Logik

Markus Lohrey

Universität Siegen

Sommersemester 2014

Organisatorisches zur Vorlesung

Informationen finden Sie unter

`http://www.eti.uni-siegen.de/ti/lehre/ss14/logik/`

z. B.

- Aktuelle Version der Folien
- Übungsblätter

Literaturempfehlung:

- Schöning: Logik für Informatiker, Spektrum Akademischer Verlag
- Ebbinghaus, Flum, Thomas: Einfhrung in die mathematische Logik, Spektrum Akademischer Verlag

Die **Übungen** werden von Herrn Moses Ganardi organisiert.

Geschichte der Logik

Beginn in Griechenland: Aristoteles (384–322 v.Chr.) untersuchte das Wesen der Argumentation und des logischen Schließens.

Verschiedene Werke, u.a.: Analytica priora, Analytica posteriora.

Aristoteles nennt die logischen Schlussfolgerungen Syllogismen.

Syllogismen (I)

Ein Syllogismus ist eine Aussage, in der bestimmte Dinge [die Prämissen] behauptet werden und in der etwas anderes [die Konsequenz], unumgänglich aus dem Behaupteten folgt. Mit dem letzten Satz meine ich, dass die Prämissen die Konsequenz zum Resultat haben, und damit meine ich, dass keine weitere Prämisse erforderlich ist, um die Konsequenz unumgänglich zu machen.

Syllogismen (II)

Wenn alle Menschen sterblich sind und
Sokrates ein Mensch ist,
dann ist Sokrates sterblich.

Wenn eine Zahl gerade und größer als zwei ist,
dann ist sie keine Primzahl.

Wenn die Leitzinsen hoch sind,
dann sind die Börsianer unzufrieden.

Syllogismen (III)

Aristoteles kompilierte eine Liste der zulässigen Syllogismen.

Alle Dackel sind Hunde Alle Hunde sind Tiere --- Dann sind alle Dackel Tiere	Alle P sind M Alle M sind S --- Alle P sind S	(Barbara)

Keine Blume ist ein Tier Alle Hunde sind Tiere --- Dann ist keine Blume ein Hund	Kein P ist M Alle S sind M --- Kein P ist S	(Cesare)

Alle Delfine leben im Meer Alle Delfine sind Säugetiere --- Dann leben einige Säugetiere im Meer	Alle M sind P Alle M sind S --- Einige S sind P	(Darapti)

Kritik an Aristoteles (aus moderner Sicht)

Es gibt viele korrekte Schlussfolgerungen, die in Aristoteles' Liste nicht vorkommen, z.B.:

$$\frac{\text{Alle Dackel sind Hunde}}{\text{Alle Dackelsschwänze sind Hundeschwänze}}$$

Aristoteles liefert keinen Kalkül für die Behandlung großer Ketten von Schlussfolgerungen.
(Leibniz war etwa 2000 Jahre später der erste, der sich einen solchen Kalkül ausgemalt hat.)

Die Aussagenlogik

Boole (1815 – 1864) entwickelt einen Kalkül zum Rechnen mit atomare Aussagen, die entweder wahr oder falsch sein können.

Verknüpfung durch Operatoren
(und; oder; nicht; wenn-dann . . .).

Keine Operatoren für Quantifizierung
(alle, einige).

Beispiel:

- Aussagen: "Anna ist Architektin", "Bruno ist Jurist".
- Vier mögliche Situationen oder "Welten":
  (1) Anna ist Architektin, Bruno ist Jurist.
  (2) Anna ist Architektin, Bruno ist kein Jurist.
  (3) Anna ist keine Architektin, Bruno ist Jurist.
  (4) Anna ist keine Architektin, Bruno ist kein Jurist.
- Einige der möglichen Verknüpfungen:
  "Anna ist Architektin oder Bruno ist Jurist".
  "Wenn Anna Architektin ist, dann ist Bruno Jurist".
  "Wenn Anna keine Architektin ist, dann ist Bruno kein Jurist".
  "Wenn Bruno kein Jurist ist, dann ist Anna keine Architektin".

"B folgt aus A": B ist wahr in allen Welten, in denen A wahr ist.

Algebraischer Kalkül um zu bestimmen, ob B aus A folgt.
Der Kalkül basiert auf der Analogie zwischen wahr und 1, falsch und 0, oder und Addition, und und Multiplikation.

Die Prädikatenlogik (Ende des 19. Jahrhunderts)

Gottlob Frege (1848–1925), Giuseppe Peano (1858–1932), Betrand Russell (1872–1970):

Logik als Grundlage der Mathematik, als formale Basis für die Vermeidung von Widersprüchen.

Entwicklung der Prädikatenlogik, die erlaubt:

- Beziehungen zwischen "Objekten" zu beschreiben
- existentielle Aussagen zu treffen: "es gibt ein x, so daß ..."
- universelle Aussage zu treffen: "für jedes x gilt, daß ..."

Beispiel: Für jede natürliche Zahl x gilt, daß es eine natürliche Zahl y gibt, so daß x kleiner als y ist.

Logik in der Informatik

Shannon (1916 – 2001) zeigt 1937 dass die boole'sche Algebra benutzt werden kann, um elektromechanische Schaltkreise zu beschreiben und zu optimieren.

Allen Newell (1927–1992), Herbert Simon (1916-2001) und Alan Robinson (1930–) entwickeln 1950-1960 die ersten Systeme für die Automatisierung des logischen Schließens als Werkzeug der Künstlichen Intelligenz.

Anwendungen in der Informatik (I)

- Schaltkreisentwurf: Schaltkreise lassen sich durch logische Formeln darstellen $\leadsto$ Entwurf und Optimierung von Schaltungen
- Modellierung und Spezifikation: Eindeutige Beschreibung von komplexen Systemen
- Verifikation: Beweisen, daß ein Programm das gewünschte Verhalten zeigt
- Datenbanken: Formulierung von Anfragen an Datenbanken $\leadsto$ Abfragesprache SQL (Structured query language)
- Künstliche Intelligenz:
  - Planung
  - Mensch-Maschine Kommunikation
  - Theorembeweiser: Der Computer beweist mathematische Sätze $\leadsto$ automatischer Beweis von wichtigen Sätzen im Bereich der Booleschen Algebren
- Logische Programmiersprachen: PROLOG

Anwendungen in der Informatik (II)

Außerdem: Logik ist ein Paradebeispiel für Syntax und formale Semantik

Ein Zitat von Edsger W. Dijkstra:

> *Informatik = VLSAL (Very large scale application of logics)*

Formale Syntax und Semantik

Auch wenn die Beispiele bisher mit natürlicher Sprache beschrieben wurden, werden wir in der Vorlesung meist auf natürliche Sprache verzichten.

Beispiele:

Natürliche Sprache	Formalisierung
Es regnet und die Straße ist naß.	$R \wedge N$
Wenn es regnet, dann ist die Straße naß.	$R \rightarrow N$
Für jede natürliche Zahl x gilt, daß es eine natürliche Zahl y gibt, so daß x kleiner als y ist.	$\forall x \exists y (x < y)$

Frage: Warum nicht natürliche Sprache?

Probleme mit natürlicher Sprache (I)

Problem: Zuordnung von Wahrheitswerten zu natürlichsprachigen Aussagen ist problematisch.

Beispiele:

- Ich habe nur ein bisschen getrunken.
- Sie hat sich in Rauch aufgelöst.
- Das gibt es doch nicht!
- Rache ist süß.

Probleme mit natürlicher Sprache (II)

Problem: Natürliche Sprache ist oft schwer verständlich.

Beispiel: Auszug aus der "Analytica Priora" von Aristoteles

Die Aussage: Wenn der Mittelbegriff sich universell auf Ober- oder Untersatz bezieht, muss ein bestimmter negativer Syllogismus resultieren, immer wenn der Mittelbegriff sich universell auf den Obersatz bezieht, sei es positiv oder negativ, und besonders wenn er sich auf den Untersatz bezieht und umgekehrt zur universellen Aussage.

Der Beweis: Denn wenn M zu keinem N gehört, aber zu einem O, ist es notwendig, dass N zu einem O nicht gehört. Denn da die negative Aussage umsetzbar ist, wird N zu keinem M gehören: Aber es war erlaubt, dass M zu einem O gehört: Deshalb wird N zu einem O nicht gehören: Denn das Ergebnis wird durch die erste Figur erreicht. Noch einmal: Wenn M zu allen N gehört, aber nicht zu einem O, ist es notwendig, dass N nicht zu einem O gehört: Denn wenn N zu allen O gehört und M auch alle N-Eigenschaften zugeschrieben werden, muss M zu allen O gehören: Aber wir haben angenommen, dass M zu einem O nicht gehört. Und wenn M zu allen N gehört, aber nicht zu allen O, können wir folgern, dass N nicht zu allen O gehört: Der Beweis ist der gleiche wie der obige. Aber wenn M alle O-Eigenschaften zugeschrieben werden, aber nicht alle N-Eigenschaften, wird es keinen Syllogismus geben.

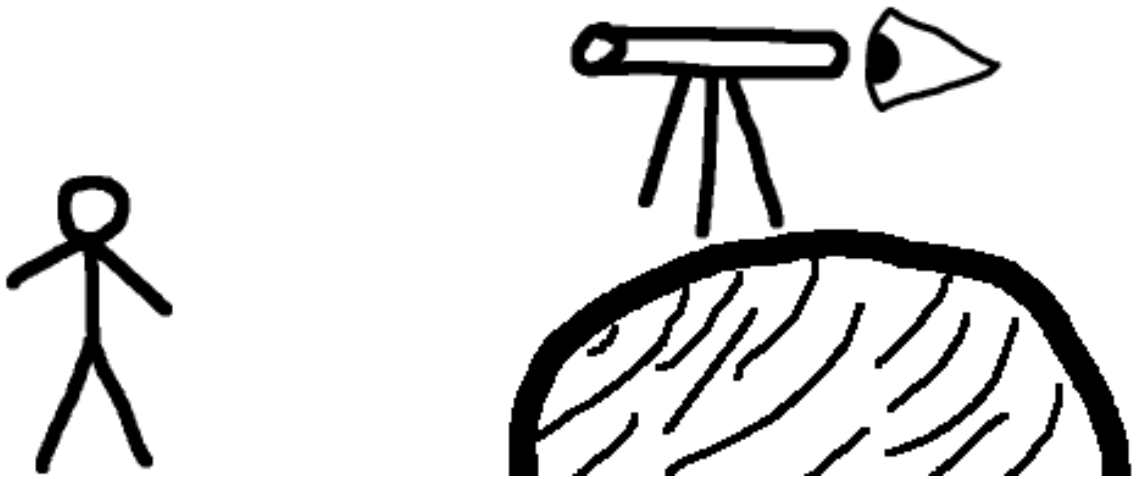
Probleme mit natürlicher Sprache (III)

Problem: Natürliche Sprache ist mehrdeutig.

Beispiel:

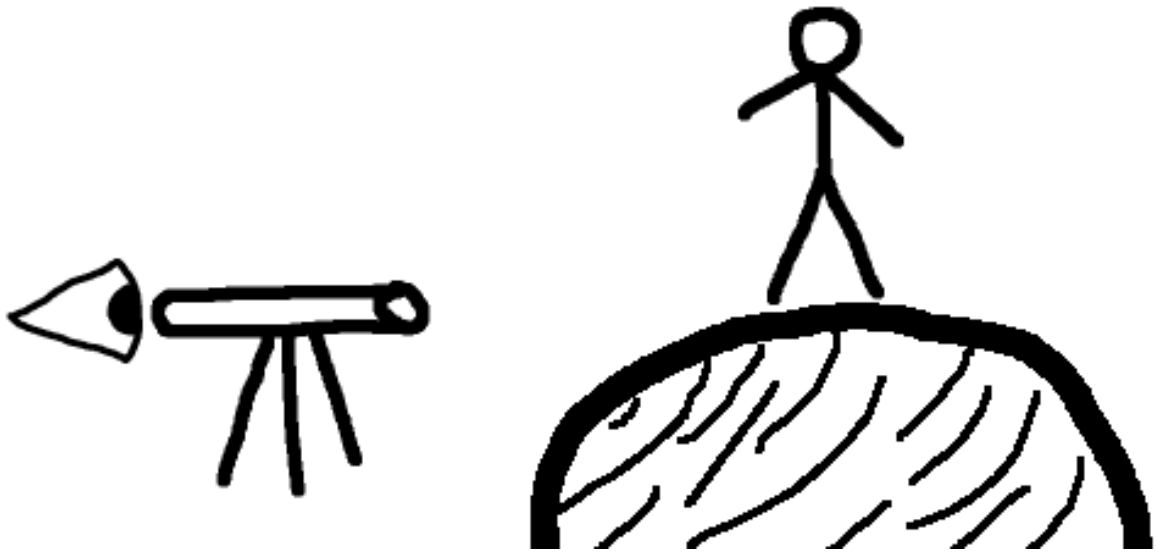
Ich sah den Mann auf dem Berg mit dem Fernrohr.

Ich sah den Mann . . .



(((Ich sah den Mann) auf dem Berg) mit dem Fernrohr)

Ich sah den Mann . . .

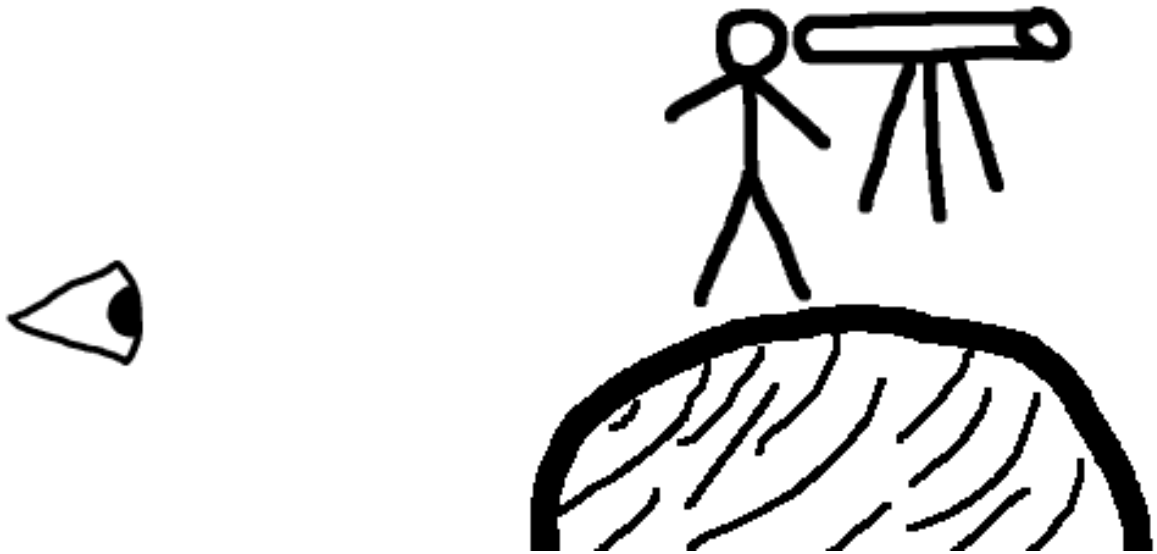


((Ich sah (den Mann auf dem Berg)) mit dem Fernrohr)

Ich sah den Mann . . .

((Ich sah den Mann) (auf dem Berg mit dem Fernrohr))

Ich sah den Mann . . .



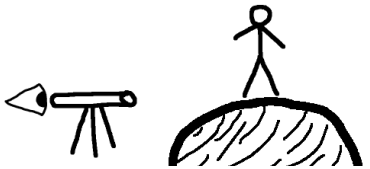
(Ich sah ((den Mann auf dem Berg) mit dem Fernrohr))

Ich sah den Mann . . .

(Ich sah (den Mann (auf dem Berg mit dem Fernrohr)))

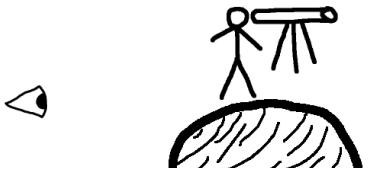
Ich sah den Mann . . .

(((Ich sah den Mann) auf dem Berg) mit dem Fernrohr)



((Ich sah (den Mann auf dem Berg)) mit dem Fernrohr)

((Ich sah den Mann) (auf dem Berg mit dem Fernrohr))



(Ich sah ((den Mann auf dem Berg) mit dem Fernrohr))

(Ich sah (den Mann (auf dem Berg mit dem Fernrohr)))

5 mögliche Interpretationen

Probleme mit natürlicher Sprache (III)

Problem: Natürliche Sprache ist nicht "kontext-frei".

Die Beatles sind Musiker
Paul McCartney ist ein Beatle

Paul McCartney ist ein Musiker

Die Beatles sind vier
Paul McCartney ist ein Beatle

Paul McCartney ist vier

Mengen, Teilmengen

Menge: Menge M von Elementen, wird beschrieben als Aufzählung

$$M = \{A_1, A_2, A_3, A_7\}$$

oder als Menge von Elementen mit einer bestimmten Eigenschaft

$$M = \{A_i \mid 1 \leq i \leq 3 \text{ oder } i = 7\}.$$

Element einer Menge: Wir schreiben $a \in M$, falls ein Element a in der Menge M enthalten ist.

Teilmengenbeziehung: Wir schreiben $A \subseteq B$, falls jedes Element von A auch in B enthalten ist. Die Relation $\subseteq$ heißt auch Inklusion.

Funktionen

Funktion:

$$\begin{aligned} f : A &\to B \\ a &\mapsto f(a) \end{aligned}$$

Die Funktion f bildet ein Element $a \in A$ auf ein Element $f(a) \in B$ ab.

Beispiel:

$$f : \{A_1, A_2, A_3, A_7\} \to \{0, 1\}$$

$$A_1 \mapsto 0,\ A_2 \mapsto 1,\ A_3 \mapsto 0,\ A_7 \mapsto 1$$

Alternativ: $f(A_1) = 0,\ f(A_2) = 1,\ f(A_3) = 0,\ f(A_7) = 1$

Syntax der Aussagenlogik

Eine atomare Formel hat die Form A_i (wobei $i = 1, 2, 3, \ldots$).
Formeln werden durch folgenden induktiven Prozeß definiert:

1. Alle atomaren Formeln sind Formeln
2. Falls F und G Formeln sind, sind auch $(F \wedge G)$ und $(F \vee G)$ Formeln.
3. Falls F eine Formel ist, ist auch $\neg F$ eine Formel.

Sprechweise:

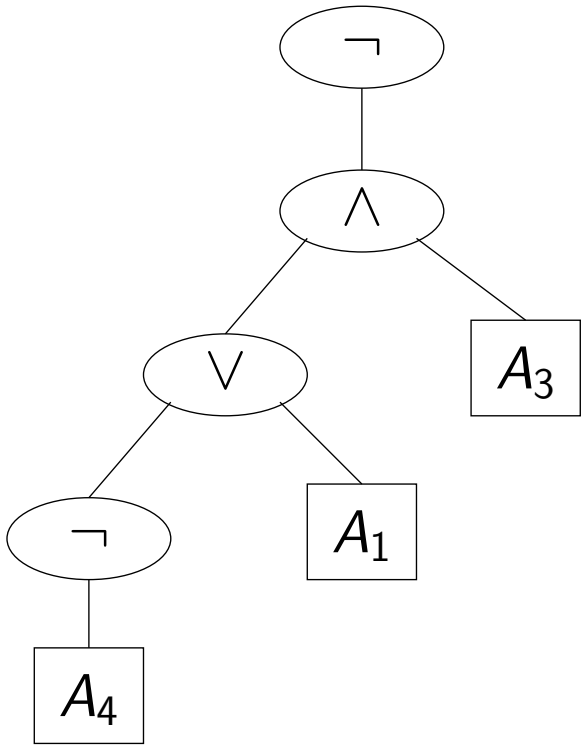
- $(F \wedge G)$: F und G, Konjunktion von F und G
- $(F \vee G)$: F oder G, Disjunktion von F und G
- $\neg F$: nicht F, Negation von F

Beispiel: $\neg((\neg A_4 \vee A_1) \wedge A_3)$ ist eine Formel.

Formel als Syntaxbaum

Jede Formel kann auch durch einen Syntaxbaum dargestellt werden.

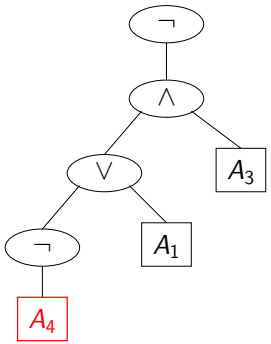
Beispiel: $F = \neg((\neg A_4 \vee A_1) \wedge A_3)$



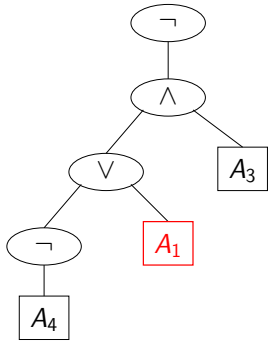
Teilformel

Die Teilformeln einer Formel F entsprechen dann den Teilbäumen.

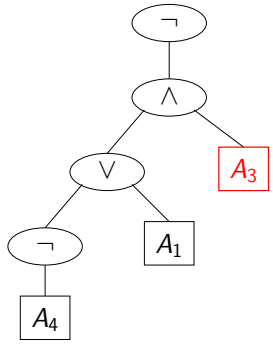
A_4



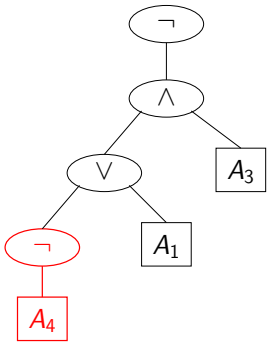
A_1



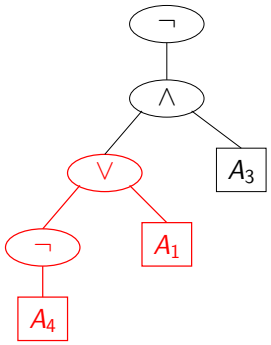
A_3



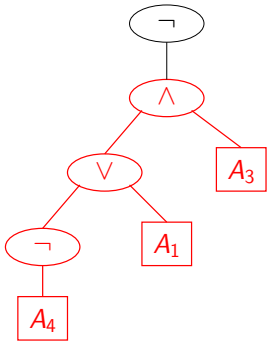
$\neg A_4$



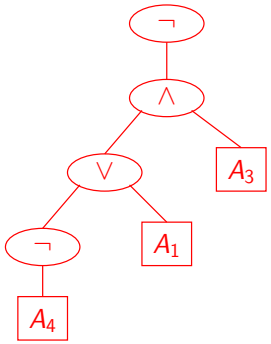
$(\neg A_4 \vee A_1)$



$((\neg A_4 \vee A_1) \wedge A_3)$



$\neg((\neg A_4 \vee A_1) \wedge A_3)$



Semantik der Aussagenlogik: Belegungen

Die Elemente der Menge $\{0, 1\}$ heißen Wahrheitswerte.

Eine Belegung ist eine Funktion $\mathcal{B}\colon D \to \{0, 1\}$, wobei $D \subseteq \{A_1, A_2, A_3, \ldots\}$ eine Teilmenge der atomaren Formeln ist.

Auf der nächsten Folie erweitern wir $\mathcal{B}$ zu einer Funktion $\widehat{\mathcal{B}}\colon E \to \{0, 1\}$, wobei $E \supseteq D$ die Menge aller Formeln ist, die nur aus den atomaren Formeln in D aufgebaut sind.

Beispiel: Sei $D = \{A_1, A_5, A_8\}$.

Dann gilt $F = \neg((\neg A_5 \vee A_1) \wedge A_8) \in E$ aber $\neg((\neg A_4 \vee A_1) \wedge A_3) \notin E$.

Ein mögliche Wahrheitsbelegung könnte definiert werden durch:
$\mathcal{B}(A_1) = 1$, $\mathcal{B}(A_5) = 0$, $\mathcal{B}(A_8) = 1$.

Frage: Was ist wohl $\widehat{\mathcal{B}}(F)$?

Semantik der Aussagenlogik: Belegungen

$$
\begin{aligned}
\widehat{\mathcal{B}}(A) &= \mathcal{B}(A) \quad \text{falls } A \in D \text{ eine atomare Formel ist} \\
\widehat{\mathcal{B}}((F \wedge G)) &= \begin{cases} 1 & \text{falls } \widehat{\mathcal{B}}(F) = 1 \text{ und } \widehat{\mathcal{B}}(G) = 1 \\ 0 & \text{sonst} \end{cases} \\
\widehat{\mathcal{B}}((F \vee G)) &= \begin{cases} 1 & \text{falls } \widehat{\mathcal{B}}(F) = 1 \text{ oder } \widehat{\mathcal{B}}(G) = 1 \\ 0 & \text{sonst} \end{cases} \\
\widehat{\mathcal{B}}(\neg F) &= \begin{cases} 1 & \text{falls } \widehat{\mathcal{B}}(F) = 0 \\ 0 & \text{sonst} \end{cases}
\end{aligned}
$$

Wir schreiben im folgenden $\mathcal{B}$ anstatt $\widehat{\mathcal{B}}$.

Verknüpfungstafeln für $\wedge$, $\vee$, und $\neg$

Berechnung von $\mathcal{B}$ mit Hilfe von Verknüpfungstafeln, auch Wahrheitstafeln genannt.

Beobachtung: Der Wert $\mathcal{B}(F)$ hängt nur davon ab, wie $\mathcal{B}$ auf den den in F vorkommenden atomaren Formeln definiert ist.

Tafeln für die Operatoren $\vee$, $\wedge$, $\neg$:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

A	$\neg A$
0	1
1	0

Abkürzungen

A, B, C oder
P, Q, R oder ... statt $A_1, A_2, A_3 \ldots$

$(F_1 \to F_2)$ statt $(\neg F_1 \vee F_2)$

$(F_1 \leftrightarrow F_2)$ statt $((F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2))$

$\left(\bigvee_{i=1}^{n} F_i\right)$ statt $(\ldots((F_1 \vee F_2) \vee F_3) \vee \ldots \vee F_n)$

$\left(\bigwedge_{i=1}^{n} F_i\right)$ statt $(\ldots((F_1 \wedge F_2) \wedge F_3) \wedge \ldots \wedge F_n)$

Verknüpfungstafeln für $\rightarrow$ und $\leftrightarrow$

Verknüpfungstafeln für die Operatoren $\rightarrow$, $\leftrightarrow$:

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Name: *Implikation*

Interpretation: Wenn A gilt, dann muß auch B gelten.

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Name: *Äquivalenz*

Interpretation: A gilt genau dann, wenn B gilt.

Achtung!!!

$A \to B$ sagt nicht, dass A eine Ursache für B ist.

> "Pinguine schwimmen $\to$ Hunde bellen"
> ist wahr (in unserer Welt).

$A \to B$ sagt nichts darüber, ob A wahr oder falsch ist.

> "Herr A ist bestechlich $\to$ Herr A gehört hinter Gitter"
> ist wahr (in unserer Welt).

Eine falsche Aussage impliziert alles.

> "Pinguine fliegen $\to$ Katzen bellen"
> ist wahr (in unserer Welt).

Formalisierung natürlicher Sprache (I)

Ein Gerät besteht aus einem Bauteil A, einem Bauteil B und einem roten Licht. Folgendes ist bekannt:

- Bauteil A oder Bauteil B (oder beide) sind kaputt.
- Wenn Bauteil A kaputt ist, dann ist auch Bauteil B kaputt.
- Wenn Bauteil B kaputt ist und das rote Licht leuchtet, dann ist Bauteil A nicht kaputt.
- Das rote Licht leuchtet.

Formalisieren Sie diese Situation als aussagenlogische Formel und stellen Sie die Wahrheitstafel zu dieser Formel auf. Verwenden Sie dazu folgende atomare Formeln: RL (rotes Licht leuchtet), AK (Bauteil A kaputt), BK (Bauteil B kaputt)

Formalisierung natürlicher Sprache (II)

Gesamte Wahrheitstafel:

RL	AK	BK	$(AK \vee BK) \wedge (AK \rightarrow BK) \wedge ((BK \wedge RL) \rightarrow \neg AK) \wedge RL$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Formalisierung von Sudoku

Formalisieren Sie das Sudoku-Problem:

4				9				2
		1				5		
	9		3	4	5		1	
		8				2	5	
7		5		3		4	6	1
	4	6				9		8
	6		1	5	9		8	
		9				6		
5				7				4

Verwenden Sie dazu eine atomare Formel $A[n, x, y]$ für jedes Tripel $(n, x, y) \in \{1, \ldots, 9\}^3$:

$A[n, x, y] = 1$, falls: Auf der Zeile x, Spalte y liegt die Zahl n.

Formalisierung von Sudoku

Beispiel: In der erste Zeile stehen alle Zahlen von 1 bis 9

$$\bigwedge_{n=1}^{9} \left(\bigvee_{y=1}^{9} A[n, 1, y] \right)$$

Die Wahrheitstabelle hat

$$2^{729} = 28240139587082174969491088422046278633513539118515775246834019308626938303611984999058739209952299969708978654982839965781232968658783909476265530884869461064307960914827161205726320724927035277237573594788345303657349 12$$

Zeilen. Warum?

Modelle

Sei F eine Formel und $\mathcal{B}$ eine Belegung.
Falls $\mathcal{B}$ für alle in F vorkommenden atomaren Formeln definiert ist so heißt $\mathcal{B}$ zu F passend.

Sei $\mathcal{B}$ passend zu F:

Falls $\mathcal{B}(F) = 1$	so schreiben wir	$\mathcal{B} \models F$
	und sagen	F gilt unter $\mathcal{B}$
	oder	$\mathcal{B}$ ist ein Modell für F
Falls $\mathcal{B}(F) = 0$	so schreiben wir	$\mathcal{B} \not\models F$
	und sagen	F gilt nicht unter $\mathcal{B}$
	oder	$\mathcal{B}$ ist kein Modell für F

Gültigkeit und Erfüllbarkeit

Erfüllbarkeit: Eine Formel F heißt erfüllbar, falls F mindestens ein Modell besitzt, andernfalls heißt F unerfüllbar.

Eine (endliche oder unendliche!) Menge von Formeln M heißt erfüllbar, falls es eine Belegung gibt, die für jede Formel in M ein Modell ist.

Gültigkeit: Eine Formel F heißt gültig (oder allgemeingültig oder Tautologie) falls jede zu F passende Belegung ein Modell für F ist. Wir schreiben $\models F$, falls F gültig ist, und $\not\models F$ sonst.

Aufgabe

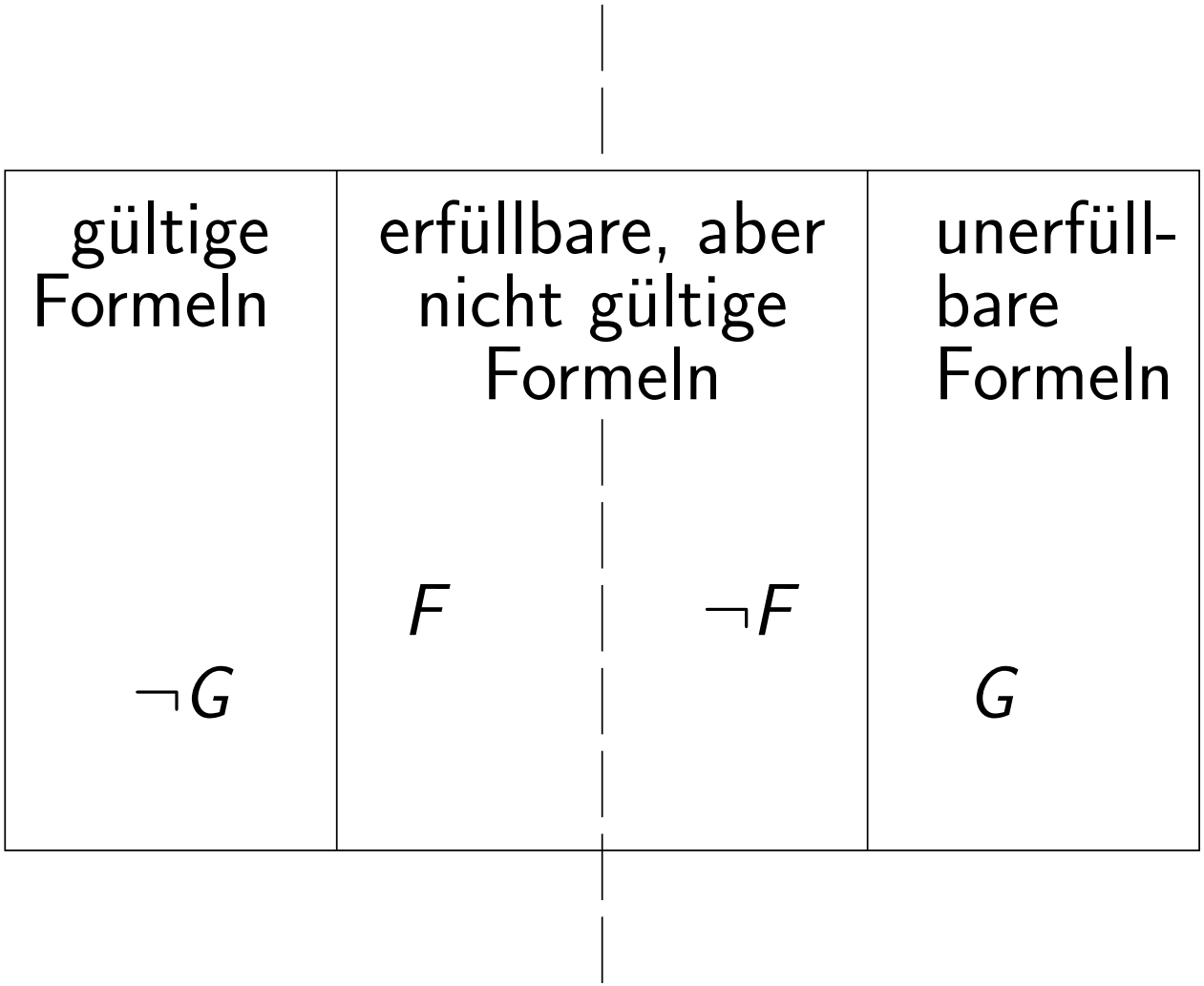
	Gültig	Erfüllbar	Unerfüllbar
A			
$A \vee B$			
$A \vee \neg A$			
$A \wedge \neg A$			
$A \rightarrow \neg A$			
$A \rightarrow B$			
$A \rightarrow (B \rightarrow A)$			
$A \rightarrow (A \rightarrow B)$			
$A \leftrightarrow \neg A$			

Aufgabe

Gelten die folgenden Aussagen?

