Automatisches Beweisen

Einführung

Erhard Konrad

Sommersemester 2004

TU Berlin

Skript zum Praktikum

Inhalt

1 Einleitung

Automatisches Beweisen ist ein Teilgebiet der *Künstlichen Intelligenz.* Die Fähigkeit, logische Schlüsse zu ziehen, ist ein Merkmal menschlicher Intelligenz. Es stellt sich die Frage, ob und inwieweit sich logisches Schließen formalisieren lässt. Dies ist Gegenstand der *formalen Logik*, auch *mathematische Logik* genannt.

Für die Möglichkeiten und Grenzen der formalen Logik gelten die folgenden Theoreme:

a) Vollständigkeit der Logik erster Stufe: *Alle logisch wahren Sätze der Logik erster Stufe sind formal beweisbar.* (Gödel 1930)

b) Unvollständigkeit der Arithmetik: *Es gibt wahre arithmetische Sätze, die nicht formal beweisbar sind.* (Gödel 1931)

c) Unvollständigkeit der Logik zweiter Stufe: *Es gibt logisch wahre Sätze der Logik zweiter Stufe, die nicht formal beweisbar sind.* (Korollar zu Gödel 1931)

d) Unentscheidbarkeit der Logik erster Stufe: *Es ist nicht formal entscheidbar, ob ein beliebiger Satz der Logik erster Stufe logisch wahr ist.* (Church 1936)

Die Idee des Beweises in Verbindung mit der deduktiven Methode und der Axiomatik in der Mathematik geht auf Euklid zurück (um 300 v. Chr.). Beweise gibt es nicht nur in der Mathematik, sondern auch in anderen Bereichen, z.B. in der Jurisprudenz, wo Indizienbeweise und Augenscheinbeweise eine wichtige Rolle spielen.

Vom Beweis allgemein ist der *Beweis in einem System* zu unterscheiden, wie er in der Mathematik benutzt wird.

Ein *mathematischer Beweis* eines Satzes S in einem Axiomemsystem besteht aus einer Folge von Schlüssen, die von Axiomen oder bereits bewiesenen Sätzen zu dem Satz S führt.

Ein *formaler Beweis* einer Formel F einer Sprache L besteht aus einer Folge von Formeln in L, die mit F endet und bei der jede Formel durch Anwendung von Regeln eines Regelsystems R aus Axiomen (fomalisiert in L) oder bereits abgeleiten Formeln entsteht.

Das Praktikum dient als Einführung in das Gebiet des *automatisches Beweisens*, auch mechanisches oder maschinelles Beweisen genannt. Im Skript werden Hilfsmittel der klassischen Prädikatenlogik bereitgestellt, die zum Verständnis des Resolutionskalkül benötigt werden. Der Resolutionskalkül ist eine Formalisierung der Logik, die für einen Automaten (Computer) besonders geeignet ist. Das Beweisprogramm OTTER, das für das Praktikum zur Verfügung steht, beruht auf dem Resolutionskalkül. Die Methoden des automatischen Beweisens werden am Beispiel einer geometrischen Anwendung eingeübt.

Das Beweisprogramm OTTER wurde Ende 1980 am Argonne National Labaratory, Argonne (Illinois), von William W. McCune entwickelt. Das Akronym OTTER steht für „Organized Techniques for Theorem proving and Effective Research".

Im Jahre 1997 gelang es McCune, mit Hilfe von OTTER zu beweisen, dass jede Robbins Algebra eine Boolesche Algebra ist. Dies ist eine Problem, das 60 Jahre lang nicht gelöst werden konnte, auch nicht von Tarski, einem der größten Logiker des zwanzigsten Jahrhunderts. Diese herausragende Leistung mit dem Beweisprogramm OTTER war eine wissenschaftliche Sensation, die auch in der *New York Times* gewürdigt wurde.

2 Aussagenlogik

2.1 Syntax

Einfache und zusammengesetzte Aussagesätze:

Der Schnee ist weiß.

Der Schnee ist schwarz.

Der Schnee ist weiß **oder** der Schnee ist schwarz.

Wenn der Schnee weiß ist, **so** ist der Schnee schwarz.

Diese Aussagen können in der Sprache der Aussagenlogik formalisiert werden. Es wird zunächst das *Alphabet* erklärt, dann werden *Formeln* induktiv definiert.

Alphabet:

Aussagenvariablen: p, q, r (evtl. mit Indizes)

Logische Symbole: $\neg$ (nicht), $\wedge$ (und), $\vee$ (oder), $\rightarrow$ (wenn-so), $\leftrightarrow$ (genau dann, wenn)

Formeln:

1. Alle Aussagenvariablen sind Formeln.
2. Wenn A eine Formel ist, so ist auch $\neg A$ eine Formel.
3. Wenn A und B Formeln sind, so sind auch $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ und $(A \leftrightarrow B)$ Formeln.
4. Das sind alle Formeln.

2.2 Semantik

2.2.1 Wahrheitsfunktionen

Den Aussagenvariablen p und q werden die Wahrheitswerte „wahr“ oder „falsch“ zugeordnet. Die *Wahrheitsfunktionen* sind wie folgt definiert:

a) *Negation*: $\neg p$ ist wahr genau dann, wenn p nicht wahr ist.

b) *Konjunktion*: $(p \wedge q)$ ist wahr genau dann, wenn p und q wahr sind.

c) *Disjunktion*: $(p \vee q)$ ist wahr genau dann, wenn p oder q wahr ist.

d) *Implikation*: $(p \rightarrow q)$ ist falsch genau dann, wenn p wahr und q falsch ist, sonst wahr.

e) *Äquivalenz*: $(p \leftrightarrow q)$ ist wahr genau dann, wenn p und q beide wahr oder beide falsch sind.

2.2.2 Definitionen und Sätze

Definitionen:

a) Eine Formel A ist *allgemeingültig* genau dann, wenn A bei jeder Belegung der Aussagevariablen mit Wahrheitswerten wahr ist.

b) Eine Formel A ist *erfüllbar* genau dann, wenn es eine Belegung der Aussagevariablen mit Wahrheitswerten gibt, so dass A wahr ist.

c) Eine Formel B *folgt logisch* aus Formeln $A_1, A_2, ..., A_n$ genau dann, wenn $A_1 \wedge A_2 \wedge ... \wedge A_n \rightarrow B$ allgemeingültig ist.
(Notation: $A_1, A_2, ..., A_n \models B$ für „B ist eine *logische Folgerung* von $A_1, A_2, ..., A_n$".)

d) Zwei Formeln A und B sind *logisch äquivalent* genau dann, wenn $A \models B$ und $B \models A$.

e) A und B sind *erfüllbarkeitsäquivalent* genau dann, wenn: A ist erfüllbar genau dann, wenn B erfüllbar ist.

Sätze:

(1) Eine Formel A ist allgemeingültig genau dann, wenn $\neg A$ unerfüllbar ist.

