen regelbasierter und kausaler Führungsautomat zu demselben Ergebnis. Obgleich sich die Ausgangslage rein taktisch für die blaue Jägergruppe verändert hat, besteht für die Automaten rein auf Grundlage der wahrzunehmenden Faktoren de facto dieselbe Lage, wie sie in Versuchsreihe 1, Versuch IV ebenfalls vorlag. Die Bewertung der Plausibilität folgt der dort bereits aufgeführten Argumentation.

Leichte Verluste aufgrund Feindfeuer im Rücken

Der im vorhergehenden Versuch vorherrschende Versuchsaufbau wird übernommen. Der bisher im nicht einsehbaren Bereich des Gehöfts befindliche zweite rote Infanterietrupp wird in den Rücken der blauen Jägergruppe gesetzt. Die Stärke der Gruppe wird um eins auf Sieben reduziert. Im regelbasierten Automaten führt dies zur Änderung der Belegung der Variablen B_3 und B_4, welche entsprechend mit $B(B_3) = \top$ und $B(B_4) = \bot$ belegt werden. Die Belegung aller weiteren Variablen bleibt unverändert. Die somit

vorliegende Belegung aller Variablen führt in der prozeduralen Wissensbasis des Automaten zur Erfüllung des Antezedenz der Regel R_1 – vgl. Tab. 6.23. Somit ermittelt der regelbasierte Automat Handlungsoption a_1 als passendstes Vorgehen.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0009	$EU(a_0) =$	0.0140
R_1 :	**true**	r_1 :	0.4142	$EU(a_1) =$	0.0102
R_2 :	false	r_2 :	0.4984	$EU(a_2) =$	**0.2180**
R_3 :	false	r_3 :	0.0199	$EU(a_3) =$	0.1958
R_4 :	false	r_4 :	0.0240	$EU(a_4) =$	-0.0156
R_5 :	false	r_5 :	0.0000		
R_6 :	false	r_6 :	0.0000		
		r_7 :	0.0187		
		r_8 :	0.0000		
		r_9 :	0.0010		
		r_{10} :	0.0225		
		r_{11} :	0.0000		

Tabelle 6.23: Ergebnis Versuchsreihe 5, Versuch II

Die Änderung der Stärke führt in der deklarativen Wissensbasis des kausalen Führungsautomaten zur Änderung der Belegung der Variablen B, deren Ausprägung von $B = b_4$ auf $B = b_3$ verändert wird. Die hierdurch ausgelöste Aktualisierung des kausalen Netzwerkes führt zur Neuberechnung der Belief States der Wurzelknoten, was sich konkret in $Bel(I) = [0.0432, 0.9567], Bel(J) = [0.4539, 0.5460]$ und $Bel(M) = [0.0000, 0.0459, 0.9539]$ niederschlägt. Der kausale Automat berechnet auf dieser Basis die Werte der probabilistischen Formeln neu, wodurch die Handlungsoption a_2 als Option mit dem gegenwärtig höchsten zu erwartenden Nutzwert ermittelt wird.

Damit weichen die ermittelten Handlungsoptionen von regelbasierten und kausalen Führungsautomaten durch die Änderung der Stärke der blauen Jägergruppe voneinander ab. Da der für die Jägergruppe sichtbare feindliche Infanterietrupp nach wie vor mit dem Rücken zur eigenen Front steht, kann dieser grundsätzlich nicht für den zugefügten Verlust verantwortlich sein. Es ist hierbei naheliegend, dass der erzeugte Ausfall durch indirektes Feuer bzw. Feuer aus dem nicht einsehbaren Bereich in der tiefen Flanke oder im Rücken zustande kam. Die Weiterführung des bisherigen Vorgehens (a_1) ist in diesem Fall durchaus plausibel, jedoch nicht zwingend zweckmäßig, da davon auszugehen ist, dass weitere Verluste durch den Verbleib an der gegenwärtigen Situation zustande kommen. Insofern ist die Entscheidung zum Umfassen des Feindes (a_2) zweckmäßiger, da diese die Gruppe in der offensichtlich ungünstigen Ausgangsposition für den Angriff anweist, einen Stellungswechsel durchzuführen und so den Feind über dessen Flanke in Halbgruppen zu werfen. Dies hat den Vorteil, dass ggf. in der Durchführung des Stellungswechsels Feindkräfte im Rücken aufgeklärt werden können. Wie der deklarativen Wissensbasis des kausalen Führungsautomaten hierfür zu entnehmen ist, folgt dessen Entscheidung hierbei logisch aus dem Inhalt des kausalen Netzwerks. Die primär ausgeprägten Belief States wechselten durch den erlittenen Ausfall zur

Annahme, dass die eigenen Kräfte sich in hoher Gefahr ($I = i_2$) befinden, das Risiko für eigene Kräfte jedoch aufgrund der unklaren Feindlage nicht eindeutig bestimmt werden kann. Gleichwohl ist auf Grundlage der vorherrschenden Wahrnehmung ein hoher Kampfwert ($M = m_3$) zu erwarten, was jedoch durch den Einfluss der hohen Gefahr und des zu erwartenden Risikos im Angriff zu einem umsichtigeren Handeln mahnt, als der direkte Angriff es zu bieten vermag.

Anhaltende Verluste aufgrund Feindfeuer im Rücken

Der Versuchsaufbau des vorhergehenden Versuches wird übernommen. Die Stärke der Jägergruppe wird fortführend um zwei auf Fünf reduziert. Im Fall des regelbasierten Automaten führt dies zur Änderung in der Belegung der Variablen H_3 und H_1, die als Folge der Reduzierung der Stärke mit $B(H_3) = \top$ und $B(H_1) = \bot$ neu belegt werden. Alle weiteren Variablen verbleiben bezüglich ihrer Belegung unverändert. In der prozeduralen Wissensbasis des Automaten erfüllt die so vorhandene Variablenbelegung die Bedingungen der Regeln R_1 und R_4. Entsprechend der geltenden Konvention fällt die Wahl hierbei auf die Regel mit dem geringeren Indexwert, wodurch R_1 und damit a_1 als momentan passendste Handlungsoption ausgewählt wird – vgl. hierzu Tab. 6.24.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0005	$EU(a_0) =$	-0.1448
R_1 :	**true**	r_1 :	0.0465	$EU(a_1) =$	0.0015
R_2 :	false	r_2 :	0.0559	$EU(a_2) =$	0.0499
R_3 :	false	r_3 :	0.0115	$EU(a_3) =$	0.3357
R_4 :	true	r_4 :	0.0139	$EU(a_4) =$	**0.6352**
R_5 :	false	r_5 :	0.4525		
R_6 :	false	r_6 :	0.3762		
		r_7 :	0.0021		
		r_8 :	0.0169		
		r_9 :	0.0006		
		r_{10} :	0.0025		
		r_{11} :	0.0204		

Tabelle 6.24: Ergebnis Versuchsreihe 5, Versuch III

Auf Basis der Ausfälle folgert der kausale Führungsautomat eine Änderung in der Belegung der Variablen H, welche von $H = h_3$ auf $H = h_1$ wechselt. Dies initiiert den Belief Propagation Algorithmus des kausalen Netzwerks, wodurch der Automat als Resultat für die Wurzelknoten I, J und M die entsprechenden Belief States $Bel(I) = [0.0432, 0.9567]$, $Bel(J) = [0.4539, 0.5460]$ bzw. $Bel(M) = [0.8662, 0.0266, 0.1071]$ berechnet. Die in Folge durchgeführte Aktualisierung der prozeduralen Wissensbasis führt anschließend dazu, dass $a = argmax_a(EU(a_x|\mathfrak{e}) = a_4$ als Handlungsoption mit dem gegenwärtig höchsten zu erwartenden Nutzen ermittelt wird.

Die bereits im letzten Versuch vorherrschende Abweichung in der Wahl der passendsten Handlungsoption setzt sich in diesem Versuch fort. Aus Sicht der blauen Jägergruppe ergibt sich hinsichtlich des sichtbaren Feindes keine konkrete Lageänderung, da der aufgeklärte Trupp weiterhin mit dem Rücken zur eigenen Front steht. Die Intensivierung der Ausfälle deutet nun umso stärker auf ein feindliches Manöver im nicht aufgeklärten Bereich hin, weswegen die Wahl zur Fortführung des Angriffes (a_1) nicht zweckmäßig und in Anbetracht der Umstände auch nicht mehr plausibel ist. Vielmehr weist die Lage darauf hin, dass eigene Kräfte sich in akuter Gefahr befinden, trotz der eigenen zahlenmäßigen Überlegenheit aufgerieben zu werden. Hierdurch ist der Gesamterfolg der Operation nicht mehr zu gewährleisten, weswegen das Ausweichen (a_4) zur Verhinderung des Aufreibens grundsätzlich zweckmäßiger ist. Diese Ableitung vermag der kausale Führungsautomat erneut auf Basis seiner deklarativen Wissensbasis zu ziehen.

So sorgten die anhaltenden Verluste für einen Wechsel in der taktischen Ausgangslage, obgleich die Eintrittswahrscheinlichkeit einer hohen Gefahr ($I = i_2$) unverändert zum vorhergehenden Versuch ist. Nichtsdestotrotz führt die nachhaltige Verringerung der Stärke zu einem Wechsel in der Bewertung des Risikos, wodurch der Automat nun annähernd indifferent ob eines vorherrschenden Risikos ist. Ausschlaggebend erweist sich dennoch die Einschätzung des Kampfwerts, welche durch die nun vorherrschende Unterschreitung der ehemals vorherrschenden zahlenmäßig notwendigen Überlegenheit als niedrig ($M = m_1$) bemessen wird. In Summe folgert der kausale Automat damit eine niedrige Erfolgschance für den Angriff, weswegen die Fortführung der Angriffsoperation außer Frage steht.

Aufreiben aufgrund Feindfeuer im Rücken

Der Versuchsaufbau des vorhergehenden Versuches wird erneut übernommen. Die Stärke der Jägergruppe wird abermals um eins reduziert, wodurch diese nun auf Vier gesetzt ist. Dies führt im Falle des regelbasierten Automaten zur Änderung in der Belegung der Variablen B_2 und B_3, welche entsprechend der Reduzierung der Stärke mit $B_2 = \top$ und $B_3 = \bot$ neu belegt werden. Alle weiteren Variablen verbleiben unverändert und entsprechend der vorherrschenden Belegungen des Vorversuchs. In der prozeduralen Wissensbasis des Führungsautomaten resultiert die beschriebene Lage in der Erfüllung der Antezedenzen der Regeln R_4 und R_6 – vgl. Tab. 6.25. Gemäß der gültigen Konvention wird nun die Regel mit dem niedrigeren Indexwert und damit R_4 gewählt. Hierdurch kommt der regelbasierte Automat zum Entschluss, dass Handlungsoption a_4 die aktuell passendste Option darstellt.

Regelbasiert		**Kausal**			
R_0 :	false	r_0 :	0.0005	$EU(a_0) =$	-0.1253
R_1 :	false	r_1 :	0.0621	$EU(a_1) =$	0.0013
R_2 :	false	r_2 :	0.0415	$EU(a_2) =$	0.0501
R_3 :	false	r_3 :	0.0154	$EU(a_3) =$	0.2431
R_4 :	**true**	r_4 :	0.0103	$EU(a_4) =$	**0.6792**
R_5 :	false	r_5 :	0.3358		
R_6 :	true	r_6 :	0.5026		
		r_7 :	0.0020		
		r_8 :	0.0166		
		r_9 :	0.0003		
		r_{10} :	0.0013		
		r_{11} :	0.0111		

Tabelle 6.25: Ergebnis Versuchsreihe 5, Versuch IV

Zur selben Zeit ändert der kausale Führungsautomat durch die erneut aufgekommene Reduzierung der eigenen Stärke die Belegung der Variablen B von $B = b_3$ auf $B = b_2$. Die hierdurch erneut in Gang gesetzte Aktualisierung des kausalen Netzwerks führt zur Neuberechnung der Belief States der Wurzelknoten, was konkret in $Bel(I) = [0.0320, 0.9679], Bel(J) = [0.5994, 0.4005]$ und $Bel(M) = [0.8662, 0.0266, 0.1071]$ gipfelt. Dies führt zur gewohnten Auffrischung der Werte der probabilistischen Formeln, woraus der Automat $a = argmax_a(EU(a_x|\mathfrak{e}) = a_4$ als Handlungsoption mit dem gegenwärtig höchsten zu erwartenden Nutzen folgert.

Damit wählen regelbasierter und kausaler Führungsautomat dieselbe Handlungsoption. Dies entspricht dem bereits aufgeführten Verhalten der Automaten unter Aufreiben, welches bereits in Versuchsreihe 3, Versuch IV entsprechend untersucht wurde. Die dort aufgeführte taktischen Bewertung ist auch an dieser Stelle gültig. In diesem Zusammenhang ist jedoch hinzuzufügen, dass die fortschreitende Lageentwicklung innerhalb des kausalen Netzwerks in der Bewertung des momentan vorherrschenden Risikos hier von besagtem vorherigen Versuch abweicht. Denn wie zu erkennen ist, bewertet der kausale Führungsautomat das momentane Risiko eher als gering ($J = j_1$), was in Anbetracht der Tatsache, dass die blaue Jägergruppe aus dem nicht aufgeklärten Bereich sukzessive aufgerieben wurde, keine korrekte Einschätzung der Lage darstellt. Hier ist vielmehr die Bewertung des vergleichbaren, vorhergehenden Versuchs ($J = j_2$) plausibel. Nichtsdestotrotz ändert dieser Umstand die letztendliche Auswahl des Automaten nicht, da nach wie vor in der gegenwärtigen Situation ein niedriger Kampfwert ($M = m_1$) und eine hohe Gefahr für die eigenen Kräfte ($I = i_2$) vorherrschen und einem Angriff damit keine hohe Erfolgschance zugeschrieben werden kann. Damit kommt der kausale Automat zur zweckmäßigen Entscheidung, dass die Gruppe auszuweichen hat (a_4), trotz der Fehleinschätzung innerhalb des Zustandsraums der Variablen J.

6.4 Auswertung des Experiments

Der folgende Abschnitt beschreibt die Auswertung der Experimentergebnisse. Die insgesamt festzustellenden Gemeinsamkeiten werden für die betrachteten Untersuchungen aus qualitativer Sicht dargestellt und die Robustheit anhand spezifischer Kriterien in den veränderten Ausgangslagen bewertet. Abschluss stellt das hieraus zu folgernde Fazit dar.

Als hauptsächliche Metrik wird die Plausibilität der getroffenen Führungsentscheidungen zur Auswertung verwendet. Die jeweilige Plausibilitätsbewertung erfolgte dabei im Fließtext der jeweiligen Einzelversuche. Sie entspricht dem Urteil des am Experiment beteiligten Domänenexperten. Entsprechend der in Kapitel 2.3 aufgestellten Definition des Plausibilitätsbegriffes, basiert dieses Urteil dabei auf der Grundlage, dass die jeweilige durch den Führungsautomaten getroffene Entscheidung genau dann als plausibel bezeichnet werden kann, wenn diese taktisch nachvollziehbar ist. Erwähnenswert ist in diesem Zusammenhang, dass innerhalb der Einzelversuche mehrere unterschiedliche Möglichkeiten des Handelns vorliegen, welche einzeln betrachtet jeweils plausibel sein können. Entscheidungen der Führungsautomaten können daher auch abweichen und dennoch gleichermaßen plausibel sein.