				J/N	Gegenb.
Wenn	F gültig,	dann	F erfüllbar		
Wenn	F erfüllbar,	dann	$\neg F$ unerfüllbar		
Wenn	F gültig,	dann	$\neg F$ unerfüllbar		
Wenn	F unerfüllbar,	dann	$\neg F$ gültig		

Spiegelungsprinzip



Ein Gültigkeitstest

Wie kann man überprüfen, ob eine Formel F gültig ist?

Eine Möglichkeit: Wahrheitstafel aufstellen

Angenommen, die Formel F enthält n verschiedene atomare Formeln. Wie groß ist die Wahrheitstafel?

Anzahl Zeilen in der Wahrheitstafel: 2^n

Geht es auch effizienter?

Wahrscheinlich nicht: Erfüllbarkeit von aussagenlogischen Formeln ist NP-vollständig und damit nicht in polynomieller Zeit möglich, es sei denn P = NP (siehe Vorlesung Komplexitätstheorie).

Folgerung

Eine Formel G heißt eine Folgerung der Formeln $F_1, \ldots, F_k$ falls für jede Belegung $\mathcal{B}$, die sowohl zu $F_1, \ldots, F_k$ als auch zu G passend ist, gilt:

> *Wenn $\mathcal{B}$ Modell von $\{F_1, \ldots, F_k\}$ ist (d.h. Modell von F_1 und Modell von F_2 und ... und Modell von F_k), dann ist $\mathcal{B}$ auch Modell von G.*

Wir schreiben $F_1, \ldots, F_k \models G$, falls G eine Folgerung von $F_1, \ldots, F_k$ ist.

Folgerung: Beispiel

$$(AK \vee BK), (AK \rightarrow BK),$$
$$((BK \wedge RL) \rightarrow \neg AK), RL \quad \models \quad (RL \wedge \neg AK) \wedge BK$$

Wenn Bauteil A oder Bauteil B kaputt ist *und* daraus, daß Bauteil A kaputt ist, immer folgt, daß Bauteil B kaputt ist *und* ...

... dann kann man die Folgerung ziehen: das rote Licht leuchtet, Bauteil A ist nicht kaputt und Bauteil B ist kaputt.

Aufgabe

M	F	Gilt $M \models F$?
A	$A \vee B$	
A	$A \wedge B$	
A, B	$A \vee B$	
A, B	$A \wedge B$	
$A \wedge B$	A	
$A \vee B$	A	
$A, A \rightarrow B$	B	

Folgerung, Gültigkeit und Unerfüllbarkeit

Theorem

Folgende Aussagen sind äquivalent:

1. $F_1, \ldots, F_k \models G$, *d.h., G ist eine Folgerung von* $F_1, \ldots, F_k$.
2. $((\bigwedge_{i=1}^{k} F_i) \rightarrow G)$ *ist gültig.*
3. $((\bigwedge_{i=1}^{k} F_i) \wedge \neg G)$ *ist unerfüllbar.*

Folgerung, Gültigkeit und Unerfüllbarkeit

Beweis:

1 $\Rightarrow$ **2**: Gelte $F_1, \ldots, F_k \models G$.

Behauptung: $((\bigwedge_{i=1}^{k} F_i) \rightarrow G)$ is gültig.

Sei $\mathcal{B}$ eine beliebige zu $((\bigwedge_{i=1}^{k} F_i) \rightarrow G)$ passende Belegung.

1.Fall: Es gibt ein $i \in \{1, \ldots, k\}$ mit $\mathcal{B}(F_i) = 0$:

Dann gilt auch $\mathcal{B}(\bigwedge_{i=1}^{k} F_i) = 0$ und somit $\mathcal{B}((\bigwedge_{i=1}^{k} F_i) \rightarrow G) = 1$.

2.Fall: Für alle $i \in \{1, \ldots, k\}$ gilt $\mathcal{B}(F_i) = 1$:

Aus $F_1, \ldots, F_k \models G$ folgt $\mathcal{B}(G) = 1$ und somit auch
$\mathcal{B}((\bigwedge_{i=1}^{k} F_i) \rightarrow G) = 1$.

Folgerung, Gültigkeit und Unerfüllbarkeit

2 $\Rightarrow$ 3: Sei $((\bigwedge_{i=1}^{k} F_i) \rightarrow G)$ gültig.

Behauptung: $((\bigwedge_{i=1}^{k} F_i) \wedge \neg G)$ ist unerfüllbar.

Sei $\mathcal{B}$ eine beliebige Belegung.

1.Fall: $\mathcal{B}(G) = 1$:

Dann gilt $\mathcal{B}((\bigwedge_{i=1}^{k} F_i) \wedge \neg G) = 0$.

2.Fall: $\mathcal{B}(\bigwedge_{i=1}^{k} F_i) = 0$:

Dann gilt wieder $\mathcal{B}((\bigwedge_{i=1}^{k} F_i) \wedge \neg G) = 0$.

3.Fall: $\mathcal{B}(\bigwedge_{i=1}^{k} F_i) = 1$ und $\mathcal{B}(G) = 0$:

Dann gilt $\mathcal{B}((\bigwedge_{i=1}^{k} F_i) \rightarrow G) = 0$, dies widerspricht jedoch der Tatsache, dass $((\bigwedge_{i=1}^{k} F_i) \rightarrow G)$ gültig ist.

Also kann Fall 3 nicht eintreten.

Folgerung, Gültigkeit und Unerfüllbarkeit

3 $\Rightarrow$ **1**: Sei $((\bigwedge_{i=1}^{k} F_i) \wedge \neg G)$ unerfüllbar.

Behauptung: $F_1, \ldots, F_k \models G$

Sei $\mathcal{B}$ eine beliebige Belegung mit $\mathcal{B}(F_i) = 1$ für alle $i \in \{1, \ldots, k\}$.

Da $((\bigwedge_{i=1}^{k} F_i) \wedge \neg G)$ unerfüllbar ist, muss $\mathcal{B}(G) = 1$ gelten (sonst wäre $\mathcal{B}((\bigwedge_{i=1}^{k} F_i) \wedge \neg G) = 1$).

Äquivalenz

Zwei Formeln F und G heißen (semantisch) äquivalent, falls für alle Belegungen $\mathcal{B}$, die sowohl für F als auch für G passend sind, gilt $\mathcal{B}(F) = \mathcal{B}(G)$. Hierfür schreiben wir $F \equiv G$.

Aufgabe

Gelten die folgenden Äquivalenzen?

$$
\begin{aligned}
(A \wedge (A \vee B)) &\equiv A \\
\neg(A \vee B) &\equiv (\neg A \wedge \neg B) \\
(A \wedge (B \vee C)) &\equiv ((A \wedge B) \vee C) \\
(A \wedge (B \vee C)) &\equiv ((A \wedge B) \vee (A \wedge C)) \\
(A \to B) \to C &\equiv A \to (B \to C) \\
(A \to B) \to C &\equiv (A \wedge B) \to C \\
(A \leftrightarrow B) \leftrightarrow C &\equiv A \leftrightarrow (B \leftrightarrow C)
\end{aligned}
$$

Die Hauptprobleme der Aussagenlogik

In der "informatischen" Aussagenlogik sucht man nach Verfahren, die folgende Aufgaben (Probleme) lösen:

- Modellprüfung
  Sei F eine Formel und sei $\mathcal{B}$ eine passende Belegung. Gilt $\mathcal{B}(F) = 1$?
- Erfüllbarkeit
  Sei F eine Formel. Ist F erfüllbar ?
- Gültigkeit
  Sei F eine Formel. Ist F gültig ?
- Folgerung
  Seien F und G Formeln. Gilt $F \models G$?
- Äquivalenz
  Seien F und G Formeln. Gilt $F \equiv G$?

Aufgabe

Zeigen Sie, dass die folgenden Aussagen gelten:

Wenn	$(F \to G)$ gültig	dann	$F \models G$.
Wenn	$F \models G$	dann	$(F \to G)$ gültig.
Wenn	$(F \leftrightarrow G)$ gültig	dann	$F \equiv G$.
Wenn	$F \equiv G$	dann	$(F \leftrightarrow G)$ gültig.

Reduktion von Problemen

Welche Probleme lassen sich auf welche reduzieren?

- Gültigkeit $\Longleftrightarrow$ (Nicht)Erfüllbarkeit:
  F gültig genau dann, wenn $\neg F$ nicht erfüllbar
  F erfüllbar genau dann, wenn $\neg F$ nicht gültig.
- Gültigkeit $\Longrightarrow$ Folgerung:
  F gültig genau dann, wenn $T \models F$ (T beliebige gültige Formel).
- Folgerung $\Longrightarrow$ Gültigkeit:
  $F \models G$ genau dann, wenn $F \rightarrow G$ gültig.
- Gültigkeit $\Longrightarrow$ Äquivalenz:
  F gültig genau dann, wenn $F \equiv T$ (T beliebige gültige Formel).
- Äquivalenz $\Longrightarrow$ Gültigkeit:
  $F \equiv G$ genau dann, wenn $F \leftrightarrow G$ gültig.

Einschub: Äquivalenzrelationen

Sei R eine binäre Relation auf der Menge A, d. h. $R \subseteq A \times A$.

- R ist reflexiv, falls für alle $a \in A$ gilt: $(a, a) \in R$.
- R ist symmetrisch, falls für alle $a, b \in A$ gilt:
  Wenn $(a, b) \in R$, dann auch $(b, a) \in R$.
- R ist transitiv, falls für alle $a, b, c \in A$ gilt:
  Wenn $(a, b) \in R$ und $(b, c) \in R$, dann auch $(a, c) \in R$.

Eine reflexive, symmetrische und transitive Relation wird auch als Äquivalenzrelation bezeichnet.

Für eine binäre Relation R schreiben wir im folgenden auch $a\,R\,b$ anstatt $(a, b) \in R$ (Infixschreibweise).

Beispiel: Für eine natürliche Zahl $k \geq 1$ definieren wir die binäre Relation $\equiv_k$ auf $\mathbb{Z}$: $n \equiv_k m$ genau dann, wenn $n - m$ durch k teilbar ist.

Übung: Zeigen Sie, dass $\equiv_k$ für jedes $k \geq 1$ eine Äquivalenzrelation ist.

Einschub: Kongruenzrelationen

Sei f ein n-stelliger Operator auf A, d. h. $f : A^n \to A$, wobei $A^n = \{(a_1, \ldots, a_n) \mid a_1, \ldots, a_n \in A\}$.

Die binäre Relation $R \subseteq A \times A$ ist abgeschlossen unter dem Operator f, falls gilt:

Für alle $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in A^n$ gilt:
Wenn $a_1 \, R \, b_1$ und $\ldots$ $a_n \, R \, b_n$, dann auch $f(a_1, \ldots, a_n) \, R \, f(b_1, \ldots, b_n)$.

Man sagt auch, dass R und f verträglich sind.

Seien $f_1, \ldots, f_n$ Operatoren auf A (beliebiger Stelligkeit).
R ist eine Kongurenzrelation auf A (bezüglich $f_1, \ldots, f_n$), falls gilt:

- R ist eine Äquivalenzrelationen.
- R ist abgeschlossen unter $f_1, \ldots, f_n$.

Beispiel: $\equiv_k$ ist eine Kongruenzrelation auf $\mathbb{Z}$ bezüglich der 2-stelligen Operatoren $+$ und $\cdot$ (mal).

Äquivalenz ist eine Kongurenzrelation

Die Äquivalenz $\equiv$ von Formeln ist eine binäre Relation auf der Menge aller Formeln: Sei $\mathcal{F}$ die Menge aller Formeln. Dann gilt $\equiv \subseteq \mathcal{F} \times \mathcal{F}$.

$\wedge$ und $\vee$ sind 2-stellige Operatoren auf $\mathcal{F}$.
$\neg$ ist ein 1-stelliger Operator auf $\mathcal{F}$.

Die Äquivalenz $\equiv$ ist eine Kongruenzrelation auf der Menge aller Formeln (bezüglich der Operatoren $\wedge, \vee$ und $\neg$):

reflexiv: Es gilt $F \equiv F$ für jede Formel F (jede Formel ist zu sich selbst äquivalent)

symmetrisch: Falls $F \equiv G$ gilt, so gilt auch $G \equiv F$

transitiv: Falls $F \equiv G$ und $G \equiv H$ gilt, so gilt auch $F \equiv H$

abgeschlossen unter Operatoren: Falls $F_1 \equiv F_2$ und $G_1 \equiv G_2$ gilt, so gilt auch $(F_1 \wedge G_1) \equiv (F_2 \wedge G_2)$, $(F_1 \vee G_1) \equiv (F_2 \vee G_2)$ und $\neg F_1 \equiv \neg F_2$.

Ersetzbarkeitstheorem

Die Abgeschlossenheit läßt sich auch folgendermaßen formulieren:

Ersetzbarkeitstheorem

Seien F und G äquivalente Formeln. Sei H eine Formel mit (mindestens) einem Vorkommen der Teilformel F. Dann ist H äquivalent zu H', wobei H' aus H hervorgeht, indem (irgend-) ein Vorkommen von F in H durch G ersetzt wird.

Beweis des Ersetzbarkeitstheorems

Beweis (durch Induktion über den Formelaufbau von H):

Induktionsanfang: Falls H eine atomare Formel ist, dann kann nur $H = F$ sein. Und damit ist klar, daß H äquivalent zu H' ist, denn $H' = G$.

Induktionsschritt: Falls F gerade H selbst ist, so trifft dieselbe Argumentation wie im Induktionsanfang zu.

Nehmen wir also an, dass F eine Teilformel von H mit $F \neq H$ ist. Dann müssen wir drei Fälle unterscheiden.

Beweis des Ersetzbarkeitstheorems

Fall 1: H hat die Bauart $H = \neg H_1$.
Nach Induktionsvoraussetzung ist H_1 äquivalent zu H_1', wobei H_1' aus H_1 durch Ersetzung von F durch G hervorgeht.
Nun ist aber $H' = \neg H_1'$.
Aus der (semantischen) Definition von „$\neg$“ folgt dann, daß H und H' äquivalent sind.

Fall 2: H hat die Bauart $H = (H_1 \vee H_2)$.

Dann kommt F entweder in H_1 oder H_2 vor. Nehmen wir den ersteren Fall an (der zweite ist völlig analog).
Dann ist nach Induktionssannahme H_1 wieder äquivalent zu H_1', wobei H_1' aus H_1 durch Ersetzung von F durch G hervorgeht.
Mit der Definition von „$\vee$“ ist dann klar, daß $H \equiv (H_1' \vee H_2) = H'$.

Fall 3: H hat die Bauart $H = (H_1 \wedge H_2)$.
Diesen Fall beweist man völlig analog zu *Fall 2*.

Äquivalenzen (I)

Satz

Es gelten die folgenden Äquivalenzen:

$$
\begin{array}{rcll}
(F \wedge F) & \equiv & F & \\
(F \vee F) & \equiv & F & \text{(Idempotenz)} \\
 & & & \\
(F \wedge G) & \equiv & (G \wedge F) & \\
(F \vee G) & \equiv & (G \vee F) & \text{(Kommutativität)} \\
 & & & \\
((F \wedge G) \wedge H) & \equiv & (F \wedge (G \wedge H)) & \\
((F \vee G) \vee H) & \equiv & (F \vee (G \vee H)) & \text{(Assoziativität)} \\
 & & & \\
(F \wedge (F \vee G)) & \equiv & F & \\
(F \vee (F \wedge G)) & \equiv & F & \text{(Absorption)} \\
 & & & \\
(F \wedge (G \vee H)) & \equiv & ((F \wedge G) \vee (F \wedge H)) & \\
(F \vee (G \wedge H)) & \equiv & ((F \vee G) \wedge (F \vee H)) & \text{(Distributivität)}
\end{array}
$$

Äquivalenzen (II)

Satz

Es gelten die folgenden Äquivalenzen:

$$\begin{array}{rcll} \neg\neg F & \equiv & F & \text{(Doppelnegation)} \\ \\ \neg(F \wedge G) & \equiv & (\neg F \vee \neg G) & \\ \neg(F \vee G) & \equiv & (\neg F \wedge \neg G) & \text{(deMorgansche Regeln)} \\ \\ (F \vee G) & \equiv & F, \text{ falls } F \text{ Tautologie} & \\ (F \wedge G) & \equiv & G, \text{ falls } F \text{ Tautologie} & \text{(Tautologieregeln)} \\ \\ (F \vee G) & \equiv & G, \text{ falls } F \text{ unerfüllbar} & \\ (F \wedge G) & \equiv & F, \text{ falls } F \text{ unerfüllbar} & \text{(Unerfüllbarkeitsregeln)} \end{array}$$

Beweis: Übung

Normalformen: **KNF**

Definition (Normalformen)
Ein Literal ist eine atomare Formel oder die Negation einer atomaren Formel. Im ersten Fall sprechen wir von einem positiven, im zweiten Fall von einem negativen Literal.

Eine Formel F ist in konjunktiver Normalform (**KNF**), falls sie eine Konjunktion von Disjunktionen von Literalen ist:

$$F = (\bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m_i} L_{i,j})),$$

wobei $L_{i,j} \in \{A_1, A_2, \ldots\} \cup \{\neg A_1, \neg A_2, \ldots\}$

Normalformen: **DNF**

Eine Formel F ist in disjunktiver Normalform (**DNF**), falls sie eine Disjunktion von Konjunktionen von Literalen ist:

$$F = (\bigvee_{i=1}^{n} (\bigwedge_{j=1}^{m_i} L_{i,j})),$$

wobei $L_{i,j} \in \{A_1, A_2, \ldots\} \cup \{\neg A_1, \neg A_2, \ldots\}$

Satz

Zu jeder Formel F existiert eine äquivalente Formel in KNF, sowie eine äquivalente Formel in DNF.

Methode 1: Ablesen aus Wahrheitstafel

Für eine atomare Formel A_i definiere

$$A_i^0 := \neg A_i \quad \text{und} \quad A_i^1 := A_i.$$

Für eine Formel F, in der genau die atomaren Formeln $A_1, \ldots, A_n$ vorkommen, definiere

$$\mathrm{DNF}(F) \quad := \bigvee_{\substack{\mathcal{B}:\{A_1,\ldots,A_n\}\to\{0,1\},\\ \mathcal{B}(F)=1}} \bigwedge_{i=1}^{n} A_i^{\mathcal{B}(A_i)}$$

$$\mathrm{KNF}(F) \quad := \bigwedge_{\substack{\mathcal{B}:\{A_1,\ldots,A_n\}\to\{0,1\},\\ \mathcal{B}(F)=0}} \bigvee_{i=1}^{n} A_i^{1-\mathcal{B}(A_i)}$$

Lemma

Für jede Formel F gilt: $F \equiv \mathrm{DNF}(F) \equiv \mathrm{KNF}(F)$.

Methode 1: Ablesen aus Wahrheitstafel

Beweis: Wir zeigen $F \equiv \mathrm{KNF}(F)$, $F \equiv \mathrm{DNF}(F)$ kann analog gezeigt werden.

Sei $\mathcal{B}'$ eine beliebige Belegung.

Wir zeigen: $\mathcal{B}'(F) = 0$ genau dann, wenn $\mathcal{B}'(\mathrm{KNF}(F)) = 0$.

1. Sei $\mathcal{B}'(F) = 0$.

Es gilt $\mathcal{B}'(\bigvee_{i=1}^{n} A_i^{1-\mathcal{B}'(A_i)}) = 0$ (dies gilt für jede passende Belegung).

Aus

$$\mathrm{KNF}(F) = \bigwedge_{\substack{\mathcal{B}:\{A_1,\ldots,A_n\}\to\{0,1\},\\ \mathcal{B}(F)=0}} \bigvee_{i=1}^{n} A_i^{1-\mathcal{B}(A_i)},$$

folgt $\mathcal{B}'(\mathrm{KNF}(F)) = 0$ (denn $\mathcal{B}'$ ist wegen $\mathcal{B}'(F) = 0$ eine der Belegungen, über die in $\mathrm{KNF}(F)$ außen das große $\bigwedge$ gebildet wird).

Methode 1: Ablesen aus Wahrheitstafel

2. Sei $\mathcal{B}'(\mathrm{KNF}(F)) = 0$.

Aus

$$\mathrm{KNF}(F) = \bigwedge_{\substack{\mathcal{B}:\{A_1,\ldots,A_n\}\to\{0,1\},\\ \mathcal{B}(F)=0}} \bigvee_{i=1}^{n} A_i^{1-\mathcal{B}(A_i)}$$

folgt, dass eine der Disjunktionen in $\mathrm{KNF}(F)$ unter $\mathcal{B}'$ gleich 0 ist.

Es gibt somit eine Belegung $\mathcal{B}$ mit: $\mathcal{B}(F) = 0$ und $\mathcal{B}'(\bigvee_{i=1}^{n} A_i^{1-\mathcal{B}(A_i)}) = 0$.

Also: $\mathcal{B}'(A_i^{1-\mathcal{B}(A_i)}) = 0$ für alle $i \in \{1, \ldots, n\}$.

Dies impliziert $\mathcal{B}'(A_i) = \mathcal{B}(A_i)$ für alle $i \in \{1, \ldots, n\}$ und somit $\mathcal{B}'(F) = 0$ (wegen $\mathcal{B}(F) = 0$). □

Methode 1: Ablesen aus Wahrheitstafel

Beachte:

- Ist F unerfüllbar, d. h. $\mathcal{B}(F) = 0$ für alle passenden Belegungen $\mathcal{B}$, so ist $\mathrm{DNF}(F)$ die leere Disjunktion. Diese soll eine unerfüllbare Formel sein.
- Ist F gültig, d. h. $\mathcal{B}(F) = 1$ für alle passenden Belegungen $\mathcal{B}$, so ist $\mathrm{KNF}(F)$ die leere Konjunktion. Diese soll eine Tautologie sein.

Methode 1: Beispiel

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

DNF: Aus jeder Zeile mit Wahrheitswert 1 wird eine Konjunktion, aus einer 0 in der Spalte A wird $\neg A$, aus einer 1 wird A.

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge C)$$
$$\vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge C)$$

KNF: Aus jeder Zeile mit Wahrheitswert 0 wird eine Disjunktion, aus einer 0 in der Spalte A wird A, aus einer 1 wird $\neg A$.

$$(A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C)$$
$$\wedge (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C)$$

Methode 2: syntaktisches Umformen

Gegeben: eine Formel F.
Wir bilden die KNF von F wie folgt:

1. Ersetze in F jedes Vorkommen einer Teilformel der Bauart

$$\begin{array}{rcl} \neg\neg G & \text{durch} & G \\ \neg(G \wedge H) & \text{durch} & (\neg G \vee \neg H) \\ \neg(G \vee H) & \text{durch} & (\neg G \wedge \neg H) \end{array}$$

   bis keine derartige Teilformel mehr vorkommt.

2. Ersetze jedes Vorkommen einer Teilformel der Bauart

$$\begin{array}{rcl} (F \vee (G \wedge H)) & \text{durch} & ((F \vee G) \wedge (F \vee H)) \\ ((F \wedge G) \vee H) & \text{durch} & ((F \vee H) \wedge (G \vee H)) \end{array}$$

   bis keine derartige Teilformel mehr vorkommt.

Mengendarstellung

Eine Klausel ist eine Disjunktion von Literalen.

Die Klausel $L_1 \vee L_2 \vee \cdots \vee L_n$, wobei $L_1, \ldots, L_n$ Literale sind, wird auch mit der Menge $\{L_1, \ldots, L_n\}$ identifiziert.

Eine Formel in **KNF** (= Konjunktion von Klauseln) wird mit einer Menge von Klauseln (d.h. Menge von Mengen von Literalen) identifiziert:
$(\bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m_i} L_{i,j}))$ wird mit $\{\{L_{i,j} \mid 1 \leq j \leq m_i\} \mid 1 \leq i \leq n\}$ identifiziert.

Die leere Klausel (= leere Disjunktion) ist äquivalent zu einer unerfüllbaren Formel.

Die leere **KNF**-Formel (= leere Konjunktion) ist äquivalent zu einer gültigen Formel.

Präzedenzen

Präzedenz der Operatoren:

$\leftrightarrow$	bindet am schwächsten
$\rightarrow$	...
$\vee$	...
$\wedge$	...
$\neg$	bindet am stärksten

Es gilt also:

$$A \leftrightarrow B \vee \neg C \rightarrow D \wedge \neg E \equiv (A \leftrightarrow ((B \vee \neg C) \rightarrow (D \wedge \neg E)))$$

Dennoch: Zu viele Klammern schaden i.A. nicht.

Erfüllbarkeit und Gültigkeit in DNF und KNF

Erfüllbarkeit ist leicht (lösbar in linearer Zeit) für Formeln in DNF:

> *Eine Formel in DNF ist erfüllbar genau dann, wenn es eine Konjunktion gibt, die nicht gleichzeitig A und $\neg A$ für eine atomare Formel A enthält.*

Erfüllbar: $(\neg B \wedge A \wedge B) \vee (\neg A \wedge C)$
Nicht erfüllbar: $(A \wedge \neg A \wedge B) \vee (C \wedge \neg C)$

Gültigkeit ist leicht (lösbar in linearer Zeit) für Formeln in KNF:

> *Eine Formel in KNF ist gültig genau dann, wenn jede Disjunktion gleichzeitig A und $\neg A$ für eine atomare Formel A enthält. (Oder es handelt sich um die leere Konjunktion.)*

Gültig: $(A \vee \neg A \vee B) \wedge (C \vee \neg C)$
Nicht gültig: $(A \vee \neg A) \wedge (\neg A \vee C)$

Effiziente Erfüllbarkeitstests

Im folgenden:

- Ein sehr effizienter Erfüllbarkeitstest für eine spezielle Klasse von Formeln, sogenannte Hornformeln
- Ein im allgemeinen effizienter Unerfüllbarkeitstest für Formeln in KNF (Resolution)

Hornformel

Eine Formel F ist eine Hornformel (benannt nach Alfred Horn, 1918–2001), falls F in **KNF** ist, und jede Klausel in F höchstens ein positives Literal enthält.

Notation:

$$\begin{array}{rll} (\neg A \vee \neg B \vee C) & \text{wird zu} & (A \wedge B \to C) \\ (\neg A \vee \neg B) & \text{wird zu} & (A \wedge B \to 0) \\ A & \text{wird zu} & (1 \to A) \end{array}$$

0: steht für eine beliebige unerfüllbare Formel

1: steht für eine beliebige gültige Formel

Allgemein:

$$\begin{array}{rcl} \neg A_1 \vee \cdots \vee \neg A_k \vee B & \equiv & \neg(A_1 \wedge \cdots \wedge A_k) \vee B \equiv (A_1 \wedge \cdots \wedge A_k) \to B \\ \neg A_1 \vee \cdots \vee \neg A_k & \equiv & \neg(A_1 \wedge \cdots \wedge A_k) \vee 0 \equiv (A_1 \wedge \cdots \wedge A_k) \to 0 \end{array}$$

$(A_1 \wedge \cdots \wedge A_k) \to B$ (bzw. $(A_1 \wedge \cdots \wedge A_k) \to 0$) ist die implikative Form der Klausel $\neg A_1 \vee \cdots \vee \neg A_k \vee B$ (bzw. $\neg A_1 \vee \cdots \vee \neg A_k$).

Erfüllbarkeitstest für Hornformeln

Markierungsalgorithmus

Eingabe: eine Hornformel F.

(1) Versehe jedes Vorkommen einer atomaren Formel A in F mit einer Markierung, falls es in F eine Teilformel der Form $(1 \to A)$ gibt;

(2) **while** es gibt in F eine Teilformel G der Form $(A_1 \wedge \ldots \wedge A_k \to B)$ oder $(A_1 \wedge \ldots \wedge A_k \to 0)$, $k \geq 1$, wobei $A_1, \ldots, A_k$ bereits markiert sind und B noch nicht markiert ist **do**:

    if G hat die erste Form **then**

        markiere jedes Vorkommen von B

    else gib "unerfüllbar" aus und stoppe;

    endwhile

(3) Gib "erfüllbar" aus und stoppe.

Induktionsprinzip

Um die Aussage

Für jedes $n \in \{0, 1, 2, 3, \ldots\}$ *gilt* $P(n)$.

zu zeigen, gehen wir im allgemeinen folgendermaßen vor:

- Wir zeigen, daß $P(0)$ gilt. (Induktionsanfang)
- Wir zeigen, daß für jedes n gilt:
  Wenn $P(n)$ gilt, dann gilt auch $P(n+1)$. (Induktionsschritt)

Dann kann man schließen, daß $P(n)$ für jedes beliebige n gilt.

Alternatives Induktionsprinzip: Wir zeigen, daß für jedes n gilt:
Wenn $P(k)$ für alle $k < n$ gilt, dann gilt auch $P(n)$.

Anwendung: Beweis, dass eine Bedingung während des Ablaufs eines Algorithmus immer erfüllt ist (Invariante). Hierzu zeigt man durch Induktion, daß die Bedingung nach n Schritten des Algorithmus erfüllt ist.

Korrektheit des Markierungsalgorithmus

Theorem

Der Markierungsalgorithmus ist korrekt und terminiert immer nach spätestens n Markierungschritten.
Dabei ist n die Anzahl der atomaren Formeln in der Eingabeformel F.

Beweis:

(A) Algorithmus terminiert:

Nach spätestens n Schritten sind alle atomare Formeln markiert.

(B) Wenn der Algorithmus eine atomare Formel A markiert, dann gilt $\mathcal{B}(A) = 1$ für jede erfüllende Belegung $\mathcal{B}$ von F.

Beweis von (B) mittels Induktion:

1.Fall: atomare Formel A wird in Schritt (1) markiert: klar

Korrektheit des Markierungsalgorithmus

2.Fall: atomare Formel A wird in Schritt (2) markiert:

Dann gibt es eine Teilformel $(A_1 \wedge \ldots \wedge A_k \rightarrow A)$, so dass $A_1, \ldots, A_k$ zu frühreren Zeitpunkten markiert wurden.

Also gilt $\mathcal{B}(A_1) = \cdots = \mathcal{B}(A_k) = 1$ für jede erfüllende Belegung $\mathcal{B}$ von F.

Dann muss aber auch $\mathcal{B}(A) = 1$ für jede erfüllende Belegung $\mathcal{B}$ von F gelten.

Dies beweist (B).

Korrektheit des Markierungsalgorithmus

(C) Wenn der Algorithmus "unerfüllbar" ausgibt, dann ist F unerfüllbar.

Sei $(A_1 \wedge \ldots \wedge A_k \to 0)$ die Teilformel von F, die die Ausgabe "unerfüllbar" verursacht.

Nach (B) gilt $\mathcal{B}(A_1) = \cdots = \mathcal{B}(A_k) = 1$ für jede erfüllende Belegung $\mathcal{B}$ von F.

Aber für solche Belegungen gilt: $\mathcal{B}(A_1 \wedge \cdots \wedge A_k \to 0) = 0$ und damit $\mathcal{B}(F) = 0$.

Also kann es keine erfüllende Belegung von F geben.

(D) Wenn der Algorithmus "erfüllbar" ausgibt, dann ist F erfüllbar.

Angenommen, der Algorithmus gibt "erfüllbar" aus.

Definiere eine Belegung $\mathcal{B}$ wie folgt:

$$\mathcal{B}(A_i) = \begin{cases} 1 & \text{der Algoritmus markiert } A_i \\ 0 & \text{sonst} \end{cases}$$

Wir behaupten, dass die Belegung $\mathcal{B}$ die Konjunktion F erfüllt:

- In $(A_1 \wedge \ldots \wedge A_k \rightarrow B)$ ist B markiert oder mindestens ein A_i nicht markiert.
- In $(A_1 \wedge \ldots \wedge A_k \rightarrow 0)$ ist mindestens ein A_i nicht markiert (sonst hätte der Algorithmus mit "unerfüllbar" terminiert).

□

Bemerkung: Mit einer geeigneten Implementierung läuft der Algorithmus in linearer Zeit.

Beispiel: MYCIN

MYCIN: Expertensystem zur Untersuchung von Blutinfektionen (entwickelt in den 70er Jahren)

Beispiel:
IF the infection is pimary-bacteremia AND the site of the culture is one of the sterile sites AND the suspected portal of entry is the gastrointestinal tract THEN there is suggestive evidence (0.7) that infection is bacteroid.

Resolution (Idee)

Resolution ist ein Verfahren, mit dem man feststellen kann, ob eine Formel F in **KNF** unerfüllbar ist.

Idee: $(F \vee A) \wedge (F' \vee \neg A) \equiv (F \vee A) \wedge (F' \vee \neg A) \wedge (F \vee F')$

Aus der Herleitung der leeren Disjunktion (= leere Klausel) folgt Unerfüllbarkeit.

Zwei Fragen:

- Kann man aus einer unerfüllbaren Formel immer die leere Klausel herleiten? (Vollständigkeit)
- Gibt es eine Möglichkeit, die Herleitung kompakter aufzuschreiben?