(2) $A_1, A_2, ..., A_n \models B$ genau dann, wenn $\{A_1, A_2, ..., A_n, \neg B\}$ unerfüllbar ist.

2.3 Resolutionskalkül

2.3.1 Definitionen und Sätze

Im Resolutionskalkül wird der Begriff der *Ableitbarkeit* für eine spezielle Klasse von Formeln definiert, nämlich *Klauseln*. Voraussetzung für die Anwendung des Resolutionskalküls ist die Überführung von Formeln in Klauseln.

Klauseln:

a) Aussagenvariablen und ihre Negate heißen *Literale*.

Beispiele: p, $\neg p$, q, $\neg q$.

b) *Klauseln* sind Disjunktionen von Literalen (auch: Mengen von Literalen).

Beispiele: $p \vee q$ bzw. $\{p, q\}$, $\neg p \vee q$ bzw. $\{\neg p, q\}$.

Überführungssatz:

Jede Formel läßt sich in eine erfüllbarkeitsäquivalente Konjunktion von Klauseln (auch: Menge von Klauseln) überführen.

Mit Hilfe der *Resolutionsregel* können Klauseln aus Klauselpaaren hergeleitet werden.

Seien K_1 und K_2 Klauseln, und sei L ein Literal mit $L \in K_1$ und $\neg L \in K_2$. Dann lautet die *Resolutionsregel* (mit $\vdash$ für ableitbar):

$K_1, K_2 \vdash (K_1 - \{L\}) \cup (K_2 - \{\neg L\})$

Das Ergebnis der Ableitung heißt *Resolvente.*

Als Resolvente kann auch die leere Menge auftreten (bei $K_1 = \{L\}$ und $K_2 = \{\neg L\}$), sie wird *leere Klausel* genannt (symbolisch: $\emptyset$). Die leere Klausel ist unerfüllbar, ebenso jede Klauselmenge, die die leere Klausel als Element enthält.

Resolutionssatz:

Eine Klauselmenge K ist unerfüllbar genau dann, wenn die leere Klausel aus K abgeleitet werden kann.

Der Resolutionssatz besagt, dass der Resolutionskalkül der Aussagenlogik *korrekt* und *vollständig* ist.

2.3.2 Ableitungsstrategien: Beispiel

Gilt $\{p \to (q \vee r), \neg q, p \vee q \vee r\} \models r$?

Dies gilt genau dann, wenn:

$\{p \to (q \vee r), \neg q, p \vee q \vee r, \neg r\}$ ist unerfüllbar.

Und dies gilt genau dann, wenn:

$\{\{\neg p, q, r\}, \{\neg q\}, \{p, q, r\}, \{\neg r\}\} \vdash \emptyset$.

Es gibt zahlreiche Ableitungstrategien, die sich bezüglich der Effizienz unterscheiden. Für das Beispiel werden zwei Strategien verglichen: die *breitenorientierte Strategie* und die *Stützmengenstrategie.*

Bei der *breitenorientierten Strategie* werden – von links nach rechts – alle Resolventen der ersten Generation erzeugt, dann die der zweiten Generation usw. bis die leere Klausel erreicht ist:

$\{\{\neg p, q, r\}, \{\neg q\}, \{p, q, r\}, \{\neg r\}\}$	Ausgangsmenge
$\cup\{\{\neg p, q\}, \{q, r\}, \{p, r\}, \{p, q\}\}$	1. Generation
$\cup\{\{q, r\}, \{\neg p\}, \{r\}, \{p\}, \{q\}\}$	2. Generation
$\cup\{\emptyset\}$	3. Generation

Zur Ableitung der leeren Klausel werden 10 Resolventen benötigt.

Bei der *Stützmengenstrategie* genügen 5 Resolventen, wie der folgende Ableitungsgraph zeigt:

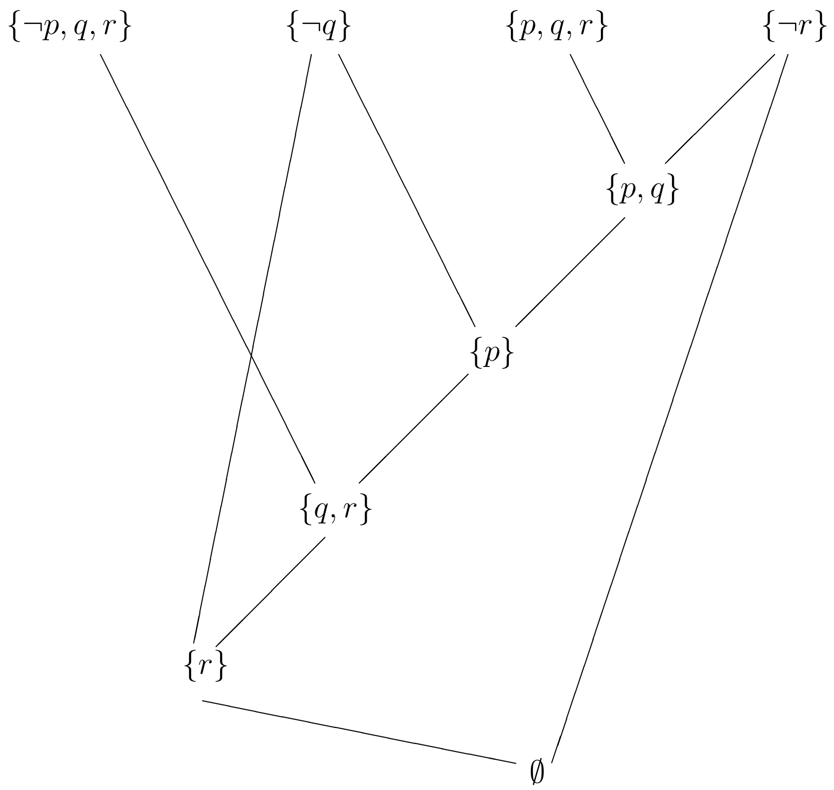

Abbildung 2.1: Ableitungsgraph mit Stützmengenstrategie

Am Anfang besteht die Stützmenge aus den Klauseln, die der vermuteten (negierten) Folgerung entsprechen (hier: $\neg r$). Bei der Ableitung sind stets Klauseln beteiligt, die in der Stützmenge vorkommen oder mit Hilfe der Stützmenge hergeleitet wurden. Beim Beweisprogramm OTTER kann die Stützmengenstrategie mit der Option *set of support* eingestellt werden. Dies ist bei großen Prämissenmengen empfehlenswert.

2.3.3 Beweisprogramm OTTER: Beispiel

Ereignis: Geburtstagsfeier

1. Wenn Anton teilnimmt, so nehmen auch Bernd oder Carola teil.
2. Bernd nimmt nicht teil.
3. Anton oder Bernd oder Carola nehmen teil.

Vermutung: Carola nimmt teil.