Erster Auswertungsgegenstand stellt die Untersuchung dar, ob die jeweils unter Sicherheit getroffenen Entscheidungen gleichmeraßen plausibel sind. Dies umfasst all jene Versuchsreihen, welche gemäß der vorgegebenen Ausgangslage das für die Automaten intendierte Verhalten zeigen. Konkret ist dies in den Versuchsreihen 1 - 3 der Fall, welche zusammen $n = 12$ Einzelversuche umfassen. Maßgeblich ist hierbei der Vergleich der Nachvollziehbarkeit in der jeweiligen gegebenen Lage und ob kausaler und regelbasierter Führungsautomat jeweils eine übereinstimmend plausible Entscheidung trafen. In diesem Sinne herrscht genau dann gemäß der o. a. Metrik eine Übereinstimmung vor, wenn entweder dieselbe oder eine taktisch ebenso plausible Entscheidung getroffen wurde. Entsprechend der taktischen Interpretation der jeweiligen Versuchsergebnisse weisen die Automaten $n_{kausal} = n_{regel} = 12$ plausible Ergebnisse vor. Kausaler und regelbasierter Führungsautomat sind somit innerhalb der betrachteten Einzelversuche qualitativ nicht zu unterscheiden. Die Entscheidungen der Automaten sind hier zwar nicht immer deckungsgleich, nichtsdestotrotz hat dies – wie bereits aufgeführt wurde – keine Auswirkung auf die Plausibilität. Somit ist festzustellen, dass beide Führungsautomaten bei den Entscheidungen unter Sicherheit im gleichen Maße plausible Ergebnisse erzeugen.

Zweiter Auswertungsgegenstand stellt die Untersuchung der Automaten hinsichtlich ihrer Robustheit in veränderten Ausgangssituationen und zusammenhängenden Lageentwicklungen dar. Dies umfasst konkret die Versuchsreihen 4 und 5, welche zusammen $n = 7$ Einzelversuche beschreiben. Erster Schritt stellt hierbei die Feststellung dar, ob zwischen den experimentell ermittelten Entscheidungen der Führungsautomaten ein qualitativer Unterschied besteht, oder ob diese im Rahmen der Betrachtung gleichwertig sind. Parallel zum ersten Auswertungsgegenstand ist der taktischen Interpretation der Einzelversuche zu entnehmen, dass bei beiden Automaten mit $n_{kausal} = n_{regel} = 6$ ein überwiegendes Verhältnis zu Gunsten der Plausibilität vorherrscht. Im Gegensatz zum ersten Auswertungsgegenstand haben hier die Abweichungen in den getroffenen Entscheidungen nun zur Folge, dass die Plausibilität in jeweils einem der betrachteten

Fälle nicht mehr gegeben ist. Qualitativ unterscheiden sich damit beide Automaten auf Basis der reinen Anzahl plausibler Führungsentscheidungen nicht. Als Ergebnis des ersten Schrittes ist somit festzustellen, dass beide Führungsautomaten sich hinsichtlich der Plausibilität nicht unterscheiden. Durch die Anzahl der plausiblen Führungsentscheidungen konnte damit kein experimenteller Nachweis dafür gefunden werden, dass sich bei Einsatz eines kausalen Führungsautomaten die Plausibilität in geänderten Ausgangslagen erhöht.

Gleichwohl ist in diesem Zusammenhang festzuhalten, dass die Bewertung der Plausibilität grundsätzlich der Einschätzung des Experten folgt, welcher in der Entwicklung der Regelsätze respektive Wissensbasen involviert war. In diesem Sinne stellt die Tatsache einer plausiblen Führungsentscheidung des jeweiligen Automaten kein hinreichendes Maß zur Bemessung der Robustheit dar, da diese keine objektive Bewertung der Robustheit im Führungsverhalten bietet. Aus diesem Grund ist es als zweiter Schritt notwendig, die Experimentergebnisse auf Grundlage anderweitiger Kriterien und damit fernab der reinen Plausibilität zu bewerten. Dies verfolgt das Ziel – insbesondere unter Einbezug der Bewertung nicht an der Entwicklung beteiligter Personen – die tatsächliche Robustheit der Ansätze ermitteln zu können. Die Entscheidung fiel hierbei auf die Kriterien eines Referenzmodells zur Bewertung militärischer Führungsansätze gem. (NATO, 2006). Folgende Kriterien wurden hierbei mit Blick auf das Experiment aufgrund ihrer Relevanz für die Aussagekraft zur Robustheit der durch die Automaten gefällten Entscheidungen identifiziert (NATO, 2006, 133):

- **Genauigkeit der Entscheidung**

  Die Entscheidung steht nicht im Gegensatz zum Auftrag.

- **Konsistenz der Entscheidung**

  Die Entscheidung ist konsistent zur formulierten Absicht der übergeordneten Führung.

- **Korrektheit der Entscheidung**

  Die Entscheidung ist konsistent zu dem durch die Jägergruppe wahrgenommenen Lagebild.

- **Relevanz der Entscheidung**

  Die Entscheidung ist der Erfüllung des momentanen Auftrags zuträglich.

- **Unsicherheit in der Entscheidung**

  Die Entscheidung bezieht Ungewissheit in der Lage mit ein.

Auf eine Bewertung der getroffenen Entscheidungen durch einen einzelnen Experten – wie sie zur Bewertung der Plausibilität stattfand – wurde verzichtet, da diese, wie es bereits erwähnt wurde, abermals nur eine subjektive Bewertung der Robustheit widerspiegeln würde. Um ein objektives Bild zu gewinnen, wurde daher auf die Bewertung mehrerer Experten zurückgegriffen. Dies erfolgte mithilfe eines Fragebogens, in welchem die einzelnen Lagebilder und getroffenen Entscheidungen der Versuchreihen 4 und 5 zur Debatte standen. Besonderes Augenmerk wurde hierbei bei der befragten Expertengruppe auf eine hohe Diversität hinsichtlich der zugehörigen Truppengattung gelegt, wobei jedoch alle Befragten sowohl eine infanteristische Vorbildung, als auch eine erfolgreich abgeschlossene Taktik- und Führungsausbildung

besaßen. Dies hatte den Zweck, trotz kleiner Stichprobengröße, eine möglichst representative Meinung abzubilden. Die Bewertung erfolgte dabei mithilfe einer Ordinalskala, welche das Zutreffen des jeweiligen Kriteriums bemisst und mit einem Punktewert hinterlegt wurde. In diesem Sinne erging die Festlegung gem. Tab. 6.26, d. h. dass mit Zutreffen des Kriteriums ein positiver und bei Nichtzutreffen ein negativer Punktewert verliehen wird.

Trifft nicht zu	**Trifft eher nicht zu**	**Indifferent**	**Trifft eher zu**	**Trifft zu**
-1.0	-0.5	0.0	+0.5	+1.0

Tabelle 6.26: Bewertungsschlüssel zur Auswertung der Umfrage

Die bewertete Robustheit des Ansatzes ergibt sich dann durch den erzielten Punktewert über alle Kriterien und Entscheidungen des betrachteten Experimentausschnitts. Maximal konnten innerhalb der Befragung 35 Punkte, minimal - 35 Punkte vergeben werden. An der Befragung waren insgesamt $n = 7$ Experten des Heeres im Dienstgrad zwischen Hauptmann und Oberst beteiligt. Der in der Befragung jeweils über alle Einzelversuche kumuliert vergebene Punktewert ist der nachfolgenden Abb. 6.9 zu entnehmen. Die Datengrundlage hierfür ist Anlage A.4 zu entnehmen.

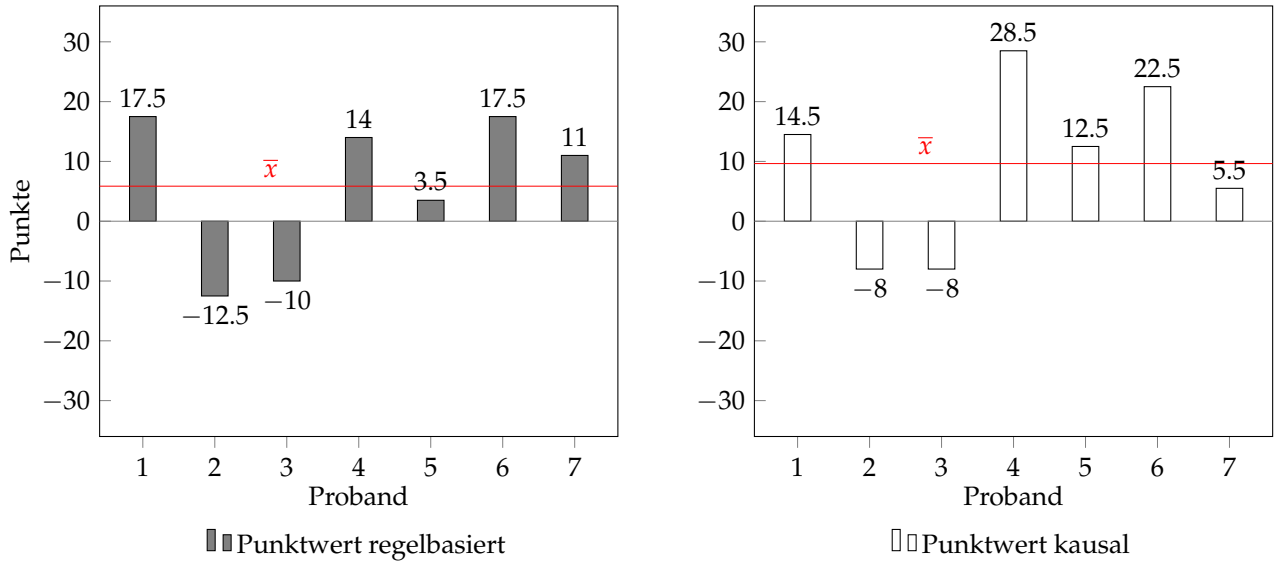


Abbildung 6.9: Umfrageergebnisse zu den Versuchsreihen 4 und 5

Insgesamt erhielt der regelbasierte Automat im Schnitt $\overline{x}_{Regel} = 5.9$ Punkte. Der kausale Führungsautomat erhielt im Vergleich hierzu im Schnitt $\overline{x}_{Kausal} = 9.6$ Punkte und somit mehr Punkte als dessen regelbasiertes Pendant. Auf dieser Bewertung aufbauend ist damit als Ergebnis des zweiten Schrittes festzuhalten, dass der intensionale Ansatz grundsätzlich durch Experten als robuster befunden wurde. Der zweite Auswertungsgegenstand lässt somit erkennen, dass regelbasierte und kausale Führungsautomaten in veränderten Ausgangssituationen und zusammenhängenden Lageentwicklungen im gleichen Maße plausible Entscheidungen erzeugen. Über die eigentliche Plausibilität hinausgehend ist festzustellen, dass kausale Automaten

auf Basis der subjektiven Meinung militärischer Experten insgesamt flexibler als klassisch regelbasierte Ansätze zu agieren scheinen.

Fazit des Experiments ist demnach, dass auf Grundlage der Versuchsreihen 1 - 3 regelbasierter und kausaler Führungsautomat im intendierten Anwendungszweck, d. h. im Einsatz innnerhalb vorab definierter und beabsichtigter Situationsräume, gleichwertig hinsichtlich der zu erwartenden Plausibilität agieren. Bei Abänderung der Situationsräume aus dem beabsichtigten Einsatzspektrum heraus – und insbesondere unter zusammenhängenden Lageentwicklungen – suggerieren die Versuchsreihen 4 und 5, dass kausale Automaten eine erhöhte Robustheit vorweisen.

7 Interpretation der Ergebnisse

Wie dem Schluss des vorherigen Kapitels zu entnehmen ist, wiesen regelbasierte Automaten innerhalb des Experiments eine geringere Robustheit im Vergleich zu den in dieser Arbeit etablierten kausalen Führungsautomaten auf. Dieses Kapitel beleuchtet im Detail, worin dieser experimentell ermittelte Unterschied begründet liegt und welche Bedeutung die gesteigerte Robustheit kausaler Führungsautomaten für zukünftige Anwendungsfälle hat. Eine abschließende Zusammenfassung der hierbei aufgeführten Fakten stellt, in Verknüpfung mit den Ergebnissen der vorhergehenden Kapitel, anschließend das konkrete Forschungsergebnis dieser Arbeit dar.

Begonnen wird mit dem regelbasierten Führungsautomaten und dem Grund dafür, warum diesem im Experiment eine niedrigere Robustheit nachgewiesen werden konnte. Als größtes Manko erwies sich hierbei das in Kapitel 3.1.2 im Kontext extensionaler Ansätze erwähnte Prinzip der Modularität, welches in allen deduktiven Systemen und damit insbesondere in regelbasierten Automaten vorzufinden ist. Wie sich anhand der Experimentergebnisse nachweisen lässt, beeinflusste die Modularität die Robustheit des Führungsautomaten negativ. Indizien lassen sich hierfür innerhalb des Experiments vor allem dort finden, wo zwei Regeln mit sich gegenseitig ausschließenden Aktionen zeitgleich zutrafen. Dies war nur in den Versuchsreihen 4 und 5 der Fall und damit also genau in jenen Versuchen, in welcher der Automat in eine unbekannte Situation versetzt wurde. Hierzu wird als erstes auf Versuchreihe 4 näher eingegangen.

Wie nämlich u.a. mit einem Blick auf Versuchsreihe 4, Versuch II (Tab. 6.20) zu sehen ist, weicht die Entscheidung des regelbasierten Automaten von der Entscheidung des kausalen Führungsautomaten ab. Dieser Umstand ist trügerisch, da der regelbasierte Automat de facto durch eine andere Auflösung der validen Regeln zu exakt denselben Entscheidungen wie dessen Pendant hätte kommen können. So führt im betrachteten Versuch die Verarbeitung der Sensorinputs dazu, dass sowohl Regel 1 als auch Regel 4 in der Wissensbasis des regelbasierten Führungsautomaten mit wahr belegt werden. Dies stellt eine taktische Zwickmühle dar, da der Führungsautomat somit gleichermaßen zwischen den sich gegenseitig ausschließenden Optionen „Angriff“ und „Ausweichen“ abwägen muss. Aus Sicht des Automaten ist dieses Dilemma logisch, da Regel 1 lediglich die Prüfung der eigenen Position zum Feind und die eigene Stärke umfasst, nicht jedoch die feindliche Stärke mit einbezieht. Dadurch sieht der Automat die im Versuch vorherrschende flankierenden Position als günstige Ausgangssituation, auf deren Grundlage gemäß Regel 1 der Angriff eine valide Handlungsoption darstellt. Gleichzeitig besagt jedoch Regel 4, dass bei stärkerem Feind – ganz der ursprünglichen Befehlsgebung folgend – die blaue Gruppe auszuweichen hat, was gerade eben jenen Fall darstellt, der angesichts des wahrgenommenen Feindes in Gruppenstärke auch eingetreten ist. Diese zeitgleiche Ableitung zweier eigentlich einander ausschließender Regeln, stellt ein klares Indiz für die Loslösung des Wissens vom Kontext dar, da die Herleitung des Zutreffens der jeweiligen Regel die situativ ebenfalls vorherrschenden Umstände (z. B. die Stärke der feindlichen Gruppe oder die Überlegenheit der eigenen Position) ignoriert.

Die Loslösung vom Kontext führt im dargestellten Beispielfall den Führungsautomaten also dazu, sich zwischen zwei validen, jedoch taktisch komplett unterschiedlichen Handlungsoptionen zu entscheiden. Da nur eine der validen Handlungsoptionen die Entscheidung darstellen kann, wird der vorherrschende Regelkonflikt auf Basis der Konvention des Testsystems durch pauschale Auswahl des niedrigeren Indexwertes aufgelöst. Damit kommt der Automat durch Auswahl von Regel 1 zur letztendlich im Versuch getroffenen Entscheidung eines Angriffes, wodurch der regelbasierte Mechanismus im weiteren Sinne auch die restlichen Inhalte der Wissensbasis – namentlich das Zutreffen von Regel 4 – ignoriert. Dies stellt die Lokalität des Mechanismus heraus, welches das zweite Charakteristikum des Modularitätsprinzips verkörpert. Hierbei ist leicht zu sehen, dass eine Präferenz zugunsten höherer Indexwerte Regel 4 bevorzugt und somit die Entscheidungen des regelbasierten Automaten innerhalb der Versuchsreihe 4 den Entscheidungen des kausalen Führungsautomaten angeglichen hätte. Generell ist in diesem Zusammenhang jedoch festzuhalten, dass die Auflösung von Regelkonflikten innerhalb eines regelbasierten Automaten keine pauschale und damit optimale Lösung bietet. Die Modularität des deduktiven Schlussmechanismus sorgt hier dafür, in Verbindung mit der Notwendigkeit zur exakten Definition der jeweiligen situativen Prämisse, dass keinerlei Flexibilität in den beschriebenen Situationen vorherrschen kann.