Mengendarstellung

Zur Erinnerung:

- Klausel: Menge von Literalen (Disjunktion).

  $\{A, B\}$ stellt $(A \vee B)$ dar.

- Formel in **KNF**: Menge von Klauseln (Konjunktion von Klauseln).

  $\{\{A, B\}, \{\neg A, B\}\}$ stellt $((A \vee B) \wedge (\neg A \vee B))$ dar.

Die leere Klausel (= leere Disjunktion) ist äquivalent zu einer unerfüllbaren Formel.
Diese wird auch mit $\square$ bezeichnet.

Die leere Formel (= leere Konjunktion) ist äquivalent zu einer gültigen Formel.

Beachte: Die Formel $\{\}$ (= leere Konjunktion = Tautologie) ist zu unterscheiden von der Formel $\{\square\}$ (= unerfüllbare Formel).

Vorteile der Mengendarstellung

Man erhält automatisch:

- Kommutativität:
  $(A \vee B) \equiv (B \vee A)$,
  beide dargestellt durch $\{A, B\}$
- Assoziativität:
  $((A \vee B) \vee C) \equiv (A \vee (B \vee C))$,
  beide dargestellt durch $\{A, B, C\}$
- Idempotenz:
  $(A \vee A) \equiv A$,
  beide dargestellt durch $\{A\}$

Resolvent

Definition: Seien K_1, K_2 und R Klauseln. Dann heißt R Resolvent von K_1 und K_2, falls es ein Literal L gibt mit $L \in K_1$ und $\overline{L} \in K_2$ und R die folgende Form hat:

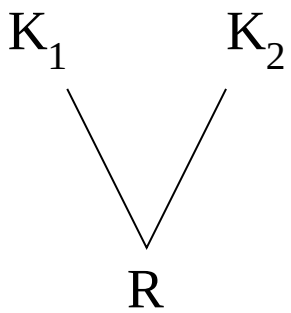
$$R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\}).$$

Hierbei ist $\overline{L}$ definiert als

$$\overline{L} = \begin{cases} \neg A_i & \text{falls } L = A_i \text{ für ein } i \geq 1, \\ A_i & \text{falls } L = \neg A_i \text{ für ein } i \geq 1 \end{cases}$$

Resolvent

Wir stellen diesen Sachverhalt durch folgendes Diagramm dar



Sprechweise: R wird aus K_1, K_2 nach L resolviert.

Ferner: falls $K_1 = \{L\}$ und $K_2 = \{\overline{L}\}$, so entsteht die leere Menge als Resolvent. Diese wird mit dem speziellen Symbol $\square$ bezeichnet, das eine unerfüllbare Formel darstellt.

Resolutionslemma

Resolutionslemma

Sei F eine Formel in **KNF**, dargestellt als Klauselmenge. Ferner sei R ein Resolvent zweier Klauseln K_1 und K_2 in F. Dann sind F und $F \cup \{R\}$ äquivalent.

Beweis: Folgt direkt aus

$$\underbrace{(F_1 \vee A)}_{K_1} \wedge \underbrace{(F_2 \vee \neg A)}_{K_2} \equiv \underbrace{(F_1 \vee A)}_{K_1} \wedge \underbrace{(F_2 \vee \neg A)}_{K_2} \wedge \underbrace{(F_1 \vee F_2)}_{R}$$

Definition von $Res(F)$

Definition: Sei F eine Menge von Klauseln. Dann ist $Res(F)$ definiert als

$$Res(F) = F \cup \{R \mid R \text{ ist Resolvent zweier Klauseln in } F\}.$$

Außerdem setzen wir:

$$\begin{aligned} Res^0(F) &= F \\ Res^{n+1}(F) &= Res(Res^n(F)) \qquad \text{für } n \geq 0 \end{aligned}$$

und schließlich sei

$$Res^*(F) = \bigcup_{n \geq 0} Res^n(F).$$

$Res^*(F)$ wird auch als die Resolutionshülle von F bezeichnet.

Aus dem Resolutionslemma folgt sofort

$$F \equiv Res^*(F).$$

Aufgabe

Angenommen, die Formel F (in **KNF**) enthält n atomare Formeln.
Wie groß kann dann $Res^*(F)$ höchstens werden?

(A) $|Res^*(F)| \leq 2^n$

(B) $|Res^*(F)| \leq 4^n$

(C) $|Res^*(F)|$ kann unendlich werden

Dabei bezeichnet $|Res^*(F)|$ die Anzahl der Elemente in $Res^*(F)$.

Resolutionssatz

Wir zeigen nun die Korrektheit und Vollständigkeit der Resolution:

Resolutionssatz der Aussagenlogik

Eine endliche Menge F von Klauseln ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$.

Beweis:

Korrektheit: Wenn $\square \in Res^*(F)$, dann ist F unerfüllbar.

Sei $\square \in Res^*(F)$.

Aus dem Resolutionslemma folgt $F \equiv Res^*(F)$.

Da $\square$ unerfüllbar ist, ist auch $Res^*(F)$ und somit F unerfüllbar.

Beweis des Resolutionssatz (Vollständigkeit)

Vollständigkeit: Wenn F unerfüllbar ist, dann gilt $\square \in Res^*(F)$.

Sei F im folgenden unerfüllbar.

Wir beweisen die Vollständigkeit durch eine Induktion über die Anzahl $n(F)$ der atomaren Formeln, die in F vorkommen.

Induktionsanfang: $n(F) = 0$. Dann muss $F = \{\square\}$ gelten. Also gilt: $\square \in F \subseteq Res^*(F)$.

Induktionsschritt: Sei $n(F) > 0$.

Wähle eine beliebige in F vorkommende atomare Formel A aus.
Wir definieren aus F die Formel F_0 wie folgt:

$$F_0 = \{K \setminus \{A\} \mid K \in F, \neg A \notin K\}.$$

Intuition: F_0 entsteht aus F indem A durch 0 ersetzt wird, und die "offensichtlichen" Vereinfachungen durchgeführt werden.

Beweis des Resolutionssatz (Vollständigkeit)

Analog definieren wir aus F Formel F_1:

$$F_1 = \{K \setminus \{\neg A\} \mid K \in F, A \notin K\}.$$

Intuition: F_1 entsteht aus F indem A durch 1 ersetzt wird, und die "offensichtlichen" Vereinfachungen durchgeführt werden.

Da F unerfüllbar ist, sind auch F_0 und F_1 unerfüllbar:

Würde z. B. F_0 durch die Belegung $\mathcal{B}$ erfüllt werden, so würde F durch die folgende Belegung $\mathcal{B}'$ erfüllt werden:

$$\mathcal{B}'(A_i) = \begin{cases} 0 & \text{falls } A_i = A \\ \mathcal{B}(A_i) & \text{falls } A_i \neq A \end{cases}$$

Da $n(F_0) = n(F_1) = n(F) - 1$ gilt, können wir aus der Induktionsannahme schließen, dass $\square \in Res^*(F_0)$ und $\square \in Res^*(F_1)$ gilt.

Beweis des Resolutionssatz (Vollständigkeit)

Somit gibt es eine Folge von Klauseln $K_1, K_2, \ldots, K_m$ mit

- $K_m = \square$
- $K_i \in F_0$ oder K_i ist Resolvent von K_j und K_ℓ mit $j, \ell < i$.

Wir definieren nun eine Folge von Klauseln $K'_1, K'_2, \ldots, K'_m$, wobei $K'_i = K_i$ oder $K'_i = K_i \cup \{A\}$, wie folgt:

1. Fall $K_i \in F_0$ und $K_i \in F$: $K'_i := K_i$
2. Fall $K_i \in F_0$ und $K_i \notin F$: $K'_i := K_i \cup \{A\} \in F$
3. Fall $K_i \notin F_0$ und K_i wird aus K_j und K_ℓ $(j, \ell < i)$ nach dem Literal L resolviert:
   K'_i wird aus K'_j und K'_ℓ bebildet, indem nach L resolviert wird.

Dann gilt entweder $K'_m = \square$ oder $K'_m = \{A\}$ und somit

$$\square \in Res^*(F) \quad \text{oder} \quad \{A\} \in Res^*(F)$$

Analog folgt:

$$\square \in Res^*(F) \quad \text{oder} \quad \{\neg A\} \in Res^*(F)$$

Hieraus folgt $\square \in Res^*(F)$. $\square$

Veranschaulichung des Induktionsschritts

$$F = \{\{A_1\}, \{\neg A_2,\ A_4\}, \{\neg A_1, A_2,\ A_4\}, \{A_3, \neg A_4\},\ \{\neg A_1, \neg A_3, \neg A_4\}\ \}$$

Veranschaulichung des Induktionsschritts

$$F = \{\{A_1\}, \{\neg A_2, \cancel{A_4}\}, \{\neg A_1, A_2, \cancel{A_4}\}, \cancel{\{A_3, \neg A_4\}}, \cancel{\{\neg A_1, \neg A_3, \neg A_4\}}\}$$

$$F_0 = \{\{A_1\}, \{\neg A_2\}, \{\neg A_1, A_2\}\}$$

Veranschaulichung des Induktionsschritts

$F = \{\{A_1\}, \{\neg A_2, A_4\}, \{\neg A_1, A_2, A_4\}, \{A_3, \neg A_4\}, \{\neg A_1, \neg A_3, \neg A_4\}\}$

$F_0 = \{\{A_1\}, \{\neg A_2\}, \{\neg A_1, A_2\}\}$

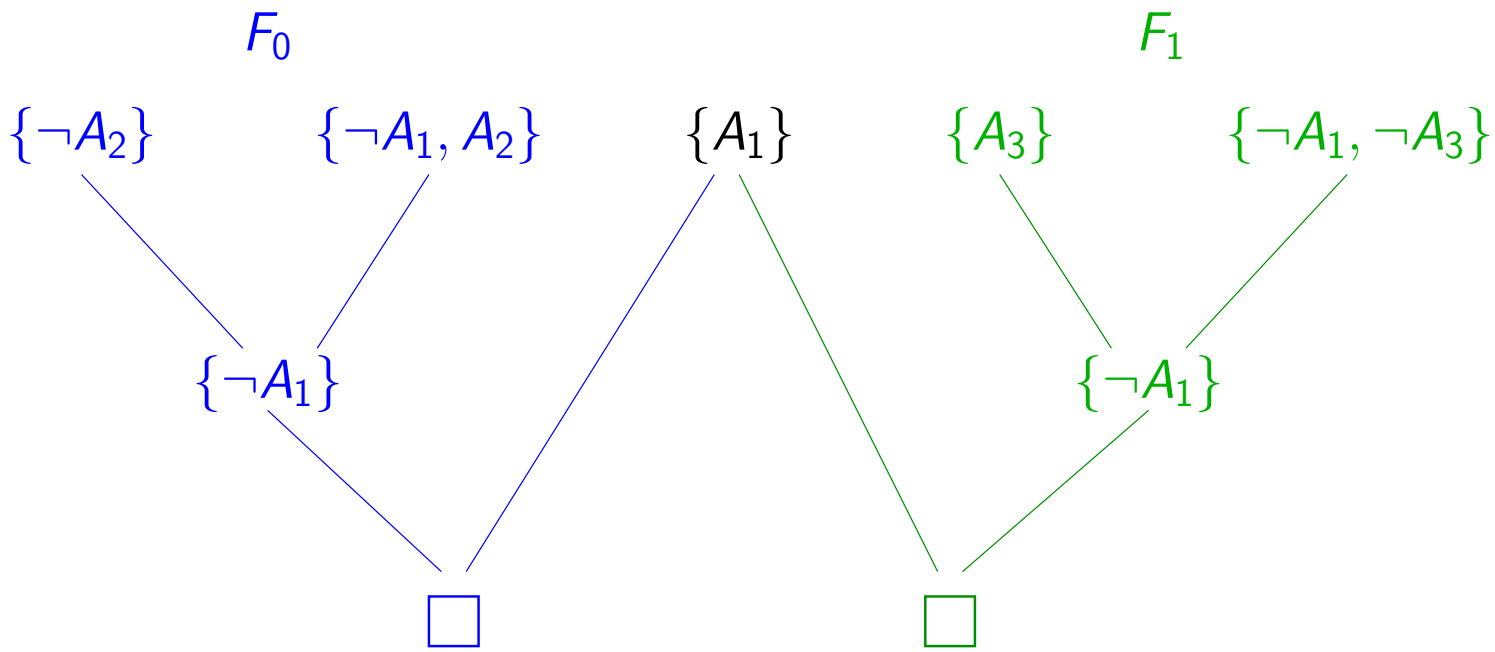
$F_1 = \{\{A_1\}, \{A_3\}, \{\neg A_1, \neg A_3\}\}$

Veranschaulichung des Induktionsschritts

$F = \{\{A_1\}, \{\neg A_2, A_4\}, \{\neg A_1, A_2, A_4\}, \{A_3, \neg A_4\}, \{\neg A_1, \neg A_3, \neg A_4\}\}$

$F_0 = \{\{A_1\}, \{\neg A_2\}, \{\neg A_1, A_2\}\}$

$F_1 = \{\{A_1\}, \{A_3\}, \{\neg A_1, \neg A_3\}\}$

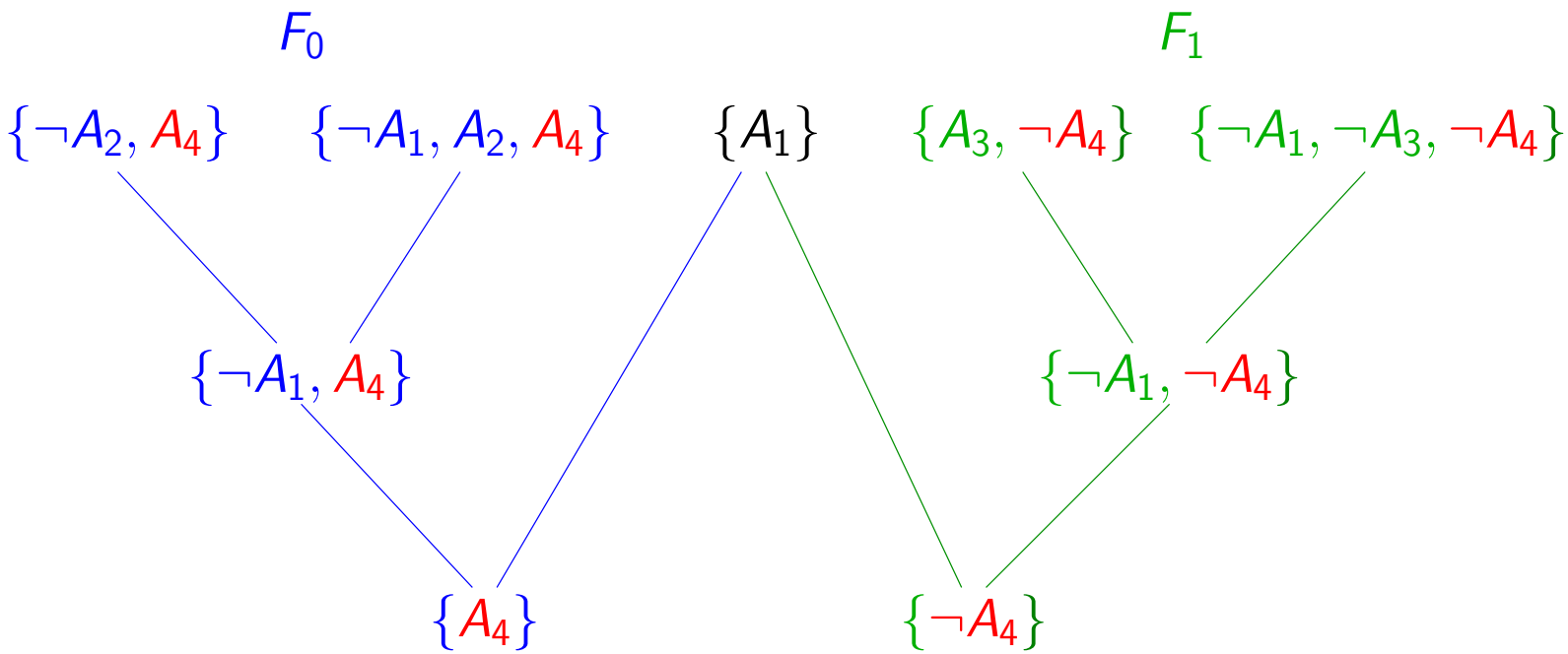


Veranschaulichung des Induktionsschritts

$F = \{\{A_1\}, \{\neg A_2, A_4\}, \{\neg A_1, A_2, A_4\}, \{A_3, \neg A_4\}, \{\neg A_1, \neg A_3, \neg A_4\}\}$

$F_0 = \{\{A_1\}, \{\neg A_2\}, \{\neg A_1, A_2\}\}$

$F_1 = \{\{A_1\}, \{A_3\}, \{\neg A_1, \neg A_3\}\}$

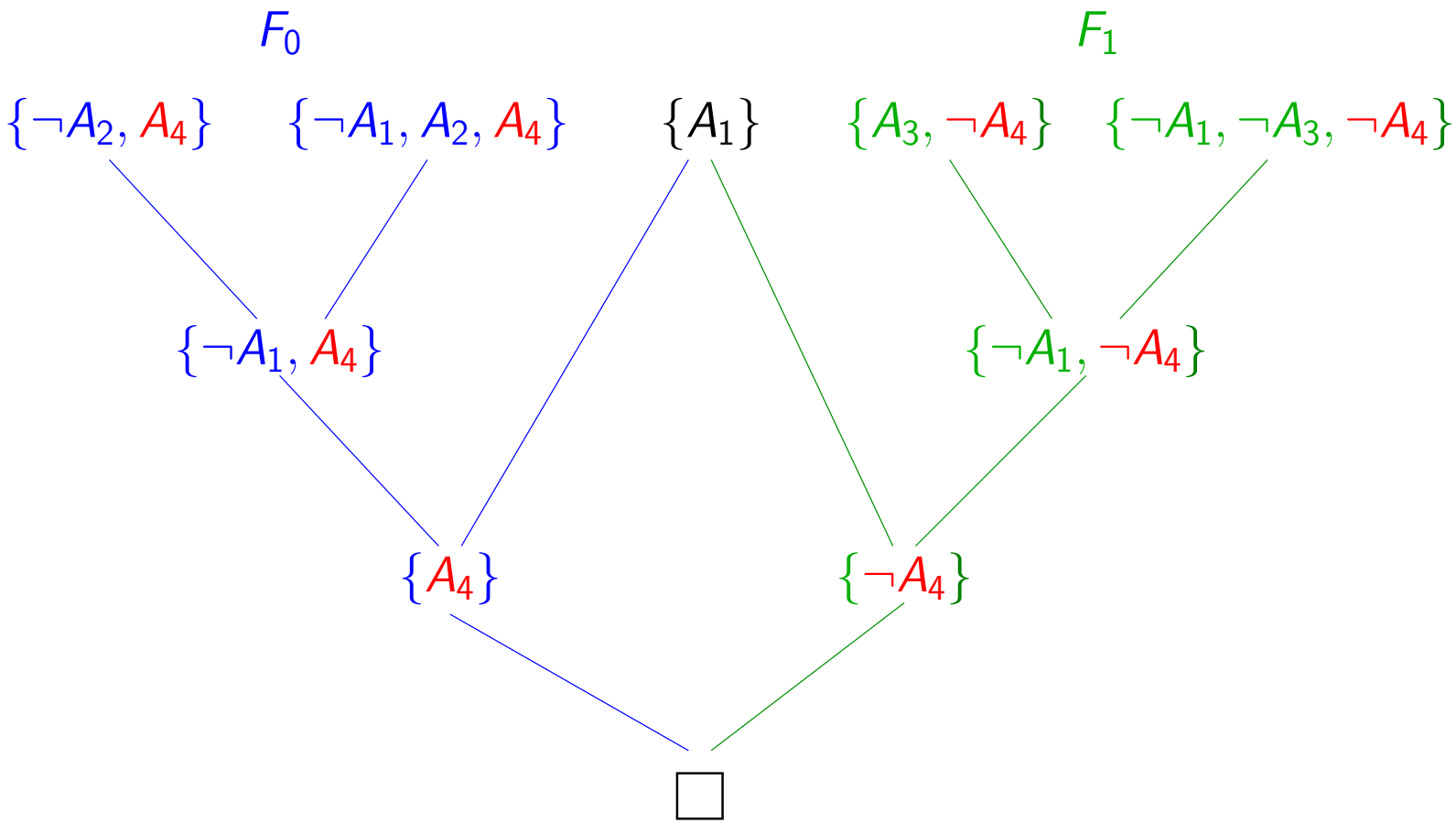


Veranschaulichung des Induktionsschritts

$F = \{\{A_1\}, \{\neg A_2, A_4\}, \{\neg A_1, A_2, A_4\}, \{A_3, \neg A_4\}, \{\neg A_1, \neg A_3, \neg A_4\}\}$

$F_0 = \{\{A_1\}, \{\neg A_2\}, \{\neg A_1, A_2\}\}$

$F_1 = \{\{A_1\}, \{A_3\}, \{\neg A_1, \neg A_3\}\}$



Deduktionen

Eine Deduktion (oder Herleitung oder Beweis) der leeren Klausel aus einer Klauselmenge F ist eine Folge von $K_1, K_2, \ldots, K_m$ von Klauseln mit folgenden Eigenschaften:

> *K_m ist die leere Klausel und für jedes $i \in \{1, \ldots, m\}$ gilt, dass K_i entweder Element von F ist oder aus gewissen Klauseln K_j,K_ℓ mit $j, \ell < i$ resolviert werden kann.*

Aus dem Resolutionssatz folgt:
Eine Klauselmenge ist unerfüllbar genau dann, wenn eine Deduktion der leeren Klausel existiert.

Bemerkung: Es kann sein, dass $Res^*(F)$ sehr groß ist, aber trotzdem eine kurze Deduktion der leeren Klausel existiert.

Resolutionskalkül

Mit dem Begriff Kalkül bezeichnet man eine Menge von syntaktischen Umformungsregeln, mit denen man semantische Eigenschaften der Eingabeformel herleiten kann.

Für den Resolutionskalkül:

- Syntaktische Umformungsregeln: Resolution, Stopp bei Erreichen der leeren Klausel
- Semantische Eigenschaft der Eingabeformel: Unerfüllbarkeit

Wünschenswerte Eigenschaften eines Kalküls:

- Korrektheit: Wenn die leere Klausel aus F abgeleitet werden kann, dann ist F unerfüllbar.
- Vollständigkeit: Wenn F unerfüllbar ist, dann ist die leere Klausel aus F ableitbar.

Beispiel zur Resolution

Wir wollen zeigen, daß

$$((AK \vee BK) \wedge (AK \rightarrow BK) \wedge (BK \wedge RL \rightarrow \neg AK) \wedge RL) \rightarrow (\neg AK \wedge BK)$$

gültig ist.

Das ist genau dann der Fall, wenn

$$(AK \vee BK) \wedge (\neg AK \vee BK) \wedge (\neg BK \vee \neg RL \vee \neg AK) \wedge RL \wedge (AK \vee \neg BK)$$

unerfüllbar ist. (Wegen: $F \rightarrow G$ gültig gdw. $F \wedge \neg G$ unerfüllbar.)

In Mengendarstellung:

$$\{\{AK, BK\},\ \{\neg AK, BK\},\ \{\neg BK, \neg RL, \neg AK\},\ \{RL\},\ \{AK, \neg BK\}\}$$

Beispiel zur Resolution

Eine mögliche Deduktion der leeren Klausel aus

$$\{\{AK, BK\},\ \{\neg AK, BK\},\ \{\neg BK, \neg RL, \neg AK\},\ \{RL\},\ \{AK, \neg BK\}\} \quad (1)$$

sieht wie folgt aus:

Beispiel zur Resolution

Eine mögliche Deduktion der leeren Klausel aus

$$\{\{AK, BK\},\ \{\neg AK, BK\},\ \{\neg BK, \neg RL, \neg AK\},\ \{RL\},\ \{AK, \neg BK\}\} \quad (1)$$

sieht wie folgt aus:

$\{AK, BK\}$	gehört zu (1)	(2)
$\{\neg AK, BK\}$	gehört zu (1)	(3)
$\{BK\}$	aus (2) und (3)	(4)
$\{\neg BK, \neg RL, \neg AK\}$	gehört zu (1)	(5)
$\{AK, \neg BK\}$	gehört zu (1)	(6)
$\{\neg BK, \neg RL\}$	aus (5) und (6)	(7)
$\{RL\}$	gehört zu (1)	(8)
$\{\neg BK\}$	aus (7) und (8)	(9)
$\square$	aus (4) und (9)	(10)

Bemerkungen zur Resolution

Armin Haken hat 1985 für jedes n eine Menge von Klauseln F_n angegeben mit:

- F_n ist unerfüllbar.
- F_n enthält $n \cdot (n+1)$ viele atomare Teilformeln.
- F_n besteht aus $\frac{n^3+n^2}{2} + n + 1$ vielen Klauseln.
- Jede Deduktion der leeren Klausel aus F_n hat Länge mindestens c^n für eine feste Konstante $c > 1$.

Fazit: Deduktionen können sehr lang werden.

Der Endlichkeitssatz der Aussagenlogik

Endlichkeitssatz (compactness theorem)

Sei M eine (eventuell unendliche) Menge von Formeln. Dann ist M genau dann erfüllbar, wenn jede endliche Teilmenge von M erfüllbar ist.

Beweis:

1. Wenn M erfüllbar ist, dann ist jede endliche Teilmenge von M erfüllbar.

Diese Aussage ist trivial, denn jedes Modell von M ist auch ein Modell für jede Teilmenge von M.

Der Endlichkeitssatz der Aussagenlogik: Beweis

2. Sei jede endliche Teilmenge von M erfüllbar. Dann ist auch M erfüllbar.

Zur Erinnerung: Die Menge der atomaren Formeln ist $\{A_1, A_2, A_3, \ldots\}$.

Sei $M_n \subseteq M$ die Menge der Formeln in M, die nur atomare Teilformeln aus $\{A_1, \ldots, A_n\}$ enthalten.

Beachte: M_n kann unendlich sein; z. B. könnte M_1 die Formeln $A_1,\ A_1 \wedge A_1,\ A_1 \wedge A_1 \wedge A_1, \ldots$ enthalten.

Aber: Es gibt nur 2^{2^n} viele verschiedene Wahrheitstafeln mit den atomaren Formeln $A_1, \ldots, A_n$.

Deshalb gibt es in M_n eine endliche Teilmenge $M'_n \subseteq M_n$ mit:

1. $|M'_n| \leq 2^{2^n}$
2. Für jede Formel $F \in M_n$ existiert eine Formel $F' \in M'_n$ mit $F \equiv F'$

Der Endlichkeitssatz der Aussagenlogik: Beweis

Nach Annahme hat M'_n ein Modell $\mathcal{B}_n$.

Also ist $\mathcal{B}_n$ auch ein Modell von M_n.

Wir konstruieren nun ein Modell $\mathcal{B}$ von M wie folgt in Stufen.

In Stufe n wird dabei der Wert $\mathcal{B}(A_n) \in \{0, 1\}$ festgelegt.

$I_0 := \{1, 2, 3, \ldots\}$
for all $n \geq 1$ **do**
    if es gibt unendlich viele Indizes $i \in I_{n-1}$ mit $\mathcal{B}_i(A_n) = 1$ **then**
        $\mathcal{B}(A_n) := 1$
        $I_n := \{i \in I_{n-1} \mid \mathcal{B}_i(A_n) = 1\}$
    else
        $\mathcal{B}(A_n) := 0$
        $I_n := \{i \in I_{n-1} \mid \mathcal{B}_i(A_n) = 0\}$
endfor

Der Endlichkeitssatz der Aussagenlogik: Beweis

Es gilt zwar $I_0 \supseteq I_1 \supseteq I_2 \supseteq I_3 \cdots$ aber:

Behauptung 1: Für jedes $n \geq 0$ ist I_n ist unendlich.

Dies zeigt man durch Induktion über n:

$I_0 = \{1, 2, 3, \ldots\}$ offensichtlich unendlich.

Ist I_{n-1} unendlich, so ist nach Konstruktion auch I_n unendlich.

Direkt aus der Konstruktion von $\mathcal{B}_i$ folgt:

Behauptung 2: Für alle $n \geq 1$ und alle $i \in I_n$ gilt:

$$\mathcal{B}_i(A_1) = \mathcal{B}(A_1), \quad \mathcal{B}_i(A_2) = \mathcal{B}(A_2), \ldots, \mathcal{B}_i(A_n) = \mathcal{B}(A_n).$$

Der Endlichkeitssatz der Aussagenlogik: Beweis

Behauptung 3: Die konstruierte Belegung $\mathcal{B}$ ist ein Modell von M.

Beweis von Behauptung 3: Sei $F \in M$.

Dann existiert ein $n \geq 1$ mit $F \in M_n$ (denn in F kommen nur endlich viele atomare Formeln vor).

Also gilt $F \in M_i$ für alle $i \geq n$.

Also ist jede der Belegungen $\mathcal{B}_i$ mit $i \geq n$ ein Modell von F.

Da I_n nach Behauptung 1 unendlich ist, existiert ein $i \geq n$ mit $i \in I_n$.

Für dieses i gilt nach Behauptung 2:

$$\mathcal{B}_i(A_1) = \mathcal{B}(A_1), \quad \mathcal{B}_i(A_2) = \mathcal{B}(A_2), \ldots, \mathcal{B}_i(A_n) = \mathcal{B}(A_n)$$

Da $\mathcal{B}_i$ wegen $i \geq n$ ein Modell von F ist, ist auch $\mathcal{B}$ ein Modell von F. □

Folgerungen aus dem Endlichkeitssatz

Die folgende Aussage haben wir bisher nur für eine endliche Klauselmenge F bewiesen.

Resolutionssatz der Aussagenlogik für beliebige Formelmengen

Eine Menge F von Klauseln ist unerfüllbar genau dann, wenn $\square \in Res^*(F)$.

Beweis:

Korrektheit: Wenn $\square \in Res^*(F)$, dann ist F unerfüllbar:
Beweis wie im endlichen Fall

Vollständigkeit: Wenn F unerfüllbar ist, dann gilt $\square \in Res^*(F)$.

Wenn F unerfüllbar ist, dann muss nach dem Endlichkeitssatz eine endliche Teilmenge $F' \subseteq F$ existieren, die ebenfalls unerfüllbar ist.

Aus dem bereits bewiesenen Resolutionssatz für endliche Formelmengen folgt $\square \in Res^*(F')$.

Also gilt auch $\square \in Res^*(F)$.

Prädikatenlogik

Aussagenlogik ist zur Formulierung mathematischer Sachverhalte im Allgemeinen zu ausdrucksschwach.

Beispiel: In der Analysis will man Aussagen der Form

> *Für alle* $x \in \mathbb{R}$ *und alle* $\varepsilon > 0$ *existiert ein* $\delta > 0$, *so dass für alle* $y \in \mathbb{R}$ *gilt: Wenn* $\mathit{abs}(x - y) < \varepsilon$, *dann* $\mathit{abs}(f(x) - f(y)) < \delta$.

Hierbei verwenden wir:

- Relationen wie z. B. $<$ oder $>$
- Funktionen wie z. B. $\mathit{abs} : \mathbb{R} \to \mathbb{R}_+$ (Absolutbetrag) oder $- : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ (Minus)
- Quantifikationen wie z. B. "für alle $x \in \mathbb{R}$" oder "existiert ein $\delta > 0$".

Die führt zur Prädikatenlogik

Syntax der Prädikatenlogik: Variablen, Terme

Die Menge der Variablen ist $\{x_0, x_1, x_2, \ldots\}$.

Ein Prädikatensymbol hat die Form P_i^k mit $i, k \in \{0, 1, 2, \ldots\}$.

Ein Funktionssymbol hat die Form f_i^k mit $i, k \in \{0, 1, 2, \ldots\}$.

In beiden Fällen heißt i der Unterscheidungsindex und k die Stelligkeit.

Variablen werden auch mit u, x, y, z bezeichnet,
Prädikatensymbole (Funktionssymbole) auch mit P, Q, R (f, g, h).

Wir definieren nun die Menge der Terme induktiv:

1. Jede Variable ist ein Term.
2. Falls f ein Funktionssymbol mit der Stelligkeit k ist, und falls $t_1, \ldots, t_k$ Terme sind, so ist auch $f(t_1, \ldots, t_k)$ ein Term.

Hierbei sollen auch Funktionssymbole der Stelligkeit 0 eingeschlossen sein, und in diesem Fall sollen die Klammern wegfallen.

Nullstellige Funktionssymbole heißen auch Konstanten
(werden meist mit a, b, c bezeichnet).

Formeln

Nun können wir (wiederum induktiv) definieren, was Formeln (der Prädikatenlogik) sind.

1. Falls P ein Prädikatsymbol der Stelligkeit k ist, und falls $t_1, \ldots, t_k$ Terme sind, dann ist $P(t_1, \ldots, t_k)$ eine Formel.
2. Für jede Formel F ist auch $\neg F$ eine Formel.
3. Für alle Formeln F und G sind auch $(F \wedge G)$ und $(F \vee G)$ Formeln.
4. Falls x eine Variable ist und F eine Formel, so sind auch $\exists x F$ und $\forall x F$ Formeln.
   Das Symbol $\exists$ wird Existenzquantor und $\forall$ Allquantor genannt.

Atomare Formeln nennen wir genau die, die gemäß 1. aufgebaut sind.

Falls F eine Formel ist und F als Teil einer Formel G auftritt, so heißt F Teilformel von G.