Formalisiert:

1. $A \to (B \vee C)$
2. $\neg B$
3. $A \vee B \vee C$

V: C

Eingabedatei (OTTER):

```
%Ereignis: Geburtstagsfeier
%1. Wenn Anton teilnimmt, so nehmen auch Bernd
%   oder Carola teil.
%2. Bernd nimmt nicht teil.
%3. Anton oder Bernd oder Carola nehmen teil.
%Verm.: Carola nimmt teil.

set(auto).              %Setzt Opt. selbst.

formula_list(usable).
   A -> (B | C).        %1. Vor.
   -B.                  %2. Vor.
   A | B | C.           %3. Vor.
    -C.                 %Neg. Verm.
end_of_list.
```

Ausgabedatei (OTTER):

```
set(auto).
     ...

formula_list(usable).
A->B|C.
-B.
A|B|C.
-C.
end_of_list.

-------> usable clausifies to:

list(usable).
0 [] -A|B|C.
0 [] -B.
0 [] A|B|C.
0 [] -C.
end_of_list.

...

The clause set is propositional; the strategy will be
ordered hyperresolution with the propositional
optimizations, with satellites in sos and nuclei in usable.

   dependent: set(hyper_res).
   dependent: set(propositional).
   dependent: set(sort_literals).

------------> process usable:
** KEPT (pick-wt=3): 1 [] -A|B|C.
** KEPT (pick-wt=1): 2 [] -B.
** KEPT (pick-wt=1): 3 [] -C.

------------> process sos:
** KEPT (pick-wt=3): 4 [] A|B|C.

======= end of input processing =======
```

```
=========== start of search ===========

given clause #1: (wt=3) 4 [] A|B|C.

given clause #2: (wt=2) 5 [hyper,4,3] A|B.

given clause #3: (wt=1) 6 [hyper,5,2] A.

given clause #4: (wt=2) 7 [hyper,6,1] B|C.

----> UNIT CONFLICT at   0.00 sec ----> 9 [binary,8.1,2.1] $F.

Length of proof is 4.  Level of proof is 4.

---------------- PROOF ----------------

1 [] -A|B|C.
2 [] -B.
3 [] -C.
4 [] A|B|C.
5 [hyper,4,3] A|B.
6 [hyper,5,2] A.
7 [hyper,6,1] B|C.
8 [hyper,7,3] B.
9 [binary,8.1,2.1] $F.

------------ end of proof -------------


Search stopped by max_proofs option.

============ end of search ============

...

That finishes the proof of the theorem.
```

3 Prädikatenlogik

3.1 Syntax

Prädikative Struktur einfacher Aussagesätze:

Der Schnee ist weiß.
Ist-weiß (Schnee).

Alle Schimmel sind weiß.
Für alle x (**wenn** ist-Schimmel (x), **so** ist-weiß (x)).

Es gibt weiße Raaben.
Es gibt ein x (ist-Raabe(x) **und** ist-weiß(x).)

Diese Aussagen können in der Sprache der Prädikatenlogik formalisiert werden. Es wird zunächst das *Alphabet* erklärt, dann werden *Terme* und *Formeln* induktiv definiert.

Alphabet:

(a) Individuensymbole:
u, v, w, x, y, z (Variablen),
a, b, c, d, e (Konstanten),
(evtl. mit Indizes).

(b) Funktionssymbole jeder Stellenzahl:
f, g, h (evtl. mit Indizes).

(c) Prädikatensymbole jeder Stellenzahl:
P, Q, R (evtl. mit Indizes).

(d) Logische Symbole: $\neg$ (nicht), $\wedge$ (und), $\vee$ (oder), $\rightarrow$ (wenn - so), $\leftrightarrow$ (genau dann, wenn), $\exists$ (es gibt), $\forall$ (für alle), = (gleich).

(e) Klammern: (,)

Terme:

(1) Jedes Individuensymbol ist ein Term.

(2) Wenn f ein n-stelliges Funktionssymbol ist und $t_1, t_2, \ldots, t_n$ Terme sind, so ist auch $f(t_1, \ldots, t_n)$ ein Term.

(3) Das sind alle Terme.

Formeln:

(1a) Wenn P ein n-stelliges Prädikatensymbol ist und $t_1, t_2, \ldots, t_n$ Terme sind, so ist $P(t_1, \ldots, t_n)$ eine Formel.

(1b) Wenn t_1 und t_2 Terme sind, so ist $t_1{=}t_2$ eine Formel.

(2) Wenn A eine Formel ist, so ist auch ¬A eine Formel.

(3) Wenn A und B Formeln sind, so sind auch (A∧B), (A∨B), (A→B) und (A↔B) Formeln.

(4) Wenn A eine Formel ist, so sind auch ∃xA und ∀xA Formeln.

(5) Das sind alle Formeln.

3.2 Semantik

3.2.1 Begriff der Interpretation

Eine *Interpretation* $\Im$ *der Sprache* L *über einem nichtleeren Individuenbereich* D ist eine Abbildung, welche jedem Individuensymbol x bzw. a genau ein Element $\Im$(x) bzw. $\Im$(a) aus D, jedem n-stelligen Funktionssymbol f genau eine n-stellige Funktion $\Im$(f) auf D, jedem n-stelligen Prädikatensymbol P genau eine n-stellige Relation $\Im$(P) auf D zuordnet.

Terme:

(1) $\Im$(x) bzw. $\Im$(a) ist schon erklärt.

(2) $\Im(\mathrm{f}(\mathrm{t}_1, \ldots, \mathrm{t}_n)) = \Im(\mathrm{f})\ (\Im(\mathrm{t}_1), \ldots, \Im(\mathrm{t}_n))$.

Formeln:

(1a) $\mathrm{P}(\mathrm{t}_1, \ldots, \mathrm{t}_n)$ ist wahr bei $\Im$ genau dann, wenn $<\Im(\mathrm{t}_1), \ldots, \Im(\mathrm{t}_n)> \in \Im(\mathrm{P})$.

(1b) $\mathrm{t}_1{=}\mathrm{t}_2$ ist wahr bei $\Im$ genau dann, wenn $\Im(\mathrm{t}_1) =_D \Im(\mathrm{t}_2)$.

(2) ¬A ist wahr bei $\Im$ genau dann, wenn A ist bei $\Im$ nicht wahr ist.

(3) (A∧B) ist wahr bei $\Im$ genau dann, wenn A und B bei $\Im$ wahr sind.

(4) (A∨B) ist wahr bei $\Im$ genau dann, wenn A oder B bei $\Im$ wahr ist.

(5) (A→B) ist falsch bei $\Im$ genau dann, wenn A bei $\Im$ wahr und B bei $\Im$ falsch ist, sonst wahr.

(6) (A↔B) ist wahr bei $\Im$ genau dann, wenn A und B bei $\Im$ wahr oder A und B bei $\Im$ falsch sind, sonst falsch.

(7) ∃xA ist wahr bei $\Im$ genau dann, wenn A bei $\Im_x^d$ für wenigstens ein d aus D wahr ist.
$\Im_x^d$ stimmt mit $\Im$ bis evtl. auf die Stelle x überein; es ist stets $\Im_x^d(\mathrm{x}) = \mathrm{d}$.