Ein weiteres Indiz für den negativen Einfluss der Modularität ist die erzeugte Unfähigkeit zur Verarbeitung einer unscharfen Lage. Dies wird insbesondere mit Blick auf die Versuche 5 II und 5 III deutlich. Die dort durch den feindlichen Beschuss ständig fortgeführte Reduktion der eigenen Stärke sollte aus taktischer Sicht dazu führen, dass der regelbasierte Führungsautomat entsprechende Maßnahmen ergreift, damit die eigenen unterstellten Kräfte aus der akuten Gefahrensituation heraus bewegt werden. Zum Erkennen der Gefahr ist der Automat jedoch nicht in der Lage. Hierfür mangelt es diesem an einen geeigneten Sensor, wodurch weder der Beschuss auf den unterstellten Bereich, noch die eigene offensichtlich ungünstige Position direkt wahrgenommen werden kann. Erneut zeigt sich auch hier auf Grundlage der Loslösung vom Kontext, dass der regelbasierte Mechanismus lediglich auf Basis der rein evidenten Informationen Schlüsse zieht, weitere hintergründige Informationen jedoch mangels zusammenführender Regeln nur in vorab fest vorgegebene Situationen klassifziert. Damit ist dieser schlichtweg in unbekannten Situationen nicht in der Lage, passende Lageeinschätzungen zu bilden und fällt daher, wie in den erwähnten Versuchen deutlich wird, den Entschluss zur ihm einzig situativ validen Entscheidung eines Angriffes. Hier ist auch erneut die Lokalität des Schlussmechanismus zu erkennen, da dabei der wissenstechnisch vorhandene und anhand der reduzierten eigenen Stärke eigentlich zwingend festzustellende Beschuss ignoriert wird.

Somit ist für Versuchsreihe 5 – simultan zu Versuchsreihe 4 – festzuhalten, dass die niedrigere Robustheit des regelbasierten Führungsautomaten primär im Modularitätsprinzip des aussagenlogischen Schlussmechanismus begründet liegt. Im Vergleich ist festzustellen, dass der kausale Führungsautomat nicht durch das Prinzip der Modularität beeinflusst wird. Dies ist besonders ersichtlich in den zuletzt betrachteten Versuchen 5 II und 5 III, in welchen trotz des Rückgriffs des Automaten auf dieselben evidenten Systemzustände, die Lagebeurteilung vom Kontext nicht losgelöst ist. Ausschlaggebend ist hier die Zusammenführung der evidenten Zustände innerhalb der deklarativen Wissensbasis, welche, neben dem situativ vorliegenden Lagebild, ebenfalls die vom unentdeckten Feind ausgehende Gefahr in die Entscheidungsfindung mit einfließen lässt. Möglich ist dem Automaten dies durch die Verknüpfung der Systemzustände mit dem probabilistischen Schlussmechanismus, welcher im kausalen Netzwerk auf Grundlage der gegebenen Verknüpfungen

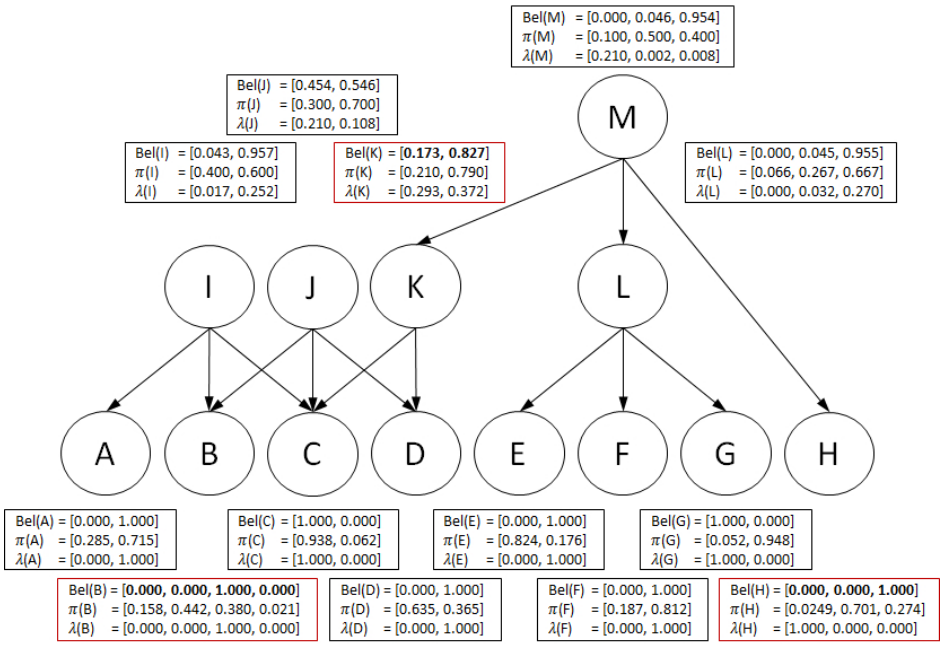


Abbildung 7.1: Belegung des kausalen Netzwerks in Versuch 5 II

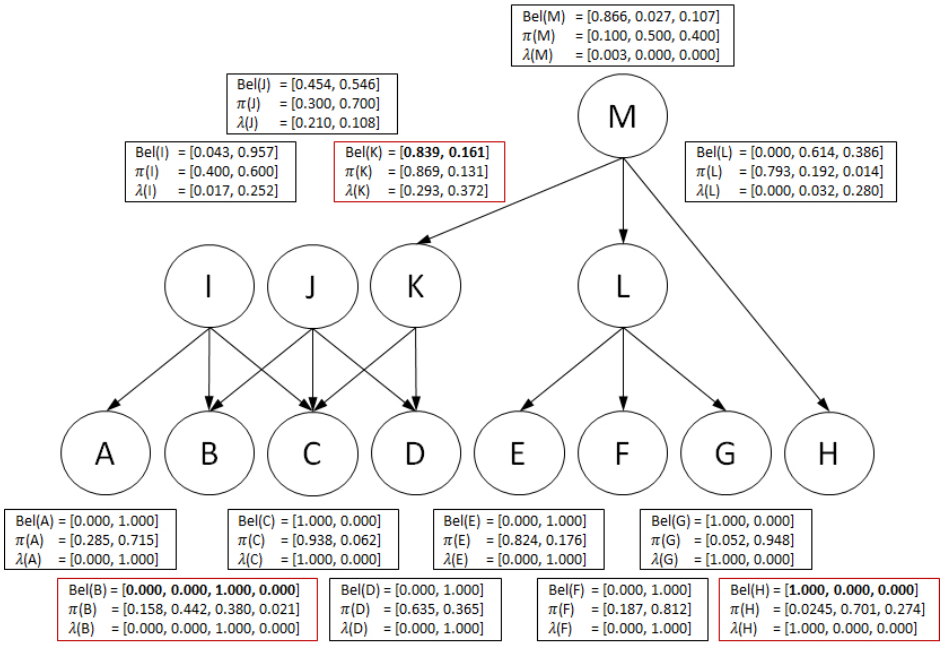


Abbildung 7.2: Belegung des kausalen Netzwerks in Versuch 5 III

immer und zu jeder Zeit alle möglichen Informationsstände abzuwiegen vermag. Als Verdeutlichung sei hier auf Abb. 7.1 und Abb. 7.2 verwiesen, welche die Belegung des kausalen Netzwerks in Versuch 5 II und 5 III und damit den fortschreitenden Einfluss des indirekt wahrgenommenen Beschusses auf die Lagebeurteilung aufzeigen. Besonderes Augenmerk sei hier auf die Beurteilung der eigenen Stärke (B), auf den Kräftevergleich (H) und des Beschusses auf eigene Kräfte (K) gelegt. Wie Abb. 7.1 zeigt, herrscht aufgrund der Reduktion der eigenen Stärke der wesentliche Belief State $B = b_3, H = h_3$ und $K = k_2$ innerhalb des kausalen Netzwerks vor. Damit schließt der Automat nicht vollends auf einen vorliegenden direkten Beschuss seitens feindlicher Kräfte, bezieht jedoch die grundsätzliche Möglichkeit mit $Bel(b_1) = 0.173$ mit ein. Diese hat jedoch realistischerweise keine tiefergehende Auswirkungen auf die Entscheidung, da die blauen Kräfte nach wie vor ein überlegenes Kräfteverhältnis besitzen.

Die Fortsetzung des Beschusses und die dadurch voranschreitende Reduktion der eigenen Kräfte zeigt jedoch in Abb. 7.2, dass dieser durchaus zur Veränderung der situativen Lageeinschätzung führt. Wie dort zu sehen ist, wechselt auf Grundlage der Reduktion der eigenen Stärke die Belegung von $H = h_3$ auf $H = h_1$, womit nun ein nachteiliges Kräfteverhältnis zum erkannten Feind vorherrscht, wohl jedoch nach wie vor die Beurteilung der Stärke der eigenen Kräfte mit $B = b_3$ unverändert bleibt. Anders als den regelbasierten Automaten führt dies nun den kausalen Führungsautomaten zur Erkenntnis, durch den ebenfalls vorliegenden Wechsel der wahrscheinlichsten Ausprägung von $K = k_2$ zu $K = k_1$, dass ein direkter Beschuss auf eigene Kräfte vorliegt. Damit ist der probabilistische Schlussmechanismus folglich nicht losgelöst vom Kontext. Zusätzlich greift der kausale Führungsautomat für Entscheidungen auf die Bewertung des zu erwartenden Nutzens in allen möglichen Situationsräumen zurück, wodurch dieser auch nicht der Lokalität unterworfen ist. Bestärkt wird letzteres Argument vor allem durch die Tatsache, dass durch die Beachtung des individuellen Nutzens aller möglichen Handlungsoptionen – über alle Situationsräume hinweg – das übrige vorhandene Wissen innerhalb der Wissensbasis nicht ignoriert wird. Der kausale Führungsautomat ist somit tatsächlich – im Gegensatz zum regelbasierten Automaten – nicht dem Modularitätsprinzip unterworfen. In Anbetracht der durch die Expertenbewertung höheren zugewiesenen Robustheit des probabilistischen Schlussmechanismus, lässt dies im Vergleich zum regelbasierten Automaten den Schluss zu, dass hierin der Grund für die höhere Robustheit im Führungsprozess begründet liegt.

Insgesamt gibt das dargestellte Verhalten des kausalen Schlussmechanismus auch einen Hinweis auf eine große Stärke: Die getroffenen Entscheidungen berücksichtigen immer den zuletzt noch innerhalb der Wissensbasis vorhandenen Wissensstand, ganz anders als dessen regelbasierter Konkurrent, welcher ohne Blick auf das im vorherigen Zeitschritt vorhandene Wissen und damit nur aus dem Moment heraus Schlüsse zu ziehen vermag. Dies ist ein weiteres mögliches Indiz für den Grund der höheren Bewertung der Robustheit des kausalen Führungsautomaten. Der Automat ist hierdurch nämlich u. a. in der Lage, den nicht direkt durch einen evidenten Zustand feststellbaren Beschuss aus dem Zustandswechsel des Kräftevergleichs abzuleiten, wie sich im Experiment innerhalb der Versuchsreihe 5 durch die bisherigen Ausführungen bestätigt. Formal liegt diese Eigenschaft in der technischen Umsetzung der Zustandsverarbeitung auf Grundlage des Belief Propagation Algorithmus gem. Kapitel 4.1 begründet. Wie dort in der Herleitung dieses Mechanismus und den zusammengefassten Ausführungen von Pearl (1988) bzw. Kim und Pearl (1983) entsprechend zu entnehmen ist, führt die lokale Veränderung eines Zustands durch den verwendeten Algorithmus zu einer aussagekräftigen und lokalen Berechnung der Auswirkungen. Die Nützlichkeit dieses Umstands für

Führungsautomaten wird entsprechend der Experimentergebnisse durch das so erhöhte Bewusstsein der eigenen Lage in nicht explizit formulierten Situationen untermauert. Gleichwohl ist festzustellen, dass der Algorithmus Charakteristika der Modularität aufweist, da dieser Änderungen lediglich lokal und losgelöst von den Werten der nicht in der unmittelbaren Nachbarschaft liegenden Knoten auslöst.

Tatsächlich ist der Belief Propagation Algorithmus somit an sich auch in gewisser Hinsicht modular, was jedoch entsprechend der Argumentation der vorhergehenden Absätze an sich nicht dafür sorgt, dass auch der probabilistische Schlussmechanismus an sich dem Modularitätsprinzip unterliegt. Viel mehr ist hierbei sogar festzustellen, dass gerade die Lokalität des Algorithmus den probabilistischen Schlussmechanismus nicht vom Kontext loslöst. Dabei mag zwar die initiierte Propagierung aufgrund der lokalen Veränderung der Variablen ausgelöst und auch alle weiteren zugleich vorliegenden Änderungen vorläufig ignoriert werden. Jedoch führt die schrittweise Abarbeitung der vorherrschenden Änderungen im Propagierunsalgorithmus zur Aktualisierung aller relevanten Knoten des Netzwerks. Als relevant erweisen sich dabei alle innerhalb der sog. Markov-Decken[1] befindlichen Knoten, welche exemplarisch für die Versuchsreihe 5 der folgenden Abb. 7.3 zu entnehmen sind. Wie in der Abbildung hierbei erkennbar ist, führt eine angestoßene Änderung innerhalb der Beurteilung des Kräftevergleichs (H) im ersten Schritt zur Änderung innerhalb der Bestimmung des Kampfwerts (M), was wiederum eine Auswirkung auf die untergeordneten Merkmale in der Beurteilung der eigenen Position (L) und des Vorliegen eines Beschusses (K) hat.