Freie und gebundene Variablen, Aussagen

Alle Vorkommen von Variablen in einer Formel, welche nicht direkt hinter einem Quantor stehen, werden in freie und gebundene Vorkommen unterteilt:

Ein Vorkommen der Variablen x in der Formel F, welches nicht direkt hinter einem Quantor steht, ist gebunden, falls dieses Vorkommen in einer Teilformel von F der Form $\exists x G$ oder $\forall x G$ vorkommt.
Andernfalls heißt dieses Vorkommen von x frei.

Eine Formel ohne Vorkommen einer freien Variablen heißt geschlossen oder eine Aussage.

Die Matrix einer Formel F ist diejenige Formel, die man aus F erhält, indem jedes Vorkommen von $\exists$ bzw. $\forall$, samt der dahinterstehenden Variablen gestrichen wird.

Wir bezeichnen die Matrix der Formel F mit F^*.

Beispiel für Formel der Prädikatenlogik

Sei F die Formel

$$\exists x P(x, f(y)) \ \vee \ \neg \forall y Q(y, g(a, h(z)))$$

Das rot markierte Vorkommen von y in F ist frei, während das grün markierte Vorkommen von y in F gebunden ist:

$$\exists x P(x, f(y)) \ \vee \ \neg \forall y Q(y, g(a, h(z)))$$

Die Matrix von F ist

$$P(x, f(y)) \ \vee \ \neg Q(y, g(a, h(z))).$$

Aufgabe

NF: Nicht-Formel F: Formel, aber nicht Aussage A: Aussage

	NF	F	A
$\forall x P(a)$			
$\forall x \exists y (Q(x,y) \vee R(x,y))$			
$\forall x Q(x,x) \rightarrow \exists x Q(x,y)$			
$\forall x P(x) \vee \forall x Q(x,x)$			
$\forall x (P(y) \wedge \forall y P(x))$			
$P(x) \rightarrow \exists x Q(x, P(x))$			
$\forall f \; \exists x P(f(x))$			

NF: Nicht-Formel F: Formel, aber nicht Aussage A: Aussage

	NF	F	A
$\forall x(\neg\forall y Q(x,y) \wedge R(x,y))$			
$\exists z(Q(z,x) \vee R(y,z)) \rightarrow \exists y(R(x,y) \wedge Q(x,z))$			
$\exists x(\neg P(x) \vee P(f(a)))$			
$P(x) \rightarrow \exists x P(x)$			
$\exists x\forall y((P(y) \rightarrow Q(x,y)) \vee \neg P(x))$			
$\exists x\forall x Q(x,x)$			

Semantik der Prädikatenlogik: Strukturen

Eine Struktur ist ein Paar $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$ mit:

$U_\mathcal{A}$ ist eine beliebige aber nicht leere Menge, die die Grundmenge von $\mathcal{A}$ (oder der Grundbereich, der Individuenbereich, das Universum) genannt wird.

Ferner ist $I_\mathcal{A}$ eine partiell definierte Abbildung, die

- jedem k–stelligen Prädikatensymbol P aus dem Definitionsbereich von $I_\mathcal{A}$ eine k–stellige Relation $I_\mathcal{A}(P) \subseteq U_\mathcal{A}^k$ zuordnet,
- jedem k–stelligen Funktionssymbol f aus dem Definitionsbereich von $I_\mathcal{A}$ eine k–stellige Funktion $I_\mathcal{A}(f) : U_\mathcal{A}^k \to U_\mathcal{A}$ zuordnet, und
- jeder Variablen x aus dem Definitionsbereich von $I_\mathcal{A}$ ein Element $I_\mathcal{A}(x) \in U_\mathcal{A}$ zuordnet.

Sei F eine Formel und $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$ eine Struktur.

$\mathcal{A}$ heißt zu F passend, falls $I_\mathcal{A}$ für alle in F vorkommenden Prädikatsymbole, Funktionssymbole und freien Variablen definiert ist.

Mit anderen Worten, der Definitionsbereich von $I_\mathcal{A}$ ist eine Teilmenge von $\{P_i^k, f_i^k, x_i \mid i, k \in \{0, 1, 2, \ldots\}$, und der Wertebereich von $I_\mathcal{A}$ ist die Menge aller Relationen, Funktionen und Elemente von $U_\mathcal{A}$.

Wir schreiben abkürzend statt $I_\mathcal{A}(P)$ einfach $P^\mathcal{A}$, statt $I_\mathcal{A}(f)$ einfach $f^\mathcal{A}$ und statt $I_\mathcal{A}(x)$ einfach $x^\mathcal{A}$.

Beispiel für Struktur

Sei F die Formel

$$\forall x P(x, f(x)) \ \wedge \ Q(g(a, z))$$

Eine zu F passende Struktur ist z. B.

$$\mathcal{A} = (\mathbb{N}, I_{\mathcal{A}})$$

wobei

$$\begin{aligned} P^{\mathcal{A}} &= \{(n, m) \mid n, m \in \mathbb{N}, n < m\} \\ Q^{\mathcal{A}} &= \{n \mid n \text{ ist Primzahl}\} \\ f^{\mathcal{A}}(n) &= n + 1 \text{ für alle } n \in \mathbb{N} \\ g^{\mathcal{A}}(n, m) &= n + m \text{ für alle } n, m \in \mathbb{N} \\ a^{\mathcal{A}} &= 2 \\ z^{\mathcal{A}} &= 3 \end{aligned}$$

In einem intuitiven Sinn gilt die Formel F in der Struktur $\mathcal{A}$.

Auswertung in einer Struktur

Sei F eine Formel und $\mathcal{A}$ eine zu F passende Struktur.

Für jeden Term t, den man aus den Bestandteilen von F bilden kann (also aus den freien Variablen und Funktionssymbolen), definieren wir induktiv den Wert $\mathcal{A}(t) \in U_{\mathcal{A}}$ von t in der Struktur $\mathcal{A}$:

1. Falls t eine Variable ist (also $t = x$), so ist $\mathcal{A}(t) = x^{\mathcal{A}}$.
2. Falls t die Form hat $t = f(t_1, \ldots, t_k)$ wobei $t_1, \ldots, t_k$ Terme und f ein k–stelliges Funktionssymbol ist, so ist $\mathcal{A}(t) = f^{\mathcal{A}}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$.

Der Fall 2 schließt auch die Möglichkeit ein, dass f nullstellig ist, also t die Form $t = a$ hat.

In diesem Fall ist also $\mathcal{A}(t) = a^{\mathcal{A}}$.

Auswertung in einer Struktur

Auf analoge Weise definieren wir induktiv den (Wahrheits–) Wert $\mathcal{A}(F)$ der Formeln F unter der Struktur $\mathcal{A}$:

- Falls F die Form hat $F = P(t_1, \ldots, t_k)$ mit den Termen $t_1, \ldots, t_k$ und k–stelligem Prädikatsymbol P, so ist

$$\mathcal{A}(F) = \begin{cases} 1, & \textit{falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in P^{\mathcal{A}} \\ 0, & \textit{sonst} \end{cases}$$

- Falls F die Form $F = \neg G$ hat, so ist

$$\mathcal{A}(F) = \begin{cases} 1, & \textit{falls } \mathcal{A}(G) = 0 \\ 0, & \textit{sonst} \end{cases}$$

Auswertung in einer Struktur

- Falls F die Form $F = (G \wedge H)$ hat, so ist

$$\mathcal{A}(F) = \begin{cases} 1, & \textit{falls } \mathcal{A}(G) = 1 \textit{ und } \mathcal{A}(H) = 1 \\ 0, & \textit{sonst} \end{cases}$$

- Falls F die Form $F = (G \vee H)$ hat, so ist

$$\mathcal{A}(F) = \begin{cases} 1, & \textit{falls } \mathcal{A}(G) = 1 \textit{ oder } \mathcal{A}(H) = 1 \\ 0, & \textit{sonst} \end{cases}$$

Auswertung in einer Struktur

- Falls F die Form $F = \forall x G$ hat, so ist

$$\mathcal{A}(F) = \begin{cases} 1, & \textit{falls für alle } d \in U_{\mathcal{A}} \textit{ gilt : } \mathcal{A}_{[x/d]}(G) = 1 \\ 0, & \textit{sonst} \end{cases}$$

- Falls F die Form $F = \exists x G$ hat, so ist

$$\mathcal{A}(F) = \begin{cases} 1, & \textit{falls es ein } d \in U_{\mathcal{A}} \textit{ gibt mit : } \mathcal{A}_{[x/d]}(G) = 1 \\ 0, & \textit{sonst} \end{cases}$$

Hierbei ist $\mathcal{A}_{[x/d]}$ diejenige Struktur $\mathcal{A}'$, die überall mit $\mathcal{A}$ identisch ist, bis auf die Definition von $x^{\mathcal{A}'}$.

Es sei nämlich $x^{\mathcal{A}'} = d$, wobei $d \in U_{\mathcal{A}} = U_{\mathcal{A}'}$ — unabhängig davon, ob $I_{\mathcal{A}}$ auf x definiert ist oder nicht.

Konvention zu $=$

Wir machen für das Weitere folgende Konvention:

Wann immer wir das Prädikatensymbol $=$ in Formeln verwenden, soll $=$ 2-stellig sein, und für jede Struktur $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$ soll gelten:

$$I_\mathcal{A}(=) = \{(a, a) \mid a \in U_\mathcal{A}\}.$$

Modell, Gültigkeit, Erfüllbarkeit

Falls für eine Formel F und eine zu F passende Struktur $\mathcal{A}$ gilt $\mathcal{A}(F) = 1$, so schreiben wir wieder $\mathcal{A} \models F$.

Sprechweise: F gilt in $\mathcal{A}$ oder $\mathcal{A}$ ist Modell für F.

$\mathcal{A}$ ist Modell für eine Menge M von Formeln, falls $\mathcal{A}$ ein Modell für jede Formel $F \in M$ ist.

Falls jede zu F passende Struktur ein Modell für F ist, so schreiben wir $\models F$, andernfalls $\not\models F$.

Sprechweise: F ist (allgemein–)gültig.

Falls es mindestens ein Modell für die Formel F gibt, so heißt F erfüllbar, andernfalls unerfüllbar.

Aufgabe

G: Gültig E: Erfüllbar, aber nicht gültig U: Unerfüllbar

	G	E	U
$\forall x P(a)$			
$\exists x(\neg P(x) \vee P(a))$			
$P(a) \rightarrow \exists x P(x)$			
$P(x) \rightarrow \exists x P(x)$			
$\forall x P(x) \rightarrow \exists x P(x)$			
$\forall x P(x) \wedge \neg \forall y P(y)$			

G: Gültig E: Erfüllbar, aber nicht gültig U: Unerfüllbar

	G	E	U
$\forall x(P(x,x) \rightarrow \exists x \forall y P(x,y))$			
$\forall x \forall y(x = y \rightarrow f(x) = f(y))$			
$\forall x \forall y(f(x) = f(y) \rightarrow x = y)$			
$\exists x \exists y \exists z(f(x) = y \wedge f(x) = z \wedge y \neq z)$			

Folgerung und Äquivalenz

Eine Formel G heißt eine Folgerung der Formeln $F_1, \ldots, F_k$, falls für jede Struktur $\mathcal{A}$, die sowohl zu $F_1, \ldots, F_k$ als auch zu G passend ist, gilt:

> *Wenn $\mathcal{A}$ Modell von $\{F_1, \ldots, F_k\}$ ist, dann ist $\mathcal{A}$ auch Modell von G.*

Wir schreiben $F_1, \ldots, F_k \models G$, falls G eine Folgerung von $F_1, \ldots, F_k$ ist.

Zwei Formeln F und G heißen (semantisch) äquivalent, falls für alle Strukturen $\mathcal{A}$, die sowohl für F als auch für G passend sind, gilt $\mathcal{A}(F) = \mathcal{A}(G)$.

Hierfür schreiben wir $F \equiv G$.

Aufgabe

1. $\forall x P(x) \vee \forall x Q(x,x)$
2. $\forall x (P(x) \vee Q(x,x))$
3. $\forall x (\forall z P(z) \vee \forall y Q(x,y))$

	J	N
1. $\models$ 2.		
2. $\models$ 3.		
3. $\models$ 1.		

Aufgabe

1. $\exists y \forall x P(x, y)$
2. $\forall x \exists y P(x, y)$

	J	N
1. $\models$ 2.		
2. $\models$ 1.		

Aufgabe

	J	N
$\forall x \forall y F \equiv \forall y \forall x F$		
$\forall x \exists y F \equiv \exists x \forall y F$		
$\exists x \exists y F \equiv \exists y \exists x F$		
$\forall x F \vee \forall x G \equiv \forall x (F \vee G)$		
$\forall x F \wedge \forall x G \equiv \forall x (F \wedge G)$		
$\exists x F \vee \exists x G \equiv \exists x (F \vee G)$		
$\exists x F \wedge \exists x G \equiv \exists x (F \wedge G)$		

Äquivalenzen

Satz (Äquivalenzen der Prädikatenlogik)

Seien F und G beliebige Formeln:

1. $\neg\forall xF \equiv \exists x\neg F$
   $\neg\exists xF \equiv \forall x\neg F$
2. Falls x in G nicht frei vorkommt, gilt:
   $(\forall xF \wedge G) \equiv \forall x(F \wedge G)$
   $(\forall xF \vee G) \equiv \forall x(F \vee G)$
   $(\exists xF \wedge G) \equiv \exists x(F \wedge G)$
   $(\exists xF \vee G) \equiv \exists x(F \vee G)$
3. $(\forall xF \wedge \forall xG) \equiv \forall x(F \wedge G)$
   $(\exists xF \vee \exists xG) \equiv \exists x(F \vee G)$
4. $\forall x\forall yF \equiv \forall y\forall xF$
   $\exists x\exists yF \equiv \exists y\exists xF$

Äquivalenzen

Wir beweisen exemplarisch die Äquivalenz

$$(\forall x F \wedge G) \equiv \forall x (F \wedge G)$$

wobei x in G nicht frei vorkommt.

Sei $\mathcal{A}$ eine beliebige zu $(\forall x F \wedge G)$ und $\forall x (F \wedge G)$ passende Struktur. Dann gilt:

$\mathcal{A}(\forall x F \wedge G) = 1$

gdw. $\mathcal{A}(\forall x F) = 1$ und $\mathcal{A}(G) = 1$

gdw. für alle $d \in U_{\mathcal{A}}$ gilt $\mathcal{A}_{[x/d]}(F) = 1$ und $\mathcal{A}(G) = 1$

gdw. für alle $d \in U_{\mathcal{A}}$ gilt $\mathcal{A}_{[x/d]}(F) = 1$ und $\mathcal{A}_{[x/d]}(G) = 1$

(beachte: da x nicht frei in G vorkommt gilt $\mathcal{A}(G) = \mathcal{A}_{[x/d]}(G)$

gdw. für alle $d \in U_{\mathcal{A}}$ gilt $\mathcal{A}_{[x/d]}(F \wedge G) = 1$

gdw. $\mathcal{A}(\forall x (F \wedge G)) = 1$

Umbenennung von gebundenen Variablen

Für eine Formel G, eine Variable x und einen Term t sei $G[x/t]$ die Formel, die aus G entsteht, indem jedes freie Vorkommen von x in G durch den Term t ersetzt wird.

Beispiel:

$$\Big(\exists x P\Big(x, f(y)\Big) \ \vee \ \neg\forall y Q\Big(y, g(a, h(z))\Big)\Big)[y/f(u)] =$$
$$\Big(\exists x P\Big(x, f(f(u))\Big) \ \vee \ \neg\forall y Q\Big(y, g(a, h(z))\Big)\Big)$$

Übung: Definiere $G[x/t]$ formal durch Induktion über den Aufbau von G.

Lemma (Umbenennung von Variablen)

Sei $F = QxG$ eine Formel mit $Q \in \{\forall, \exists\}$. Sei y eine Variable, die in G nicht vorkommt. Dann gilt $F \equiv QyG[x/y]$.

Bereinigte Formeln

Eine Formel heißt bereinigt, sofern es

- keine Variable gibt, die in der Formel sowohl gebunden als auch frei vorkommt, und sofern
- hinter allen vorkommenden Quantoren verschiedene Variablen stehen.

Wiederholte Anwendung des Lemmas "Umbennung von Variaben" liefert:

Lemma

Zu jeder Formel F gibt es eine äquivalente bereinigte Formel.

Pränexform

Eine Formel heißt pränex oder in Pränexform, falls sie die Bauart

$$Q_1 y_1 Q_2 y_2 \cdots Q_n y_n F$$

hat, wobei

- $n \geq 0$, $Q_1, \ldots, Q_n \in \{\exists, \forall\}$, $y_1, \ldots, y_n$ Variablen sind, und
- in F kein Quantor vorkommt.

Eine bereinigte Formel in Pränexform ist in **BPF**.

Satz

Für jede Formel gibt es eine äquivalente Formel in **BPF**.

Beweis: Wiederholte Anwendung der Äquivalenzen (1) und (2) aus dem Satz "Äquivalenzen der Prädikatenlogik"

Bildung der Pränexform

Sei F eine beliebige Formel.

Wir definieren eine zu F äquivalente **BPF**-Formel F' durch Induktion über den Aufbau von F:

- F ist atomar:
  Dann ist F bereits in **BPF** und wir können $F' = F$ setzen.
- $F = \neg G$:
  Nach Induktion existiert eine zu G äquivalente **BPF**-Formel

$$G' = Q_1 y_1 Q_2 y_2 \cdots Q_n y_n H,$$

  wobei $Q_1, \ldots, Q_n \in \{\exists, \forall\}$ und H keine Quantoren enthält.
  Aus Punkt (1) im Satz "Äquivalenzen der Prädikatenlogik" folgt:

$$\begin{aligned} F = \neg G \equiv \neg G' &= \neg Q_1 y_1 Q_2 y_2 \cdots Q_n y_n H \\ &\equiv \overline{Q}_1 y_1 \overline{Q}_2 y_2 \cdots \overline{Q}_n y_n \neg H \end{aligned}$$

  wobei $\overline{\exists} = \forall$ und $\overline{\forall} = \exists$. Letztere Formel ist in **BPF**.

Bildung der Pränexform

- $F = F_1 \wedge F_2$:
  Nach Induktion existieren zu F_1 und F_2 äquivalente **BPF**-Formeln

  $$F_1' = Q_1 y_1 Q_2 y_2 \cdots Q_m y_m G_1 \text{ und } F_2' = P_1 z_1 P_2 z_2 \cdots P_n z_n G_2,$$

  wobei $Q_1, \ldots, Q_m, P_1, \ldots, P_n \in \{\exists, \forall\}$ und G_1, G_2 keine Quantoren enthalten.
  Auf Grund des Umbennungslemmas können wir annehmen, dass $y_1, \ldots, y_m$ nicht in F_2' vorkommen und $z_1, \ldots, z_n$ nicht in F_1' vorkommen.
  Aus Punkt (2) im Satz "Äquivalenzen der Prädikatenlogik" folgt dann

  $$\begin{aligned} F &= F_1 \wedge F_2 \equiv F_1' \wedge F_2' \\ &= (Q_1 y_1 Q_2 y_2 \cdots Q_m y_m G_1) \wedge F_2' \\ &\equiv Q_1 y_1 Q_2 y_2 \cdots Q_m y_m (G_1 \wedge F_2') \\ &= Q_1 y_1 Q_2 y_2 \cdots Q_m y_m (G_1 \wedge P_1 z_1 P_2 z_2 \cdots P_n z_n G_2) \\ &\equiv Q_1 y_1 Q_2 y_2 \cdots Q_m y_m P_1 z_1 P_2 z_2 \cdots P_n z_n (G_1 \wedge G_2) \end{aligned}$$

Bildung der Pränexform

- $F = F_1 \vee F_2$: gleiche Argumentation wie bei $\wedge$
- $F = QxG$ mit $Q \in \{\exists, \forall\}$:
  Nach Induktion existiert eine zu G äquivalente **BPF**-Formel

$$G' = Q_1 y_1 Q_2 y_2 \cdots Q_n y_n H$$

Auf Grund des Umbennungslemmas können wir annehmen, dass $x \notin \{y_1, \ldots, y_n\}$ gilt.

Damit gilt

$$F = QxG \equiv QxQ_1 y_1 Q_2 y_2 \cdots Q_n y_n H$$

und letztere Formel ist in **BPF**. □

Bildung der Pränexform: Beispiel

$$\Big(\forall x \exists y\ P(x, g(y, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \neg \forall x\ R(x, y)$$

$$\equiv \Big(\forall x \exists y\ P(x, g(y, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \exists x\ \neg R(x, y)$$

$$\equiv \forall x \exists y \Big(P(x, g(y, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \exists x\ \neg R(x, y)$$

$$\equiv \forall x \exists y \Big(P(x, g(y, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \exists w\ \neg R(w, y)$$

$$\equiv \forall x \Big(\exists y \Big(P(x, g(y, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \exists w\ \neg R(w, y)\Big)$$

$$\equiv \forall x \Big(\exists v \Big(P(x, g(v, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \exists w\ \neg R(w, y)\Big)$$

$$\equiv \forall x \exists v \Big(\Big(P(x, g(v, f(x)))\ \vee\ \neg Q(z)\Big)\ \vee\ \exists w\ \neg R(w, y)\Big)$$

$$\equiv \forall x \exists v \exists w \Big(P(x, g(v, f(x)))\ \vee\ \neg Q(z)\ \vee\ \neg R(w, y)\Big)$$

Skolemform

Für jede Formel F in **BPF** definieren wir ihre Skolemform(–el) als das Resultat der Anwendung von folgenden Algorithmus auf F:

while F enthält einen Existenzquantor **do**
begin

F habe die Form $F = \forall y_1 \forall y_2 \cdots \forall y_n \exists z G$ für eine Formel G in **BPF** und $n \geq 0$ (der Allquantorblock kann auch leer sein);

Sei f ein neues bisher in F nicht vorkommendes n–stelliges Funktionssymbol;

$F := \forall y_1 \forall y_2 \cdots \forall y_n G[z/f(y_1, y_2, \ldots, y_n)]$;
(d. h. der Existenzquantor in F wird gestrichen und jedes Vorkommen der Variablen z in G durch $f(y_1, y_2, \ldots, y_n)$ ersetzt)

end

Skolemform: Beispiel

Wir wollen die Skolemform von

$$\forall x \exists v \exists w \Big(P(x, g(v, f(x))) \ \vee \ \neg Q(z) \ \vee \ \neg R(w, y) \Big)$$

bilden.

Nach 1. Durchlauf durch **while**-Schleife:

$$\forall x \exists w \Big(P(x, g(f_1(x), f(x))) \ \vee \ \neg Q(z) \ \vee \ \neg R(w, y) \Big)$$

Nach 2. Durchlauf durch **while**-Schleife:

$$\forall x \Big(P(x, g(f_1(x), f(x))) \ \vee \ \neg Q(z) \ \vee \ \neg R(f_2(x), y) \Big)$$

Skolemform(F) erfüllbar gdw. F erfüllbar.

Satz

Für jede Formel F in **BPF** gilt: F ist erfüllbar genau dann, wenn die Skolemform von F erfüllbar ist.

Für den Beweis benötigen wir das folgende einfache Überführungslemma:

Überführungslemma

Sei F eine Formel, x eine Variable, und t ein Term, der keine in F gebundene Variable enthält. Dann gilt für jede zu F und $F[x/t]$ passende Struktur $\mathcal{A}$:

$$\mathcal{A}(F[x/t]) = \mathcal{A}_{[x/\mathcal{A}(t)]}(F)$$

Beweis des Überführungslemmas

Wir zeigen zunächst durch Induktion über den Aufbau von Termen:

Für jeden Term t' gilt: $\mathcal{A}(t'[x/t]) = \mathcal{A}_{[x/\mathcal{A}(t)]}(t')$

- $t' = y$ für eine Variable y.
  - $y = x$, d. h. $t' = x$:

    Dann gilt $t'[x/t] = t$.

    Also: $\mathcal{A}_{[x/\mathcal{A}(t)]}(t') = \mathcal{A}_{[x/\mathcal{A}(t)]}(x) = \mathcal{A}(t) = \mathcal{A}(t'[x/t])$
  - $y \neq x$.

    Dann gilt $t'[x/t] = t' = y$.

    Also: $\mathcal{A}_{[x/\mathcal{A}(t)]}(t') = \mathcal{A}(t') = \mathcal{A}(t'[x/t])$.

Beweis des Überführungslemmas

- $t' = f(t_1, \ldots, t_n)$.

  Dann gilt:

$$\begin{aligned}
\mathcal{A}(t'[x/t]) &= \mathcal{A}(f(t_1[x/t], \ldots, t_n[x/t])) \\
&= f^{\mathcal{A}}(\mathcal{A}(t_1[x/t]), \ldots, \mathcal{A}(t_n[x/t])) \\
&= f^{\mathcal{A}_{[x/\mathcal{A}(t)]}}(\mathcal{A}_{[x/\mathcal{A}(t)]}(t_1), \ldots, \mathcal{A}_{[x/\mathcal{A}(t)]}(t_n)) \\
&= \mathcal{A}_{[x/\mathcal{A}(t)]}(f(t_1, \ldots, t_n)) \\
&= \mathcal{A}_{[x/\mathcal{A}(t)]}(t')
\end{aligned}$$

Beweis des Überführungslemmas

Nun können wir $\mathcal{A}(F[x/t]) = \mathcal{A}_{[x/\mathcal{A}(t)]}(F)$ für eine Formel F durch Induktion über den Aufbau von F beweisen:

- F atomar, d. h. $F = P(t_1, \ldots, t_n)$ für ein Prädikatensymbol P und Terme $t_1, \ldots, t_n$.

  Dann gilt:

$$
\begin{aligned}
\mathcal{A}(F[x/t]) = 1 \quad & \text{gdw.} \quad \mathcal{A}(P(t_1[x/t], \ldots, t_n[x/t])) = 1 \\
& \text{gdw.} \quad (\mathcal{A}(t_1[x/t]), \ldots, \mathcal{A}(t_n[x/t])) \in P^{\mathcal{A}} \\
& \text{gdw.} \quad (\mathcal{A}_{[x/\mathcal{A}(t)]}(t_1), \ldots, \mathcal{A}_{[x/\mathcal{A}(t)]}(t_n)) \in P^{\mathcal{A}_{[x/\mathcal{A}(t)]}} \\
& \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(P(t_1, \ldots, t_n)) = 1 \\
& \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(F) = 1
\end{aligned}
$$

Beweis des Überführungslemmas

- $F = \neg G$.

  Dann gilt:

$$\begin{aligned} \mathcal{A}(F[x/t]) = 1 \quad & \text{gdw.} \quad \mathcal{A}(\neg G[x/t]) = 1 \\ & \text{gdw.} \quad \mathcal{A}(G[x/t]) = 0 \\ & \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(G) = 0 \\ & \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(\neg G) = 1 \\ & \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(F) = 1 \end{aligned}$$

- $F = F_1 \wedge F_2$ oder $F = F_1 \vee F_2$:

  Analoges Argument wie im vorherigen Fall.

Beweis des Überführungslemmas

- $F = \exists y G$, wobei y nicht in t vorkommt (denn t soll keine in F gebundene Variable enthalten).

  Wenn $y = x$, dann $\mathcal{A}(F[x/t]) = \mathcal{A}(F) = \mathcal{A}_{[x/\mathcal{A}(t)]}(F)$.

  Die letzte Gleichung gilt, weil x nicht frei in F vorkommt.

  Sei nun $y \neq x$. Dann gilt:

$$
\begin{aligned}
\mathcal{A}(F[x/t]) = 1 \quad & \text{gdw.} \quad \mathcal{A}(\exists y G[x/t]) = 1 \\
& \text{gdw.} \quad \text{es gibt } d \in U_{\mathcal{A}} \text{ mit } \mathcal{A}_{[y/d]}(G[x/t]) = 1 \\
& \text{gdw.} \quad \text{es gibt } d \in U_{\mathcal{A}} \text{ mit } \mathcal{A}_{[y/d][x/\mathcal{A}_{[y/d]}(t)]}(G) = 1 \\
& \text{gdw.} \quad \text{es gibt } d \in U_{\mathcal{A}_{[x/\mathcal{A}(t)]}} \text{ mit } \mathcal{A}_{[x/\mathcal{A}(t)][y/d]}(G) = 1 \\
& \qquad (\mathcal{A}_{[y/d]}(t) = \mathcal{A}(t), \text{ da } y \text{ nicht in } t \text{ vorkommt}) \\
& \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(\exists y G) = 1 \\
& \text{gdw.} \quad \mathcal{A}_{[x/\mathcal{A}(t)]}(F) = 1
\end{aligned}
$$

Beweis von Skolemform(F) erfüllbar gdw. F erfüllbar

Beweis des Satzes:

Wir zeigen: Nach jedem Durchlauf durch die **while**-Schleife ist die erhaltene Formel erfüllbar, gdw. die Formel vor dem Durchlauf erfüllbar ist.

Formel vor dem Durchlauf durch die **while**-Schleife:

$$F = \forall y_1 \forall y_2 \cdots \forall y_n \exists z G$$

Formel nach dem Durchlauf durch die **while**-Schleife:

$$F' = \forall y_1 \forall y_2 \cdots \forall y_n G[z/f(y_1, y_2, \ldots, y_n)]$$

Zu zeigen: F ist erfüllbar genau dann, wenn F' ist erfüllbar.

Beweis von Skolemform(F) erfüllbar gdw. F erfüllbar

(1) Sei $F' = \forall y_1 \forall y_2 \cdots \forall y_n G[z/f(y_1, y_2, \ldots, y_n)]$ erfüllbar.

Dann gibt es eine Struktur $\mathcal{A}$ (passend zu F') mit $\mathcal{A}(F') = 1$.

Dann ist $\mathcal{A}$ auch zu F passend und es gilt:

Für alle $d_1, \ldots, d_n \in U_\mathcal{A}$ gilt:

$$\mathcal{A}_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n]}(G[z/f(y_1, y_2, \ldots, y_n)]) = 1$$

Mit dem Überführungslemma folgt:

Für alle $d_1, \ldots, d_n \in U_\mathcal{A}$ gilt:

$$\mathcal{A}_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n][z/d]}(G) = 1,$$

wobei $d = f^\mathcal{A}(d_1, \ldots, d_n)$. Dies impliziert

Für alle $d_1, \ldots, d_n \in U_\mathcal{A}$ gibt es ein $d \in U_\mathcal{A}$ mit:

$$\mathcal{A}_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n][z/d]}(G) = 1$$

d.h. $\mathcal{A}(\forall y_1 \forall y_2 \cdots \forall y_n \exists z G) = 1$.

Beweis von Skolemform(F) erfüllbar gdw. F erfüllbar

(2) Sei $F = \forall y_1 \forall y_2 \cdots \forall y_n \exists z G$ erfüllbar.

Dann existiert eine Struktur $\mathcal{A}$ mit

$$\text{Für alle } d_1, \ldots, d_n \in U_{\mathcal{A}} \text{ gibt es ein } d \in U_{\mathcal{A}} \text{ mit:}$$
$$\mathcal{A}_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n][z/d]}(G) = 1$$

Wir können also für alle $d_1, \ldots, d_n \in U_{\mathcal{A}}$ ein $u(d_1, \ldots, d_n) \in U_{\mathcal{A}}$ mit

$$\mathcal{A}_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n][z/u(d_1,\ldots,d_n)]}(G) = 1$$

auswählen. Wir definieren nun eine Struktur $\mathcal{A}'$ wie folgt:

- $\mathcal{A}'$ ist identisch zu $\mathcal{A}$ außer
- $f^{\mathcal{A}'}(d_1, \ldots, d_n) = u(d_1, \ldots, d_n)$ für alle $d_1, \ldots, d_n \in U_{\mathcal{A}}$.

Dann gilt:

$$\text{Für alle } d_1, \ldots, d_n \in U_{\mathcal{A}} = U_{\mathcal{A}'} \text{ gilt:}$$
$$\mathcal{A}'_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n][z/f^{\mathcal{A}'}(d_1,\ldots,d_n)]}(G) = 1$$

Beweis von Skolemform(F) erfüllbar gdw. F erfüllbar

Mit dem Überführungslemma folgt:

Für alle $d_1, \ldots, d_n \in U_{\mathcal{A}'}$ gilt:

$$\mathcal{A}'_{[y_1/d_1][y_2/d_2]\cdots[y_n/d_n]}(G[z/f(y_1, y_2, \ldots, y_n)]) = 1$$

und somit $\mathcal{A}'(\forall y_1 \forall y_2 \cdots \forall y_n G[z/f(y_1, y_2, \ldots, y_n)]) = 1$. □

Bemerkungen:

- Die Skolemform einer Formel F ist i.A. nicht äquivalent zu F.

  Beispiel: Die Skolemform von $\exists x P(x)$ ist $P(a)$ für eine Konstante a. Aber: $\exists x P(x) \not\equiv P(a)$
- Der obige Beweis zeigt sogar, dass jedes Modell der Skolemform von F auch ein Modell von F ist.
- Ausserdem erhalten wir ein Modell der Skolemform von F indem wir ein Modell von F erweitern um eine Interpretation der neuen Funktionssymbole (die bei der Bildung der Skolemform eingeführt werden).

Skolemform einer Menge von Formeln

Sei nun M eine (i.A. unendliche) Menge von prädikatenlogischen Formeln, die o.B.d.A. alle in **BPF** sind.

Wir definieren die Skolemform von M als die Menge M' der Skolemformeln, die wir wie folgt erhalten:

Wir ersetzen in M jede Formel F durch seine Skolemform. Dabei achten wir darauf, dass wir für verschiedene Formeln $F, G \in M$ disjunkte Mengen von neuen Funktionssymbolen einführen.

Dann erhalten wir:

Satz

Sei M eine (i.A. unendliche) Menge von prädikatenlogischen Formeln in **BPF**. Dann ist M erfüllbar, genau dann, wenn die Skolemform von M erfüllbar ist.