(8) ∀xA ist wahr bei $\Im$ genau dann, wenn A bei $\Im_x^d$ für alle d aus D wahr ist.

3.2.2 Definitionen und Sätze

Definitionen:

a) Eine Interpretation $\Im$, bei der eine Formel A wahr ist, heißt *Modell von A*.

b) Eine Formel A heißt *erfüllbar* genau dann, wenn es eine Interpretation gibt, bei der A wahr ist.

c) Eine Formel A heißt *allgemeingültig* genau dann, wenn A bei jeder Interpretation wahr ist.

d) Zwei Formeln A und B heißen *äquivalent* genau dann, wenn bei jeder Interpretation A und B entweder gleichzeitig wahr oder falsch sind.

e) Zwei Formeln A und B heißen *erfüllbarkeitsäquivalent* genau dann, wenn: A ist genau dann erfüllbar, wenn B erfüllbar ist.

Definition der *logischen Folgerung* mit dem Modellbegriff:

f) Eine Formel A *folgt* aus einer Formelmenge M ($M \models A$) genau dann, wenn jedes Modell von M auch ein Modell von A ist.

Sätze:

(1) Eine Formel A ist allgemeingültig genau dann, wenn $\neg A$ unerfüllbar ist.

(2) $M \models B$ genau dann, wenn $M \cup \{\neg B\}$ unerfüllbar ist.

3.3 Resolutionskalkül

3.3.1 Definitionen und Sätze

Die Begriffe in der Aussagenlogik werden erweitert. Literale sind jetzt atomare Formeln und ihre Negate. Um die Resolution anwenden zu können, müssen Literale vereinheitlicht werden.

Klauseln:

a) Atomare Formeln und ihre Negate heißen *Literale*.

Beispiele: $P(a)$, $\neg a = b$, $P(f(x))$, $\neg P(fx))$, $Q(a,x)$, $Q(x, g(a,y))$, $\neg Q(x,y,z)$, $a = g(a,b,y)$.

b) *Klauseln* sind Disjunktionen von Literalen (auch: Mengen von Literalen).

Beispiele: $P(f(x)) \vee \neg Q(a)$ bzw. $\{P(f(x)), \neg Q(a)\}$, $a = g(x,b,y) \vee Q(y)$ bzw. $\{a = g(x,b,y), Q(y)\}$, $P(a) \vee \neg Q(a, g(x,b))$ bzw. $\{P(a), \neg Q(a, g(x,b))\}$.

Überführungssatz:

Jede Formel läßt sich in eine erfüllbarkeitsäquivalente Konjunktion von Klauseln (Menge von Klauseln) überführen.

Unifikation:

Sei $\sigma = [(x_1/t_1, x_2/t_2, ..., x_n/t_n]$ eine *Substitution*, Dann bedeutet $L\sigma$ bzw. $K\sigma$, dass die Variablen $x_1, x_2, ..., x_n$ überall, wo sie in dem Literal L bzw. der Klausel K vorkommen, durch die Terme $t_1, t_2, ..., t_n$ ersetzt werden.

Beispiele: $g(a, y)[y/f(a)] = g(a, f(a)), f(x, y)[x/a, y/g(y)] = f(a, g(y))$,
$P(x, f(y, b))[x/a, y/g(y)] = P(a, f(g(y), b))$,
$\{\{P(b, x)\}, \{\neg Q(y), R(a)\}\}[x/a, y/b] = \{\{P(b, a)\}, \{\neg Q(b), R(a)\}\}$.

Seien t_1 und t_2 Terme. Wenn es eine Substitution σ gibt, so dass $t_1\sigma = t_2\sigma$, dann heißt σ *Unifikator* von t_1 und t_2. Es wird gesagt, dass t_1 und t_2 *unifizierbar* sind.

Beispiele: Die Terme f(x, b) und f(a, y) sind unifizierbar mit dem Unifikator [x/a, y/b], die Terme f(x, f(y, z)) und f(f(u, v), u) sind unifizierbar mit dem Unifikator [x/f(f(y, z), v), u/f(y, z)].
Die Terme f(x) und g(y) sind nicht unifizierbar.

Eine Substitution σ heißt *allgemeinster Unifikator* zweier Terme t_1 und t_2. wenn σ ein Unifikator von t_1 und t_2 ist und wenn gilt: zu jedem Unifikator τ von t_1 und t_2 gibt es eine Substitution ρ so dass $\tau = \rho \circ \sigma$.

Die Definitionen zur Unifikation werden auf atomare Formeln und Literale ausgedehnt:

Zwei atomare Formeln F_1 und F_2 heißen *unifizierbar*, wenn es eine Substitution σ gibt, so dass $F_1\sigma = F_2\sigma$. Bei Literalen wird vom Negationszeichen abgesehen: Zwei Literale F_1 und $\neg F_2$ heißen *unifizierbar*, wenn es eine Substitution σ gibt, so dass $F_1\sigma = F_2\sigma$. F_1 und $\neg F_2$ heißen *komplementäre Literale*.

Resolutionsregel:

Seien K_1 und K_2 Klauseln ohne gemeinsame Variablen, und seien L_1 und L_2 komplementäre Literale mit $L_1 \in K_1$ und $L_2 \in K_2$. Wenn L_1 und L_2 einen allgemeinsten Unifikator σ haben, so gilt:

$K_1, K_2 \vdash (K_1\sigma - \{L_1\sigma\}) \cup (K_2\sigma - \{L_2\sigma\})$

Das Ergebnis der Ableitung heißt *Resolvente*.

Resolutionssatz:

Eine Klauselmenge K ist unerfüllbar genau dann, wenn die leere Klausel aus K abgeleitet werden kann.