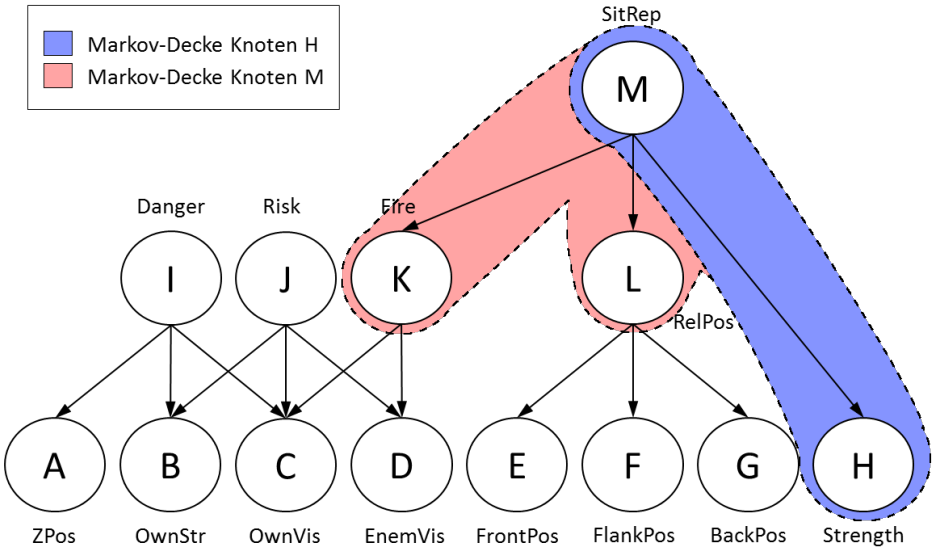


Abbildung 7.3: Relevante Markov-Decken des kausalen Netzwerks in Versuchsreihe 5

Im Umkehrschluss bedeutet dies, dass kausale Führungsautomaten somit – wie in Kapitel 3 bereits entsprechend antizipiert wurde – für weitaus weniger Situationen im vorhinein kalibriert werden müssen,

1 Eine Markov-Decke beschreibt auf Basis der innerhalb des probabilistischen Netzwerks explizit dargestellten Abhängigkeitsbeziehungen, welche Knoten entsprechend durch die Änderung des Belief State eines im Netzwerk befindlichen Knoten beeinflusst werden. Gem. Pearl (1988, 121) liegen für einen beliebigen Knoten X alle Elternknoten, direkten Nachfolger und weitere Elternknoten der direkten Nachfolger innerhalb einer Markov-Decke.

als es regelbasierte Automaten bedürfen. Die rein auf Grundlage der kausal ableitbaren Abhängigkeiten definierten Zusammenhänge innerhalb der deklarativen Wissensbasis genügten, um im Entscheidungsverhalten unter Unsicherheit von Experten eine höhere Robustheit bescheinigt zu bekommen. Dies birgt großes Potenzial für zukünftige Anwendungsfälle, ganz besonders unter Berücksichtigung des Gesichtspunktes, dass innerhalb der prozeduralen Wissensbasis nur die drei Variablen der Wurzelknoten (I, J und M) für das aufgezeigte Entscheidungsverhalten kombiniert werden mussten. Der regelbasierte Automat stützte sich hier im Vergleich auf eine Kombination der sechs Zustandsvariablen ab, was u. a. zeigt, dass die vorgeschaltene deklarative Wissensbasis tatsächlich eine effizientere semantischen Erfassung möglicher Situationen erlaubt. Gleichwohl verbleibt es jedoch eine Tatsache, dass für explizit im Vorhinein definierte Situationen, wie im Experiment aufgezeigt wurde, zwischen regelbasierten und kausalen Führungsautomat kein größerer Unterschied hinsichtlich der situativen Beschreibungsfähigkeit vorherrscht.

In diesem Zusammenhang ist festzustellen, dass das Einbringen zusätzlicher Sensoren bzw. die Erweiterung der Wissensbasis den regelbasierten Führungsautomaten robuster gemacht hätte. Im Vergleich zum kausalen Automaten erscheint diese Erweiterung aus Gründen des Ceteris paribus zunächst gerechtfertigt, da der kausale Automat ein erheblich komplexeres Modell der Umwelt beschreibt. Hier ist jedoch klar festzustellen, dass das Hinzufügen von Kontext einen regelbasierten Automaten zwar für mehr intendierte Situationen wappnet, die Übertragung des Wissens auf allgemeine oder unbekannte Situationen jedoch nach wie vor nicht gegeben ist. Wie auch zu Beginn dieser Interpretation bereits festgestellt wurde, ist es insbesondere die Auflösung von Regelkonflikten, welche die Entscheidungen des regelbasierten Automaten in unbekannten Situationen von denen des kausalen Automaten unterschieden haben. Die Komplexität des unterliegenden Modells scheint demnach keine wirkliche Auswirkung auf die Flexibilität eines Automaten zu haben - es ist vielmehr der Schlussmechanismus an sich, welcher maßgeblich für die Robustheit verantwortlich ist. Hier zeigt sich anhand der Versuchsergebnisse der probabilistische Mechanismus der klassischen Aussagenlogik überlegen. Diese Erkenntnis erfolgt vollkommen unbeachtet der tatsächlichen Komplexität des jeweiligen Umweltmodells, da beide Modelle letztendlich Entscheidungen auf Grundlage der semantischen Kombination derselben evidenten Systemzustände treffen. Eine Erweiterung der Wissensbasis des regelbasieren Automaten, ggf. unter Einbringen neuer Regeln, würde diesen Umstand kaschieren.

Mit Blick auf die benötigte Kalibierungszeit und des dadurch entstehenden Aufwandes zur Feinjustierung, ist der regelbasierte Automat bei Ausschluss von Unschärfen effizienter, da dieser mangels deklarativer Wissensbasis schneller brauchbaren Output erzeugt. So benötigte bpsw. das Erstellen der Formeln für den regelbasierten Automaten im Experiment weniger als einen Arbeitstag. Die Kalibrierung und Feinjustierung des kausalen Automaten bedingte dort hingegen mit ca. drei Arbeitstagen gleichwohl mehr Aufwand. Damit deckt sich interessanterweise der zeitliche Bedarf beider Automaten mit den Erkenntnissen von Rochel (1990, 151). Im Kontrast hierzu sind jedoch hinsichtlich des benötigten Umfangs an Quellcode zwischen regelbasiertem und kausalem Ansatz nur marginale Unterschiede festzustellen. Als Folgerung ist daher festzuhalten, dass beide Umsetzungen durchaus ihre Berechtigung haben. Die Auswahl zwischen kausalen und regelbasierten Automaten sollte jedoch wohl auf Grundlage des zeitlichen Bedarfs mit Blick auf den Verwendungszweck der Simulation abgewogen werden.

Unter besonderer Berücksichtigung der vorhergehenden Kapitel folgt daher in Zusammenfassung als Forschungsergebnis:

- Regelbasierte Automaten sind als geläufigste Umsetzung von Führungsautomaten nicht in der Lage, Ungewissheit in der Lageeinschätzung zu adressieren. Dies begründet sich darauf, dass diese nur explizit benannte Situationen zu kategorisieren vermögen. In ihrer Schlussbildung sind diese dem Prinzip der Modularität unterworfen, welches dafür sorgt, dass diese vor allem in unbekannten Situationen Kontextwissen entweder nicht einzuordnen vermögen, oder gar komplett ignorieren.
- Im Vergleich möglicher Ansätze zur Abbildung von Ungewissheit wurden diagnostische Systeme untersucht, die in ihrem Aufbau zwischen extensionalen und intensionalen Ansätzen unterteilt werden können. Intensionale Ansätze erwiesen sich hier als geeigneter, da diese nicht nur eine ausdrucksstärkere Wissensgenerierung ermöglichen, sondern im Gegensatz zu extensionalen Ansätzen auch abduktive Schlüsse zulassen. Insbesondere Letzteres zeigte sich als äußerst vorteilhaft, da unter dem Einsatz eines probabilistischen Modells und dessen Zuordnung zu einem Graphen ein ausdrucksstarker Formalismus zur Umsetzung von Führungsautomaten gefunden werden konnte. Die Verknüpfung des Formalismus mit den klassischen Ansätzen eines hybriden wissensbasierten Automaten gipfelte im Entwurf der sogenannten kausalen Führungsautomaten.
- Innerhalb eines Experiments wurden sowohl regelbasierter und kausaler Führungsautomat miteinander verglichen. Hier konnte ermittelt werden, dass beide Automaten in vorab definierten Situationen und damit bei Entscheidungen unter Sicherheit gleichwertig agieren. Wie eine Befragung von Experten ergab, konnte sich in unbekannten Situationen der kausale Führungsautomat hinsichtlich der Robustheit vom regelbasierten Automaten absetzen.
- Der Grund für das bessere Abschneiden konnte im Rahmen der Interpretation darin identifiziert werden, dass der kausale Führungsautomat – ganz im Gegensatz zum regelbasierten Automaten – nicht dem Prinzip der Modularität unterliegt.
- Der probabilistische Schlussmechanismus des kausalen Führungsautomaten ist in der Formulierung der Entscheidungsregeln innerhalb der prozeduralen Wissensbasis effizienter. Nichtsdestotrotz ist festzustellen, dass der Einsatz von Regeln einen zeitlichen Vorteil gegenüber der Kalibrierung und Justierung von deklarativer und prozeduraler Wissensbasis aufweist. Damit ist insbesondere für Entscheidungen unter Sicherheit herauszustellen, dass der regelbasierte Automat hier grundsätzlich weiterhin seine Daseinsberechtigung besitzt.

8 Schlussbemerkungen

Diese Arbeit verfolgte das Ziel, die bisherigen Diskrepanzen bei der Einbindung von Unschärfen in momentan gebräuchlichen Führungsautomaten aufzuzeigen. Dies konnte mit Blick auf die Darstellung der unterschiedlichen Implementierungsformen, insbesondere im Kontext regelbasierter Automaten erreicht werden. Die hierbei gewonnenen Erkenntnisse sollten nicht sonderlich überraschen, da in vergangenen Untersuchungen bereits in vielerlei Hinsicht auf die stochastische Natur des Führungsprozesses aufmerksam gemacht wurde – vgl. hierzu (Rochel, 1990; Schnurer, 1995; Hofmann, 2000b). Die in diesem Sinne aufgezeigten Schwächen regelbasierter Führungsautomaten sind umso eindrücklicher, da diese – wie den entsprechenden Kapiteln zu entnehmen ist – ein Resultat der unterliegenden logischen Schlussmechanismen darstellen. Dies wird entsprechend auch von Reidelhuber (1979, 272) konstatiert:

> „*Für eine experimentelle Anwendung* [...] *geschlossener Simulationsmodelle stellt sich die kritische Frage, inwieweit die zuvor beschriebenen Entscheidungsregeln für Experimente gültig bleiben, denen ein geänderter Anfangszustand oder eine veränderte Struktur zugrunde liegt, selbst wenn man das taktische Konzept als unverändert ansieht. Diese Frage kann generell nicht beantwortet werden. Je besser jedoch die Menge der taktischen Entscheidungsregeln die für die jeweilige Entscheidungsebene charakteristischen Entscheidungsprozesse abbildet, desto flexibler und dadurch plausibler wird das damit gesteuerte Simulationsmodell auf taktisch unterschiedliche Situationen reagieren können.*“

Reidelhuber verweist hier auf die vielmals diskutierte Unstimmigkeit bisheriger Ansätze in der Modellierung militärischer Führungsprozesse. Unbestreitbar ist dabei der signifikante Einfluss von Ungewissheit auf den Führungsprozess, welcher in bisher entwickelten Ansätzen jedoch nur am Rande adressiert wurde. So legt u. a. auch Rochel (1990, 143) entsprechend nahe:

> „*Die Ungewißheit in militärischen Führungsprozessen resultiert im wesentlichen aus der Ungewißheit von Lagedaten und deren Verarbeitung. Da im Rahmen dieser Arbeit kein Modell zur (Multi-) Sensordaten- bzw. Lagedatenfusion entwickelt wurde, fand auch der Einfluß von Ungewißheit im vorgestellten Ansatz keine explizite Berücksichtigung. In der Literatur existieren verschiedene Ansätze zur Repräsentation von Ungewißheit* [...]*, deren Diskussion bezüglich der Modellierung militärischer Führungsprozesse weiteren Arbeiten vorbehalten bleiben soll.*“

In vielerlei Hinsicht baut diese Arbeit auf diesem fachlichen Fundament auf. Die Notwendigkeit für eine Lösung von den klassischen regelbasierten Systemen ist durchaus evident, was mitunter auch durch Russell und Norvig (2010, 548) entsprechend gestützt wird: „*The bad news for rule-based systems is the properties of locality, detachement, and truth-functionality are simply not appropriate for uncertainty reasoning.*“ Die in diesem Sinne aufgezeigten Forschungsergebnisse zugunsten intensionaler Ansätze folgen diesem mahnenden Grundgedanken. Inwiefern die Bundeswehr oder Streitkräfte anderer Nationen entsprechende Schritte zur Etablierung dieser Ansätze unternehmen, liegt jedoch sehr wohl außerhalb des Einflussvermögens dieser Arbeit. Dass die Grundsteine hierfür grundsätzlich aber bereits gelegt sind, wird u. a. im Konzept zur Simulationslandschaft des Heeres (KonzSimLdschH) 2020+ ersichtlich:

> *„Modularität der Komponenten und Flexibilität bezüglich der Ausbildungsinhalte von SimSys sind über die Fortschreibung und Umsetzung von Standards konsequent voranzutreiben. Dazu sind:*
> *[...]*
>
> - *querschnittliche Funktionalitäten der SimLdschH (z. B. Waffenwirkung, Wetter, Visualisierung,* ***künstliche Intelligenz/Führungsautomaten****) in Modulen zusammenzufassen und in die Systeme modulbasiert zu integrieren oder als Services vorzuhalten,*
>
> *[...]*
> *Die dazu notwendigen konzeptionellen Grundlagen sind zu erarbeiten und in Vorgaben und Maßnahmen zur Weiterentwicklung umzusetzen.“* (AHEntwg, 2016, 23f)
> [Absatz 405; Hervorhebungen d. d. Verf.]

Es ist jedoch zu betonen, dass die in dieser Arbeit betrachteten intensionalen Ansätze lediglich einen Anfang für die Betrachtung der Möglichkeiten unscharfer Wissensverarbeitung im Kontext der Modellierung von Führungsprozessen darstellen.

Inwieweit kausale Führungsautomaten hier den Test der Zeit bestehen, werden zukünftige Untersuchungen zeigen müssen. Im Lichte momentaner Trends scheint für die Streitkräfte jedoch eher der Fokus auf lernfähigen regelbasierten Ansätzen zu liegen, welche im Bereich des maschinellen Lernens zu verorten sind. Aus Sicht des Autors stellt dies einen Rückgang zu alten Mustern dar, da schon Tolk (1995) den Benefit lernfähiger Systeme für regelbasierte Automaten untersucht hat. Rein regelbasierte Mechanismen sind aus den in den vorhergehenden Kapiteln genannten Gründen im Kontext von Führungsautomaten als durchaus kritisch zu bewerten. Zusätzlich verspricht der momentane Trend nur die bereits vorliegenden Ergebnisse von Tolk zu bestätigen. Eine vertiefende Betrachtung des Einsatzes lernfähiger Verfahren unter Nutzung intensionaler Ansätze wird hier eher als zielführend gesehen, da diese – wie nachgewiesen wurde – eine höhere Robustheit vorweisen und zur Verarbeitung unscharfen Wissens geeignet sind. Kausale Führungsautomaten böten in diesem Sinne eine ideale Ausgangsbasis, da deren deklarative Wissensbasis eine Sonderform eines Bayes'schen Netzes darstellt, welches auch grundsätzlich lernfähig ist[1]. Alternativ zeigen die anderen intensionale Ansätze, z. B. die Fuzzy-Logik[2] in Verbindung mit neuronalen Netzen[3] ein vielversprechendes Potenzial für den Einsatz zur Modellierung von Führungsprozessen.

Letztlich – und dies ist an dieser Stelle zu betonen – ist jedoch nicht entscheidend, ob in nächster Zeit nun regelbasierte oder gar kausale Führungsautomaten im Schwerpunkt der Forschung und Weiterentwicklung stehen. Die Gefechtssimulation als solche ist auf eine stetige Weiterenwicklung der formalen Modelle des militärischen Führungsprozesses angewiesen, da das militärische Bedrohungsspektrum sich ebenfalls fortwährend weiterentwickelt. Nach wie vor vorherrschende Defizite wie die simulative Abbildung der Auftragstaktik, des geführten Feuerkampfes oder des infanteristischen Kampfes zeigen hier

1 Bayes'sche Netze sind in der Lage, deren Netzstruktur auf der Grundlage von Daten abzuleiten – vgl. (Russell und Norvig, 2010; Neapolitan et al., 2004; Pearl, 1988). Des Weiteren lassen sich mit hinterlegten relativen Häufigkeiten die unterliegenden Wahrscheinlichkeiten fallbezogen aktualisieren. Wie Pearl (1988, 125) entsprechend zu entnehmen ist, sind die Bezeichnungen Bayes'sches (Glaubens-)Netz und kausales Netzwerk gleichbedeutend. Damit weisen die im Schwerpunkt betrachteten kausalen Netzwerke ebenfalls diese grundsätzliche Lernfähigkeit auf.
2 Vgl. bspw. (Ernest et al., 2016) für die erfolgreiche Nutzung innerhalb von Ausbildungssimulatoren.
3 Vgl. hierzu (Wörsdörfer, 2014).

insbesondere einen zeitnahen Entwicklungsbedarf auf (Caussanel et al., 2007, 2ff). Welcher Ansatz hierfür geeignet sein wird, wird nur durch weitere Untersuchungen zu zeigen sein. Eine geeignetes Fundament zur Adressierung dieser Herausforderungen ist durch vergangene und aktuelle Forschungsarbeiten nun aber gewiss gebildet. Als abschließender Ausblick ist es jedoch erwähnenswert, dass ein fortwährender Stillstand bei der Entwicklung der Führungsautomaten der Zukunft der Gefechtssimulation nicht zuträglich ist.