Skolemform einer Menge von Formeln

Beweis: Sei $M' = \{F' \mid F \in M\}$ die Skolemform von M, wobei F' eine Skolemform von F ist.

(1) Sei $\mathcal{A}$ ein Modell von M' und sei $F \in M$ beliebig.

Dann ist $\mathcal{A}$ also ein Modell von F' und damit ein Modell von F.

(2) Sei $\mathcal{A}$ ein Modell von M

Für eine Formel $F \in M$ sei $\text{fun}(F)$ die Menge der neuen Funktionssymbole, die wir bei der Bildung der Skolemform F' von F eingeführt haben.

Also gilt $\text{fun}(F) \cap \text{fun}(G) = \emptyset$ für $F \neq G$ aus $\mathcal{F}$.

Wir erhalten ein Modell von $F' \in M'$ indem wir $\mathcal{A}$ um eine Interpretation der Symbole aus $\text{fun}(F)$ erweitern.

Diese Erweiterungen liefern dann ein Modell von M'. □

Klauselform

Eine Aussage (= geschlossene Formel) heißt in Klauselform, falls sie die Bauart

$$\forall y_1 \forall y_2 \cdots \forall y_n \, F$$

hat, wobei F keine Quantoren enthält und in **KNF** ist, und $y_i \neq y_j$ für $i \neq j$.

Eine Ausage in Klauselform kann als Menge von Klauseln dargestellt werden.

Umformung einer beliebigen Formel in eine Aussage in Klauselform

Gegeben: eine prädikatenlogische Formel F (mit eventuellen Vorkommen von freien Variablen).

1. Bereinige F durch systematisches Umbenennen der gebundenen Variablen. Es entsteht eine zu F äquivalente Formel F_1.
2. Seien $y_1, y_2, \ldots, y_n$ die in F bzw. F_1 vorkommenden freien Variablen. Ersetze F_1 durch $F_2 = \exists y_1 \exists y_2 \ldots \exists y_n F_1$. Dann ist F_2 erfüllbar genau dann, wenn F_1 erfüllbar ist, und F_2 enthält keine freien Variablen mehr.
3. Stelle eine zu F_2 äquivalente (und damit zu F erfüllbarkeitsäquivalente) Aussage F_3 in Pränexform her.
4. Eliminiere die vorkommenden Existenzquantoren durch Übergang zur Skolemform von F_3. Diese sei F_4 und ist dann erfüllbarkeitsäquivalent zu F_3 und damit auch zu F.
5. Forme die Matrix von F_4 um in **KNF** (und schreibe diese Formel F_5 dann als Klauselmenge auf).

Aufgabe zu Normalformen

Welche dieser Formel sind bereinigt, in Pränexform, in Skolemform, in Klauselform?

	B	P	S	K
$\forall x(Tet(x) \vee Cube(x) \vee Dodec(x))$				
$\exists x \exists y(Cube(y) \vee BackOf(x, y))$				
$\forall x(\neg FrontOf(x, x) \wedge \neg BackOf(x, x))$				
$\neg \exists x Cube(x) \leftrightarrow \forall x \neg Cube(x)$				
$\forall x(Cube(x) \rightarrow Small(x)) \rightarrow \forall y(\neg Cube(y) \rightarrow \neg Small(y))$				
$(Cube(a) \wedge \forall x Small(x)) \rightarrow Small(a)$				
$\exists x(Larger(a, x) \wedge Larger(x, b)) \rightarrow Larger(a, b)$				

Herbrand-Universum

Sei $\mathcal{F} \subseteq \{f_i^k \mid i, k \geq 0\}$ eine Menge von Funktionssymbolen, die mindestens eine Konstante enthält.

Das Herbrand-Universum $D(\mathcal{F})$ ist die Menge aller variablenfreien Terme, die aus den Symbolen in $\mathcal{F}$ gebildet werden können.

Etwas formaler wird $D(\mathcal{F})$ wird wie folgt induktiv definiert:

Für jedes n-stellige Funktionssymbol $f \in \mathcal{F}$ ($n = 0$ ist hier möglich) und Terme $t_1, t_2, \ldots, t_n \in D(\mathcal{F})$ gehört auch der Term $f(t_1, t_2, \ldots, t_n)$ zu $D(\mathcal{F})$.

Beispiel: $D(\{f, a\}) = \{a, f(a), f(f(a)), f(f(f(a))), \ldots\}$

Herbrand-Universum

Eine Herbrand-Struktur ist eine Struktur $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$, so dass eine Menge $\mathcal{F} \subseteq \{f_i^k \mid i, k \geq 0\}$ von Funktionssymbolen existiert mit:

- $\mathcal{F}$ enthält eine Konstante.
- $U_\mathcal{A} = D(\mathcal{F})$
- Ein Funktionssymbol f gehört zu $\text{dom}(I_\mathcal{A})$ genau dann, wenn $f \in \mathcal{F}$.
- Für alle $f \in \mathcal{F}$ (n-stellig) und $t_1, t_2, \ldots, t_n \in D(\mathcal{F})$ gilt $f^\mathcal{A}(t_1, t_2, \ldots, t_n) = f(t_1, t_2, \ldots, t_n)$

Für jede Herbrand-Struktur $\mathcal{A}$ wie oben und alle $t \in D(\mathcal{F})$ gilt $\mathcal{A}(t) = t$.

Sei M eine Menge von Formeln. Ein Herbrand-Modell von M ist ein Modell von M, welches gleichzeitig eine Herbrand-Struktur ist.

Der fundamentale Satz der Prädikatenlogik:

Satz

Sei M eine Menge von Aussagen in Skolemform.
M ist genau dann erfüllbar, wenn M ein Herbrand-Modell besitzt.

M erfüllbar gdw. M hat ein Herbrand-Modell

Beweis:

Falls M ein Herbrand-Modell hat, ist M natürlich erfüllbar.

Sei nun M erfüllbar und sei $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$ ein Modell von M.

O.B.d.A. gehört eine Konstante zum Definitionsbereich von $I_\mathcal{A}$.

Sei $\mathcal{F}$ die Menge aller Funktionssymbole, die im Definitionsbereich von $I_\mathcal{A}$ liegen.

Wir definieren nun ein Herbrand-Modell $\mathcal{B} = (D(\mathcal{F}), I_\mathcal{B})$:

Wir müssen $I_\mathcal{B}$ noch auf den Prädikatensymbolen in F definieren.

Für alle n-stelligen Prädikatensymbole P, die in M vorkommen, und alle $t_1, \ldots, t_n \in D(\mathcal{F})$ sei:

$$(t_1, \ldots, t_n) \in P^\mathcal{B} \text{ gdw. } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n)) \in P^\mathcal{A}$$

M erfüllbar gdw. M hat ein Herbrand-Modell

Behauptung: Für jede Aussage F in Skolemform, die aus den in M vorkommenden Funktions- und Prädikatensymbolen aufgebaut ist, gilt: Wenn $\mathcal{A}(F) = 1$, dann auch $\mathcal{B}(F) = 1$.

Falls F keine Quantoren enthält, zeigen wir sogar $\mathcal{A}(F) = \mathcal{B}(F)$ durch Induktion über den Aufbau von F:

- F ist atomar, d. h. $F = P(t_1, \ldots, t_n)$ für ein Prädikatensymbol P und variablenfreie Terme $t_1, \ldots, t_n \in D(\mathcal{F})$.
  (beachte: F ist eine Aussage, d. h. F enthält keine freien Variablen).

$$\begin{aligned} \mathcal{A}(F) = 1 \quad & \textbf{gdw.} \quad (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n)) \in P^{\mathcal{A}} \\ & \textbf{gdw.} \quad (t_1, \ldots, t_n) \in P^{\mathcal{B}} \\ & \textbf{gdw.} \quad (\mathcal{B}(t_1), \ldots, \mathcal{B}(t_n)) \in P^{\mathcal{B}} \\ & \textbf{gdw.} \quad \mathcal{B}(F) = 1 \end{aligned}$$

- $F = \neg G$: $\mathcal{A}(F) = 1$ **gdw.** $\mathcal{A}(G) = 0$ **gdw.** $\mathcal{B}(G) = 0$ **gdw.** $\mathcal{B}(F) = 1$
- $F = F_1 \wedge F_2$ oder $F = F_1 \vee F_2$: Analoges Argument wie für $F = \neg G$.

M erfüllbar gdw. M hat ein Herbrand-Modell

Damit ist der Fall, dass F keine Quantoren enthält, abgehandelt.

Den allgemeinen Fall behandeln wir durch Induktion über die Anzahl n der Quantoren in F.

Beachte: Da F in Skolemform ist, ist F von der Form $\forall y_1 \cdots \forall y_n H$, wobei H keine Quantoren enthält.

Induktionsanfang: $n = 0$:

Aus $\mathcal{A}(F) = 1$ folgt $\mathcal{B}(F) = 1$, siehe vorherige Folie.

Induktionsschritt: Sei $F = \forall x G$.

Aus $\mathcal{A}(\forall x G) = 1$ folgt

$$\text{für alle } d \in U_{\mathcal{A}} \text{ gilt } \mathcal{A}_{[x/d]}(G) = 1$$

Wegen $\{\mathcal{A}(t) \mid t \in D(\mathcal{F})\} \subseteq U_{\mathcal{A}}$ folgt:

$$\text{für alle } t \in D(\mathcal{F}) \text{ gilt } \mathcal{A}_{[x/\mathcal{A}(t)]}(G) = 1$$

M erfüllbar gdw. M hat ein Herbrand-Modell

Mit dem Überführungslemma folgt:

$$\text{für alle } t \in D(\mathcal{F}) \text{ gilt } \mathcal{A}(G[x/t]) = 1.$$

Die Aussage $G[x/t]$ ist wieder in Skolemform, und hat nur $n-1$ Quantoren.

Nach Induktionsannahme gilt daher:

$$\text{für alle } t \in D(\mathcal{F}) \text{ gilt } \mathcal{B}(G[x/t]) = 1.$$

Mit dem Überführungslemma folgt wieder:

$$\text{für alle } t \in D(\mathcal{F}) \text{ gilt } \mathcal{B}_{[x/\mathcal{B}(t)]}(G) = \mathcal{B}_{[x/t]}(G) = 1$$

und damit $\mathcal{B}(F) = \mathcal{B}(\forall x G) = 1$. □

Bemerkung: Im soeben durchgeführten Beweis ist wichtig, dass alle $F \in M$ in Skolemform sind, und damit keine Existenzquantoren enthalten.

Der Satz von Löwenheim und Skolem

Eine Menge A ist abzählbar, falls A endlich ist, oder eine Bijektion $f : \mathbb{N} \to A$ existiert.

Beachte: Das Universum $D(\mathcal{F})$ einer Herbrand-Struktur ist abzählbar.

Satz von Löwenheim und Skolem

Jede erfüllbare Menge von Aussagen der Prädikatenlogik besitzt ein Modell mit einer abzählbaren Grundmenge (ein abzählbares Modell).

Beweis: Sei M eine erfüllbare Menge von Aussagen der Prädikatenlogik.

Sei M' die Skolemform von M.

Satz auf Folie 153 $\implies$ M' ist erfüllbar.

Satz auf Folie 159 $\implies$ M' hat ein Herbrand-Modell $\mathcal{B}$.

Bemerkung auf Folie 152 $\implies$ $\mathcal{B}$ ist auch ein Modell von M.

Außerdem ist $\mathcal{B}$ abzählbar. □

Herbrand-Expansion

Sei M eine Menge von Aussagen in Skolemform.

Sei $\mathcal{F}$ die Menge der Funktionssymbole, die in Formeln aus M vorkommen, und sei a eine fest gewählte Konstante.

Wir definieren:

$$D(M) = \begin{cases} D(\mathcal{F}) & \text{falls } \mathcal{F} \text{ eine Konstante enthält} \\ D(\mathcal{F} \cup \{a\}) & \text{sonst} \end{cases}$$

Die Menge von Aussagen

$$E(M) = \{F^*[y_1/t_1][y_2/t_2]\ldots[y_n/t_n] \mid \forall y_1 \forall y_2 \cdots \forall y_n F^* \in M, \\ t_1, t_2, \ldots, t_n \in D(M)\}$$

ist die Herbrand-Expansion von M.

Die Formeln in $E(M)$ entstehen also, indem die Terme aus $D(M)$ in jeder möglichen Weise für die Variablen in F^* ($F \in M$) substituiert werden.

Herbrand-Expansion

Beachte: Wenn M erfüllbar ist, gibt es ein Herbrand-Modell von M mit Universum $D(M)$.

Beispiel: Für $M = \{\forall x \forall y (P(a, x) \wedge \neg R(f(y)))\}$ gilt

$$D(M) = \{a, f(a), f(f(a)), \ldots\}.$$

Die Herbrand-Expansion von M ist damit

$$\begin{aligned} E(M) = \{ & P(a, a) \wedge \neg R(f(a)), \\ & P(a, f(a)) \wedge \neg R(f(a)), \\ & P(a, a) \wedge \neg R(f(f(a))), \\ & P(a, f(a)) \wedge \neg R(f(f(a))), \ldots\} \end{aligned}$$

Herbrand-Expansion

Wir betrachten die Herbrand-Expansion von M im folgenden als eine Menge von aussagenlogischen Formeln.

Die atomaren Formeln sind hierbei von der Gestalt $P(t_1, \ldots, t_n)$, wobei P ein in M vorkommendes Prädikatensymbol ist und $t_1, \ldots, t_n \in D(M)$.

Im Beispiel auf der vorherigen Folie kommen in der Herbrand-Expansion

$$E(M) = \{P(a, f^n(a)) \wedge \neg R(f^m(a)) \mid n \geq 0, m \geq 1\}$$

(hierbei ist $f^n(a)$ eine Abkürzung für den Term $f(f(\cdots f(a)\cdots))$, wobei f genau n-mal vorkommt) genau die atomoren Formeln aus der Menge

$$\{P(a, f^n(a)) \mid n \geq 0\} \cup \{R(f^m(a)) \mid m \geq 1\}$$

vor. Die Belegung $\mathcal{B}$ mit

$$\mathcal{B}(P(a, f^n(a))) = 1 \text{ für } n \geq 0 \quad \text{und} \quad \mathcal{B}(R(f^m(a))) = 0 \text{ für } m \geq 0$$

erfüllt offenbar auch $E(M)$ im aussagenlogischen Sinn: $\mathcal{B}(G) = 1$ für alle $G \in E(M)$.

Herbrand-Expansion

Auch die (einzige) Formel $\forall x \forall y (P(a, x) \wedge \neg R(f(y))) \in M$ ist erfüllbar.
Wie der folgende Satz zeigt, ist die kein Zufall.

Satz von Gödel-Herbrand-Skolem

Satz von Gödel-Herbrand-Skolem

Sei M eine Menge von Aussagen in Skolemform. Dann ist M ist erfüllbar genau dann, wenn die Formelmenge $E(M)$ (im aussagenlogischen Sinn) erfüllbar ist.

Beweis: Es genügt zu zeigen, daß M ein Herbrand-Modell mit Universum $D(M)$ besitzt genau dann, wenn $E(F)$ erfüllbar ist:

$\mathcal{A}$ *ist ein Herbrand-Modell für* M *mit Universum* $D(M)$

gdw. *für alle* $\forall y_1 \forall y_2 \cdots \forall y_n F^* \in M,\ t_1, t_2, \ldots, t_n \in D(M)$ *gilt*

$$\mathcal{A}_{[y_1/t_1][y_2/t_2]\ldots[y_n/t_n]}(F^*) = 1$$

gdw. *für alle* $\forall y_1 \forall y_2 \cdots \forall y_n F^* \in M,\ t_1, t_2, \ldots, t_n \in D(M)$ *gilt*

$$\mathcal{A}(F^*[y_1/t_1][y_2/t_2]\ldots[y_n/t_n]) = 1$$

gdw. *für alle* $G \in E(F)$ *gilt* $\mathcal{A}(G) = 1$

gdw. $\mathcal{A}$ *ist ein Modell für* $E(F)$

Endlichkeitssatz der Prädikatenlogik

Endlichkeitssatz der Prädikatenlogik (Gödel 1930)

Eine Menge M von prädikatenlogischen Formeln ist erfüllbar genau dann, wenn jede endliche Teilmenge von M erfüllbar ist.

Beweis: Es genügt die "wenn"-Richtung zu zeigen.

Nach dem Satz von Folie 153 können wir annehmen, dass M eine Menge von Aussagen in Skolemform ist.

Sei jede endliche Teilmenge N von M erfüllbar.

Satz von Gödel-Herbrand-Skolem $\Rightarrow$ Für jede endliche Teilmenge $N \subseteq M$ ist die Herbrand-Expansion $E(N)$ im aussagenlogischen Sinn erfüllbar.

Insbesondere ist jede endliche Teilmenge von $E(M)$ im aussagenlogischen Sinn erfüllbar.

Endlichkeitssatz der Aussagenlogik (Folie 104) $\Rightarrow$ $E(M)$ erfüllbar.

Satz von Gödel-Herbrand-Skolem $\Rightarrow$ M erfüllbar. □

Satz von Herbrand

Satz von Herbrand

Eine Aussage F in Skolemform ist unerfüllbar genau dann, wenn es eine endliche Teilmenge von $E(F)$ gibt, die (im aussagenlogischen Sinn) unerfüllbar ist.

Beweis: Ummittelbare Folge des Satzes von Gödel-Herbrand-Skolem und des Endlichkeitssatzes der Aussagenlogik (Folie 104). □

Algorithmus von Gilmore

Sei F eine prädikatenlogische Aussage in Skolemform und sei $\{F_1, F_2, F_3, \ldots,\}$ eine Aufzählung von $E(F)$.

Algorithmus von Gilmore

Eingabe: F
$n := 0$;
repeat $n := n + 1$;
until $(F_1 \wedge F_2 \wedge \ldots \wedge F_n)$ ist unerfüllbar;
Gib "unerfüllbar" aus und stoppe.

Die gültigen Formeln sind semi-entscheidbar

Aus dem Satz von Herbrand folgt:

Satz

Sei F eine prädikatenlogische Aussage in Skolemform. Dann gilt:

- Wenn die Eingabeformel F unerfüllbar ist, dann terminiert der Algorithmus von Gilmore nach endlicher Zeit mit dem Output "unerfüllbar".
- Wenn die Eingabeformel F erfüllbar ist, dann terminiert der Algorithmus von Gilmore nicht, d. h. er läuft unendlich lange.

Die Menge der unerfüllbaren prädikatenlogischen Aussagen ist also semi-entscheidbar (rekursive aufzählbar) (siehe GTI).

Korollar

Die Menge der gültigen prädikatenlogischen Aussagen ist semi-entscheidbar.

Das Erfüllbarkeitsproblem ist nicht entscheidbar

Beweis: F is gültig, genau dann, wenn $\neg F$ unerfüllbar ist.

Wir werden noch sehen: Es gibt keinen Algorithmus, der als Eingabe eine prädikatenlogische Aussage bekommt, und folgende Eigenschaften hat:

- Wenn F erfüllbar ist, dann terminiert der Algorithmus nach endlicher Zeit mit dem Output "erfüllbar".
- Wenn F unerfüllbar ist, dann terminiert der Algorithmus nach endlicher Zeit mit dem Output "unerfüllbar".

Anders gesagt: Die Menge der (un)erfüllbaren prädikatenlogischen Aussagen ist unentscheidbar.

Resolution in der Prädikatenlogik

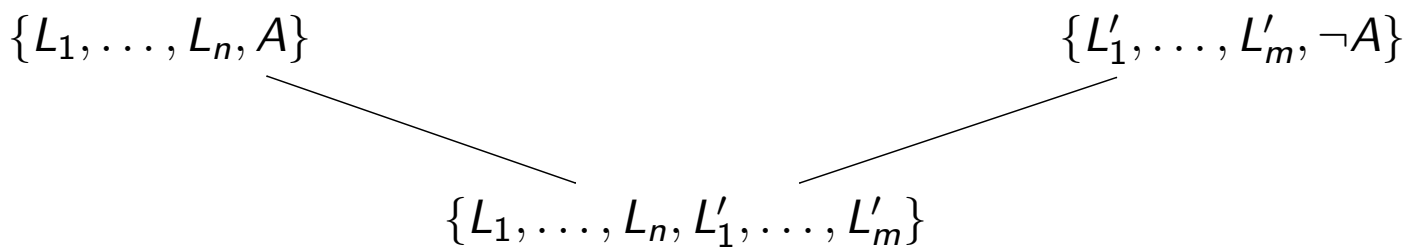
Der Algorithmus von Gilmore funktioniert zwar, ist in der Praxis aber unbrauchbar.

Daher ist unser Programm der nächsten Stunden:

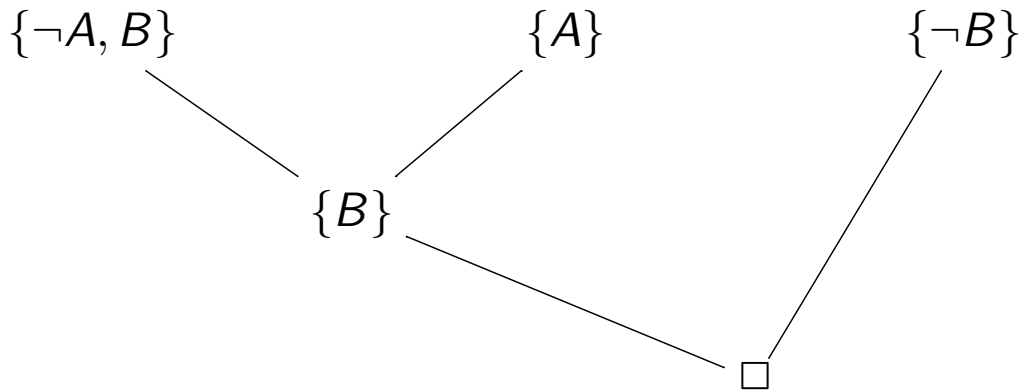
Wie sieht Resolution in der Prädikatenlogik aus?

Wiederholung: Resolution in der Aussagenlogik

Resolutionsschritt:



Mini-Beispiel:



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

Anpassung des Algorithmus von Gilmore

Algorithmus von Gilmore:
Sei F eine prädikatenlogische Aussage in Skolemform und sei $\{F_1, F_2, F_3, \ldots\}$ eine Aufzählung von $E(F)$.

Eingabe: F
$n := 0$;
repeat $n := n + 1$;
until $(F_1 \wedge F_2 \wedge \ldots \wedge F_n)$ ist unerfüllbar;
(dies kann mit Mitteln der Aussagenlogik, z.B. Wahrheitstafeln, getestet werden)
Gib "unerfüllbar" aus und stoppe.

"Mittel der Aussagenlogik" $\rightsquigarrow$ wir verwenden Resolution für den Unerfüllbarkeitstest

Definition von $Res(F)$ (Wiederholung)

Definition: Sei F eine Klauselmenge. Dann ist $Res(F)$ definiert als

$$Res(F) = F \cup \{R \mid R \text{ ist Resolvent zweier Klauseln in } F\}.$$

Außerdem setzen wir:

$$\begin{aligned} Res^0(F) &= F \\ Res^{n+1}(F) &= Res(Res^n(F)) \qquad \text{für } n \geq 0 \end{aligned}$$

Schließlich sei

$$Res^*(F) = \bigcup_{n \geq 0} Res^n(F).$$

Grundresolutionsalgorithmus

Sei $F_1, F_2, F_3, \ldots$ wieder eine Aufzählung der Herbrand-Expansion von F.

F sei in Klauselform, d.h. $F = \forall y_1 \forall y_2 \cdots \forall y_n F^*$, wobei F^* in **KNF** ist.

Wir betrachten F^* als eine Menge von Klauseln, dann ist auch jedes F_i eine Menge von Klauseln.

Eingabe: eine Aussage F in Skolemform mit der Matrix F^* in **KNF**
$i := 0$;
$M := \emptyset$;
repeat
$\quad i := i + 1$; $M := M \cup F_i$; $M := Res^*(M)$
until $\square \in M$
Gib "unerfüllbar" aus und stoppe.

Warum der Name Grundresolution?
Im Gegensatz zu späteren Verfahren werden Terme ohne Variable (= Grundterme) substituiert, um die Formeln der Herbrand-Expansion zu erhalten.

Grundresolutionssatz

Den Grundresolutionsalgorithmus kann man auch in folgenden Grundresolutionssatz umformulieren:

Grundresolutionssatz

Eine Aussage in Skolemform $F = \forall y_1 \ldots \forall y_k F^*$ mit der Matrix F^* in **KNF** ist unerfüllbar genau dann, wenn es eine Folge von Klauseln $K_1, \ldots, K_n$ gibt mit der Eigenschaft:

- K_n ist die leere Klausel
- Für alle $i \in \{1, \ldots, n\}$ gilt:
  - entweder ist K_i eine Grundinstanz einer Klausel $K \in F^*$,d.h. $K_i = K[y_1/t_1] \ldots [y_k/t_k]$ mit $t_i \in D(F)$
  - oder K_i ist (aussagenlogischer) Resolvent zweier Klauseln K_a, K_b mit $a < i$ und $b < i$

Weglassen von Klauseln und Resolutionsschritten, die nicht zur Herleitung der leeren Klausel beitragen.

Substitutionen

Eine Substitution sub ist eine Abbildung von einer endlichen Menge von Variablen in die Menge aller Terme.

Sei $\mathrm{Def}(\mathrm{sub})$ der Definitionsbereich der Substitution sub.

Für einen Term t definieren wir den Term $t\,\mathrm{sub}$ (Anwendung der Substitution sub auf den Term t) wie folgt induktiv.

- $x\,\mathrm{sub} = \mathrm{sub}(x)$, falls $x \in \mathrm{Def}(\mathrm{sub})$.
- $y\,\mathrm{sub} = y$, falls $y \notin \mathrm{Def}(\mathrm{sub})$.
- $f(t_1, \ldots, t_n)\,\mathrm{sub} = f(t_1\,\mathrm{sub}, \ldots, t_n\,\mathrm{sub})$ für Terme $t_1, \ldots, t_n$ und ein n-stelliges Funktionssymbol f
  (dies impliziert $a\,\mathrm{sub} = a$, falls a eine Konstante ist)

Für ein Literal F (= evtl. negierte atomare Formel) definieren wir $F\,\mathrm{sub}$ wie folgt, wobei P ein n-stelliges Prädikatensymbol ist, und $t_1, \ldots, t_n$ Terme sind:

$$\begin{aligned} P(t_1, \ldots, t_n)\,\mathrm{sub} &= P(t_1\,\mathrm{sub}, \ldots, t_n\,\mathrm{sub}) \\ \neg P(t_1, \ldots, t_n)\,\mathrm{sub} &= \neg P(t_1\mathrm{sub}, \ldots, t_n\mathrm{sub}) \end{aligned}$$

Substitutionen und Ersetzungen

Eine Ersetzung $[x/t]$ (x ist eine Variable, t ein Term) kann mit der Substitution sub mit $\mathrm{Def}(\mathrm{sub}) = \{x\}$ und $\mathrm{sub}(x) = t$ identifiziert werden.

Eine Substitution sub mit $\mathrm{Def}(\mathrm{sub}) = \{y_1, \ldots, y_n\}$ (jedes y_i ist eine Variable) kann auch als Folge von Ersetzungen $[y_1/t_1][y_2/t_2]\cdots[y_n/t_n]$ geschrieben werden.

Beachte: Ersetzungen werden von links nach rechts durchgeführt!

Beispiel: Die Substitution sub mit $\mathrm{Def}(\mathrm{sub}) = \{x, y, z\}$ und

$$\mathrm{sub}(x) = f(h(w)), \quad \mathrm{sub}(y) = g(a, h(w)), \quad \mathrm{sub}(z) = h(w)$$

ist gleich der Substitution

$$[x/f(z)]\,[y/g(a,z)]\,[z/h(w)].$$

Verknüpfung von Substitutionen: Bei $\mathrm{sub}_1\mathrm{sub}_2$ wird zuerst sub_1 angewandt, anschließend sub_2.

Vertauschen von Substitutionen

Lemma (Regel für das Vertauschen von Substitutionen)

Falls (i) $x \notin \mathrm{Def}(\mathrm{sub})$ und (ii) x in keinem der Terme $y\,\mathrm{sub}$ mit $y \in \mathrm{Def}(\mathrm{sub})$ vorkommt, so gilt

$$[x/t]\mathrm{sub} = \mathrm{sub}[x/t\,\mathrm{sub}].$$

Beispiele:

- $[x/f(y)]\underbrace{[y/g(z)]}_{\mathrm{sub}} = \underbrace{[y/g(z)]}_{\mathrm{sub}}[x/f(g(z))]$
- aber $[x/f(y)]\underbrace{[x/g(z)]}_{\mathrm{sub}} \neq \underbrace{[x/g(z)]}_{\mathrm{sub}}[x/f(y)]$

Vertauschen von Substitutionen

Beweis des Lemmas:

Wir zeigen $t'[x/t]\mathrm{sub} = t'\mathrm{sub}[x/t\,\mathrm{sub}]$ für alle Terme t' durch Induktion über den Aufbau von t'.

- $t' = x$:
  Dann gilt $x[x/t]\mathrm{sub} = t\,\mathrm{sub}$ und ebenso
  $x\,\mathrm{sub}[x/t\,\mathrm{sub}] = x[x/t\,\mathrm{sub}] = t\,\mathrm{sub}$, da $x\,\mathrm{sub} = x$ wegen
  $x \notin \mathrm{Def}(\mathrm{sub})$.
- $t' = y$ für eine Variable $y \neq x$:
  Dann gilt $y[x/t]\mathrm{sub} = y\,\mathrm{sub}$ und ebenso $y\,\mathrm{sub}[x/t\,\mathrm{sub}] = y\,\mathrm{sub}$, da x in $y\,\mathrm{sub}$ nicht vorkommt.
- $t' = f(t_1, \ldots, t_n)$:
  Diesen Fall kann man sofort mit der Induktionsannahme für $t_1, \ldots, t_n$ erledigen. □

Unifikator/Allgemeinster Unifikator

Gegeben sei eine Menge $\mathbf{L} = \{L_1, \ldots, L_k\}$ von Literalen (= evtl. negierte atomare prädikatenlogische Formeln).

Eine Substitution sub heißt Unifikator von $\mathbf{L}$, falls

$$L_1\mathrm{sub} = L_2\mathrm{sub} = \cdots = L_k\mathrm{sub}$$

Das ist gleichbedeutend mit $|\mathbf{L}\,\mathrm{sub}| = 1$, wobei $\mathbf{L}\,\mathrm{sub} = \{L_1\mathrm{sub}, \ldots, L_k\mathrm{sub}\}$.

Ein Unifikator sub von $\mathbf{L}$ heißt allgemeinster Unifikator von $\mathbf{L}$, falls für jeden Unifikator sub' von $\mathbf{L}$ eine Substitution s mit $\mathrm{sub}' = \mathrm{sub}\,s$ existiert.

Aufgabe

Unifizierbar?			Ja	Nein
	$P(f(x))$	$P(g(y))$		
	$P(x)$	$P(f(y))$		
	$P(x, f(y))$	$P(f(u), z)$		
	$P(x, f(y))$	$P(f(u), f(z))$		
	$P(x, f(x))$	$P(f(y), y)$		
	$P(x, g(x), g^2(x))$	$P(f(z), w, g(w))$		
$P(x, f(y))$	$P(g(y), f(a))$	$P(g(a), z)$		

Unifikationsalgorithmus

Eingabe: eine endliche Literalmenge $\mathbf{L} \neq \emptyset$

$\mathrm{sub} := []$; (leere Substitution, d. h. $\mathrm{Def}([]) = \emptyset$)
while $|\mathbf{L}\,\mathrm{sub}| > 1$ **do**
    Suche die erste Position, an der sich zwei Literale $L_1, L_2 \in \mathbf{L}\,\mathrm{sub}$ unterscheiden
    if keines der beiden Symbole an dieser Position ist eine Variable **then**
        stoppe mit "nicht unifizierbar"
    else sei x die Variable und t der Term im anderen Literal (möglicherweise auch eine Variable)
        if x kommt in t vor **then**
            stoppe mit "nicht unifizierbar"
        else $\mathrm{sub} := \mathrm{sub}\,[x/t]$
endwhile

Ausgabe: sub

Korrektheit des Unifikationsalgorithmus

Satz

Es gilt:

(A) Der Unifikationsalgorithmus terminiert für jede Eingabe **L**.

(B) Wenn die Eingabe **L** nicht unifizierbar ist, so terminiert der Unifikationsalgorithmus mit der Ausgabe "nicht unifizierbar".

(C) Wenn die Eingabe **L** unifizierbar ist, dann findet der Unifikationsalgorithmus immer einen allgemeinsten Unifikator von **L**.

(C) impliziert insbesondere, daß jede unifizierbare Menge von Literalen einen allgemeinsten Unifikator hat.

Beweis der Korrektheit des Unifikationsalgorithmus

Beweis:

(A) Der Unifikationsalgorithmus terminiert für jede Eingabe **L**.

Dies gilt, denn die Anzahl der in $\mathbf{L}\,\mathrm{sub}$ vorkommenden Variablen wird in jedem Schritt kleiner.

Betrachte hierzu einen Durchlauf durch die **while**-Schleife.

Falls der Algorithmus in diesem Durchlauf nicht terminiert, so wird sub auf $\mathrm{sub}\,[x/t]$ gesetzt.

Hierbei kommt x in $\mathbf{L}\,\mathrm{sub}$ vor und der Term t enthält x nicht.

Also kommt x in $\mathbf{L}\,\mathrm{sub}\,[x/t]$ nicht mehr vor.