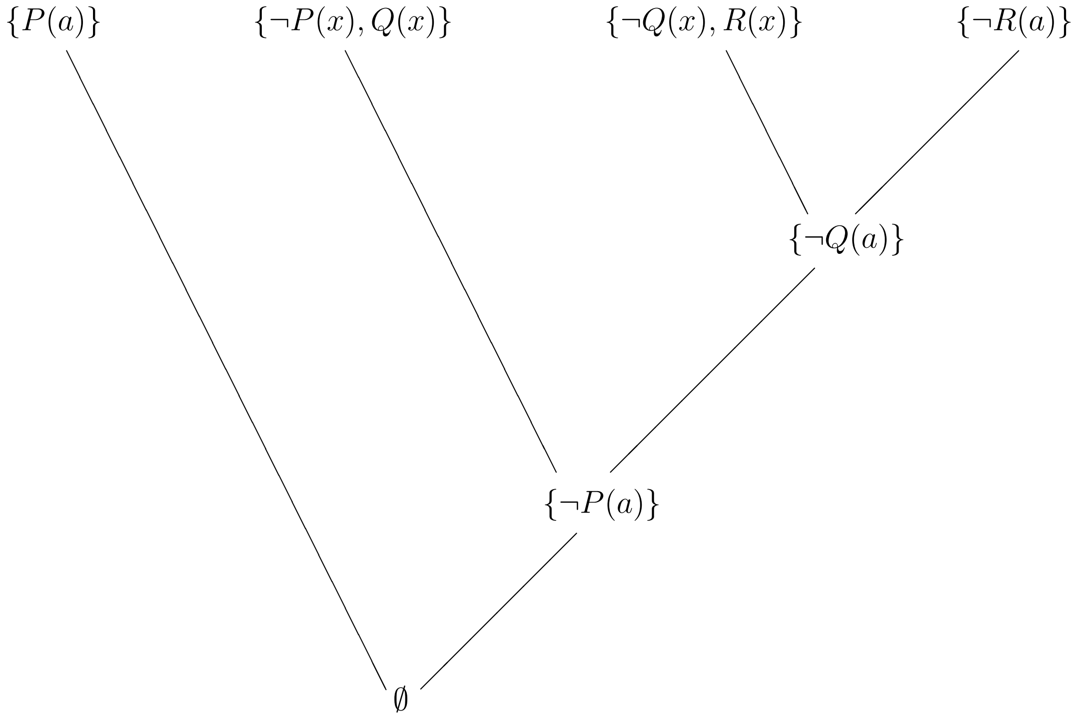

Abbildung 3.1: Ableitungsgraph mit Stützmengenstrategie

Mit dem Unifikator [x/a] werden die Literale $R(x)$ und $\neg R(a)$ unifiziert, so dass die Klausel $\{\neg Q(a)\}$ abgeleitet werden kann. Die Literale $\neg Q(a)$ und $Q(x)$ werden dann ebenfalls mit [x/a] unifiziert. Nun bekommt man die Klausel $\{\neg P(a)\}$ und danach mit $\{P(a)\}$ die leere Klausel als Resolvente.

3.3.2 Beweisprogramm OTTER: Beispiel

Thema: Klassifikation

1. Alle Enten sind Vögel.
2. Alle Vögel haben Federn.
3. Donald ist eine Ente.

Vermutung: Donald hat Federn.

Formalisiert:

1. $\forall x(E(x) \rightarrow V(x))$
2. $\forall x(V(x) \rightarrow F(x))$
3. E(d)

V: F(d)

Eingabedatei (OTTER):

```
%Thema: Klassifikation
%1. Alle Enten sind Voegel.
%2. Alle Voegel haben Federn.
%3. Donald ist eine Ente.
%Verm.: Donald hat Federn.



set(auto).             %Setzt Opt. selbst.



formula_list(usable).
  all x (E(x) -> V(x)).  %1. Vor.
  all x (V(x) -> F(x)).  %2. Vor.
  E(d).                  %3. Vor.
  -F(d).                 %Neg. Verm.
end_of_list.
```

Ausgabedatei (OTTER):

```
set(auto).
     ...
formula_list(usable).
all x (E(x)->V(x)).
all x (V(x)->F(x)).
E(d).
-F(d).
end_of_list.
```

```
-------> usable clausifies to:

list(usable).
0 [] -E(x)|V(x).
0 [] -V(x)|F(x).
0 [] E(d).
0 [] -F(d).
end_of_list.
...

This is a Horn set without equality.  The strategy will
be hyperresolution, with satellites in sos and nuclei
in usable.

   dependent: set(hyper_res).
   dependent: clear(order_hyper).

------------> process usable:
** KEPT (pick-wt=4): 1 [] -E(x)|V(x).
** KEPT (pick-wt=4): 2 [] -V(x)|F(x).
** KEPT (pick-wt=2): 3 [] -F(d).

------------> process sos:
** KEPT (pick-wt=2): 4 [] E(d).

======= end of input processing =======



=========== start of search ===========

given clause #1: (wt=2) 4 [] E(d).

given clause #2: (wt=2) 5 [hyper,4,1] V(d).

----> UNIT CONFLICT at   0.00 sec ----> 7 [binary,6.1,3.1] $F.

Length of proof is 2.  Level of proof is 2.
```

```
---------------- PROOF ----------------

1 [] -E(x)|V(x).
2 [] -V(x)|F(x).
3 [] -F(d).
4 [] E(d).
5 [hyper,4,1] V(d).
6 [hyper,5,2] F(d).
7 [binary,6.1,3.1] $F.

------------ end of proof -------------

Search stopped by max_proofs option.

============ end of search ============
...
That finishes the proof of the theorem.
```

4 Geometrische Anwendung

4.1 Geometrische Welt

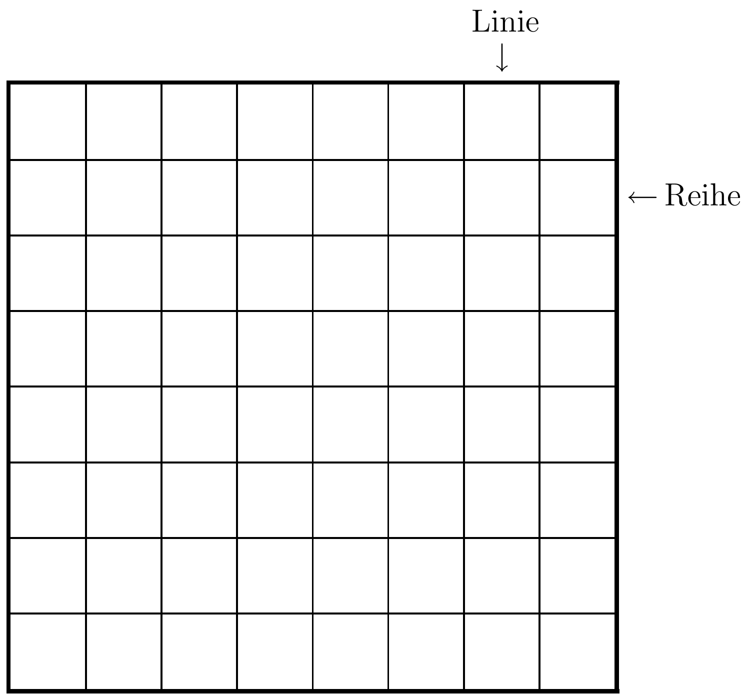

Abbildung 4.1 Geometrische Welt

Auf einem Schachbrett (8x8 Felder) können geometrische Figuren unterschiedlicher Größe aufgestellt werden, auf jedes Feld höchstens eine Figur. Figuren können einen oder mehrere Namen haben oder namenlos sein. Verschiedene Figuren dürfen nicht den gleichen Namen haben. In der Sprache GEO werden Form und Größe der Figuren sowie ihre räumlichen Beziehungen zueinander beschrieben. Die 8 horizontalen Felderfolgen heißen *Reihen* und die 8 vertikalen Felderfolgen *Linien*.