A Anhang

A.1 Maximum Cardinality Search

Sei $G = (V, E)$ ein beliebiger ungerichteter Graph. Die Nachbarschaft einer Ecke $v \in V$ ist definiert als die Menge $N_G(v) = \{v \neq v' | (v, v') \in E\}$. Die bijektive Funktion $\alpha : V \rightarrow \{1, \ldots, |V|\}$ heißt Eliminationsordnung (engl. elimination ordering) auf G. Zusätzlich beschreibe $w(v) \in \mathbb{Z}$ das Gewicht einer Ecke v. Der **MCS-Algorithmus** (siehe Abb. A.1) weist jeder Ecke $v \in V$ einen Ordnungswert $\alpha(v)$ und allen adjazenten Ecken $v'_i \in N_G(v)$ ein Gewicht $w(v'_i)$ zu, welche der Kardinalität ihrer zugehörigen Nachbarschaft entspricht.

Algorithmus *MaximumCardinalitySearch - MCS*

Input: A graph G.
Output: An elimination ordering α of G;
Begin
  for all vertices v in G **do**
    $w(v) = 0$;
  end for
  for $i = |V| \ldots 1$ **do**
    Choose an unnumbered vertex z of maximum weight;
    $\alpha(z) = i$;
    for all unnumbered vertices $y \in N(z)$ **do**
      $w(y) = w(y) + 1$;
    end for
  end for
End

Abbildung A.1: Maximum Cardinality Search Algorithmus (Berry et al., 2002, 4)

Die mithilfe des MCS-Algorithmus erzeugte Eliminationsordnung eines Graphen beschreibt in absteigender Reihenfolge, welche Ecken entfernt werden müssen, um einen Graph in endlich viele, paarweise disjunkte Teilgraphen zu zerlegen. Gleichzeitig kann durch die Zuweisung der Kardinalitäten eine lexikographische Ordnung der Nachbarschaften erfolgen. Den einzelnen Cliquen von G kann somit – unter Zuweisung der jeweilig höchsten Ordnungswerts der inhärenten Ecken – ebenfalls eine Eleminitationsordnung zugewiesen werden. Den nachfolgenden Abschnitten ist die Anwendung des Algorithmus im Zusammenhang mit den Beispielen dieser Arbeit zu entnehmen.

Anwendung im Rahmen des Kapitels 3.2

Ergebnisse und Teilschritte des Algorithmus werden mithilfe eines Trace Tables dargestellt. Dabei beschreibe die Variable z entsprechend des Algorithmus die momentan gewählte (Ausgangs-)Ecke mit dem höchsten zugeordneten Kantengewicht $w(z)$. Im Falle eines Gleichstandes wird grundsätzlich immer erst nach links und möglichst immer nach oben aufgelöst, d. h. in der bereits gegebenen lexikographischen Anordnung des Ausgangsgraphen, wird als Konvention immer die Ecke mit der lexikographisch kleinsten zugehörigen Variable gewählt. Alle bereits mit einer Eliminationsordnung $\alpha(z)$ versehenen Ecken werden innerhalb der Zuweisung eines höheren Kantengewichts übersprungen und führen in Folge zu keinem Unterarbeitsschritt. Nachfolgender Tab. A.1 ist auf dieser Grundlage das Ergebnis der Ausführung des Algorithmus auf den Gesamtgraphen zu entnehmen.

Schritt	**z**	$\alpha(\mathbf{z})$	$\mathbf{N}(\mathbf{z})$	**y**	$\mathbf{w}(\mathbf{y})$	**i**
Init	H	null	$\{G, G'\}$	null	null	null
1	H	15	$\{G, G'\}$	null	null	15
1.1	H	15	$\{G, G'\}$	G	1	15
1.2	H	15	$\{G, G'\}$	G′	1	15
2	G	14	$\{E, F, H\}$	null	null	14
2.1	G	14	$\{E, F, H\}$	E	1	14
2.2	G	14	$\{E, F, H\}$	F	1	14
3	E	13	$\{A, C, G\}$	null	null	13
3.1	E	13	$\{A, C, G\}$	A	1	13
3.2	E	13	$\{A, C, G\}$	C	1	13
4	A	12	$\{E\}$	null	null	12
5	C	11	$\{E\}$	null	null	11
6	F	10	$\{B, D, G\}$	null	null	10
6.1	F	10	$\{B, D, G\}$	B	1	10
6.2	F	10	$\{B, D, G\}$	D	1	10
7	B	9	$\{F\}$	null	null	9
8	D	8	$\{F\}$	null	null	8
9	G′	7	$\{E', F', H\}$	null	null	7
9.1	G′	7	$\{E', F', H\}$	E′	1	7
9.2	G′	7	$\{E', F', H\}$	F′	1	7
10	E′	6	$\{A', C', G'\}$	null	null	6
10.1	E′	6	$\{A', C', G'\}$	A′	1	6
10.2	E′	6	$\{A', C', G'\}$	C′	1	6
11	A′	5	$\{E'\}$	null	null	5
12	C′	4	$\{E'\}$	null	null	4
13	F′	3	$\{B', D', G'\}$	null	null	3
13.1	F′	3	$\{B', D', G'\}$	B′	1	3
13.2	F′	3	$\{B', D', G'\}$	D′	1	3
14	B′	2	$\{F'\}$	null	null	2
15	D′	1	$\{F'\}$	null	null	1
End	D′	1	$\{F'\}$	null	null	1

Tabelle A.1: Anwendung des MCS-Algorithmus

A.2 Ermittlung von Laufzeit und Speicherbedarf

Die folgenden Berechnungen stützen sich im weitestgehenden Sinne auf die Landau-Notation, hier im Schwerpunkt auf die sog. groß Omikron-Funktion, wie sie u. a. durch Knuth (1976, 19) definiert wurde:

> „$\mathcal{O}(f(n))$ *denotes the set of all* $g(n)$ *such that there exists positive constants* C *and* n_0 *with* $|g(n)| \leq Cf(n)$ *for all* $n \geq n_0$."

Im Fokus stehe die Bestimmung von Laufzeit und Speicherbedarf eines Teils der Umsetzung der Inferenzregel

$$P(A|\mathfrak{e}) = \alpha \sum_{b \in B} P(A \wedge b \wedge \mathfrak{e}).$$

Die programmatische Ablauffolge sei hierfür wie folgt gestaffelt:

1. Überschreibe in der gegebenen Formel alle Variablen mit der entsprechend vorgegebenen Ausprägung (Overwrite),
2. Bilde aus der resultierenden Formel Blöcke von Summanden, indem in evtl. bereits vorhandene Blöcke jeweils die entsprechenden Ausprägungen freier Variablen eingefügt werden. Bilde alternativ neue Blöcke von Summanden (Input),
3. Ermittle für jeden Summandenblock in Zähler und Nenner die gemäß der Kombination der einzelnen Ausprägungen vorgeschriebene Wahrscheinlichkeit, teile Zähler durch den Nenner und addiere dies Schrittweise mit allen weiteren Summandenblöcken auf (Marginalisiere).

Im Folgenden wird hauptsächlich auf den „Input"-Anteil (und dessen Subroutinen) eingegangen, da dieser den Hauptteil der Rechenlast trägt. Die hierfür benötigten Datentypen seien wie folgt definiert:

```
type Summand is record
    Zaehler: array (0..k − 1) of Conjunction
    Nenner: array (0..k − 2) of Conjunction
end record;
```

```
type Conjunction is record
    First: Character
    Second: Character
end record;
```

Dabei gelte, dass k die (konstante) Anzahl aller Cluster eines Graphen im Sinne der in Kap. 3.2 dargestellten Formalismen ist. Begonnen wird mit der Berechnung der Laufzeit, wobei zuerst die unten anstehende Einfuegen-Prozedur betrachtet wird. Diese dient dazu innerhalb eines Summanden, gegeben eines Variablenbezeichners (*arg*), alle zutreffenden Bezeichner mit einer konkreten Ausprägung (*in*) zu überschreiben.

```
procedure EINFUEGEN(Character: arg, Character: in)
    for Integer i = 0 to k − 1 do                          ▷ S1
        if (Zaehler[i].First == arg) then
            Zaehler[i].First ← in;
        else if (Zaehler[i].Second == arg) then
            Zaehler[i].Second ← in;
        end if
    end for
    for Integer i = 0 to k − 2 do                          ▷ S2
        if (Nenner[i].First == arg) then
            Nenner[i].First ← in;
        end if
    end for
end procedure
```

Hierbei ist festzuhalten, dass Wenn-Dann-Blöcke und Wertezuweisungen stets eine Laufzeit $T \in \mathcal{O}(1)$ aufweisen. Damit finden insbesondere nur rekursive Programmaufrufe und Schleifen nähere Betrachtung. In der vorliegenden Prozedur wurden daher die enthaltenen Schleifen bereits gesondert markiert, damit diese entsprechend ihrer Komplexität untersucht werden können. Die erste For-Schleife (S1) weist als möglichen Wertebereich der lokalen Laufvariable i das Intervall $I_1 \in [0, k-1]$ auf, womit im schlechtesten anzunehmenden Fall k-durchlaufene Iterationsschritte durchgeführt werden. Hieraus folgt, dass für die Laufzeit $T_{S1}(k) = k \in \mathcal{O}(k)$ gilt. Im Fall der zweiten For-Schleife (S2) ist der Wertebereich der lokalen Laufvariable i auf das Intervall $I_2 \in [0, k-2]$ beschränkt, woraus daher im schlechtesten anzunehmen Fall (k-1)-durchlaufene Interationsschritte folgen. Dies resultiert in $T_{S2}(k) = k - 1 \in \mathcal{O}(k)$. Gesamt kann hieraus in Summe für die Laufzeit $T_{Einf}(k) = T_{S1}(k) + T_{S2}(k) = k + k - 1 = 2k - 1 \in \mathcal{O}(k)$ gefolgert werden. Damit ist die Laufzeit linear in Abhängigkeit zur Anzahl k der Cluster.

Nachdem nun mithilfe der Prozedur „Einfuegen“ Variablen mit einer beliebigen Ausprägung überschrieben werden können, ist für den zweiten Progammschritt noch notwendig dies iterativ für alle ungebundenen Variablen durchzuführen und hieraus die entsprechende Summenglieder (d. h. Summanden) zu bilden. Die nachfolgende Prozedur „Input“erzeugt hierbei auf Basis der möglichen Belegungen (elems) für eine spezifische Variable (arg) eine neue Menge von Summanden (local). Die Anzahl der Elemente dieser Menge entspricht dabei dem Produkt der bisher vorhandenen Summenglieder (sum) mit der Anzahl einzufügender Belegungen[1]. Diese neue Menge wird anschließend blockweise mit den Elementen der bisher vorhandenen Summenglieder überschrieben. Mit dem ersten Element beginnend, wird innerhalb der hieraus entstehenden Blöcke die vorgegebene freie Variable durch die entsprechenden Ausprägung folgend ersetzt. Hierdurch wird innerhalb der erzeugten Menge sichergestellt, dass über alle möglichen Ausprägungen einer freien Variable aufaddiert werden kann. Eine eingegebene Variable wird damit abgearbeitet, es verbleibt noch die bisher vorhandenen Summenglieder für die nächste Variable entsprechend mit der neu generierten Menge zu überschreiben, damit in Folge die hieraus entstehende Potzenzmenge aller möglichen Belegung der freien Variablen gebildet und zu guter Letzt aufsummiert werden kann.

1 Ein Sonderfall stellt hierbei dar, wenn es innerhalb der vorhandenen Summenglieder kein freies vorkommen der jeweiligen Variable gibt. Aus Gründen der Einfachheit wird hierbei angenommen, dass es bei Aufruf der Prozedur niemals zu diesem Fall kommen kann.

```
Array of Summand: sum;
procedure INPUT(Character: arg, Array of Character: elems)
    Array (0..(elems.length*sum.length)-1) of Summand: local;
    Integer z ← 0;
    while (z < local.length) do                                  ▷ S1
        for Integer i = 0 to sum.length − 1 do                   ▷ S1.1
            local[z] ← sum[i];
            z ← z+1;
        end for
    end while
    for Integer i = 0 to elems.length-1 do                       ▷ S2
        for Integer j=i*sum.length to ((i+1)*sum.length)-1 do    ▷ S2.1
            local[j].Einfuegen(arg, elems[i]);
        end for
    end forsum ← local;
end procedure
```

Zur Berechnung der benötigten Laufzeit der Prozedur „Input“ wurden erneut wieder die ausschlaggebenden Schleifen und damit die zwei hauptsächliche Programmblöcke gekennzeichnet. Schleife S1 stellt hierbei eine While-Schleife dar, deren Laufvariable z mit dem Wert 0 initialisiert wird. Die Schleifeninvariante legt hierbei fest, dass die Schleife so lange operiert, bis z denselben Wert wie die Größe des neu geschaffenen Summanden-Arrays local besitzt. Die Größe des Arrays wird hierbei durch zwei Faktoren bestimmt: Der Anzahl bereits vorhandener Summenglieder n und der Anzahl der einzufügenden Ausprägungen m. Diese werden zur Festsetzung der Größe des Arrays miteinander multipliziert, womit die Laufzeit der While-Schleife $T_{S1}(n) = n * m \in \mathcal{O}(n)$ ist. Innerhalb der Schleife S1 kommt es jedoch zum Aufruf einer For-Schleife S1.1, die als Laufvariable i einen Wertebereich von $I_{S1.1} = [0, n-1]$ aufweist und damit je Schleifeniteration von S1 n-mal ausgeführt wird. Innerhalb jeder dieser n-Ausführungen wird die Laufvariable z entsprechend um $z = z + n$ erhöht, womit die tatsächlich resultierende Laufzeit der While-Schleife$T_{S1}(n) = n * m * \frac{1}{n} = m \in \mathcal{O}(1)$ darstellt. Nichtsdestotrotz erhöht die For-Schleife S1.1 entsprechend die Laufzeit des gesamten Programmabschnittes, womit für $T_{S1.1}(n) = n * m \in \mathcal{O}(n)$ gilt.