Beweis der Korrektheit des Unifikationsalgorithmus

(B) Wenn die Eingabe **L** nicht unifizierbar ist, so terminiert der Unifikationsalgorithmus mit der Ausgabe "nicht unifizierbar".

Sei die Eingabe **L** nicht unifizierbar.

Falls die Bedingung $|\mathbf{L}\,\mathrm{sub}| > 1$ der **while**-Schleife irgendwann verletzt wäre, so wäre **L** doch unifizierbar.

Da nach (A) der Algorithmus bei Eingabe **L** terminiert, muss schließlich "nicht unifizierbar" ausgegeben werden.

Beweis der Korrektheit des Unifikationsalgorithmus

(C) Wenn die Eingabe **L** unifizierbar ist, dann findet der Unifikationsalgorithmus immer einen allgemeinsten Unifikator von **L**.

Sei **L** unifizierbar und sei sub_i ($i \geq 0$) die nach dem i-ten Durchlauf der **while**-Schleife berechnete Substituiton.

Angenommen der Algorithmus macht N Durchläufe durch die **while**-Schleife.

Beachte: $\mathrm{sub}_0 := []$ und sub_N ist die Ausgabe des Algorithmus (falls diese existiert).

Behauptung:

(1) Für jeden Unifikator sub' von **L** und für alle $0 \leq i \leq N$ existiert eine Substitution s_i mit $\mathrm{sub}' = \mathrm{sub}_i \; s_i$.

(2) Im i-ten Durchlauf durch die **while**-Schleife ($1 \leq i \leq N$) terminiert der Algorithmus entweder erfolgreich (und gibt die Substitution sub_N aus) oder der Algotrithmus betritt die beiden **else**-Zweige.

Beweis der Korrektheit des Unifikationsalgorithmus

Beweis der Behauptung:

Sei sub' ein Unifikator von **L**.

Zunächst betrachten wir durch Induktion über i den Fall, dass **L** und $\{y\,\mathrm{sub}' \mid y \in \mathrm{Def}(\mathrm{sub}')\}$ keine gemeinsamen Variablen enthalten.

Wir werden s_i als eine Einschränkung von sub' wählen, d. h. $\mathrm{Def}(s_i) \subseteq \mathrm{Def}(\mathrm{sub}')$ und $s_i(x) = \mathrm{sub}'(x)$ für alle $x \in \mathrm{Def}(s_i)$.

Induktionsanfang: $i = 0$.

Es gilt: $\mathrm{sub}' = []\,\mathrm{sub}' = \mathrm{sub}_i\,\mathrm{sub}'$. Wir können also $s_0 = \mathrm{sub}'$ setzen.

Beweis der Korrektheit des Unifikationsalgorithmus

Induktionsschritt: Sei $i > 0$ und sei (1) und (2) bereits für $i-1$ bewiesen.

Nach Induktionshypothese existiert eine Einschränkung s_{i-1} von sub' mit $\mathrm{sub}' = \mathrm{sub}_{i-1}s_{i-1}$.

Falls $|\mathbf{L}\,\mathrm{sub}_{i-1}| = 1$, so terminiert der Algorithmus im i-ten Durchlauf.

Sei nun $|\mathbf{L}\,\mathrm{sub}_{i-1}| > 1$.

Betrachte die erste Position p, an der sich zwei Literale L_1 und L_2 aus $\mathbf{L}\mathrm{sub}_{i-1}$ unterscheiden.

Wegen $|\mathbf{L}\,\mathrm{sub}_{i-1}s_{i-1}| = |\mathbf{L}\,\mathrm{sub}'| = 1$ gilt $L_1s_{i-1} = L_2s_{i-1}$.

Also können an Position p in L_1 und L_2 nicht zwei verschiedene Funktionssymbole stehen.

Stehe in L_1 an Position p etwa eine Variable x und in L_2 an Position p ein Term $t \neq x$.

Beweis der Korrektheit des Unifikationsalgorithmus

Dann gilt $x\, s_{i-1} = t\, s_{i-1}$.

In t kann die Variable x nicht vorkommen:

Dies ist klar, wenn t eine Variable (da $t \neq x$) oder Konstante ist.

Ist t von der Form $f(t_1, \ldots, t_n)$ mit $n \geq 1$, so muss
$x\, s_{i-1} = t\, s_{i-1} = f(t_1 s_{i-1}, \ldots, t_n s_{i-1})$ gelten.

Würde x in einem der Terme t_i vorkommen, so würde $f(t_1 s_{i-1}, \ldots, t_n s_{i-1})$ mehr Symbole als $x\, s_{i-1}$ enthalten.

Also werden die beiden **else**-Zweige im Rumpf der **while**-Schleife betreten (dies zeigt (2)).

Es gilt $\mathrm{sub}_i = \mathrm{sub}_{i-1}[x/t]$.

Sei s_i die Einschränkung von s_{i-1} auf alle von x verschiedenen Variablen.

Beweis der Korrektheit des Unifikationsalgorithmus

Dann gilt:

$$
\begin{aligned}
\mathrm{sub}_i\, s_i &= \mathrm{sub}_{i-1}\,[x/t]\, s_i \\
&= \mathrm{sub}_{i-1}\, s_i\,[x/t s_i] && \text{(denn } x \notin \mathrm{Def}(s_i) \text{ und } x \text{ kommt in keinem der} \\
& && \text{Terme } y\, s_i \text{ für } y \in \mathrm{Def}(s_i) \text{ vor)} \\
&= \mathrm{sub}_{i-1}\, s_i\,[x/t\, s_{i-1}] && \text{(denn } x \text{ kommt in } t \text{ nicht vor)} \\
&= \mathrm{sub}_{i-1}\, s_{i-1} && \text{(wegen } x\, s_{i-1} = t\, s_{i-1} \text{ und Def. von } s_i\text{)} \\
&= \mathrm{sub}' && \text{(Induktionshypothese)}
\end{aligned}
$$

Dies zeigt (1).

Der Fall, dass **L** und $\{y\,\mathrm{sub}' \mid y \in \mathrm{Def}(\mathrm{sub}')\}$ keine gemeinsamen Variablen enthalten, ist damit abgeschlossen.

Beweis der Korrektheit des Unifikationsalgorithmus

Im allgemeinen Fall (sub' ist ein beliebiger Unifikator von **L**) sei X die Menge aller Variablen, die in $\{y\,\mathrm{sub}' \mid y \in \mathrm{Def}(\mathrm{sub}')\}$ vorkommen.

Sei Y eine Menge von Variablen mit $|X| = |Y|$, so dass Y und **L** keine gemeinsamen Variablen enthalten.

Sei u eine beliebige Bijektion zwischen X und Y.
Wir nennen u eine Variablenumbenennung.

Dann ist auch $\mathrm{sub}'u$ ein Unifikator von **L**, so dass **L** und $\{y\,\mathrm{sub}'u \mid y \in \mathrm{Def}(\mathrm{sub}')\}$ keine gemeinsame Variablen enthalten.

Also gibt es für alle $0 \leq i \leq N$ eine Substitution s_i mit $\mathrm{sub}'u = \mathrm{sub}_i\ s_i$.

Also gilt $\mathrm{sub}' = \mathrm{sub}_i(s_i u^{-1})$.

Beweis der Korrektheit des Unifikationsalgorithmus

Aus (1) und (2) folgt nun:

Der Algorithmus terminiert nach N Durchläufen durch die **while**-Schleife mit einem Unifikator $\mathrm{sub} = \mathrm{sub}_N$.

Ist sub' ein beliebiger Unifikator von **L**, so existiert wegen (1) eine Substitution s mit $\mathrm{sub}' = \mathrm{sub}\, s$.

Also ist sub ein allgemeinster Unifikator von **L**. □

Beispiel zum Unifikationsalgorithmus

Betrachte $\mathbf{L} = \{P(f(z, g(a, y)), h(z)),\ P(f(f(u, v), w), h(f(a, b)))\}$

Beispiel zum Unifikationsalgorithmus

Betrachte $\mathbf{L} = \{P(f(z, g(a, y)), h(z)),\ P(f(f(u, v), w), h(f(a, b)))\}$

$P(f(z, g(a, y)), h(z))$
$P(f(f(u, v), w), h(f(a, b)))$ $\quad \text{sub} = []$

$P(f(f(u, v), g(a, y)), h(f(u, v)))$
$P(f(f(u, v), w), h(f(a, b)))$ $\quad \text{sub} = [z/f(u, v)]$

$P(f(f(u, v), g(a, y)), h(f(u, v)))$
$P(f(f(u, v), w), h(f(a, b)))$ $\quad \text{sub} = [z/f(u, v)]$

$P(f(f(u, v), g(a, y)), h(f(u, v)))$
$P(f(f(u, v), g(a, y)), h(f(a, b)))$ $\quad \text{sub} = [z/f(u, v)][w/g(a, y)]$

Beispiel zum Unifikationsalgorithmus

$P(f(f(u,v),g(a,y)),h(f(u,v)))$

$P(f(f(u,v),g(a,y)),h(f(a,b)))$ $\quad \text{sub} = [z/f(u,v)][w/g(a,y)]$

$P(f(f(a,v),g(a,y)),h(f(a,v)))$

$P(f(f(a,v),g(a,y)),h(f(a,b)))$ $\quad \text{sub} = [z/f(u,v)][w/g(a,y)][u/a]$

$P(f(f(a,v),g(a,y)),h(f(a,v)))$

$P(f(f(a,v),g(a,y)),h(f(a,b)))$ $\quad \text{sub} = [z/f(u,v)][w/g(a,y)][u/a]$

$P(f(f(a,b),g(a,y)),h(f(a,b)))$

$P(f(f(a,b),g(a,y)),h(f(a,b)))$ $\quad \text{sub} = [z/f(u,v)][w/g(a,y)][u/a][v/b]$

Komplexität des Unifikationsalgorithmus

Zwar ist die Anzahl der Durchläufe durch die while-Schleife in dem Unifikationsalgorithmus durch die Eingabelänge beschränkt, aber:

Durch das wiederholte Einsetzen von Termen knnen sehr große Terme entstehen.

In der Tat ist die Laufzeit unseres Unifikationsalgorithmus i.A. exponentiell in der Eingabelänge.

Andererseits gilt folgendes Resultat:

Paterson, Wegman 1976

Es gibt einen Unifikationsalgorithmus, dessen Laufzeit linear in der Eingabelänge beschränkt ist.

Prädikatenlogische Resolution

Eine Klausel R heißt prädikatenlogischer Resolvent zweier Klauseln K_1 und K_2, wenn folgendes gilt:

- Es gibt Variablenumbenennungen s_1 und s_2, so dass $K_1 s_1$ und $K_2 s_2$ keine gemeinsamen Variablen enthalten.
- Es gibt $m, n \geq 1$ und Literale $L_1, \ldots, L_m$ aus $K_1 s_1$ und Literale $L'_1, \ldots, L'_n$ aus $K_2 s_2$, so dass

$$\mathbf{L} = \{\overline{L_1}, \ldots, \overline{L_m}, L'_1, \ldots, L'_n\}$$

  unifizierbar ist. Sei sub ein allgemeinster Unifikator von $\mathbf{L}$.
  ($\overline{L}$ bezeichnet das zu L negierte Literal)
- Es gilt

$$R = ((K_1 s_1 - \{L_1, \ldots, L_m\}) \cup (K_2 s_2 - \{L'_1, \ldots, L'_n\}))\mathrm{sub}.$$

Beispiel für prädikatenlogischen Resolventen

Sei

$$\begin{aligned} K_1 &= \{P(f(x)), \neg Q(z), P(z)\} \\ K_2 &= \{\neg P(x), R(g(x), a)\} \end{aligned}$$

Für die Variablenumbenennungen $s_1 = []$ und $s_2 = [x/u]$ gilt:

$$\begin{aligned} K_1 s_1 &= \{P(f(x)), \neg Q(z), P(z)\} \\ K_2 s_2 &= \{\neg P(u), R(g(u), a)\} \end{aligned}$$

Diese Klauseln haben keine gemeinsamen Variablen.

Sei $L_1 = P(f(x)) \in K_1 s_1$, $L_2 = P(z) \in K_1 s_1$ und $L_1' = \neg P(u) \in K_2 s_2$.

Die Menge $\mathbf{L} = \{\overline{L_1}, \overline{L_2}, L_1'\} = \{\neg P(f(x)), \neg P(z), \neg P(u)\}$ ist unifizierbar.

Ein allgemeinster Unifikator ist $\mathrm{sub} = [z/f(x)][u/f(x)]$.

Somit ist

$$((K_1 s_1 - \{L_1, L_2\}) \cup (K_2 s_2 - \{L_1'\}))\mathrm{sub} = \{\neg Q(f(x)), R(g(f(x)), a)\}$$

Resolvent von K_1 und K_2.

Korrektheit und Vollständigkeit

Zwei Fragen:

- Wenn man mit prädikatenlogischer Resolution aus einer Formel F die leere Klausel $\square$ ableiten kann, ist F dann unerfüllbar? (Korrektheit)
- Kann man für eine unerfüllbare Formel F immer durch prädikatenlogische Resolution die leere Klausel herleiten? (Vollständigkeit)

Aufgabe

Sind diese Klauseln resolvierbar?
Wieviele mögliche Resolventen gibt es?

K_1	K_2	Möglichkeiten
$\{P(x), Q(x,y)\}$	$\{\neg P(f(x))\}$	
$\{Q(g(x)), R(f(x))\}$	$\{\neg Q(f(x))\}$	
$\{P(x), P(f(x))\}$	$\{\neg P(y), Q(y,z)\}$	

Lifting-Lemma

Eine Grundinstanz eines Literals L ist ein Literal $L\mathrm{sub}$, welches keine Variablen enthält.

Eine Grundinstanz einer Klausel $K = \{L_1, \ldots, L_n\}$ ist ein Klausel $K\mathrm{sub} = \{L_1\mathrm{sub}, \ldots, L_n\mathrm{sub}\}$, welche keine Variablen enthält.

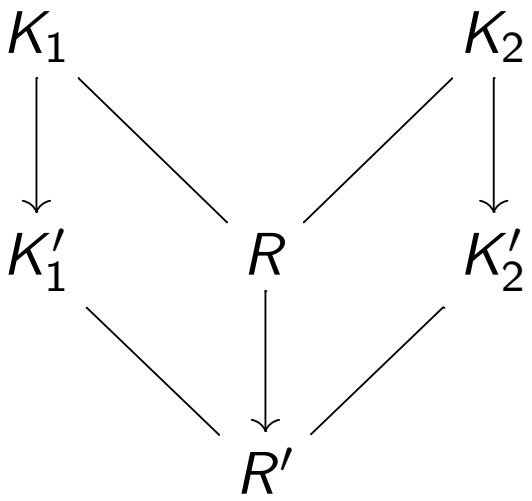
Beispiel: $P(f(a), f(f(a)), g(a, b))$ ist eine Grundinstanz des Literals $P(x, f(x), g(a, y))$.

Lifting-Lemma

Seien K_1, K_2 zwei prädikatenlogische Klauseln und seien K_1', K_2' zwei Grundinstanzen hiervon, die aussagenlogisch resolvierbar sind und den Resolventen R' ergeben.
Dann gibt es einen prädikatenlogischen Resolventen R von K_1, K_2, so dass R' eine Grundinstanz von R ist.

Lifting-Lemma

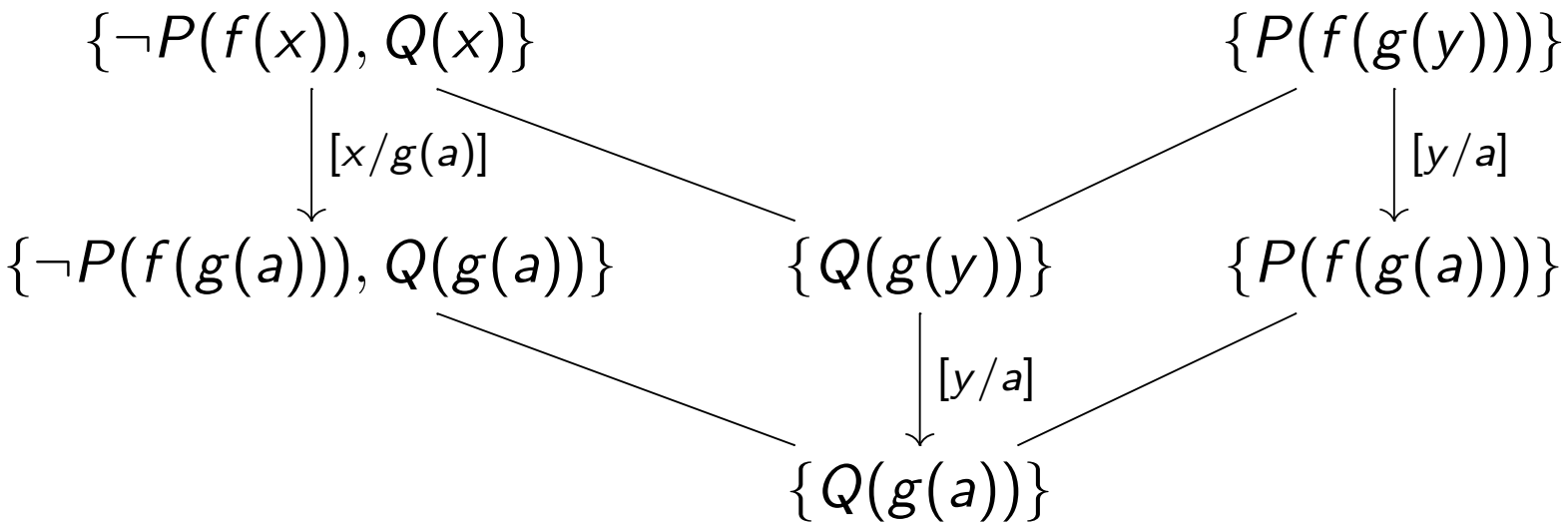
Veranschaulichung des Liftig-Lemma:



—: Resolution

$\rightarrow$: Substitution

Beispiel:



Beweis des Lifting-Lemmas

Beweis:

Seien s_1 und s_2 Variablenumbennungen, so dass $K_1 s_1$ und $K_2 s_2$ keine gemeinsamen Variablen haben.

K_i' Grundinstanz von K_i. $\Longrightarrow$ K_i' Grundinstanz von $K_i s_i$.

Sei sub_i eine Substitution mit $K_i' = K_i s_i \, \mathrm{sub}_i$.

Dabei gilt o.B.d.A. für $i \in \{1, 2\}$:

1. $\mathrm{Def}(\mathrm{sub}_i)$ ist die Menge der in $K_i s_i$ vorkommenden Variablen.
2. Für alle $x \in \mathrm{Def}(\mathrm{sub}_i)$ enthält der Term $\mathrm{sub}_i(x)$ keine Variablen (sub_i ist eine Grundsubstitution).

Aus (1) folgt insbesondere $\mathrm{Def}(\mathrm{sub}_1) \cap \mathrm{Def}(\mathrm{sub}_2) = \emptyset$.

Sei $\mathrm{sub} = \mathrm{sub}_1 \mathrm{sub}_2 = \mathrm{sub}_2 \mathrm{sub}_1$.

Es gilt $K_i' = K_i s_i \, \mathrm{sub}_i = K_i s_i \, \mathrm{sub}$.

Nach Voraussetzung ist R' aussagenlogischer Resolvent von K_1' und K_2'.

Beweis des Lifting-Lemmas

Also gibt es $L \in K_1' = K_1 s_1 \,\mathrm{sub}$ und $\overline{L} \in K_2' = K_2 s_2 \,\mathrm{sub}$ mit

$$R' = (K_1' \setminus \{L\}) \cup (K_2' \setminus \{\overline{L}\}).$$

Seien $L_1, \ldots, L_m \in K_1 s_1$ $(m \geq 1)$ alle Literale aus $K_1 s_1$ mit

$$L = L_1 \mathrm{sub} = \cdots = L_m \mathrm{sub}.$$

Seien $L_1', \ldots, L_n' \in K_2 s_2$ $(n \geq 1)$ alle Literale aus $K_2 s_2$ mit

$$\overline{L} = L_1' \mathrm{sub} = \cdots = L_n' \mathrm{sub}.$$

Somit ist sub ein Unifikator der Literalmenge

$$\mathbf{L} = \{\overline{L_1}, \ldots, \overline{L_m}, L_1', \ldots, L_n'\}$$

und die Klauseln K_1 und K_2 sind prädikatenlogisch resolvierbar.

Sei sub_0 ein allgemeinster Unifikator von $\mathbf{L}$.

Beweis des Lifting-Lemmas

Dann ist

$$R = ((K_1 s_1 - \{L_1, \ldots, L_m\}) \cup (K_2 s_2 - \{L'_1, \ldots, L'_n\}))\mathrm{sub}_0.$$

ein prädikatenlogischer Resolvent von K_1 und K_2.

Da sub_0 allgemeinster Unifikator von $\mathbf{L}$ ist und sub ein Unifikator von $\mathbf{L}$ ist, existiert eine Substitution s mit $\mathrm{sub}_0\, s = \mathrm{sub}$. Es folgt

$$\begin{aligned}
R' &= (K'_1 - \{L\}) \cup (K'_2 - \{\overline{L}\}) \\
&= (K_1 s_1\, \mathrm{sub} - \{L\}) \cup (K_2 s_2\, \mathrm{sub} - \{\overline{L}\}) \\
&= (K_1 s_1\, \mathrm{sub} - \{L_1 \mathrm{sub}, \ldots, L_m \mathrm{sub}\}) \cup (K_2 s_2\, \mathrm{sub} - \{L'_1 \mathrm{sub}, \ldots, L'_n \mathrm{sub}\}) \\
&= \Big((K_1 s_1 - \{L_1, \ldots, L_m\}) \cup (K_2 s_2 - \{L'_1, \ldots, L'_n\})\Big)\, \mathrm{sub} \\
&= \Big((K_1 s_1 - \{L_1, \ldots, L_m\}) \cup (K_2 s_2 - \{L'_1, \ldots, L'_n\})\Big)\, \mathrm{sub}_0 s \\
&= R\, s
\end{aligned}$$

Damit ist gezeigt, dass R' eine Grundinstanz von R ist. □

Resolutionssatz

Resolutionssatz der Prädikatenlogik

Sei F eine Aussage in Skolemform mit einer Matrix F^* in **KNF**. Dann gilt: F ist unerfüllbar genau dann, wenn $\square \in Res^*(F^*)$.

Für den Beweis des Resolutionssatzes benötigen wir folgenden Begriff:

Für eine Formel H mit freien Variablen $x_1, \ldots, x_n$ bezeichnen wir mit

$$\forall H = \forall x_1 \forall x_2 \cdots \forall x_n H$$

ihren Allabschluss.

Lemmata zum Allabschluss

Lemma

Sei F eine Aussage in Skolemform, deren Matrix F^* in **KNF** ist. Dann gilt:

$$F \equiv \forall F^* \equiv \bigwedge_{K \in F^*} \forall K$$

Beweis: Da F eine Aussage ist (also keine freien Variablen hat), gilt $F = \forall F^*$.

Da F^* in **KNF** ist, gilt

$$F^* \equiv \bigwedge_{K \in F^*} K.$$

Das Lemma folgt somit aus der Äquivalenz $\forall y(G \wedge H) \equiv \forall y G \wedge \forall y H$. □

Beispiel:

$$\begin{aligned} F^* &= P(x,y) \wedge \neg Q(y,x) \\ F &\equiv \forall x \forall y (P(x,y) \wedge \neg Q(y,x)) \equiv \forall x \forall y P(x,y) \wedge \forall x \forall y (\neg Q(y,x)) \end{aligned}$$

Lemmata zum Allabschluss

Lemma

Sei R Resolvent zweier Klauseln K_1 und K_2. Dann ist $\forall R$ eine Folgerung von $\forall K_1 \ \wedge \ \forall K_2$.

Beweis:

Sei $\mathcal{A}$ ein Modell von $\forall K_1$ und von $\forall K_2$: $\mathcal{A}(\forall K_1) = \mathcal{A}(\forall K_2) = 1$

Sei

$$R = ((K_1 s_1 - \{L_1, \ldots, L_m\}) \cup (K_2 s_2 - \{L'_1, \ldots, L'_n\}))\mathrm{sub}$$

wobei $L_1, \ldots, L_m \in K_1 s_1$, $L'_1, \ldots, L'_n \in K_2 s_2$ und sub allgemeinster Unifikator von

$$\mathbf{L} = \{\overline{L_1}, \ldots, \overline{L_m}, L'_1, \ldots, L'_n\}$$

ist.

Sei $L = \overline{L_1}\,\mathrm{sub} = \cdots = \overline{L_m}\,\mathrm{sub} = L'_1\,\mathrm{sub} = \cdots = L'_n\,\mathrm{sub}$. Dann gilt

$$(K_1 s_1\,\mathrm{sub} - \{\overline{L}\}) \cup (K_2 s_2\,\mathrm{sub} - \{L\}) \subseteq R.$$

Lemmata zum Allabschluss

Angenommen, es gilt $\mathcal{A}(\forall R) = 0$.

Dann gibt es eine Struktur $\mathcal{A}'$ mit:

- $\mathcal{A}'$ ist identisch zu $\mathcal{A}$ bis auf die Werte $I_{\mathcal{A}'}(x)$ für die in R vorkommenden Variablen x.
- $\mathcal{A}'(R) = 0$.

Also gilt

$$\mathcal{A}'(K_1 s_1 \,\mathrm{sub} - \{\overline{L}\}) = \mathcal{A}'(K_2 s_2 \,\mathrm{sub} - \{L\}) = 0. \tag{11}$$

Aus $\mathcal{A}(\forall K_1) = \mathcal{A}(\forall K_2) = 1$ folgt

$$\mathcal{A}'(K_1 s_1 \,\mathrm{sub}) = \mathcal{A}'(K_2 s_2 \,\mathrm{sub}) = 1. \tag{12}$$

(11) und (12) ergibt zusammen: $\mathcal{A}'(L) = \mathcal{A}'(\overline{L}) = 1$. Widerspruch! □

Beweis des Resolutionssatzes

Beweis des Resolutionssatzes:

(A) Korrektheit: Wenn $\square \in Res^*(F^*)$, dann ist F unerfüllbar.

Gelte $\square \in Res^*(F^*)$.

Aus den soeben bewiesenen Lemmata folgt: $\square = \forall\square$ ist eine Folgerung von $\bigwedge_{K \in F^*} \forall K \equiv F$.

Da $\square$ kein Modell hat, kann auch F kein Modell haben.

(B) Vollständigkeit: Wenn F unerfüllbar ist, dann gilt $\square \in Res^*(F^*)$.

Sei F unerfüllbar.

Beweis des Resolutionssatzes

Aus dem Grundresolutionssatz folgt, dass es eine Folge von Klauseln $K'_1, \ldots, K'_n$ mit folgender Eigenschaft gibt:

- K'_n ist die leere Klausel
- Für $i = 1, \ldots, n$ gilt:
  - K'_i ist eine Grundinstanz einer Klausel $K \in F^*$,d.h. $K'_i = K[y_1/t_1] \ldots [y_k/t_k]$ mit $t_i \in D(F)$
  - **oder** K'_i ist (aussagenlogischer) Resolvent zweier Klauseln K'_a, K'_b mit $a < i$ und $b < i$

Für alle $i \in \{1, \ldots, n\}$ geben wir eine Klausel K_i an, so dass K'_i eine Grundinstanz von K_i ist und $(K_1, \ldots, K_n)$ eine prädikatenlogische Resolutionsherleitung der leeren Klausel $K_n = \square$ aus den Klauseln in F^* ist.

Betrachte ein $i \in \{1, \ldots, n\}$ und seien $K_1, \ldots, K_{i-1}$ bereits konstruiert.

Beweis des Resolutionssatzes

1.Fall: K'_i eine Grundinstanz einer Klausel $K \in F^*$.

Definiere $K_i = K$.

2.Fall: K'_i ist aussagenlogischer Resolvent zweier Klauseln K'_a, K'_b mit $a < i$ und $b < i$.

Aus dem Lifting-Lemma ergibt sich ein prädikatenlogischer Resolvent R von K_a und K_b, so dass K'_i eine Grundinstanz von R ist.

Definiere $K_i = R$. □

Beispiel I

Ist die Klauselmenge

$$\{\{P(f(x))\},\ \{\neg P(x), Q(x, f(x))\},\ \{\neg Q(f(a), f(f(a)))\}\}$$

unerfüllbar?

Beispiel I

Ist die Klauselmenge

$$\{\{P(f(x))\},\ \{\neg P(x), Q(x, f(x))\},\ \{\neg Q(f(a), f(f(a)))\}\}$$

unerfüllbar?

Ja, hier ist eine Resolutionsableitung der leeren Klausel:

$\{P(f(x))\}$ und $\{\neg P(x), Q(x, f(x))\}$ ergeben den Resolventen $\{Q(f(x), f(f(x)))\}$

$\{Q(f(x), f(f(x)))\}$ und $\{\neg Q(f(a), f(f(a)))\}$ ergeben den Resolventen $\{\} = \square$.

Beispiel II

Wir betrachten folgende Klauselmenge
(Beispiel aus dem Buch von Schöning):

$$\begin{aligned} F \quad = \quad & \{\{\neg P(x), Q(x), R(x, f(x))\}, \{\neg P(x), Q(x), S(f(x))\}, \{T(a)\}, \\ & \{P(a)\}, \{\neg R(a, x), T(x)\}, \{\neg T(x), \neg Q(x)\}, \{\neg T(x), \neg S(x)\}\} \end{aligned}$$

Verfeinerung der Resolution (Ausblick)

Probleme bei der prädikatenlogischen Resolution:

- Zu viele Wahlmöglichkeiten
- Immer noch zu viele Sackgassen
- Kombinatorische Explosion des Suchraums

Lösungsansätze:

Strategien und Heuristiken: Verbieten bestimmter Resolutionsschritte, Suchraum wird dadurch eingeschränkt

Vorsicht: Die Vollständigkeit darf dadurch nicht verloren gehen!

Unentscheidbarkeit der Prädikatenlogik

Wir wollen nun zeigen, dass es keinen Algorithmus, der als Eingabe ein prädikatenlogische Formel F bekommt, und folgende Eigenschaften hat:

- Wenn F gültig ist, dann terminiert der Algorithmus mit der Ausage "Ja".
- Wenn F nicht gültig ist, dann terminiert der Algorithmus mit der Ausage "Nein".

Mit anderen Worten, wir wollen den folgenden Satz beweisen:

Satz von Church

Die Menge der gültigen prädikatenlogischen Formeln ist unentscheidbar.

Korollar

Die Menge der erfüllbaren prädikatenlogischen Formeln ist nicht semi-entscheidbar.

Beweis: Die Menge der unerfüllbaren Formeln ist semi-entscheidbar.

Registermaschinen

Wir beweisen den Satz von Church durch eine Reduktion vom Halteproblem für Registermachinenprogramme.

Seien $R_1, R_2, \ldots$ Bezeichner für Register.

Intuition: Jedes Register speichert eine natürliche Zahl ab.

Eine Registermachinenprogramm (kurz RMP) P besteht aus einer Folge $A_1; A_2; \ldots; A_l$ von Anweisungen, wobei A_l die Anweisung STOP ist, und für alle $1 \leq i \leq l-1$ die Anweisung A_i von einem der folgenden Typen ist:

- $R_j := R_j + 1$ für ein $1 \leq j \leq l$
- $R_j := R_j - 1$ für ein $1 \leq j \leq l$
- IF $R_j = 0$ THEN k_1 ELSE k_2 für $1 \leq j, k_1, k_2 \leq l$,

Eine Konfiguration von P ist ein Tupel $(i, n_1, \ldots n_l) \in \mathbb{N}^{l+1}$ mit $1 \leq i \leq l$.

Intuition: i ist die Nummer der Anweisung, die als nächste ausgeführt wird, und n_j ist der aktuelle Inhalt von Register R_j.

Registermaschinen

Für Konfigurationen $(i, n_1, \ldots, n_l)$ und $(i', n'_1, \ldots, n'_l)$ schreiben wir

$$(i, n_1, \ldots, n_l) \to_P (i', n'_1, \ldots, n'_l)$$

genau dann, wenn $1 \leq i \leq l-1$ und einer der folgenden Fälle gilt:

- $A_i = (R_j := R_j + 1)$ für ein $1 \leq j \leq l$, $i' = i + 1$, $n'_j = n_j + 1$, $n'_k = n_k$ für $k \neq j$.
- $A_i = (R_j := R_j - 1)$ für ein $1 \leq j \leq l$, $i' = i + 1$, $n_j = n'_j = 0$ oder $(n_j > 0,\ n'_j = n_j - 1)$, und $n'_k = n_k$ für $k \neq j$.
- $A_i = (\text{IF } R_j = 0 \text{ THEN } k_1 \text{ ELSE } k_2)$ für ein $1 \leq j, k_1, k_2 \leq l$, $n'_k = n_k$ für alle $1 \leq k \leq l$, $i' = k_1$ falls $n_j = 0$, $i' = k_2$ falls $n_j > 0$.

Wir definieren

$$\begin{aligned}\text{HALT} = \{P \mid\ & P = A_1; A_2; \ldots; A_l \text{ ist ein RMP mit } l \text{ Anweisungen},\\ & (1, 0, \ldots, 0) \to_P^* (l, n_1, \ldots, n_l) \text{ für } n_1, \ldots, n_l \geq 0\}\end{aligned}$$

Beweis des Satzes von Church

Unentscheidbarkeit des Halteproblems

Die Menge HALT ist unentscheidbar.

Wir beweisen den Satz von Church, indem wir jedem RMP P effektiv eine prädikatenlogischen Aussage F_P zuordnen, so dass gilt:

$$F_P \text{ ist gültig} \iff P \in \text{HALT}$$

Sei $P = A_1; A_2; \ldots; A_l$ ein RMP.