4.2 Syntax der Sprache GEO

Alphabet:

a) *Namen (Konstanten):* a, b, c, d, e (evtl. mit Indizes)
b) *Einstellige Funktionssymbole:*
fm (frontmost), bm (backmost), lm (leftmost), rm (rightmost)

(c) *Prädikatensymbole*:
1-stellig: Cube, Tet, Dodec, Small, Medium, Large.
2-stellig: Smaller, Larger, Leftof, RightOf, FrontOf, BackOf, SameSize, SameShape, SameRow, SameCol, Adjoins.
3-stellig: Between.

Terme und Formeln werden wie unter 3.1 aufgebaut.

4.3 Semantik der Sprache GEO

Einfache Terme:

Term	*Interpretation*
fm(a)	die vorderste Figur in der Linie, in der die Figur a steht
bm(a)	die hinterste Figur in der Linie, in der die Figur a steht
lm(a)	die am weitesten links stehende Figur in der Reihe, in der die Figur a steht
rm(a)	die am weitesten rechts stehende Figur in der Reihe, in der die Figur a steht

Atomare Formeln:

Formel	*Interpretation*
Cube(a)	a ist ein Würfel
Tet(a)	a ist ein Tetraeder
Dodec(a)	a ist ein Dodekaeder
Small(a)	a ist klein
Medium(a)	a ist mittelgroß
Large(a)	a ist groß
Smaller(a, b)	a ist kleiner als b
Larger(a, b)	a ist größer als b
LeftOf(a, b)	a ist links von b (näher am linken Rand als b)
RightOf(a, b)	a ist rechts von b (näher am rechten Rand als b)
FrontOf(a, b)	a ist weiter vorne als b (näher am vorderen Rand als b)
BackOf(a, b)	a ist weiter hinten als b (näher am hinteren Rand als b)
SameSize(a, b)	a hat die gleiche Größe wie b
SameShape(a, b)	a hat die gleiche Form wie b
SameRow(a, b)	a ist in der gleichen Reihe wie b
SameCol(a, b)	a ist in der gleichen Linie wie b
Adjoins(a, b)	a ist auf einem Nachbarfeld von b (aber nicht diagonal)
Between(a, b, c)	a, b und c sind in der gleichen Reihe, Linie oder Diagonalen, und a ist zwischen b und c

4.4 Dreidimensionale Beispielwelt

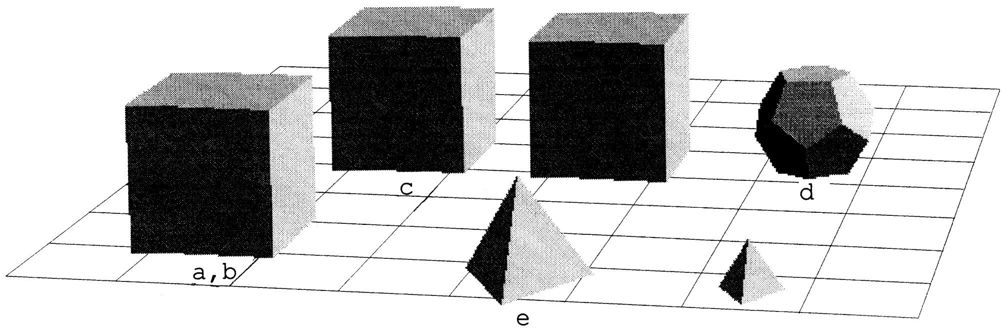

Abbildung 4.2: Dreidimensionale Welt

Die Individuenkonstanten a, b, c, d und e bezeichnen Figuren der Welt in Abb. 4.2. Figuren können keine, eine oder mehrere Bezeichnungen (Namen) haben. Die Beispielwelt enthält einen großen Würfel mit zwei Namen (a,b), einen großen Würfel mit einem Namen (c), einen großen Würfel ohne Namen, ein mittelgroßes Tetraeder mit einem Namen (e), ein kleines Tetraeder ohne Namen und ein mittelgroßes Dodekaeder mit einem Namen (d).

Wir betrachten die folgenden Aussagen:

(1) a=b

(2) c=e

(3) Cube(a)

(4) Tet(e)

(5) Dodec(c)

(6) Between(c, a, e)

(7) ¬Cube(d)

(8) a=b ∨ ¬Tet(e)

(9) Tet(e) ∨ Dodec(c)

(10) c=e $\rightarrow$ Cube(d)

(11) $\neg$Smaller(a, c) $\vee$ $\neg$Larger(e, c)

(12) $\forall$x Cube(c)

(13) $\forall$x(Cube(x) $\rightarrow$ Large(x))

(14) $\exists$x(Tet(x) $\wedge$ Small(x))

(15) $\exists$x Between(x, c, d)

Die Formeln (1) bis (6) sind atomar. Ihre Wahrheit oder Falschheit bestimmen wir durch Beobachtung, ebenso verfahren wir bei den atomaren Bestandteilen der zusammengesetzen Formeln. Verifikation ist jedoch keine Aufgabe der Logik. Ausgangpunkt der logischen Semantik ist eine Interpretation, durch die alle Wahrheitswerte festgelegt sind. Das Prädikat *Cube* z. B. kann mit der Klasse aller Würfel identifiziert werden (extensionale Auffassung). Dann trifft das Prädikat *Cube* auf alle Figuren zu, die dieser Klassen angehören. Analog kann die Extension der übrigen Prädikate bestimmt werden.

Gemäß Beobachtung und logischer Analyse sind die folgenden Aussagen wahr: (1), (3), (7), (8), (9), (10), (11), (13), (14) und (15). Die übrigen Aussagen sind falsch. Bemerkenswert ist, daß sich die Aussagen (14) und (15) auf namenlose Figuren beziehen.

5 Übungsaufgaben

1. Aufgabenblatt

Aufgabe 1 (4 Punkte)

Welche der folgenden Formeln sind allgemeingültig? Jeweils Beweis mit OTTER oder Angabe einer falsifizierenden Belegung.

a) $p \to (q \to p)$
b) $(p \to (q \to r)) \to (q \to (p \to r))$
c) $(p \to (q \to r)) \to ((p \lor q) \to r)$
d) $(p \to (q \to r)) \to ((p \land q) \to r)$

Aufgabe 2 (4 Punkte)

Welche der folgenden Formelpaare sind logisch äquivalent? Jeweils Beweis mit OTTER oder Angabe einer falsifizierenden Belegung.

a) $p \to q$ und $(p \land q) \leftrightarrow p$
b) $p \to q$ und $(p \lor q) \leftrightarrow q$
c) $p \leftrightarrow q$ und $\neg p \leftrightarrow \neg q$
d) $p \leftrightarrow q$ und $(r \lor p) \leftrightarrow (r \lor q)$

Aufgabe 3 (4 Punkte)

Die folgenden Formeln seien als Voraussetzungen gegeben:

a) $\neg p \land q$
b) $r \to s$
c) $(q \lor r) \to (p \to t)$
d) $s \to p$

Welche der folgenden Formeln (Vermutungen) folgen logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe einer falsifizierenden Belegung zum Nachweis der Unabhängigkeit.