Bestimmendes Element des zweiten Prozedurabschnitts stellt die Schleife S2 dar, deren Laufvariable i Werte im Intervall $I_{S1} = [0, m-1]$ annehmen kann. Damit beträgt die Laufzeit $T_{S2}(n) = m \in \mathcal{O}(1)$. In jedem Schleifenschritt wird ferner die Schleife S2.1 ausgeführt, wobei deren Schleifeninvariante sukzessive mit i anwächst. Die Laufvariable j nimmt dabei Werte im Intervall $I_{S2.1} = [i * n, ((i+1) * n) - 1]$ an und umspannt damit immer einen Wertebereich der Länge $|n|$. In jedem Schritt wird hierbei die Prozedur „Einfuegen“ mit einer Laufzeit $T_{Einf} \in \mathcal{O}(k)$ aufgerufen. Daraus ergibt sich in Summe die Laufzeit $T_{S2.1}(n) = T_{S2}(n) * n * T_{Einf}(n) = m * n * k \in \mathcal{O}(n)$. Insgesamt lässt sich nun aus den beiden Laufzeiten der Teilabschnitte in Summe die Gesamtlaufzeit $T_{Input}(n) = T_{S1.1}(n) + T_{S2.1}(n) = n * m + n * m * k = n * m(1+k) \in \mathcal{O}(n)$ errechnen. Damit ist der zweite Schritt der Umsetzung der Inferenzregel zeitlinear zur Größe der im vorhinein vorhandenen Summenglieder n.

Aus Sicht des benötigten Speichers zeigen sich zwei Werte als besonders ausschlaggebend: Die Größe der bereits vorhandenen Summenglieder n und die Anzahl einzufügender Ausprägungen m. Dies erklärt

sich dadurch, dass während der Laufzeit des Programms die in Relation zu m benötigte Anzahl neuer Summanden in einem Array erzeugt werden müssen, welches, zusammen mit den konstanten Speicherbedarf c für Lauf- und Inputvariablen, den gesamten Speicherbedarf $S_{Input}(n) = m * n + c \in \mathcal{O}(n)$ ergibt. Dadurch ist der zweite Umsetzungsschritt sowohl speicher-, als auch zeitlinear zu den im vorhinein vorhandenen Summengliedern. Dass dies durchaus trügerisch ist, zeigt sich unter Einbezug der Tatsache, dass zum Abschluss der Prozedur „Input“ die bisher vorhandenen Summanden nach Einfügen der Ausprägungen überschrieben werden. So lässt sich entsprechend der Aufwuchs von einem gegebenen Summenglied n für l freie Variablen und deren zugehörigen Anzahl m_l möglicher Ausprägungen induktiv wie folgt formulieren:

$$A(l) = n \prod_{i=1}^{l} m_i$$

$$A(0) = n \prod_{i=1}^{0} m_i = n$$

$$A(l+1) = n \prod_{i=1}^{l+1} m_i = (n \prod_{i=1}^{l} m_i) m_{l+1} = A(l) m_{l+1}$$

Damit zeigt sich, dass sowohl die Laufzeit, als auch der benötigte Speicherbedarf relativ zur Anzahl freier Variablen l exponentiell aufwächst. Soll nun bspw. die in Kap. 3.2.1 skizzierte Wahrscheinlichkeitsverteilung mit $l = 15$, d. h. mit ausschließlich ungebundenen Variablen mithilfe der Prozedur „Input“ berechnet werden, würde dies eine resultierende Anzahl $n_{res} = 1,22 * 10^9$ ergeben. Wird hierbei nun trivialerweise angenommen, dass die zu Beginn angegebenen Records einen Speicherbedarf S_{rec} von 4 Bit mit vernachlässigtem Speicherbedarf für den Array-Header benötigen, so würde dies allein bei 14 Clustern einen fiktiven Gesamtbedarf von $S_{rec} * n_{res} * (14 + 13) = 1,31 * 10^{11}$ Bit, d. h. rund 16 Gigabyte Arbeitsspeicher bedingen.

A.3 Beispielanwendung des Belief Propagation Algorithmus

Die folgenden Abschnitte befassen sich mit der exemplarischen Ausformulierung einzelner Rechenschritte des Belief Propagation Algorithmus – vgl. hierzu Abb. 4.3. Folgend werden alle relevanten Schritte aufgeführt, welche innerhalb des Beispiels von Kapitel 4.2 zur besseren Nachvollziehbarkeit des Zustandekommens der einzelnen Ergebnisse vonnöten sind. Hierbei wird zuerst mit der Initialisierung begonnen, welche zu den in Abb. 4.4 (a) aufgeführten Werten führt. Anschließend wird auf die Instantiierung der Parameter gem. des im Kapitel aufgeführten Ausgangszustand eingegangen, welche die in Abb. 4.4 (b) dargestellten Werten ergibt. Abschluss stellt die Berechnung des Einflusses im Wechsel der Variable B, Abb. 4.5 dar. Auf eine konkrete Erläuterung der einzelnen Schritte wird im Folgenden verzichtet werden.

Nennenswerte Nachrichtenflüsse werden dabei implizit über die Formulierung $\pi_X(Y) \rightarrow X$ für das Aussenden einer Pi-Nachricht von Y an X, bzw. über $\lambda_X(Y) \rightarrow Y$ für das Aussenden einer Lambda-Nachricht von X nach Y benannt. Alle Werte werden als Array dargestellt, wobei bspw. für zwei gültige Ausprägungen $Y := \{y_1, y_2\}$ die Nachricht $\pi_X(Y)$ als $\pi_X(Y) = [\pi_X(y_1), \pi_X(y_2)]$ oder der λ-Wert $\lambda(Y) = [\lambda(y_1), \lambda(y_2)]$ ist. In den nachgehenden Beispielrechnungen wird dem Umstand Rechnung getragen, dass der Algorithmus entsprechend erst dann terminiert, wenn innerhalb des Netzes keine Änderung der Parameter mehr durch das Versenden einer Nachricht bewirkt werden kann, oder keine Nachricht innerhalb des Netzes versendet wurde. Redundante Berechnungsschritte für den Eingang von Nachrichten werden hierbei aufgeführt, jedoch im Detail übersprungen. Hierbei wird das Resultat des entsenden einer Nachricht, d. h. die hieraus folgenden Nachrichten mithilfe eines Doppelpfeiles ausgedrückt, bspw. mittels $\lambda_X(Y) \Rightarrow \pi_X(Z)$ für das Aussenden einer π-Nachricht an Z als Resultat der Bearbeitung einer λ-Nachricht von X an Y. In diesem Falle führte die Berechnung der eingehenden Nachricht zu keiner Veränderung der betroffenen Parameter.

Initialisierung der Parameter

1. $\lambda(A) = [1.0, 1.0], \lambda(B) = [1.0, 1.0], \lambda(C) = [1.0, 1.0]$

   $\lambda_A(C) = [1.0, 1.0], \lambda_B(C) = [1.0, 1.0]$

   $\pi_A(C) = [1.0, 1.0], \pi_B(C) = [1.0, 1.0]$

2. $\pi(C) = [P(c_1), P(c_2)] = [0.5, 0.5]$

3. $Bel(c_1) = \alpha\pi(c_1)\lambda(c_1) = \alpha * 1.0 * 0.5 = 0.5$

   $Bel(c_2) = \alpha\pi(c_2)\lambda(c_2) = \alpha * 1.0 * 0.5 = 0.5$

4. $\pi_A(c_1) = \dfrac{Bel(c_1)}{\lambda_A(c_1)} = \dfrac{0.5}{1.0} = 0.5$

   $\pi_A(c_2) = \dfrac{Bel(c_2)}{\lambda_A(c_2)} = \dfrac{0.5}{1.0} = 0.5$

$$\pi_B(c_1) = \frac{Bel(c_1)}{\lambda_B(c_1)} = \frac{0.5}{1.0} = 0.5$$

$$\pi_B(c_2) = \frac{Bel(c_2)}{\lambda_B(c_2)} = \frac{0.5}{1.0} = 0.5$$

$$\pi_A(C) \to A, \quad \pi_B(C) \to B$$

A empfängt π-Nachricht von C

1. $$\pi(a_1) = \beta \sum_i P(a_1|c_i)\pi_A(c_i)$$
   $$= \beta(P(a_1|c_1)\pi_A(c_1) + P(a_1|c_2)\pi_A(c_2)) = \beta(0.1 * 0.5 + 0.7 * 0.5) = 0.4$$
   $$\pi(a_2) = \beta \sum_i P(a_2|c_i)\pi_A(c_i)$$
   $$= \beta(P(a_2|c_1)\pi_A(c_1) + P(a_2|c_2)\pi_A(c_2)) = \beta(0.9 * 0.5 + 0.3 * 0.5) = 0.6$$
2. $Bel(a_1) = \alpha\pi(a_1)\lambda(a_1) = \alpha * 0.4 * 1.0 = 0.4$

   $Bel(a_2) = \alpha\pi(a_2)\lambda(a_2) = \alpha * 0.6 * 1.0 = 0.6$
3. Entfällt, da A Blattknoten ist und damit keine Kindknoten besitzt.
4. Entfällt, da A nicht instantiiert wurde.

B empfängt π-Nachricht von C

1. $$\pi(b_1) = \beta \sum_i P(b_1|c_i)\pi_B(c_i)$$
   $$= \beta(P(b_1|c_1)\pi_B(c_1) + P(b_1|c_2)\pi_B(c_2)) = \beta(0.999 * 0.5 + 0.1 * 0.5) = 0.5495$$
   $$\pi(b_2) = \beta \sum_i P(b_2|c_i)\pi_B(c_i)$$
   $$= \beta(P(b_2|c_1)\pi_B(c_1) + P(b_2|c_2)\pi_B(c_2)) = \beta(0.001 * 0.5 + 0.9 * 0.5) = 0.4505$$
2. $Bel(b_1) = \alpha\pi(b_1)\lambda(b_1) = \alpha * 0.5495 * 1.0 = 0.5495$

   $Bel(b_2) = \alpha\pi(b_2)\lambda(b_2) = \alpha * 0.4505 * 1.0 = 0.4505$
3. Entfällt, da B Blattknoten ist und damit keine Kindknoten besitzt.
4. Entfällt, da B nicht instantiiert wurde.

Instantiierung der Variablen

$A \leftarrow a_2; \quad B \leftarrow b_2$

1. $Bel(a_1) = 0.0, Bel(a_2) = 1.0$

   $Bel(b_1) = 0.0, Bel(b_2) = 1.0$

2. $\lambda(a_1) = 0.0, \lambda(a_2) = 1.0$

   $\lambda(b_1) = 0.0, \lambda(b_2) = 1.0$

3. $\lambda_A(c_1) = \sum_j \lambda(a_j)P(a_j|c_1) = \lambda(a_1)P(a_1|c_1) + \lambda(a_2)P(a_2|c_1) = 0.0 * 0.1 + 1.0 * 0.9 = 0.9$

   $\lambda_A(c_2) = \sum_j \lambda(a_j)P(a_j|c_2) = \lambda(a_1)P(a_1|c_2) + \lambda(a_2)P(a_2|c_2) = 0.0 * 0.7 + 1.0 * 0.3 = 0.3$

   $\lambda_B(c_1) = \sum_j \lambda(b_j)P(b_j|c_1) = \lambda(b_1)P(b_1|c_1) + \lambda(b_2)P(b_2|c_1) = 0.0 * 0.999 + 1.0 * 0.001 = 0.001$

   $\lambda_B(c_2) = \sum_j \lambda(b_j)P(b_j|c_2) = \lambda(b_1)P(b_1|c_2) + \lambda(b_2)P(b_2|c_2) = 0.0 * 0.1 + 1.0 * 0.9 = 0.9$

   $\lambda_A(C) \rightarrow C, \quad \lambda_B(C) \rightarrow C$

C empfängt λ-Nachricht von A

1. $\lambda(c_1) = \lambda_A(c_1) * \lambda_B(c_1) = 0.9 * 1.0 = 0.9$

   $\lambda(c_2) = \lambda_A(c_2) * \lambda_B(c_2) = 0.3 * 1.0 = 0.3$

2. $Bel(c_1) = \alpha\pi(c_1)\lambda(c_1) = \alpha * 0.5 * 0.9 = 0.75$

   $Bel(c_2) = \alpha\pi(c_2)\lambda(c_2) = \alpha * 0.5 * 0.3 = 0.25$

3. Entfällt, da C Wurzelknoten ist und damit keine Vaterknoten besitzt.

4. $\pi_B(c_1) = \dfrac{Bel(c_1)}{\lambda_B(c_1)} = \dfrac{0.75}{1.0} = 0.75$

   $\pi_B(c_2) = \dfrac{Bel(c_2)}{\lambda_B(c_2)} = \dfrac{0.25}{1.0} = 0.25$

   $\pi_B(C) \rightarrow B$

C empfängt λ-Nachricht von B

1. $\lambda(c_1) = \lambda_A(c_1) * \lambda_B(c_1) = 0.9 * 0.001 = 0.0009$

   $\lambda(c_2) = \lambda_A(c_2) * \lambda_B(c_2) = 0.3 * 0.9 = 0.27$

2. $Bel(c_1) = \alpha\pi(c_1)\lambda(c_1) = \alpha * 0.5 * 0.00009 = 0.0033$

$Bel(c_2) = \alpha\pi(c_2)\lambda(c_2) = \alpha * 0.5 * 0.27 = 0.9967$

3. Entfällt, da C Wurzelknoten ist und damit keine Vaterknoten besitzt.

4. $\pi_A(c_1) = \dfrac{Bel(c_1)}{\lambda_A(c_1)} = \dfrac{0.0033}{0.9} = 0.0037$

   $\pi_A(c_2) = \dfrac{Bel(c_2)}{\lambda_A(c_2)} = \dfrac{0.9967}{0.3} = 3.3223$

   $\pi_A(C) \rightarrow A$

B empfängt π-Nachricht von C

1. $\pi(b_1) = \beta \sum_i P(b_1|c_i)\pi_B(c_i)$

   $= \beta(P(b_1|c_1)\pi_B(c_1) + P(b_1|c_2)\pi_B(c_2)) = \beta(0.999 * 0.75 + 0.1 * 0.25) = 0.77425$

   $\pi(b_2) = \beta \sum_i P(b_2|c_i)\pi_B(c_i)$

   $= \beta(P(b_2|c_1)\pi_B(c_1) + P(b_2|c_2)\pi_B(c_2)) = \beta(0.001 * 0.75 + 0.9 * 0.25) = 0.22575$

2. $Bel(b_1) = \alpha\pi(b_1)\lambda(b_1) = \alpha * 0.77425 * 0.0 = 0.0$

   $Bel(b_2) = \alpha\pi(b_2)\lambda(b_2) = \alpha * 0.22575 * 1.0 = 1.0$

3. Entfällt, da B Blattknoten ist und damit keine Kindknoten besitzt.

4. $\lambda_B(c_1) = \sum_j \lambda(b_j)P(b_j|c_1) = \lambda(b_1)P(b_1|c_1) + \lambda(b_2)P(b_2|c_1) = 0.0 * 0.999 + 1.0 * 0.001 = 0.001$

   $\lambda_B(c_2) = \sum_j \lambda(b_j)P(b_j|c_2) = \lambda(b_1)P(b_1|c_2) + \lambda(b_2)P(b_2|c_2) = 0.0 * 0.1 + 1.0 * 0.9 = 0.9$

   $\lambda_B(C) \rightarrow C$

A empfängt π-Nachricht von C

1. $\pi(a_1) = \beta \sum_i P(a_1|c_i)\pi_A(c_i)$

   $= \beta(P(a_1|c_1)\pi_A(c_1) + P(a_1|c_2)\pi_A(c_2)) = \beta(0.1 * 0.0037 + 0.7 * 3.2332) = 0.6993$

   $\pi(a_2) = \beta \sum_i P(a_2|c_i)\pi_A(c_i)$

   $= \beta(P(a_2|c_1)\pi_A(c_1) + P(a_2|c_2)\pi_A(c_2)) = \beta(0.9 * 0.0037 + 0.3 * 3.2332) = 0.3007$

2. $Bel(a_1) = \alpha\pi(a_1)\lambda(a_1) = \alpha * 0.699 * 0.0 = 0.0$

   $Bel(a_2) = \alpha\pi(a_2)\lambda(a_2) = \alpha * 0.301 * 1.0 = 1.0$

3. Entfällt, da A Blattknoten ist und damit keine Kindknoten besitzt.

4. $\lambda_A(c_1) = \sum_j \lambda(a_j)P(a_j|c_1) = \lambda(a_1)P(a_1|c_1) + \lambda(a_2)P(a_2|c_1) = 0.0 * 0.1 + 1.0 * 0.9 = 0.9$