Wir fixieren folgende Symbole:

- $<$: 2-stelliges Prädikatensymbol
- c: Konstante
- f, g: 1-stellige Funktionssymbole
- R: $(l+2)$-stelliges Prädikatensymbol

Beweis des Satzes von Church

Wir definieren eine Struktur $\mathcal{A}_P$ durch Fallunterscheidung:

1. Fall: $P \notin$ HALT:

- Universum $U_{\mathcal{A}_P} = \mathbb{N}$
- $<^{\mathcal{A}_P} = \{(n, m) \mid n < m\}$ (gewöhnliche Ordnung auf $\mathbb{N}$)
- $c^{\mathcal{A}_P} = 0$
- $f^{\mathcal{A}_P}(n) = n + 1$, $g^{\mathcal{A}_P}(n + 1) = n$, $g^{\mathcal{A}_P}(0) = 0$
- $R^{\mathcal{A}_P} = \{(s, i, n_1, \ldots, n_l) \mid (1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)\}$

2. Fall: $P \in$ HALT:

Sei t so, dass $(1, 0, \ldots, 0) \to_P^t (l, n_1, \ldots, n_l)$ und $e = \max\{t, l\}$.

- Universum $U_{\mathcal{A}_P} = \{0, 1, \ldots, e\}$
- $<^{\mathcal{A}_P} = \{(n, m) \mid n < m\}$ (gewöhnliche Ordnung auf $\{0, 1, \ldots, e\}$)
- $c^{\mathcal{A}_P} = 0$
- $f^{\mathcal{A}_P}(n) = n + 1$ für $0 \leq n \leq e - 1$ und $f^{\mathcal{A}_P}(e) = e$.
- $g^{\mathcal{A}_P}(n + 1) = n$ für $0 \leq n \leq e - 1$ und $g^{\mathcal{A}_P}(0) = 0$.
- $R^{\mathcal{A}_P} = \{(s, i, n_1, \ldots, n_l) \mid 0 \leq s \leq t, (1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)\}$

Beweis des Satzes von Church

Im folgenden verwenden wir die Abkürzung $\overline{m}$ für den Term $f^m(c)$.

Wir definieren nun eine Aussage G_P (in der $<, c, f, g$ und R vorkommen) mit folgenden Eigenschaften:

(A) $\mathcal{A}_P \models G_P$

(B) Für jedes Modell $\mathcal{A}$ von G_P gilt Folgendes:

Wenn $(1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)$, dann:

$$\mathcal{A} \models R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_l}) \wedge \bigwedge_{i=0}^{s-1} \overline{i} < \overline{i+1}.$$

Wir definieren

$$G_P = G_0 \wedge R(\overline{0}, \overline{1}, \overline{0}, \ldots, \overline{0}) \wedge G_1 \wedge \cdots \wedge G_{l-1}$$

wobei die Aussagen $G_0, G_1, \ldots, G_{l-1}$ wie folgt definiert sind.

Beweis des Satzes von Church

G_0 sagt aus:

- $<$ ist eine lineare Ordnung mit kleinstem Element c,
- $x \leq f(x)$ und $g(x) \leq x$ für alle x,
- für jedes x, das nicht das größte Element bzgl. $<$ ist, ist $f(x)$ der unmittelbare Nachfolger von x, und
- für jedes x, das nicht das kleinste Element c ist, ist $g(x)$ der unmittelbare Vorgänger von x.

$$\begin{aligned}\forall x, y, z\ & (\neg x < x) \wedge (x = y \vee x < y \vee y < x) \wedge ((x < y \wedge y < z) \rightarrow x < z) \\ & \wedge (x = c \vee c < x) \\ & \wedge (x = f(x) \vee x < f(x)) \\ & \wedge (x = g(x) \vee g(x) < x) \\ & \wedge \big(\exists u(x < u) \rightarrow (x < f(x) \wedge \forall u(x < u \rightarrow (u = f(x) \vee f(x) < u)))\big) \\ & \wedge \big(\exists u(u < x) \rightarrow (g(x) < x \wedge \forall u(u < x \rightarrow (u = g(x) \vee u < g(x))))\big)\end{aligned}$$

Beweis des Satzes von Church

Bemerkung: Für jedes Modell $\mathcal{A}$ von G_0 gilt:

- $\mathcal{A} \models g(c) = c$
- $\mathcal{A} \models \forall x \, (\exists u(x < u) \rightarrow g(f(x)) = x)$

Beweis des Satzes von Church

G_i für $1 \leq i \leq l-1$ beschreibt die Wirkung der Anweisung A_i.

1. Fall: $A_i = (R_j := R_j + 1)$. Dann sei

$$G_i = \forall x \forall x_1 \cdots \forall x_l \Big(R(x, \overline{i}, x_1, \ldots, x_l) \rightarrow (x < f(x) \wedge R(f(x), \overline{i+1}, x_1, \ldots, x_{j-1}, f(x_j), x_{j+1}, \ldots, x_l)) \Big)$$

2. Fall: $A_i = (R_j := R_j - 1)$. Dann sei

$$G_i = \forall x \forall x_1 \cdots \forall x_l \Big(R(x, \overline{i}, x_1, \ldots, x_l) \rightarrow (x < f(x) \wedge R(f(x), \overline{i+1}, x_1, \ldots, x_{j-1}, g(x_j), x_{j+1}, \ldots, x_l)) \Big)$$

Beweis des Satzes von Church

3. Fall: $A_i = (\text{IF } R_j = 0 \text{ THEN } k_1 \text{ ELSE } k_2)$ für ein $1 \leq j, k_1, k_2 \leq l$.
Dann sei

$$G_i = \forall x \forall x_1 \cdots \forall x_l \Big(R(x, \overline{i}, x_1, \ldots, x_l) \rightarrow (x < f(x) \wedge (x_j = c \wedge R(f(x), \overline{k_1}, x_1, \ldots, x_l)) \vee (x_j > c \wedge R(f(x), \overline{k_2}, x_1, \ldots, x_l))) \Big)$$

Aussage (A) folgt sofort aus der Definition von $\mathcal{A}_P$ und G_P.

Aussage (B) beweisen wir durch eine Induktion über s.

IA: $s = 0$. Gelte $(1, 0, \ldots, 0) \rightarrow_P^0 (i, n_1, \ldots, n_l)$, d.h. $i = 1$ und $n_1 = n_2 = \cdots = n_l = 0$.

Aus $\mathcal{A} \models G_P$ folgt $\mathcal{A} \models R(\overline{0}, \overline{1}, \overline{0}, \ldots, \overline{0})$, d. h. $\mathcal{A} \models R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_l})$.

Beweis des Satzes von Church

IS: Sei nun $s > 0$ und gelte Aussage (B) für $s - 1$.

Sei $(1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)$.

Dann gibt es $j, m_1, \ldots, m_l$ mit

$$(1, 0, \ldots, 0) \to_P^{s-1} (j, m_1, \ldots, m_l) \to_P (i, n_1, \ldots, n_l)$$

Aus der IH folgt

$$\mathcal{A} \models R(\overline{s-1}, \bar{j}, \overline{m_1}, \ldots, \overline{m_l}) \wedge \bigwedge_{i=0}^{s-2} \bar{i} < \overline{i+1}.$$

Wir machen nun eine Fallunterscheidung bezüglich der Anweisung A_j, wobei wir nur den Fall betrachten, dass A_j von der Form $R_k := R_k - 1$ ist.

Es gilt dann $i = j + 1$, $n_1 = m_1, \ldots, n_{k-1} = m_{k-1}$, $n_{k+1} = m_{k+1}, \ldots, n_l = m_l$, ($n_k = m_k = 0$ oder $m_k > 0$ und $n_k = m_k - 1$).

Beweis des Satzes von Church

Wegen $\mathcal{A} \models G_j$ gilt:

$$\mathcal{A} \models \forall y, y_1, \ldots, y_l \Big(R(y, \bar{j}, y_1, \ldots, y_l) \rightarrow (y < f(y) \wedge R(f(y), \overline{j+1}, y_1, \ldots, y_{k-1}, g(y_k), y_{k+1}, \ldots, y_l)) \Big)$$

Wegen $\mathcal{A} \models R(\overline{s-1}, \bar{j}, \overline{m_1}, \ldots, \overline{m_l})$ folgt

$$\mathcal{A} \models \overline{s-1} < f(\overline{s-1}) \wedge R(f(\overline{s-1}), \overline{j+1}, \overline{m_1}, \ldots, \overline{m_{k-1}}, g(\overline{m_k}), \overline{m_{k+1}}, \ldots, \overline{m_l})$$

d.h.

$$\mathcal{A} \models \overline{s-1} < \bar{s} \wedge R(\bar{s}, \bar{i}, \overline{n_1}, \ldots, \overline{n_{k-1}}, g(\overline{m_k}), \overline{n_{k+1}}, \ldots, \overline{n_l})$$

Beweis des Satzes von Church

Wegen $\mathcal{A} \models \overline{s-1} < \overline{s}$ gilt

$$\mathcal{A} \models \bigwedge_{i=0}^{s-1} \overline{i} < \overline{i+1}.$$

Ausserdem folgt aus $\mathcal{A} \models G_0$, dass $\mathcal{A} \models g(\overline{m_k}) = \overline{n_k}$.

Also gilt auch $\mathcal{A} \models R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_l})$.

Damit sind (A) und (B) gezeigt.

Beweis des Satzes von Church:

Setze $F_P = (G_P \to \exists x \exists x_1 \cdots \exists x_l R(x, \overline{l}, x_1, \ldots, x_l))$

Behauptung: F_P ist gültig $\iff P \in \text{HALT}$.

Beweis des Satzes von Church

Ist F_P gültig, so gilt insbesondere $\mathcal{A}_P \models F_P$.

Wegen (A) gilt $\mathcal{A}_P \models \exists x \exists x_1 \cdots \exists x_l R(x, \overline{l}, x_1, \ldots, x_l)$.

Also gibt es $s, n_1, \ldots, n_l \geq 0$ mit $(s, l, n_1, \ldots, n_l) \in R^{\mathcal{A}_P}$.

Es folgt $P \in \text{HALT}$.

Sein nun $P \in \text{HALT}$ und gelte $(1, 0, \ldots, 0) \to_P^s (l, n_1, \ldots, n_l)$

Sei $\mathcal{A}$ eine Struktur mit $\mathcal{A} \models G_P$.

Aus (B) folgt $\mathcal{A} \models R(\overline{s}, \overline{l}, \overline{n_1}, \ldots, \overline{n_l})$.

Also ist F_P gültig. □

Der Satz von Trachtenbrot

Eine Formel F ist im Endlichen erfüllbar genau dann, wenn F ein Modell mit einem endlichen Universum hat, sonst ist F im Endlichen unerfüllbar.

Lemma

Die Menge der im Endlichen erfüllbaren Formeln ist semi-entscheidbar.

Beweis:

Sei $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \ldots$ eine systematische Auflistung aller endlichen zu F passenden Strukturen (o.B.d.A. ist $I_{\mathcal{A}_i}$ nur auf den in F vorkommenden Prädikaten- und Funktionssymbolen definiert).

Folgender Algorithmus terminiert genau dann, wenn F im endlichen erfüllbar ist:

$i := 1$;
while true do
    if $\mathcal{A}_i \models F$ **then STOP else** $i := i + 1$
end

□

Der Satz von Trachtenbrot

Eine Formel F ist im Endlichen gültig genau dann, wenn jede endliche zu F passende Struktur ein Modell von F ist.

Beispiel: Die Formel

$$\forall x \forall y (f(x) = f(y) \rightarrow x = y) \;\leftrightarrow\; \forall y \exists x (f(x) = y)$$

ist im Endlichen gültig, aber nicht (allgemein) gültig.

Satz von Trachtenbrot

Die Menge der im Endlichen erfüllbaren Formeln ist unentscheidbar.

Korollar

Die Menge der im Endlichen unerfüllbaren Formeln sowie die Menge der im Endlichen gültigen Formeln ist nicht semi-entscheidbar.

Der Satz von Trachtenbrot

Beweis des Satzes von Trachtenbrot:

Wir verwenden die Konstruktion aus dem Beweis des Satzes von Church.

Behauptung: G_P ist im Endlichen erfüllbar $\iff P \in \text{HALT}$.

(1) Gelte $P \in \text{HALT}$.

Dann ist $\mathcal{A}_P$ endlich und es gilt $\mathcal{A}_P \models G_P$ nach Aussage (A).

Also ist G_P im Endlichen erfüllbar.

Der Satz von Trachtenbrot

(2) Sei G_P im Endlichen erfüllbar.

Sei $\mathcal{A}$ eine endliche Struktur mit $\mathcal{A} \models G_P$.

Angenommen $P \notin \text{HALT}$ gilt.

Also gibt es für jede Zahl $s \geq 0$ Zahlen $i, n_1, \ldots, n_l$ mit $(1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)$.

Aussage (B) impliziert, dass $\mathcal{A} \models \bar{i} < \overline{i+1}$ für alle $i \geq 0$.

Da $<^{\mathcal{A}}$ eine lineare Ordnung ist (wegen $\mathcal{A} \models G_0$) ist die Menge $\{\mathcal{A}(\bar{i}) \mid i \geq 0\}$ unendlich, was ein Widerspruch ist. □

(Un)entscheidbare Theorien

Sei $\mathcal{A}$ eine Struktur, wobei der Definitionsbereich von $I_{\mathcal{A}}$ endlich sei und keine Variablen enthält.

Sei $f_1, \ldots, f_n, R_1, \ldots, R_m$ der Definitionsbereich von $I_{\mathcal{A}}$.

Wir identifizieren dann $\mathcal{A}$ mit dem Tupel $(U^{\mathcal{A}}, f_1^{\mathcal{A}}, \ldots, f_n^{\mathcal{A}}, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$, wofür wir auch $(U^{\mathcal{A}}, f_1, \ldots, f_n, R_1, \ldots, R_m)$ schreiben.

Definition

Die Theorie von $\mathcal{A}$ ist die Menge von Formeln

$$\mathrm{Th}(\mathcal{A}) = \{F \mid F \text{ ist eine Aussage, } \mathcal{A} \text{ passt zu } F, \mathcal{A} \models F\}.$$

Wir interessieren uns für die Frage, ob eine Struktur eine entscheidbare Theorie hat.

(Un)entscheidbare Theorien

Satz

Sei $\mathcal{A}$ eine beliebige Struktur. Dann ist $\mathrm{Th}(\mathcal{A})$ *entscheidbar genau dann, wenn* $\mathrm{Th}(\mathcal{A})$ *semi-entscheidbar ist.*

Beweis: Sei $\mathrm{Th}(\mathcal{A})$ semi-entscheidbar und sei F eine beliebige Aussage.

Dann gilt entweder $F \in \mathrm{Th}(\mathcal{A})$ oder $\neg F \in \mathrm{Th}(\mathcal{A})$.

Wir können daher einen Semi-Entscheidungsalgorithmus für $\mathrm{Th}(\mathcal{A})$ mit Eingabe F und $\neg F$ parallel laufen lassen.

Einer der beiden Läufe wird irgendwann mit der Antwort terminieren. □

(Un)entscheidbare Theorien

Für die Frage nach der Entscheidbarkeit einer Struktur können wir uns auf sogenannte relationale Strukturen beschränken.

Eine Struktur $\mathcal{A} = (A, f_1, \ldots, f_n, R_1, \ldots, R_m)$ ist relational, falls $n = 0$ gilt.

Für eine beliebige Struktur $\mathcal{A} = (A, f_1, \ldots, f_n, R_1, \ldots, R_m)$ definieren wir

$$\mathcal{A}_{\text{rel}} = (A, P_1, \ldots, P_n, R_1, \ldots, R_m)$$

wobei

$$P_i = \{(a_1, \ldots, a_n, a) \mid f_i(a_1, \ldots, a_n) = a\}.$$

Lemma

$\text{Th}(\mathcal{A})$ *ist entscheidbar genau dann, wenn* $\text{Th}(\mathcal{A}_{rel})$ *entscheidbar ist.*

Beweis: Übung.

Unentscheidbarkeit der Arithmetik (nach Ebbinhaus,Flum,Thomas)

Satz (Gödel 1931)

$\mathrm{Th}(\mathbb{N}, +, \cdot)$ ist unentscheidbar.

Korollar

$\mathrm{Th}(\mathbb{N}, +, \cdot)$ ist nicht semi-entscheidbar, also nicht rekursiv aufzählbar.

Für den Beweis reduzieren wir die Menge HALT von terminierenden RMPs auf $\mathrm{Th}(\mathbb{N}, +, \cdot)$.

Um den Beweis etwas komfortabler zu machen, betrachten wir $\mathrm{Th}(\mathbb{N}, +, \cdot, s, 0)$ mit $s(n) = n + 1$.

Übung: $\mathrm{Th}(\mathbb{N}, +, \cdot, s, 0)$ ist unentscheidbar genau dann, wenn $\mathrm{Th}(\mathbb{N}, +, \cdot)$ unentscheidbar ist.

Unentscheidbarkeit der Arithmetik

Sei nun $P = A_1; A_2; \cdots ; A_l$ ein RMP, in dem die Register $R_1, \ldots, R_l$ verwendet werden.

Wir konstruieren eine arithmetische Formel F_P mit den freien Variablen $x, x_1, \ldots, x_l$, so dass für alle $1 \leq i \leq l$ und $n_1, \ldots, n_l \in \mathbb{N}$ folgende beiden Aussagen äquivalent sind:

- $(\mathbb{N}, +, \cdot, s, 0)_{[x/i,\ x_1/n_1, \ldots, x_l/n_l]} \models F_P$
- $(1, 0, \ldots, 0) \to_P^* (i, n_1, \ldots, n_l)$

Dann gilt $P \in \text{HALT} \iff (\mathbb{N}, +, \cdot, s, 0) \models \exists x_1 \cdots \exists x_l \, F_P[x/s^l(0)]$.

Unentscheidbarkeit der Arithmetik

Intuitiv sagt die Formel F_P Folgendes aus:

Es gibt ein $s \geq 0$ und Konfigurationen $C_0, C_1, \ldots, C_s$ mit:

- $C_0 = (1, 0, \ldots, 0)$
- $C_s = (x, x_1, \ldots, x_l)$
- $C_i \to_P C_{i+1}$ für alle $0 \leq i \leq s - 1$

Wir können die $(l+1)$-Tupel $C_0, C_1, \ldots, C_s$ durch ein $(s+1)(l+1)$-Tupel kodieren, und müssen dann Folgendes ausdrücken, wobei $k = l + 1$ sei.

Es gibt ein $s \geq 0$ und ein Tupel
$(y_0, y_1, \ldots, y_{k-1},\ y_k, y_{k+1}, \ldots, y_{2k-1}, \ldots, y_{sk}, y_{sk+1}, \ldots, y_{(s+1)k-1})$ mit:

- $y_0 = 1,\ y_1 = 0, \ldots, y_{k-1} = 0$
- $y_{sk} = x,\ y_{sk+1} = x_1, \ldots, y_{(s+1)k-1} = x_l$
- $(y_{ik}, \ldots, y_{(i+1)k-1}) \to_P (y_{(i+1)k}, \ldots, y_{(i+2)k-1})$ für alle $0 \leq i \leq s - 1$

Unentscheidbarkeit der Arithmetik

Will man dies durch eine arithmetische Formel ausdrücken, hat man das Problem, dass man nicht über Folgen von Zahlen quantifizieren kann ($\exists y \exists x_1 \cdots \exists x_y$ ist nicht zulässig).

Um trotzdem eine Quantifizierung über beliebig lange Folgen zu simulieren, benötigen wir Gödels β-Funktion.

Lemma

Es gibt eine Funktion $\beta : \mathbb{N}^3 \to \mathbb{N}$ mit:

- Für jede Folge $(a_0, \ldots, a_r)$ über $\mathbb{N}$ gibt es $t, p \in \mathbb{N}$, so dass $\beta(t, p, i) = a_i$ für alle $0 \leq i \leq r$
- Es gibt eine arithmetische Formel B mit freien Variablen v, x, y, z, so dass für alle $t, p, i, a \in \mathbb{N}$ gilt:

$$(\mathbb{N}, +, \cdot, s, 0)_{[v/t,\, x/p,\, y/i,\, z/a]} \models B \iff \beta(t, p, i) = a$$

Man sagt auch: β ist arithmetisch definierbar.

Beweis des Lemmas:

Sei $(a_0, \ldots, a_r)$ eine beliebige Folge über $\mathbb{N}$.

Sei p eine Primzahl mit $p > r + 1$ und $p > a_i$ für alle i.

Sei weiter

$$t = 1p^0 + a_0p^1 + 2p^2 + a_1p^3 + \cdots + (i+1)p^{2i} + a_ip^{2i+1} + \cdots + (r+1)p^{2r} + a_rp^{2r+1}.$$

D.h. $(1, a_0, 2, a_1, \ldots, (i+1), a_i, \ldots, (r+1), a_r)$ ist die Darstellung von t zur Basis p.

Behauptung: Für alle $a \in \mathbb{N}$ und alle $0 \leq i \leq r$ gilt $a = a_i$ genau dann, wenn es $b_0, b_1, b_2 \in \mathbb{N}$ gibt mit:

(a) $t = b_0 + b_1((i+1) + ap + b_2p^2)$

(b) $a < p$

(c) $b_0 < b_1$

(d) Es gibt ein m mit $b_1 = p^{2m}$.

Unentscheidbarkeit der Arithmetik

$\Rightarrow$: Wenn $a = a_i$ dann können wir b_0, b_1, b_2 wie folgt wählen:

$$\begin{aligned} b_0 &= 1p^0 + a_0p^1 + 2p^2 + a_1p^3 + \cdots + ip^{2i-2} + a_{i-1}p^{2i-1} \\ b_1 &= p^{2i} \\ b_2 &= (i+2) + a_{i+1}p + \cdots + a_rp^{2(r-i)-1} \end{aligned}$$

$\Leftarrow$: Gelte (a)-(d), d.h.

$$\begin{aligned} t &= b_0 + b_1((i+1) + ap + b_2p^2) \\ &= b_0 + (i+1)p^{2m} + ap^{2m+1} + p^{2m+2}b_2. \end{aligned}$$

wobei $b_0 < b_1 = p^{2m}$, $a < p$ und $(i+1) < p$.

Ein Vergleich mit

$$t = 1p^0+a_0p^1+2p^2+a_1p^3+\cdots+(i+1)p^{2i}+a_ip^{2i+1}+\cdots+(r+1)p^{2r}+a_rp^{2r+1}$$

liefert $m = i$ und $a = a_i$.

Unentscheidbarkeit der Arithmetik

Da p eine Primzahl ist, ist (d) äquivalent zu: b_1 ist ein Quadrat, und für alle $d \geq 2$ mit $d|b_1$ gilt $p|d$.

Wir definieren nun für alle Zahlen $t, p, i \in \mathbb{N}$ die Zahl $\beta(t, p, i)$ als die kleinste Zahl a, so dass $b_0, b_1, b_2 \in \mathbb{N}$ existieren mit:

(a) $t = b_0 + b_1((i+1) + ap + b_2p^2)$,

(b) $a < p$,

(c) $b_0 < b_1$,

(d) b_1 ist ein Quadrat, und für alle $d \geq 2$ mit $d|b_1$ gilt $p|d$.

Sollten solche Zahlen $b_0, b_1, b_2 \in \mathbb{N}$ nicht existieren, so setzen wir $\beta(t, p, i) = 0$.

Aus der gerade gezeigten Behauptung folgt dann: Für jede Folge $(a_0, \ldots, a_r)$ über $\mathbb{N}$ gibt es $t, p \in \mathbb{N}$, so dass $\beta(t, p, i) = a_i$ für alle $0 \leq i \leq r$.

Außerdem ist β offensichtlich arithmetisch definierbar. □

Unentscheidbarkeit der Arithmetik

Wir können nun den Beweis für die Unentscheidbarkeit der Arithmetik beenden.

Wir müssen folgende Aussage durch eine arithmetische Formel (mit freien Variablen $x, x_1, \ldots, x_l$) ausdrücken:

Es gibt ein s und ein Tupel
$(y_0, y_1, \ldots, y_{k-1},\ y_k, y_{k+1}, \ldots, y_{2k-1}, \ldots, y_{sk}, y_{sk+1}, \ldots, y_{(s+1)k-1})$ mit:

- $y_0 = 1,\ y_1 = 0, \ldots, y_{k-1} = 0$
- $y_{sk} = x,\ y_{sk+1} = x_1, \ldots, y_{(s+1)k-1} = x_l$
- $(y_{ik}, \ldots, y_{(i+1)k-1}) \to_P (y_{(i+1)k}, \ldots, y_{(i+2)k-1})$ für alle $0 \leq i \leq s-1$

Beachte: $k = l + 1$ ist hierbei eine Konstante, die durch das RMP P festgelegt ist.

Unentscheidbarkeit der Arithmetik

Dies ist äquivalent zu: Es gibt s, t, p mit:

- $\beta(t, p, 0) = 1$, $\beta(t, p, 1) = 0$, $\ldots$, $\beta(t, p, k-1) = 0$
- $\beta(t, p, sk) = x$, $\beta(t, p, sk+1) = x_1, \ldots, \beta(t, p, (s+1)k-1) = x_l$
- Für alle $0 \leq i \leq s-1$ gilt:

$$\Big(\beta(t, p, ik), \ldots, \beta(t, p, (i+1)k-1)\Big) \to_P \Big(\beta(t, p, (i+1)k), \ldots, \beta(t, p, (i+2)k-1)\Big)$$

Eine arithmetische Formel für $(y, y_1, \ldots, y_l) \to_P (x, x_1, \ldots, x_l)$ ist einfach als Disjunktion über alle Anweisungen A_i des RMPs P anzugeben (Übung). □

Automatische Strukturen

Wir werden im folgenden automatische Strukturen einführen.

Die Hauptresultate zu automatische Strukturen, die wir beweisen, sind:

- Jede automatische Struktur hat eine entscheidbare Theorie.
- $(\mathbb{N}, +)$ ist automatisch.
- $(\mathbb{Q}, \leq)$ ist automatisch.

Konvolution von Wörtern

Sei $n \geq 1$. Sei Σ ein endliches Alphabet und sei $\# \notin \Sigma$.

Sei $\Sigma_\# = \Sigma \cup \{\#\}$ im Weiteren.

Seien $w_1, w_2 \ldots, w_n \in \Sigma^*$. Wir definieren die Konvolution

$$w_1 \otimes w_2 \otimes \cdots \otimes w_n \in \left(\Sigma_\#^n\right)^*$$

wie folgt:

- Sei $w_i = a_{i,1} a_{i,2} \cdots a_{i,\ell_i}$, d.h. $\ell_i = |w_i|$.
- Sei $\ell = \max\{\ell_1, \ldots, \ell_n\}$
- Für alle $1 \leq i \leq n$ und $\ell_i < j \leq \ell$ sei $a_{i,j} = \#$
- $w_1 \otimes w_2 \otimes \cdots \otimes w_n := (a_{1,1}, \ldots, a_{n,1})(a_{1,2}, \ldots, a_{n,2}) \cdots (a_{1,\ell}, \ldots, a_{n,\ell})$.

Beispiel: $abba \otimes babaaa = (a,b)(b,a)(b,b)(a,a)(\#,a)(\#,a)$

Synchrone Mehrbandautomaten

Ein synchroner n-Bandautomat A über dem Alphabet Σ ist ein gewöhnlicher endlicher Automat über dem Alphabet $\Sigma_{\#}^n$.

$\rightsquigarrow \; L(A) \subseteq \left(\Sigma_{\#}^n\right)^*$.

Es sei $K(A) = \{(w_1, \ldots, w_n) \mid w_1, \ldots, w_n \in \Sigma^*, w_1 \otimes \cdots \otimes w_n \in L(A)\}$.

Eine n-stellige Relation R über Σ^* ist synchron-rational, falls ein synchroner n-Bandautomat A mit $K(A) = R$ existiert.

Beachte: Elemente in $L(A)$ die nicht zu $\{w_1 \otimes \cdots \otimes w_n \mid w_1, \ldots, w_n \in \Sigma^*\}$ gehören, haben keinen Einfluss auf die Relation $K(A)$ (es handelt sich sozusagen um Müll).

Man kann aus A jedoch leicht einen synchronen n-Bandautomaten B mit $L(B) = L(A) \cap \{w_1 \otimes \cdots \otimes w_n \mid w_1, \ldots, w_n \in \Sigma^*\}$ konstruieren.

Beachte: $\{w_1 \otimes \cdots \otimes w_n \mid w_1, \ldots, w_n \in \Sigma^*\} \subseteq \left(\Sigma_{\#}^n\right)^*$ ist regulär.

Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:

v	b_0	b_1	b_2	$\cdots$	b_{m-1}	b_m	#	$\cdots$	#
u	a_0	a_1	a_2	$\cdots$	a_{m-1}	a_m	a_{m+1}	$\cdots$	a_n

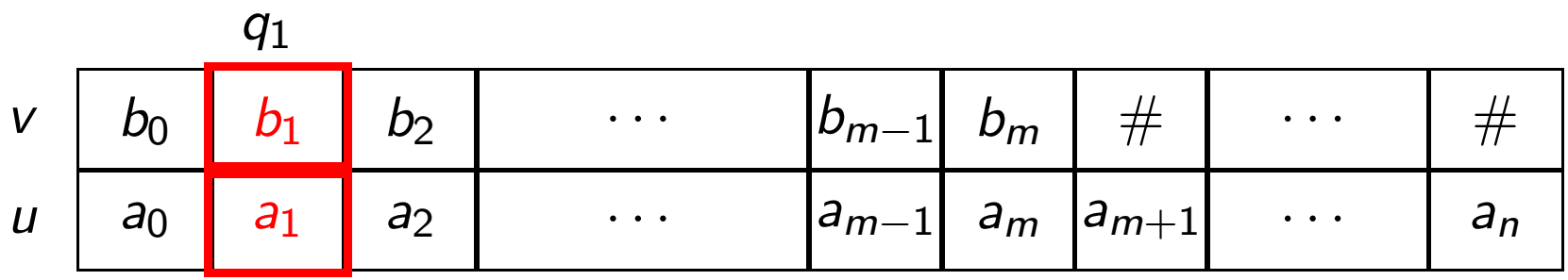
Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:

	q_0								
v	b_0	b_1	b_2	$\cdots$	b_{m-1}	b_m	$\#$	$\cdots$	$\#$
u	a_0	a_1	a_2	$\cdots$	a_{m-1}	a_m	a_{m+1}	$\cdots$	a_n

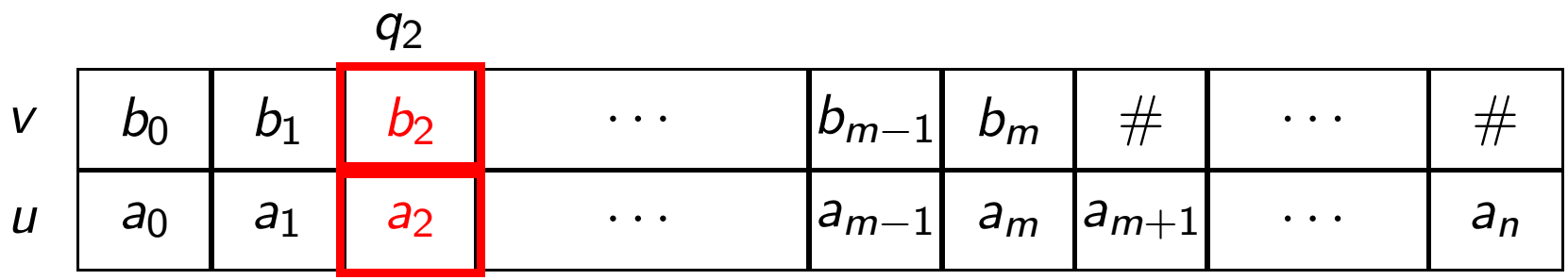
Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:



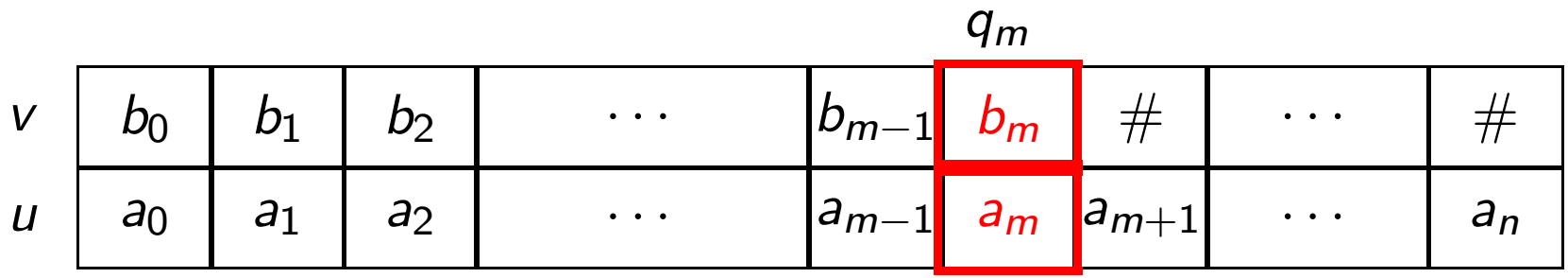
Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:



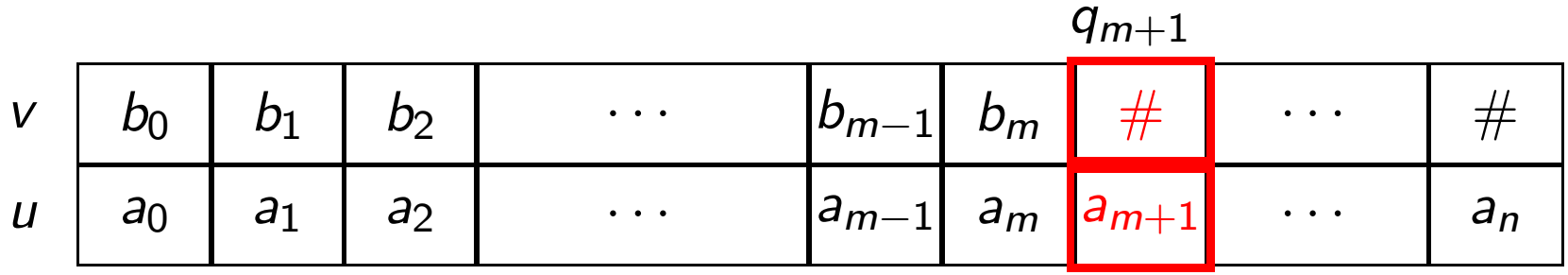
Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:



Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:



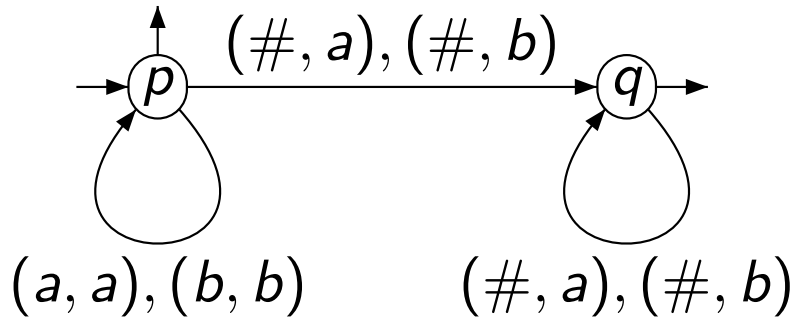
Synchrone Mehrbandautomaten

Veranschaulichung der Arbeitsweise eines synchronen Mehrbandautomaten:

									q_n
v	b_0	b_1	b_2	$\cdots$	b_{m-1}	b_m	$\#$	$\cdots$	$\#$
u	a_0	a_1	a_2	$\cdots$	a_{m-1}	a_m	a_{m+1}	$\cdots$	a_n

Synchrone Mehrbandautomaten

Beispiel 1: Sei A der folgende synchrone 2-Bandautomat:



Es gilt $K(A) = \{(u, v) \mid u, v \in \{a, b\}^*, u \text{ ist Präfix von } v\}$.