v1) $s \to t$
v2) $q \to p$
v3) $q \to (s \to t)$
v4) $p \to s$

Aufgabe 4 (4 Punkte)

Welche der folgenden Formeln sind von den übrigen vier Formeln logisch abhängig? Beweis mit OTTER.

a) $(\neg p \vee q) \rightarrow r$
b) $(s \wedge r) \rightarrow t$
c) $t \rightarrow k$
d) $k \wedge q \wedge s$
e) $q \vee t$

2. Aufgabenblatt

Aufgabe 1 (4 Punkte)

Welche der folgenden Formeln sind allgemeingültig? Jeweils Beweis mit OTTER oder Angabe einer falsifizierenden geometrischen Interpretation.

a) $\forall x(P(x) \rightarrow Q(x)) \rightarrow (\forall x P(x) \rightarrow \forall x Q(x))$
b) $(\forall x P(x) \rightarrow \forall x Q(x)) \rightarrow \forall x(P(x) \rightarrow Q(x))$
c) $(\exists x P(x) \wedge \exists x Q(x)) \rightarrow \exists x(P(x) \wedge Q(x))$
d) $\exists x(P(x) \wedge Q(x)) \rightarrow (\exists x P(x) \wedge \exists x Q(x))$

Aufgabe 2 (4 Punkte)

Welche der folgenden Formelpaare sind logisch äquivalent? Jeweils Beweis mit OTTER oder Angabe einer falsifizierenden geometrischen Interpretation.

a) $\exists x(P(x) \vee Q(x))$ und $\exists x P(x) \vee \exists x Q(x)$
b) $\forall x(P(x) \wedge Q(x))$ und $\forall x P(x) \wedge \forall x Q(x)$
c) $\forall x(P(x) \vee Q(x))$ und $\forall x P(x) \vee \forall x Q(x)$
d) $\exists x(P(x) \wedge Q(x))$ und $\forall x P(x) \wedge \forall x Q(x)$

Aufgabe 3 (5 Punkte)

a) Welche der Formeln A_3 oder A_4 folgt logisch aus der Formelmenge $\{A_1, A_2\}$? Jeweils Beweis mit OTTER oder Angabe einer falsifizierenden geometrischen Interpretation.

A_1: $\forall x(P(x) \rightarrow Q(x))$
A_2: $\neg\forall x(Q(x)$
A_3: $\exists x P(x)$
A_4: $\exists x \neg P(x)$

b) Analog zu a) für B_3 oder B_4 oder B_5 bezüglich $\{B_1, B_2\}$.

B_1: $\neg\exists x(P(x) \rightarrow Q(x))$
B_2: $\forall x(R(x) \rightarrow Q(x))$
B_3: $\forall x(R(x) \rightarrow \neg P(x))$
B_4: $\forall x(\neg R(x) \rightarrow \neg P(x))$
B_5: $\forall x(R(x) \rightarrow P(x))$

Aufgabe 4 (3 Punkte)

Gegeben seien die folgenden Voraussetzungen:

a) $\forall y(Cube(y) \lor Dodec(y))$
b) $\forall x(Cube(x) \to Large(x))$
c) $\exists x \neg Large(x)$

Folgt $\exists x Dodec(x)$ oder $\exists x(Dodec(x) \land Small(x))$ logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe einer geometrischen Welt zum Nachweis der Unabhängigkeit.

3. Aufgabenblatt

Aufgabe 1 (3 Punkte)

Gegeben seien die folgenden Voraussetzungen:

a) $\forall x(Cube(x) \to \exists y LeftOf(x, y))$
b) $\neg\exists x\exists z(Cube(x) \land Cube(z) \land LeftOf(x, z))$
c) $\exists x\exists y(Cube(x) \land Cube(y) \land x \neq y)$

Folgt $\exists x\exists y\exists z(BackOf(y, z) \land LeftOf(x, z))$ oder $\exists x\neg Cube(x)$ logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe einer geometrischen Welt als Gegenbeispiel.

Aufgabe 2 (5 Punkte)

Welche der folgenden Schlüsse sind gültig bzw. ungültig? Jeweils Formalisierung in der Sprache GEO, dann entweder Beweis mit OTTER oder Angabe eines Gegenbeispiels.

a) Alle Würfel sind groß.
Alle Tetraeder sind groß.
Also sind alle Würfel Tetraeder.

b) Einige Würfel sind groß.
Einige Tetraeder sind groß.
Also sind einige Würfel Tetraeder.

c) Alle Würfel sind groß.
Nicht alle Objekte sind groß.
Also sind einige Objekte keine Würfel.

d) Kein Würfel ist ein Tetraeder.
Kein Dodekaeder ist ein Würfel.
Also ist kein Dodekaeder ein Tetraeder.

e) Alle großen Objekte sind Würfel.
Alle Tetraeder sind groß.
Also sind einige Tetraeder Würfel.

Aufgabe 3 (4 Punkte)

Gegeben seien die folgenden Voraussetzungen:

a) $\forall x(Tet(lm(x)) \rightarrow Cube(x))$
b) $\forall x(lm(lm(x)) = lm(x))$
c) $\neg Cube(lm(a))$

Welche der folgenden Formeln folgen logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe einer geometrischen Welt als Gegenbeispiel.

f1) $Tet(a)$
f2) $\neg Tet(a)$
f3) $\neg Tet(lm(a))$
f4) $\neg Tet(lm(lm(a)))$

Aufgabe 4 (4 Punkte)

Gegeben seien die folgenden Voraussetzungen:

a) $\forall x(Small(x) \leftrightarrow fm(x) = x)$
b) $\forall x(Small(bm(x)) \leftrightarrow Small(fm(x)))$
c) $\forall x(fm(x) = b)$

Welche der folgenden Formeln folgen logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe einer geometrischen Welt als Gegenbeispiel.

f1) $\exists x Small(x)$
f2) $\forall x Small(x)$
f3) $\exists x Small(fm(x))$
f4) $\exists x Small(bm(x))$

4. Aufgabenblatt

Aufgabe 1 (6 Punkte)

Welche der folgenden Formeln sind allgemeingültig? Jeweils Beweis mit OTTER oder Angabe eines geometrischen bzw. arithmetischen Gegenbeispiels.

a) $\forall x(SameShape(x, y) \rightarrow SameShape(y, x))$
b) $\forall x(x = y \rightarrow y = x)$
c) $Tet(a) \rightarrow \exists x Tet(x)$
d) $\exists x Tet(x) \rightarrow Tet(a)$
e) $\exists x \forall y Larger(x, y) \rightarrow \forall y \exists x Larger(x, y)$
f) $\forall y \exists x Larger(x, y) \rightarrow \exists x \forall y Larger(x, y)$

Aufgabe 2 (4 Punkte)

Gegeben seien die folgenden Formeln als Voraussetzungen:

a) $\neg \exists x Larger(x, a)$.
b) $\neg \exists x Larger(b, x)$.
c) $Larger(c, d)$
d) $Small(d)$
e) $SameSize(d, e)$

Welche der folgenden Formeln folgen logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe eines geometrischen bzw. arithmetischen Gegenbeispiels.

f1) $Larger(a, b)$
f2) $\neg Larger(b, a)$
f3) $\neg Larger(b, d)$
f4) $Small(e)$

Aufgabe 3 (6 Punkte)

Gegeben seien die folgenden Voraussetzungen:

a) a und b sind Würfel.
b) a ist groß und b ist klein.
c) Kein Objekt ist sowohl groß als auch klein.