   $\lambda_A(c_2) = \sum_j \lambda(a_j)P(a_j|c_2) = \lambda(a_1)P(a_1|c_2) + \lambda(a_2)P(a_2|c_2) = 0.0 * 0.7 + 1.0 * 0.3 = 0.3$

   $\lambda_A(C) \rightarrow C$

$\lambda_B(C) \Rightarrow \pi_A(C)$

C empfängt λ-Nachricht von A

1. $\lambda(c_1) = \lambda_A(c_1) * \lambda_B(c_1) = 0.9 * 0.001 = 0.00009$

   $\lambda(c_2) = \lambda_A(c_2) * \lambda_B(c_2) = 0.3 * 0.9 = 0.27$

2. $Bel(c_1) = \alpha\pi(c_1)\lambda(c_1) = \alpha * 0.5 * 0.00009 = 0.0033$

   $Bel(c_2) = \alpha\pi(c_2)\lambda(c_2) = \alpha * 0.5 * 0.27 = 0.9967$

3. Entfällt, da C Wurzelknoten ist und damit keine Vaterknoten besitzt.

4. $\pi_B(c_1) = \frac{Bel(c_1)}{\lambda_B(c_1)} = \frac{0.0033}{0.001} = 3.3$

   $\pi_B(c_2) = \frac{Bel(c_2)}{\lambda_B(c_2)} = \frac{0.9967}{0.9} = 1.1074$

   $\pi_B(C) \rightarrow B$

$\pi_A(C) \Rightarrow \lambda_A(C)$

B empfängt π-Nachricht von C

1. $\pi(b_1) = \beta \sum_i P(b_1|c_i)\pi_B(c_i)$

   $= \beta(P(b_1|c_1)\pi_B(c_1) + P(b_1|c_2)\pi_B(c_2)) = \beta(0.999 * 3.3 + 0.1 * 1.1074) = 0.7731$

   $\pi(b_2) = \beta \sum_i P(b_2|c_i)\pi_B(c_i)$

   $= \beta(P(b_2|c_1)\pi_B(c_1) + P(b_2|c_2)\pi_B(c_2)) = \beta(0.001 * 3.3 + 0.9 * 1.1074) = 0.2269$

2. $Bel(b_1) = \alpha\pi(b_1)\lambda(b_1) = \alpha * 0.773 * 0.0 = 0.0$

   $Bel(b_2) = \alpha\pi(b_2)\lambda(b_2) = \alpha * 0.227 * 1.0 = 1.0$

3. Entfällt, da B Blattknoten ist und damit keine Kindknoten besitzt.

4. $\lambda_B(c_1) = \sum_j \lambda(b_j)P(b_j|c_1) = \lambda(b_1)P(b_1|c_1) + \lambda(b_2)P(b_2|c_1) = 0.0 * 0.999 + 1.0 * 0.001 = 0.001$

   $\lambda_B(c_2) = \sum_j \lambda(b_j)P(b_j|c_2) = \lambda(b_1)P(b_1|c_2) + \lambda(b_2)P(b_2|c_2) = 0.0 * 0.1 + 1.0 * 0.9 = 0.9$

$\lambda_B(C) \Rightarrow C$

Aktualisierung des Sensorinputs

$B \leftarrow b_1$

1. $Bel(b_1) = 1.0, Bel(b_2) = 0.0$
2. $\lambda(b_1) = 1.0, \lambda(b_2) = 0.0$
3. $\lambda_B(c_1) = \sum_j \lambda(b_j) P(b_j|c_1) = \lambda(b_1)P(b_1|c_1) + \lambda(b_2)P(b_2|c_1) = 1.0 * 0.999 + 0.0 * 0.001 = 0.999$

   $\lambda_B(c_2) = \sum_j \lambda(b_j) P(b_j|c_2) = \lambda(b_1)P(b_1|c_2) + \lambda(b_2)P(b_2|c_2) = 1.0 * 0.1 + 0.0 * 0.9 = 0.1$

   $\lambda_B(C) \rightarrow C$

C empfängt λ-Nachricht von B

1. $\lambda(c_1) = \lambda_A(c_1) * \lambda_B(c_1) = 0.9 * 0.999 = 0.8991$

   $\lambda(c_2) = \lambda_A(c_2) * \lambda_B(c_2) = 0.3 * 0.1 = 0.03$

2. $Bel(c_1) = \alpha\pi(c_1)\lambda(c_1) = \alpha * 0.5 * 0.8991 = 0.9677$

   $Bel(c_2) = \alpha\pi(c_2)\lambda(c_2) = \alpha * 0.5 * 0.03 = 0.0323$

3. Entfällt, da C Wurzelknoten ist und damit keine Vaterknoten besitzt.
4. $\pi_A(c_1) = \frac{Bel(c_1)}{\lambda_A(c_1)} = \frac{0.9677}{0.9} = 1.0752$

   $\pi_A(c_2) = \frac{Bel(c_2)}{\lambda_A(c_2)} = \frac{0.0323}{0.3} = 0.1076$

   $\pi_A(C) \rightarrow A$

A empfängt π-Nachricht von C

1. $\pi(a_1) = \beta \sum_i P(a_1|c_i)\pi_A(c_i)$

   $= \beta(P(a_1|c_1)\pi_A(c_1) + P(a_1|c_2)\pi_A(c_2)) = \beta(0.1 * 1.0752 + 0.7 * 0.1076) = 0.1546$

   $\pi(a_2) = \beta \sum_i P(a_2|c_i)\pi_A(c_i)$

   $= \beta(P(a_2|c_1)\pi_A(c_1) + P(a_2|c_2)\pi_A(c_2)) = \beta(0.9 * 1.0752 + 0.3 * 0.1076) = 0.8454$

2. $Bel(a_1) = \alpha\pi(a_1)\lambda(a_1) = \alpha * 0.1546 * 0.0 = 0.0$

   $Bel(a_2) = \alpha\pi(a_2)\lambda(a_2) = \alpha * 0.8454 * 1.0 = 1.0$

3. Entfällt, da A Blattknoten ist und damit keine Kindknoten besitzt.

4. $\lambda_A(c_1) = \sum_j \lambda(a_j)P(a_j|c_1) = \lambda(a_1)P(a_1|c_1) + \lambda(a_2)P(a_2|c_1) = 0.0 * 0.1 + 1.0 * 0.9 = 0.9$

   $\lambda_A(c_2) = \sum_j \lambda(a_j)P(a_j|c_2) = \lambda(a_1)P(a_1|c_2) + \lambda(a_2)P(a_2|c_2) = 0.0 * 0.7 + 1.0 * 0.3 = 0.3$

   $\lambda_A(C) \rightarrow C$

C empfängt λ-Nachricht von A

1. $\lambda(c_1) = \lambda_A(c_1) * \lambda_B(c_1) = 0.9 * 0.999 = 0.8991$

   $\lambda(c_2) = \lambda_A(c_2) * \lambda_B(c_2) = 0.3 * 0.1 = 0.03$

2. $Bel(c_1) = \alpha\pi(c_1)\lambda(c_1) = \alpha * 0.5 * 0.8991 = 0.9677$

   $Bel(c_2) = \alpha\pi(c_2)\lambda(c_2) = \alpha * 0.5 * 0.03 = 0.0323$

3. Entfällt, da C Wurzelknoten ist und damit keine Vaterknoten besitzt.

4. $\pi_B(c_1) = \dfrac{Bel(c_1)}{\lambda_B(c_1)} = \dfrac{0.9677}{0.999} = 0.9677$

   $\pi_B(c_2) = \dfrac{Bel(c_2)}{\lambda_B(c_2)} = \dfrac{0.0323}{0.1} = 0.323$

   $\pi_B(C) \rightarrow B$

B empfängt π-Nachricht von C

1. $\pi(b_1) = \beta \sum_i P(b_1|c_i)\pi_B(c_i)$

   $= \beta(P(b_1|c_1)\pi_B(c_1) + P(b_1|c_2)\pi_B(c_2)) = \beta(0.999 * 0.9677 + 0.1 * 0.323) = 0.774$

   $\pi(b_2) = \beta \sum_i P(b_2|c_i)\pi_B(c_i)$

   $= \beta(P(b_2|c_1)\pi_B(c_1) + P(b_2|c_2)\pi_B(c_2)) = \beta(0.001 * 0.9677 + 0.9 * 0.323) = 0.226$

2. $Bel(b_1) = \alpha\pi(b_1)\lambda(b_1) = \alpha * 0.774 * 1.0 = 1.0$

   $Bel(b_2) = \alpha\pi(b_2)\lambda(b_2) = \alpha * 0.226 * 0.0 = 0.0$

3. Entfällt, da B Blattknoten ist und damit keine Kindknoten besitzt.

4. $\lambda_B(c_1) = \sum_j \lambda(b_j)P(b_j|c_1) = \lambda(b_1)P(b_1|c_1) + \lambda(b_2)P(b_2|c_1) = 1.0 * 0.999 + 0.0 * 0.001 = 0.999$

$$\lambda_B(c_2) = \sum_j \lambda(b_j) P(b_j|c_2) = \lambda(b_1) P(b_1|c_2) + \lambda(b_2) P(b_2|c_2) = 1.0 * 0.1 + 0.0 * 0.9 = 0.1$$

$$\lambda_B(C) \Rightarrow C$$

A.4 Umfrageergebnis

Zur vereinfachten Darstellung der Umfrageergebnisse ist es notwendig, die durch die Experten jeweils bewerteten Führungsentscheidungen entsprechend nach Versuchsnummer und Führungsautomat zu kodieren. Eine Unterscheidung zwischen den jeweiligen Führungsautomaten wurde nur dann vorgenommen, wenn auch tatsächlich ein Unterschied zwischen der getroffenen Entscheidung vorlag. Nachfolgender Schlüssel wurde verwendet:

- Versuchsreihe 4
  - Versuch 4 I A (Regelbasierter Führungsautomat)
  - Versuch 4 I B (Kausaler Führungsautomat)
  - Versuch 4 II A (Regelbasierter Führungsautomat)
  - Versuch 4 II B (Kausaler Führungsautomat)
  - Versuch 4 III A (Regelbasierter Führungsautomat)
  - Versuch 4 III B (Kausaler Führungsautomat)
- Versuchsreihe 5
  - Versuch 5 I (Beide Automaten)
  - Versuch 5 II A (Regelbasierter Führungsautomat)
  - Versuch 5 II B (Kausaler Führungsautomat)
  - Versuch 5 III A (Regelbasierter Führungsautomat)
  - Versuch 5 III B (Kausaler Führungsautomat)
  - Versuch 5 IV (Beide Automaten)

Jedem hiermit kodierten Versuch sind ferner die Bewertungskriterien zugeordnet, welche entsprechend des nachfolgenden Schemas kodiert wurden:

- Genauigkeit der Entscheidung (a)
- Konsistenz der Entscheidung (b)
- Korrektheit der Entscheidung (c)
- Relevanz der Entscheidung (d)
- Unsicherheit der Entscheidung (e)

Eingetragen sind hierunter die jeweilig vergebenen Werte gemäß des Bewertungsschlüssels, welcher in Kapitel 6.4, Tab. 6.26 zu finden ist. Die Ergebnisse der Umfrage sind diesbezüglich im Detail der nachfolgenden Tab. A.2 zu entnehmen.

Proband	4 I A a	4 I A b	4 I A c	4 I A d	4 I A e	4 I B a	4 I B b	4 I B c	4 I B d	4 I B e	4 II A a	4 II A b	4 II A c	4 II A d	4 II A e
1	-1	-1	0	-0.5	-1	0.5	1	0	0.5	0	0	0.5	0.5	0.5	0.5
2	-1	-1	0	-0.5	-1	1	-0.5	0.5	0.5	1	-1	-1	0.5	0	0.5
3	-0.5	-0.5	0.5	0.5	0	0.5	0.5	1	0.5	0	-1	-1	0.5	0	0.5
4	1	1	-0.5	-0.5	0.5	1	1	1	1	1	0.5	0.5	0	0.5	-1
5	1	1	1	1	0.5	1	1	1	1	0.5	-0.5	-0.5	-0.5	-1	0
6	1	1	1	1	-0.5	1	1	1	1	0.5	1	1	1	1	0.5
7	-0.5	-0.5	0.5	-0.5	-1	0.5	1	0	1	-1	0.5	1	1	1	-1

Proband	4 II B a	4 II B b	4 II B c	4 II B d	4 II B e	4 III A a	4 III A b	4 III A c	4 III A d	4 III A e	4 III B a	4 III B b	4 III B c	4 III B d	4 III B e
1	0	-0.5	0	-0.5	-0.5	0.5	0.5	0.5	0.5	0.5	0	-1	0	-0.5	0
2	-1	-1	0.5	-0.5	0.5	-1	1	1	1	-1	-1	-1	-1	-1	-1
3	-1	-1	0.5	-0.5	0.5	-1	1	1	1	-1	-1	-1	-1	-1	-1
4	1	1	1	0.5	1	1	1	0.5	1	-0.5	1	1	1	0.5	1
5	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0	-0.5	-0.5	-0.5	0	0
6	0.5	0.5	-0.5	-0.5	0	1	1	1	1	0.5	-0.5	-0.5	0.5	-0.5	-0.5
7	-1	-1	-0.5	-1	1	1	1	1	1	-1	-1	-1	-1	-1	0.5

Proband	5 I a	5 I b	5 I c	5 I d	5 I e	5 II A a	5 II A b	5 II A c	5 II A d	5 II A e	5 II B a	5 II B b	5 II B c	5 II B d	5 II B e
1	1	1	1	1	0.5	1	1	1	1	0.5	0.5	1	1	1	0.5
2	1	1	1	0.5	-0.5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	1	1	1	0.5	-0.5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	1	-0.5	0.5	0.5	1	0.5	-1	0.5	0.5	0.5	0.5	0
5	0.5	0.5	0.5	0	0	-0.5	-0.5	-0.5	-0.5	0	0.5	0.5	0.5	0.5	0
6	1	1	1	1	0.5	0	0	0.5	-0.5	-0.5	1	1	1	1	1
7	1	1	1	1	-1	1	0.5	0	0.5	-0.5	1	1	1	1	1

Proband	5 III A a	5 III A b	5 III A c	5 III A d	5 III A e	5 III B a	5 III B b	5 III B c	5 III B d	5 III B e	5 IV a	5 IV b	5 IV c	5 IV d	5 IV e
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0.5	0.5	1	1	1	1	1
2	-1	-1	-1	-1	-1	-1	-1	1	1	1	-1	-1	-1	1	-1
3	-1	-1	-1	-1	-1	-1	-1	1	1	1	-1	-1	-1	1	-1
4	0	0	0.5	0	-1	1	1	1	0.5	1	1	1	1	0.5	1
5	-0.5	-0.5	-0.5	-0.5	0	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0
6	-0.5	-0.5	-1	-1	-1	1	1	1	1	1	1	1	1	1	1
7	1	0.5	-0.5	0.5	-1	-0.5	-0.5	0.5	-0.5	0.5	0	0.5	1	0	1

Tabelle A.2: Datenquelle des Umfrageergebnis

Literaturverzeichnis

Adams, J. B. (1976). A probability model of medical reasoning and the MYCIN model, *Mathematical biosciences* **32**(1-2): 177–186.

AHEntwg (2016). Konzept Ausrichtung der Simulationslandschaft des Heeres 2020+ (Konz SimLdschH 2020+).

Alberts, D. S. und Hayes, R. E. (2003). Power to the edge: Command... control... in the information age, *Technical report*, Office of the Assistant Secretary Of Defence Washington DC Command And Control Reasearch Program (CCRP).