Welche der folgenden Aussagen folgen logisch aus den Voraussetzungen?

f1) Es gibt mindestens zwei Würfel.
f2) Es gibt höchstens zwei Würfel.
f3) Es gibt genau zwei Würfel.

Jeweils Formalisierung in der Sprache GEO, dann entweder Beweis mit OTTER oder Angabe eines Gegenbeispiels. Wie hängen die Aussagen f1), f2) und f3) logisch untereinander zusammen? Jeweils Beweis mit OTTER.

5. Aufgabenblatt

Aufgabe 1 (4 Punkte)

Gegeben seien die folgenden Formeln als Voraussetzungen:

a) $\forall x \forall y(LeftOf(x,y) \rightarrow Larger(x,y))$
b) $\forall x(Cube(x) \rightarrow Small(x))$
c) $\forall x(Tet(x) \rightarrow Large(x))$
d) $\forall x \forall y((Small(x) \wedge Small(y)) \rightarrow \neg Larger(x,y))$

Welche der folgenden Formeln folgen logisch aus den Voraussetzungen? Jeweils Beweis mit OTTER oder Angabe eines Gegenbeispiels.

f1) $\neg \exists z \exists y(Cube(z) \wedge Cube(y) \wedge LeftOf(z,y))$
f2) $\forall z(Medium(z) \rightarrow Tet(z))$
f3) $\forall z \forall w((Tet(z) \wedge Cube(w)) \rightarrow LeftOf(z,w))$
f4) $\forall z \forall y(LeftOf(z,y) \rightarrow (\neg Small(z) \vee \neg Small(y)))$

Aufgabe 2 (6 Punkte)

Gegeben seien die folgenden Aussagen als Voraussetzungen:

a) $a \neq b \wedge b \neq c \wedge a \neq c$
b) $Larger(a,b) \wedge \neg Larger(a,c)$
c) $\neg \exists x Small(x)$
d) $\forall x(\neg Small(x) \rightarrow \neg Cube(x))$
e) $Dodec(b) \wedge Dodec(c)$
f) $\exists x(Dodec(x) \wedge \exists y \exists z(Tet(y) \wedge Tet(z) \wedge Between(x,y,z)))$
g) $\forall x(Medium(x) \leftrightarrow \exists y FrontOf(x,y)$
h) $\forall x(LeftOf(x,c) \rightarrow (Medium(x) \vee Large(x)))$
i) $\forall x(Cube(x) \vee Tet(x) \vee Dodec(x))$
j) $\forall x(Small(x) \vee Medium(x) \vee Large(x))$
k) $\forall x \forall y \forall z((Larger(x,y) \wedge \neg Larger(x,z)) \rightarrow Medium(x))$
l) $\forall x \forall y(Larger(x,y) \rightarrow (\neg Small(x) \wedge \neg Large(y)))$
m) $\forall x \forall y(FrontOf(x,y) \leftrightarrow BackOf(y,x))$
n) $\forall x(Medium(x) \rightarrow \exists y \exists z Between(x,y,z))$

Welche Schlüsse (mind. 12) können aus den Aussagen a) bis n) über die Objekte a, b und c gezogen werden (Form, Größe, Lage zu anderen Objekten)? Jeweils Beweis mit OTTER.

Aufgabe 3 (6 Punkte)

Die zweistellige Beziehung der Formgleichheit von geometrischen Objekten (SameShape) wird durch das folgende Axiomensystem beschrieben:

a) $\forall x(Cube(x) \vee Dodec(x) \vee Tet(x))$
b) $\neg\exists x(Cube(x) \wedge Tet(x))$
c) $\neg\exists x(Tet(x) \wedge Dodec(x))$
d) $\neg\exists x(Dodec(x) \wedge Cube(x))$
e) $\forall x\forall y((Cube(x) \wedge Cube(y)) \rightarrow SameShape(x, y))$
f) $\forall x\forall y((Dodec(x) \wedge Dodec(y)) \rightarrow SameShape(x, y))$
g) $\forall x\forall y((Tet(x) \wedge Tet(y)) \rightarrow SameShape(x, y))$
h) $\forall x\forall y((SameShape(x, y) \wedge Cube(x)) \rightarrow Cube(y))$
i) $\forall x\forall y((SameShape(x, y) \wedge Dodec(x)) \rightarrow Dodec(y))$
j) $\forall x\forall y((SameShape(x, y) \wedge Tet(x)) \rightarrow Tet(y))$
k) $\forall x\forall y(SameShape(x, y) \leftrightarrow ((Cube(x) \wedge Cube(y)) \vee (Tet(x) \wedge Tet(y))\vee$ $(Dodec(x) \wedge Dodec(y)))$

Einige Axiome sind logische Folgerungen der übrigen Axiome. Es soll eine minimale Teilmenge angegeben werden, aus der alle aufgeführten Axiome logisch folgen. Beweis mit OTTER.

Literatur

BARWISE, JON; ETCHEMENDY, JOHN: *The Language of First Order Logic – Tarskis World 4.0*, 3. Auflage, CSLI Publications, Stanford 1992.

BARWISE, JON; ETCHEMENDY, JOHN: *Language, Proof and Logic*, CSLI Publications, Stanford 1999/2000.

K.H. BLÄSIUS; H.-J. BÜRCKERT (HRSG.): *Deduktionssysteme – Automatisierung des logischen Denkens*, 2. Auflage, Oldenbourg Verlag, München 1992.

MCCUNE, WILLIAM: *OTTER 3.3 Reference Manual*, Argonne National Laboratory, Tech. Mem. No. 263, Argonne (Illinois) 2003.

EBBINGHAUS, HEINZ-DIETER; FLUM, JÖRG; THOMAS, WOLFGANG: *Einführung in die mathematische Logik*, 3. Auflage, BI Wissenschaftsverlag, Mannheim 1992.

HERMES, HANS: *Einführung in die mathematische Logik*, 2. Auflage, Teubner, Stuttgart 1972.

KALMAN, JOHN A.: *Automated Reasoning with OTTER*, Rinton Press, Princeton (New Jersey) 2001.

SCHÖNING, UWE: *Logik für Informatiker*, 3. Auflage, BI Wissenschaftsverlag, Mannheim 1992.