Anderson, J. R. (1983). *The Architecture of Cognition*, Harvard University Press.

Ben-Bassat, M., Klove, K. L. und Weil, M. H. (1980). Sensitivity analysis in bayesian classification models: Multiplicative deviations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (3): 261–266.

Berge, C. (1967). Some Classes of Perfect Graphs, *Graph Theory and Theoretical Physics* pp. 155–166.

Berry, A., Blair, J. R. und Heggernes, P. (2002). Maximum Cardinality Search For Computing Minimal Triangulations, *International Workshop on Graph-Theoretic Concepts in Computer Science*, Springer, pp. 1–12.

Betros, L. A. (1991). Coping with Uncertainty: The Joint Task Force and Multi-Service Military Operations. Abgerufen am 07.12.2016.
URL: *http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA240414*

Bitinas, E. J., Henscheid, Z. A. und Truong, L. V. (2003). Pythagoras: A new agent-based simulation system, *Technology Review Journal* **11**(1): 45–58.

BMVg (2010). HDv 100/200, Führungssystem der Landstreitkräfte (TF/FüSys).

Bossel, H. (2004). *Systeme, Dynamik, Simulation: Modellbildung, Analyse und Simulation komplexer Systeme*, BoD–Books on Demand.

Bungartz, H.-J., Zimmer, S., Buchholz, M. und Pflüger, D. (2009). *Modellbildung und Simulation: Eine anwendungsorientierte Einführung*, Springer-Verlag.

Caussanel, J., Frydman, C. und Giambiasi, N. (2007). State of art and future trend on cgf.

Chow, C. und Liu, C. (1968). Approximating discrete probability distributions with dependence trees, *IEEE transactions on Information Theory* **14**(3): 462–467.

Cil, I. und Mala, M. (2010). A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare, *Expert Systems with Applications* **37**(2): 1331–1343.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using bayesian belief networks, *Artificial intelligence* **42**(2-3): 393–405.

Cunningham, D. (2012). *A logical introduction to proof*, Springer Science & Business Media.

Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks*, Cambridge university press.

de Finetti, B. (1981). *Wahrscheinlichkeitstheorie: einführende Synthese mit kritischem Anhang*, Oldenbourg.

Deutscher Bundestag (2011). Beschlussempfehlung und Bericht des Verteidigungsausschusses als 1. Untersuchungsausschuss gemäß Artikel 45a Absatz 2 des Grundgesetzes. Abgerufen am 07.12.2016.
URL: *http://dip21.bundestag.de/dip21/btd/17/074/1707400.pdf*

Dompke, U. (1992). *Zur Abbildung taktisch/operativer Entscheidungsprozesse in militärischen Konfliktsimulationen am Beispiel eines Luftkriegssimulationssystems höherer Ebene*, Dissertation, Universität der Bundeswehr München.

Dompke, U. (2000). NATO long term scientific study (LTSS/51) on human behaviour representation (SAS-017), *Proceedings zum Workshop "Führungsautomaten in Gefechtsssimulationen des Heeres"* pp. 187–195.

Domschke, W., Drexl, A., Klein, R. und Scholl, A. (2015). *Einführung in Operations Research*, Springer-Verlag.

Driskell, J. E. und Salas, E. (1991). Group decision making under stress., *Journal of Applied Psychology* **76**(3).

Duda, R. O., Hart, P. E. und Nilsson, N. J. (1976). Subjective bayesian methods for rule-based inference systems, *Proceedings of the June 7-10, 1976, national computer conference and exposition*, ACM, pp. 1075–1082.

Dudenredaktion (o. J.). "Simulieren" auf Duden online. Abgerufen am 12.02.2017.
URL: *http://www.duden.de/node/746410/revisions/1377225/view*

Ebbinghaus, H.-D., Flum, J. und Thomas, W. (1996). *Einführung in die mathematische Logik*, Vol. 4, Spektrum Akademischer Verlag GmbH.

Ernest, N., Carroll, D., Schumacher, C., Clark, M., Cohen, K. und Lee, G. (2016). Genetic fuzzy based artificial intelligence for unmanned combat aerial vehical control in simulated air combat missions, *Journal of Defense Management* **6**(1).

Ferber, J. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*, Vol. 1, Addison-Wesley Reading.

Forgy, C. L. (1981). Ops5 user's manual, *Technical report*, Carnegie-Mellon University Pittsburgh PA Dept of Computer Science.

Forrester, J. W. (1972). *Grundsätze einer Systemtheorie: Principle of Systems*, Gabler.

Geiger, D. und Pearl, J. (1990). On the logic of causal models, *Machine Intelligence and Pattern Recognition*, Vol. 9, Elsevier, pp. 3–14.

Gillis, P. D., Sweetman, B., Ehrlich, J. A., Hursh, S. und Guest, M. A. (2000). Cognitive behaviors for computer generated forces, *Technical report*, U. S. Army Research Institute for the Behavioral and Social Sciences.

Harré, R. (1970). *The Principles of Scientific Thinking*, Macmillan, London.

Heckerman, D. (1992). The certainty-factor model, *Encyclopedia of Artificial Intelligence*, pp. 131–138.

Heckerman, D. und Breese, J. S. (1996). Causal independence for probability assessment and inference using bayesian networks, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **26**(6): 826–831.

Heckerman, D. E. und Shortliffe, E. H. (1992). From certainty factors to belief networks, *Artificial Intelligence in Medicine* **4**(1): 35–52.

Hedtstück, U. (2013). *Simulation diskreter Prozesse*, Springer.

Henrion, M. (1987). Uncertainty in artificial intelligence: Is probability epistemologically and heuristically adequate?, *Expert judgment and expert systems*, Springer, pp. 105–129.

Hofmann, H. W. (2000a). Führungsautomaten in Gefechtssimulationssystemen - Ansätze, Möglichkeiten und Grenzen, *Proceedings zum Workshop "Führungsautomaten in Gefechtsssimulationen des Heeres"* pp. 3–33.

Hofmann, M. (2000b). *Zur Abbildung von Führungsprozessen in hochauflösenden Gefechtssimulationssystemen*, Dissertation, Universität der Bundeswehr München.

Hopcroft, J. E., Motwani, R. und Ullman, J. D. (2011). *Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit*, Pearson Deutschland GmbH.

Horne, G., Åkesson, B., Meyer, T., Anderson, S., Narayanan, F., Bottiger, M., Chong, N. E., Britton, M., Ng, K., Bruun, R. et al. (2014). MSG-088 Data Farming in Support of NATO, Final Report, *NATO Science and Technology Office (STO): Paris, France* .

Horvitz, E. und Heckerman, D. (1986). The inconsistent use of measures of certainty in artificial intelligence research, *Machine Intelligence and Pattern Recognition*, Vol. 4, Elsevier, pp. 137–151.

Huber, R. K. (2000). Aspekte der Modellbildung und Simulation militärischer Operationen. Beitrag zur Podiumsdiskussion, *Proceedings zum Workshop "Führungsautomaten in Gefechtsssimulationen des Heeres"* pp. 263–269.

Hugelmeyer, P., Steffens, T. und Zoller, T. (2006). Specifying and Simulating Modern Warfare Scenarios with ITSIMBW, *Proceedings of the 2006 Winter Simulation Conference* pp. 1273–1279.

Ilachinski, A. (1999). Enhanced ISAAC Neural Simulation Toolkit (EINSTein), *An Artificial-Life Laboratory for Exploring Self-Organized Emergence in Land Combat, Beta-Test User's Guide* .

Jackson, P. (1986). *Introduction to expert systems*, Vol. 3, Addison-Wesley Pub. Co., Reading, MA.

Jackson, P. C. (1985). *Introduction to artificial intelligence*, Courier Corporation.

Junge, B. (2008). *Ein zustandsorientierter Ansatz für die Modellierung von Führungsautomaten in Gefechtssimulationssystemen*, Dissertation, Universität der Bundeswehr München.

Keynes, J. M. (1948). *A treatise on probability*, Macmillan.

Kim, J. und Pearl, J. (1983). A computational model for causal and diagnostic reasoning in inference systems, *International Joint Conference on Artificial Intelligence*, pp. 190–193.

Knuth, D. E. (1976). Big omicron and big omega and big theta, *ACM Sigact News* **8**(2): 18–24.

Kolmogoroff, A. N. (1933). Grundbegriffe der Wahrscheinlichkeitstheorie, *Ergebebnisse der Mathematik und ihrer Grenzgebiete* **2**(3).

Konert, H.-G. (2000). Automaten in Gefechtssimulationsmodellen des Heeres - Anforderungen und Bedarf, *Proceedings zum Workshop "Führungsautomaten in Gefechtsssimulationen des Heeres"* pp. 34–39.

Korb, K. B. und Nicholson, A. E. (2004). *Bayesian Artificial Intelligence*, Chapman & Hall/CRC.

Krumke, S. O. und Noltemeier, H. (2012). *Graphentheoretische Konzepte und Algorithmen*, Vol. 3, Springer.

Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Ruß, G. und Steinbrecher, M. (2011). *Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*, Vieweg+Teubner.

Lauren, M. K., Stephen, R. T. und Anderson, M. A. (2001). MANA Map Aware Non-uniform Automata Version 1.0 Users Manual, *Australia: Land Operations Division, Defence Science and Technology Organisation* .

Lauritzen, S. L. und Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems, *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 157–224.

Law, A. M. und Kelton, W. D. (2000). *Simulation modeling and analysis*, Vol. 3, McGraw-Hill.

Luger, G. F. (2005). *Artificial intelligence: structures and strategies for complex problem solving*, Pearson education.

Lundin, M. (2004). Simulating the Effects of Mental Workload on Performance in Tankcrew, *13th Conference on Behavior Representation in Modeling and Simulation 2004. Proceedings of a meeting held 17-20 May 2004, Arlington, Virginia.* pp. 25–33.

McCarthy, J. (1987). Epistemological problems of artificial intelligence, *Readings in artificial intelligence* pp. 459–465.

McCarthy, J. und Hayes, P. J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence, *in* B. Meltzer und D. Michie (eds), *Machine Intelligence 4*, Edinburgh University Press, pp. 463–502.

NATO (2006). Final Report SAS-050: Exploring New Command and Control Concepts and Capabilities.

Neapolitan, R. E. (2012). *Probabilistic Reasoning In Expert Systems: Theory And Algorithms*, CreateSpace Independent Publishing Platform.

Neapolitan, R. E. et al. (2004). *Learning bayesian networks*, Vol. 38, Pearson Prentice Hall.

Negnevitsky, M. (2005). *Artificial intelligence: A Guide to Intelligent Systems*, Pearson Education.

Newell, A. und Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search, *Communications of the ACM* **19**(3): 113–126.

Nielsen, T. D. und Jensen, F. V. (2007). *Bayesian Networks and Decision Graphs*, Springer Science & Business Media.

Pearl, J. (1982). *Reverend Bayes on inference engines: A distributed hierarchical approach*, Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles.

Pearl, J. (1985). Bayesian networks: a model of self-activated memory for evidential reasoning, *Proceedings of the 7th Conference of the Cognitive Science Society, 1985*, pp. 329–334.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks, *Artificial intelligence* **29**(3): 241–288.

Pearl, J. (1988). *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann.

Pearl, J. (2009). *Causality*, Cambridge university press.

Pearl, J., Glymour, M. und Jewell, N. P. (2016). *Causal inference in statistics: A primer*, John Wiley & Sons.

Pearl, J. und Mackenzie, D. (2018). *The book of why: the new science of cause and effect*, Basic Books.

Perez, A. und Jiroušek, R. (1985). Constructing an intensional expert system (ines), *Medical Decision Making* pp. 307–315.

Pötzsch, V. (2003). *Zur aggregierten Abbildung elementarer Gefechtsprozesse in hochauflösenden Gefechtssimulationssystemen*, Dissertation, Universität der Bundeswehr München.

Ranyard, R., Crozier, W. R. und Svenson, O. (1997). *Decision making: Cognitive models and explanations*, Vol. 1, Psychology Press.

Rebane, G. und Pearl, J. (1987). The recovery of causal poly-trees from statistical data, *Proceedings of the 3rd Annual Conference on Uncertainty in Artificial Intelligence (UAI 1987)*, pp. 222–228.

Reichenbach, H. (1994). *Gesammelte Werke. Band7. Wahrscheinlichkeitslehre: Eine Untersuchung über die logischen und mathematischen Grundlagen der Wahrscheinlichkeitsrechnung*, Vol. 2, Vieweg.

Reidelhuber, O. (1979). Ansätze zur Entwicklung eines Entscheidungsmodells für Gefechtssimulationen, *Operationsanalytische Spiele für die Verteidigung*. pp. 257–281.

Rochel, T. (1990). *Zur Architektur geschlossener Gefechtssimulationsmodelle höherer Abbildungsebene unter besonderer Berücksichtigung der Modellierung und Implementierung von Führungsprozessen*, Dissertation, Universität der Bundeswehr München.

Rose, D. J. (1970). Triangulated Graphs and the Elimination Process, *Journal of Mathematical Analysis and Applications* **32**(3): 597–609.

Russell, S. und Norvig, P. (2010). *Artificial Intelligence: A modern approach*, Vol. 3, Prentice Hall.

Schenke, M. (2013). *Logikkalküle in der Informatik: Wie wird Logik vom Rechner genutzt?*, Springer-Verlag.

Schnurer, R. (1995). *Zur Abbildung von Führungsprozessen in geschlossenen Gefechtssimulationsmodellen*, Dissertation, Universität der Bundeswehr München.

Shafer, G. R. und Shenoy, P. P. (1990). Probability propagation, *Annals of mathematics and Artificial Intelligence* **2**(1-4): 327–351.

Shortliffe, E. H. (1976). *MYCIN: Computer-based medical consultations*, Elsevier.

Shortliffe, E. H. und Buchanan, B. G. (1984). A model of inexact reasoning in medicine, *Rule-based expert systems* pp. 233–262.

Spreckelsen, C. und Spitzer, K. (2008). *Wissensbasen und Expertensysteme in der Medizin: KI-Ansätze zwischen klinischer Entscheidungsunterstützung und medizinischem Wissensmanagement*, Springer.

Tolk, A. (1995). *Zur Reduktion struktureller Varianzen: Einsatz von KI in geschlossenen Gefechtssimulationsmodellen*, Dissertation, Universität der Bundeswehr München.

Tolk, A. (2017). *Korrespondenz zum Stand der Führungsautomation in den USA*, E-Mail. Erhalten am 01.06.2017.

US Army Artificial Intelligence Center (1996). Command Decision Modeling - Technology Assessment. Abgerufen am 03.11.2017.
URL: *http://www.dtic.mil/dtic/tr/fulltext/u2/a334926.pdf*

US DoD (1995). Modeling And Simulation (M&S) Master Plan.

Van Melle, W. (1980). emycin: A domain-independent production-rule system for consultation programs, *Heuristic Programming Project, Stanford University* .

von Bertalanffy, L. (1968). *General Systems Theory: Foundations, Development, Applications (revised edition)*, George Braziller.

von Clausewitz, C. (1832). *Hinterlassene Werke über Krieg und Kriegführung: Vom Kriege*, Vol. 1, Dümmler.

von Mises, R. (1972). *Wahrscheinlichkeit, Statistik und Wahrheit*, Vol. 4, Springer.

Weichselberger, K. und Pöhlmann, S. (1990). *A methodology for uncertainty in knowledge-based systems*, Berlin (u. a.): Springer.

Winston, P. H. und Prendergast, K. A. (1986). Caduceus: An experimental expert system for medical diagnosis.

Wörsdörfer, R. (2014). *Modulare Architektur für militärische Führungsautomaten*, Masterarbeit, Universität der Bundeswehr München.

Zsambok, C. E. und Klein, G. (2014). *Naturalistic decision making*, Psychology Press.