Automatische Strukturen

Definition

Eine relationale Struktur $\mathcal{A} = (A, R_1, \ldots, R_m)$ (wobei R_i eine n_i-stellige Relation ist) ist automatisch, falls ein endliches Alphabet Σ, ein endlicher Automat B über dem Alphabet Σ und synchrone n_i-Bandautomaten B_i über dem Alphabet Σ $(1 \leq i \leq m)$ existieren mit:

- $L(B) = A$
- $K(B_i) = R_i$ für $1 \leq i \leq m$

Definition

Eine Struktur $\mathcal{A}$ ist automatisch präsentierbar, falls $\mathcal{A}$ isomorph zu einer automatischen Struktur ist.

$(\mathbb{N}, +)$ ist automatisch

Satz

$(\mathbb{N}, +)$ mit $+ = \{(a, b, c) \mid a + b = c\}$ ist automatisch präsentierbar.

Beweis: Sei A ein endlicher Automat mit $L(A) = \{0\} \cup \{0, 1\}^* 1$.

Dann ist die folgende Abbildung $h : L(A) \to \mathbb{N}$ eine Bijektion:

$$h(w) = \text{die durch } w \text{ repräsentierte Binärzahl, rückwärts gelesen}$$

Sei B_+ der synchrone 3-Bandautomat auf der nächsten Folie.

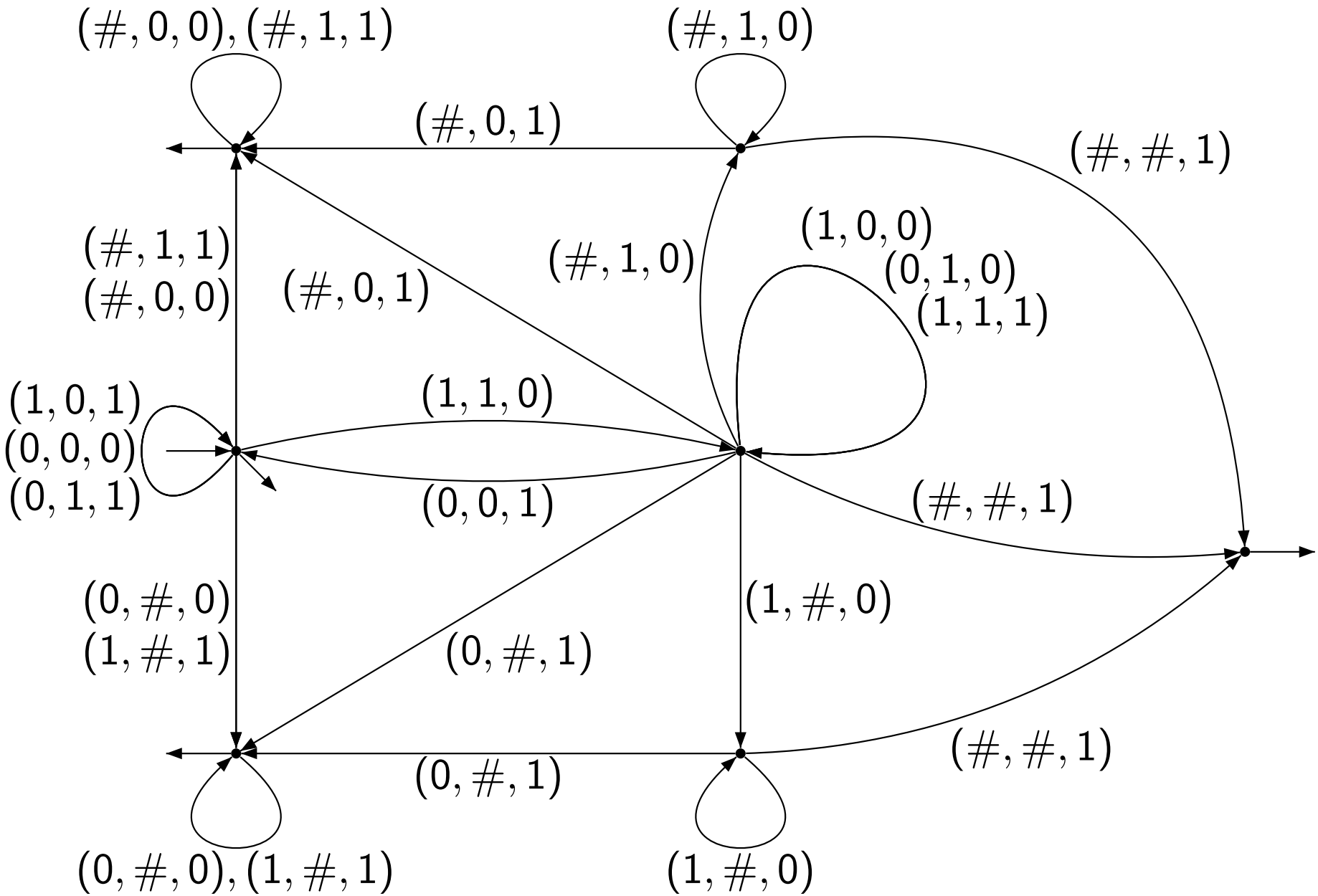
B_+ erkennt "fast" die Relation $\{(u, v, w) \in L(A)^3 \mid h(u) + h(v) = h(w)\}$, es gilt z. B. $(00, 0000, 0000) \in K(B_+)$.

Sei A_+ ein synchroner 3-Bandautomat mit

$$L(A_+) = L(B_+) \cap \{u \otimes v \otimes w \mid u, v, w \in L(A)\}.$$

Dann gilt $K(A_+) = \{(u, v, w) \in L(A)^3 \mid h(u) + h(v) = h(w)\}$. □

$(\mathbb{N}, +)$ ist automatisch



Weitere Beispiele

Man kann den vorherigen Satz noch erweitern: Sei $p > 1$ und $(\mathbb{N}, +, |_p)$, wobei $x |_p y$ genau dann, wenn $\exists n, k \in \mathbb{N} : x = p^n,\ y = k \cdot x$, ist automatisch präsentierbar.

Satz

$(\mathbb{Q}, \leq)$ ist automatisch präsentierbar.

Für den Beweis benutzen wir den Satz von Cantor.

Eine lineare Ordnung $(A, \leq)$ ist dicht falls gilt:

$$\forall x \forall y (x < y \rightarrow \exists z (x < z < y)).$$

Satz von Cantor

Seien $(A, \leq_A)$ und $(B, \leq_B)$ zwei abzählbare dichte lineare Ordnungen ohne kleinstes Element und ohne größtes Element. Dann sind $(A, \leq_A)$ und $(B, \leq_B)$ isomorph.

Satz von Cantor

Beweis des Satzes von Cantor:

Wir konstruieren Auflistungen

$$a_1, a_2, a_3, a_4, \ldots \text{ und } b_1, b_2, b_3, b_4, \ldots$$

mit folgenden Eigenschaften:

- $a_i \neq a_j$ und $b_i \neq b_j$ für $i \neq j$
- $A = \{a_i \mid i \geq 1\}$ und $B = \{b_i \mid i \geq 1\}$
- $a_i < a_j$ genau dann wenn $b_i < b_j$ für alle i, j.

Dann ist $f : A \to B$ mit $f(a_i) = b_i$ ein Isomorphismus.

Da A und B abzählbar unendlich sind, können wir beide Mengen auflisten:

$$A = \{x_1, x_2, x_3, \ldots\} \text{ und } B = \{y_1, y_2, y_3, \ldots\}$$

Der folgende "Algorithmus" konstruiert die Auflistungen:

Satz von Cantor

$L_A := [x_1, x_2, x_3, \ldots]$; $L_B := [y_1, y_2, y_3, \ldots]$

for all $i \geq 1$ **do** ($a_1, \ldots a_{i-1}$, $b_1, \ldots b_{i-1}$ sind bereits definiert)

if i ist ungerade **then**

sei x das erste Element aus L_A

entferne x aus der Liste L_A

sei y ein Element aus L_B mit folgender Eigenschaft:

$\forall 1 \leq j \leq i-1 : a_j < x \longleftrightarrow b_j < y$

entferne y aus der Liste L_B

$a_i := x$; $b_i := y$

else

sei y das erste Element aus L_B

entferne y aus der Liste L_B

sei x ein Element aus L_A mit folgender Eigenschaft:

$\forall 1 \leq j \leq i-1 : a_j < x \longleftrightarrow b_j < y$

entferne x aus der Liste L_A

$a_i := x$; $b_i := y$

endfor

$(\mathbb{Q}, \leq)$ ist automatisch

Beweis, dass $(\mathbb{Q}, \leq)$ automatisch ist:

Auf Grund des Satzes von Cantor genügt es, eine abzählbare dichte automatische lineare Ordnung ohne kleinstes und größtes Element anzugeben.

Sei hierzu $L = \{0,1\}^*1$.

Sei $\leq$ die lexikographische Ordnung auf L, d.h. für $x, y \in L$ gilt $x \leq y$ genau dann, wenn einer der folgenden Fälle gilt:

- Es gibt ein $u \in \{0,1\}^*$ mit $y = xu$ (x ist Anfangsstück von y)
- Es gibt $z, u, v \in \{0,1\}^*$ mit $x = z0u$ und $y = z1v$.

Offensichtlich ist $(L, \leq)$ eine lineare Ordnung.

- $(L, \leq)$ hat kein größtes Element:

  Sei $x \in L$ beliebig. Dann gilt $x < x1 \in L$

$(\mathbb{Q}, \leq)$ ist automatisch

- $(L, \leq)$ hat kein kleinstes Element:

  Sei $x = u1 \in L$ beliebig. Dann gilt $u01 < u1 = x$

- $(L, \leq)$ ist dicht:

  Seien $x, y \in L$ mit $x < y$ beliebig.

  1. Fall: $x = u1$, $y = u1v1$:

  Dann gilt: $x = u1 < u10^{|v|+1}1 < u1v1 = y$

  2. Fall: $x = u0v1$, $y = u1w$:

  Dann gilt: $x = u0v1 < u01^{|v|+2} < u1w = y$

- $(L, \leq)$ ist automatisch: Übung

Nicht automatisch präsentierbare Strukturen

Von den folgenden Strukturen kann man zeigen, dass sie nicht automatisch sind:

- $(\mathbb{R}, +)$ (denn jede automatische Struktur ist abzählbar)
- jede Struktur mit einer unentscheidbaren Theorie (siehe nächste Folie).

  Beispiele hierfür:

  - $(\mathbb{N}, +, \cdot)$ (Satz von Gödel)
  - $(\Sigma^*, \circ)$ (das freie Monoid über Σ) falls $|\Sigma| > 1$ (Quine 1946)

- $(\mathbb{N}, \cdot)$ und $(\mathbb{N}, |)$
- $(\mathbb{Q}, +)$ (Tsankov 2009)

Theorie einer automatischen Struktur

Unser Hauptresultat über automatische Strukturen lautet:

Satz (Khoussainov, Nerode 1994)

Für jede automatisch präsentierbare Struktur $\mathcal{A}$ ist $\mathrm{Th}(\mathcal{A})$ entscheidbar.

Korollar (Presburger 1929)

$\mathrm{Th}(\mathbb{N}, +)$ ist entscheidbar.

Korollar

$\mathrm{Th}(\mathbb{Q}, \leq)$ ist entscheidbar.

Beweis des Satzes von Khoussainov und Nerode:

Sei $\mathcal{A} = (L, R_1, \ldots, R_m)$ eine automatische Struktur mit $L \subseteq \Sigma^*$.

Für jede Formel F mit höchstens den freien Variablen $x_1, \ldots, x_n$ werden wir durch Induktion einen synchronen n-Bandautomaten B_F konstuieren, so dass gilt:

$$K(B_F) = \{(w_1, \ldots, w_n) \in L^n \mid \mathcal{A}_{[x_1/w_1]\cdots[x_n/w_n]} \models F\}.$$

Theorie einer automatischen Struktur

Fall 1: $F = R_i(x_{i_1}, \ldots, x_{i_k})$, wobei $1 \leq i_1, \ldots, i_k \leq n$:

Definiere den Homomorphismus $f : \left(\Sigma_\#^n\right)^* \to \left(\Sigma_\#^k\right)^*$ wie folgt, wobei $a_1, \ldots, a_n \in \Sigma_\#$:

$$f(a_1, \ldots, a_n) = \begin{cases} \varepsilon & \text{falls } a_{i_1} = \cdots = a_{i_k} = \# \\ (a_{i_1}, \ldots, a_{i_k}) & \text{sonst} \end{cases}$$

Beachte: $f(w_1 \otimes \cdots \otimes w_n) = w_{i_1} \otimes \cdots \otimes w_{i_k}$ für alle $w_1, \ldots, w_n \in \Sigma^*$.

Sei B_i der synchrone k-Bandautomat für R_i. Aus B_i konstruieren wir nun einen n-Bandautomaten B_F mit

$$L(B_F) = f^{-1}(L(B_i)) \cap \{w_1 \otimes \cdots \otimes w_n \mid w_1, \ldots, w_n \in L\}.$$

Beachte: Die regulären Sprachen sind unter inversen Homomorphismen abgeschlossen.

Theorie einer automatischen Struktur

Fall 2: $F = (x_i = x_j)$, wobei $1 \leq i, j \leq n$:

Analog zu Fall 1, da $\{(v, v) \mid v \in L\}$ synchron rational ist.

Fall 3: $F = \neg G$:

IH $\rightsquigarrow$ n-Bandautomat B_G für G

Wir wählen dann B_F so, dass gilt:

$$L(B_F) = \{w_1 \otimes \cdots \otimes w_n \mid w_1, \ldots, w_n \in L\} \setminus L(B_G)$$

Fall 4: $F = G \vee H$, wobei F höchstens freie Variablen $x_1, \ldots, x_n$ enthält:

IH $\rightsquigarrow$ n-Bandautomaten B_G, B_H für G und H

Wir wählen dann B_F so, dass gilt:

$$L(B_F) = L(B_G) \cup L(B_H)$$

Theorie einer automatischen Struktur

Fall 5: $H = \exists x_{n+1} : G(x_1, \ldots, x_n, x_{n+1})$:

IH $\rightsquigarrow$ $(n+1)$-Bandautomat B_G für G

Definiere den Homomorphismus $f : \left(\Sigma_{\#}^{n+1}\right)^* \to \left(\Sigma_{\#}^{n}\right)^*$ wie folgt, wobei $a_1, \ldots, a_n, a_{n+1} \in \Sigma_{\#}$:

$$f(a_1, \ldots, a_n, a_{n+1}) = \begin{cases} \varepsilon & \text{falls } a_1 = \cdots = a_n = \# \\ (a_1, \ldots, a_n) & \text{sonst} \end{cases}$$

Beachte: $f(w_1 \otimes \cdots \otimes w_n \otimes w_{n+1}) = w_1 \otimes \cdots \otimes w_n$ für alle $w_1, \ldots, w_{n+1} \in \Sigma^*$.

Dann wählen wir für B_F einen n-Bandautomaten mit $L(B_F) = f(L(B_G))$.

Beachte: Die regulären Sprachen sind unter Homomorphismen abgeschlossen.

Dies beendet die Konstruktion von B_F.

Theorie einer automatischen Struktur

Sei nun F eine Aussage (keine freien Variablen).

O.B.d.A. können wir davon ausgehen, dass F von der Form $F = \exists x\, G(x)$ ist (wir können immer einen Dummy-$\exists$-Quantor hinzufügen).

Dann gilt: $\mathcal{A} \models F \iff L(B_G) \neq \emptyset$.

Letzteres ist entscheidbar, da Leerheit der von einem endlichen Automaten akzeptierten Sprache entscheidbar ist. $\square$

Theorie einer automatischen Struktur

Bemerkungen zur Komplexität:

Unser Algorithmus, der $F \in \mathrm{Th}(\mathcal{A})$ entscheidet, ist nicht sehr effzient.

Grund: Für jede Negation $\neg$ in F müssen wir einen Automaten komplementieren. Dies verursacht einen exponentiellen Blow-Up in der Automatengröße.

Die Laufzeit unseres Algorithmus ist deshalb in etwa $f_{|F|}(O(1))$, wobei $f_0(n) = n$ und $f_{i+1}(n) = 2^{f_i(n)}$ für $i \geq 0$.

Dies ist jedoch auch nicht vermeidbar:

Sei $T_2 = (\{0,1\}^*, S_0, S_1, \leq)$ wobei:

- $S_0 = \{(w, w0) \mid w \in \{0,1\}^*\}$
- $S_1 = \{(w, w1) \mid w \in \{0,1\}^*\}$
- $\leq = \{(w, wu) \mid w, u \in \{0,1\}^*\}$

Beachte: T_2 ist eine automatische Struktur.

Theorie einer automatischen Struktur

Meyer 1974

Es gibt kein $i \in \mathbb{N}$ und einen Algorithmus, der $\mathrm{Th}(T_2)$ korrekt entscheidet und dessen Laufzeit durch $f_i(n)$ (bei einer Eingabeformel der Länge n) beschränkt ist.

Man sagt auch: Es existiert kein elementarer Algorithmus für $\mathrm{Th}(T_2)$.

Es gibt jedoch viele Spezialfälle von automatischen Strukturen, für die ein elementarer Algorithmus zur Entscheidung der Theorie existiert.

Hier sind zwei Beispiele:

Oppen 1978

Es existiert ein Algorithmus, der $\mathrm{Th}(\mathbb{N}, +)$ in Zeit $2^{2^{2^{O(n)}}}$ entscheidet.

Entscheidbarkeit der reellen Arithmetik

Satz (Tarski 1948)

$\mathrm{Th}(\mathbb{R}, +, \cdot)$ ist entscheidbar.

Beweis:

Zunächst betrachten wir anstatt $\mathrm{Th}(\mathbb{R}, +, \cdot)$ die Theorie $\mathrm{Th}(\mathbb{R}, +, \cdot, <, 0, 1, -1)$.

Wir schreiben im Folgenden $\mathbb{R}$ für $(\mathbb{R}, +, \cdot, <, 0, 1, -1)$.

Quantorenelimination: Wir konstruieren zu einer gegebenen prädikatenlogischen Formel F eine quantorenfreie Formel F' mit: $F \in \mathrm{Th}(\mathbb{R}) \iff F' \in \mathrm{Th}(\mathbb{R})$.

Es genügt, dies für eine Formel $F = \exists x\, G$ zu zeigen, wobei G quantorenfrei ist.

Seien $y_1, \ldots, y_n$ die freien Variablen von F.

Entscheidbarkeit der reellen Arithmetik

Außerdem können wir annehmen, dass G folgende Gestalt hat:

$$G \;=\; s(x, y_1, \ldots, y_n) = 0 \wedge \bigwedge_{i=1}^{m} t_i(x, y_1, \ldots, y_n) > 0,$$

wobei $s, t_1, \ldots, t_m \in \mathbb{Z}[x, y_1, \ldots, y_n]$.

Beachte hierzu:

- $\exists x(G_1 \vee G_2) \equiv (\exists x G_1) \vee (\exists x G_2)$
- $s_1 = s_2 \iff s_1 - s_2 = 0$
- $s_1 < s_2 \iff s_2 - s_1 > 0$
- $\neg(s = 0) \iff (s > 0 \vee -s > 0)$
- $\neg(s > 0) \iff (s = 0 \vee -s > 0)$
- $\bigwedge_{i=1}^{k} s_i = 0 \iff \sum_{i=1}^{k} s_i^2 = 0$

Entscheidbarkeit der reellen Arithmetik

Wir unterscheiden nun 3 Fälle:

- Fall 1: x kommt in s vor und $m = 1$
- Fall 2: x kommt in s vor und $m > 1$
- Fall 3: x kommt in s nicht vor.

Fall 1: $G = (s = 0 \wedge t > 0)$ mit $s = p_m x^m + \cdots + p_1 x + p_0$, $m \geq 1$, $p_i \in \mathbb{Z}[y_1, \ldots, y_n]$

Notation: Für $k \geq 0$ sei $(\#x : G) = k$ eine neue Formel mit:
Für alle $a_1, \ldots, a_n \in \mathbb{R}$ gilt $\mathbb{R}_{[y_1/a_1,\ldots,y_n/a_n]} \models (\#x : G) = k$ g.d.w.

$$|\{a \in \mathbb{R} \mid \mathbb{R}_{[x/a,y_1/a_1,\ldots,y_n/a_n]} \models G\}| = k.$$

Beachte: $\exists x G$ ist in $\mathbb{R}$ äquivalent zu

$$(\#x : G) = 1 \vee (\#x : G) = 2 \vee \cdots \vee (\#x : G) = m$$

Neues Ziel: Finde eine quantorenfreie Formel, welche in $\mathbb{R}$ äquivalent ist zu $(\#x : G) = k$.

Entscheidbarkeit der reellen Arithmetik

Unser Hilfsmittel sind sogenannte Sturmfolgen.

Für $\overline{a} = (a_1, \ldots, a_n) \in (\mathbb{R} \setminus \{0\})^n$ sei

$$\mathrm{Var}(\overline{a}) = |\{i < n \mid a_i a_{i+1} < 0\}|$$

(Anzahl der Vorzeichenwechsel).

Für $\overline{a} \in \mathbb{R}^n$ sei $\mathrm{Var}(\overline{a}) = \mathrm{Var}(\overline{b})$, wobei $\overline{b}$ aus $\overline{a}$ durch Löschen aller Nullen entsteht.

Für $\overline{f} = (f_1, \ldots, f_n) \in (\mathbb{R}[x])^n$ und $a \in \mathbb{R}$ sei

$$\mathrm{Var}_a(\overline{f}) = \mathrm{Var}(f_1(a), \ldots, f_n(a)).$$

Entscheidbarkeit der reellen Arithmetik

Seien $f, g \in \mathbb{R}[x]$. Definiere die Polynome $h_0(x), \ldots, h_n(x)$ eindeutig wie folgt (Euklidischer Algorithmus):

$$\begin{aligned}
h_0(x) &= f(x), \qquad h_1(x) = g(x) \\
h_0(x) &= q_1(x)h_1(x) - h_2(x) \qquad \deg(h_2) < \deg(h_1) \\
h_1(x) &= q_2(x)h_2(x) - h_3(x) \qquad \deg(h_3) < \deg(h_2) \\
&\vdots \\
h_{n-1}(x) &= q_n(x)h_n(x)
\end{aligned}$$

Dann gilt:

- $h_n(x) = \mathrm{ggT}(f, g)$
- Für alle $0 \leq i \leq n$ ist das Polynom $h_n(x)$ ein Teiler von $h_i(x)$.

Dann ist $[f, g] = (h_0(x), h_1(x), \ldots, h_n(x))$ die Sturmfolge von f und g.

Entscheidbarkeit der reellen Arithmetik

Die gekürzte Sturmfolge von f und g ist

$$\left(\frac{h_0(x)}{h_n(x)}, \frac{h_1(x)}{h_n(x)}, \ldots, \frac{h_{n-1}(x)}{h_n(x)}, 1\right).$$

Für eine quantorenfreie Formel H mit der einzigen freien Variablen x und $a, b \in \mathbb{R}$ mit $a < b$ sei

$$(\#x : H)_a^b = |\{c \in (a, b) \mid \mathbb{R}_{[x/c]} \models H\}|.$$

Mit $f'(x)$ bezeichnen wir die formale Ableitung des Polynoms $f(x) \in \mathbb{R}[x]$.

Satz von Sturm und Tarski

Seien $f, g \in \mathbb{R}[x]$, $\mathrm{ggT}(f, g) = \mathrm{ggT}(f, f') = 1$, $a, b \in \mathbb{R}$, $a < b$, $f(a) \neq 0 \neq f(b)$ (insbedondere $f \neq 0$). Dann gilt

$$(\#x : f(x) = 0 \wedge g(x) > 0)_a^b - (\#x : f(x) = 0 \wedge g(x) < 0)_a^b = \mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]).$$

Entscheidbarkeit der reellen Arithmetik

Für den Beweis des Satzes von Sturm und Tarski benötigen wir zwei Lemmata.

Lemma A

Seien $f, g \in \mathbb{R}[x]$, $a, b \in \mathbb{R}$, $a < b$, und $\forall c \in [a, b] : f(c) \neq 0$. Dann gilt $\mathrm{Var}_a([f, g]) = \mathrm{Var}_b([f, g])$.

Beweis von Lemma A: Sei

$$[f, g] = S = (h_0, h_1, \ldots, h_s)$$

und sei

$$S' = (h'_0, h'_1, \ldots, h'_s)$$

die gekürzte Sturmfolge, d.h. $h'_s = 1$ und $h'_i = \frac{h_i}{h_s}$.

Sei $N = \{c \in [a, b] \mid \exists 0 \leq i \leq s : h'_i(c) = 0\}$.

Dann ist N endlich.

Sei $[a', b'] \subseteq [a, b]$ ein Intervall mit $|N \cap [a', b']| \leq 1$.

Entscheidbarkeit der reellen Arithmetik

Es genügt es folgende Behauptung zu zeigen:

$$\mathrm{Var}_{a'}([f,g]) = \mathrm{Var}_{b'}([f,g]).$$

Fall 1: Kein h_i' hat eine Nullstelle in $[a', b']$.

Nach dem Zwischenwertsatz gilt $h_i'(a') \cdot h_i'(b') > 0$ für alle $0 \leq i \leq s$.

Also gilt $\mathrm{Var}_{a'}(S') = \mathrm{Var}_{b'}(S')$.

Wegen $h_s(a') \neq 0 \neq h_s(b')$ (denn $f(a') \neq 0 \neq f(b')$ und $h_s = \mathrm{ggT}(f,g)$ ist Teiler von f) folgt

$$\mathrm{Var}_{a'}(S) = \mathrm{Var}_{a'}(S') = \mathrm{Var}_{b'}(S') = \mathrm{Var}_{b'}(S).$$

Fall 2: Mindestens ein h'_i hat eine Nullstelle $c \in [a', b']$, d.h. $N \cap [a', b'] = \{c\}$.

Wegen $h'_s = 1$ und der Annahme, dass $f = h_0$ keine Nullstelle in $[a, b]$ hat (damit hat auch h'_0 keine Nullstelle in $[a, b]$) gilt $1 \leq i \leq s - 1$.

Es gilt $h'_{i-1}(c) = q_i(c)h'_i(c) - h'_{i+1}(c) = -h'_{i+1}(c)$.

Würde $h'_{i+1}(c) = 0 = h'_i(c)$ gelten, so wäre $h'_j(c) = 0$ für alle $j \geq i$, was $h'_s = 1$ widerspricht.

Also gilt $h'_{i+1}(c) \neq 0$ und damit

$$h'_{i-1}(c)h'_{i+1}(c) = -(h'_{i+1}(c))^2 < 0,$$

d.h. $h'_{i-1}(c)$ und $h'_{i+1}(c)$ haben verschiedene Vorzeichen.

Da h'_{i-1} und h'_{i+1} keine Nullstelle in $[a', b']$ haben, gilt

$$h'_{i-1}(a')h'_{i+1}(a') < 0 \quad \text{und} \quad h'_{i-1}(b')h'_{i+1}(b') < 0.$$

Entscheidbarkeit der reellen Arithmetik

Es folgt:

$$\begin{aligned}
\mathrm{Var}_{a'}(S') &= \mathrm{Var}(h'_0(a'), \ldots, h'_{i-1}(a'), h'_i(a'), h'_{i+1}(a'), \ldots, h'_s(a')) \\
&= \mathrm{Var}(h'_0(a'), \ldots, h'_{i-1}(a'), h'_{i+1}(a'), \ldots, h'_s(a')) \\
\mathrm{Var}_{b'}(S') &= \mathrm{Var}(h'_0(b'), \ldots, h'_{i-1}(b'), h'_i(b'), h'_{i+1}(b'), \ldots, h'_s(b')) \\
&= \mathrm{Var}(h'_0(b'), \ldots, h'_{i-1}(b'), h'_{i+1}(b'), \ldots, h'_s(b'))
\end{aligned}$$

Auf diese Weise können wir für alle j mit $h'_j(c) = 0$ die Einträge $h'_j(a')$ und $h'_j(b')$ eliminieren.

Wir erhalten somit

$$\begin{aligned}
\mathrm{Var}_{a'}(S') &= \mathrm{Var}(g_0(a'), \ldots, g_t(a')) \\
\mathrm{Var}_{b'}(S') &= \mathrm{Var}(g_0(b'), \ldots, g_t(b'))
\end{aligned}$$

wobei die Polynome $g_0, \ldots, g_t$ keine Nullstelle in $[a', b']$ haben.

Also gilt:

$$\begin{aligned} \mathrm{Var}_{a'}(S) = \mathrm{Var}_{a'}(S') &= \mathrm{Var}(g_0(a'), \ldots, g_t(a')) \\ &= \mathrm{Var}(g_0(b'), \ldots, g_t(b')) = \mathrm{Var}_{b'}(S') = \mathrm{Var}_{b'}(S) \end{aligned}$$

□

Lemma B

Seien $f, g \in \mathbb{R}[x]$, $\mathrm{ggT}(f,g) = \mathrm{ggT}(f,f') = 1$, $a, b, c \in \mathbb{R}$, $a < c < b$, $f(c) = 0$, $\forall d \in [a,b] \setminus \{c\} : f(d) \neq 0$. Dann gilt

$$\mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]) = \begin{cases} 1 & \text{falls } g(c) > 0 \\ -1 & \text{falls } g(c) < 0 \end{cases}$$

Beweis von Lemma B:

Wegen $\mathrm{ggT}(f, g) = \mathrm{ggT}(f, f') = 1$ haben f und g keine gemeinsamen Nullstellen, und f hat keine mehrfache Nullstelle.

Insbesondere gilt $g(c) \neq 0$ und es gibt ein Polynom $h(x)$ mit $f(x) = (x - c) \cdot h(x)$, $h(c) \neq 0$.

Also gilt $f\, f'\, g = (x - c) \cdot \underbrace{(h^2 g + (x - c)\, h\, h'\, g)}_{u(x)}$.

Sei $[f, f'g] = (f, f'g, h_2, \ldots, h_s)$ mit $s \geq 1$.

Gelte $g(c) > 0$ (der Fall $g(c) < 0$ kann analog analysiert werden).

Es gilt $u(c) = (h(c))^2 g(c) > 0$ und $f'g(c) \neq 0$.

Also gibt es $a' < b'$ mit $a \leq a' < c$, $c < b' \leq b$ und $\forall x \in [a', b'] : u(x) > 0$ und $f'g(x) \neq 0$.

Es folgt $f(a') \cdot f'g(a') < 0$ und $f(b') \cdot f'g(b') > 0$.

Entscheidbarkeit der reellen Arithmetik

Wir erhalten damit im Fall $s \geq 2$:

$$\begin{aligned}
\mathrm{Var}_a([f, f'g]) &\overset{\text{Lemma A}}{=} \mathrm{Var}_{a'}([f, f'g]) \\
&= 1 + \mathrm{Var}_{a'}([f'g, h_2]) \\
&\overset{\text{Lemma A}}{=} 1 + \mathrm{Var}_{b'}([f'g, h_2]) \\
&= 1 + \mathrm{Var}_{b'}([f, f'g]) \\
&\overset{\text{Lemma A}}{=} 1 + \mathrm{Var}_{b}([f, f'g])
\end{aligned}$$

Im Fall $s = 1$ (d.h. $[f, f'g] = (f, f'g)$) gilt

$$\begin{aligned}
\mathrm{Var}_a([f, f'g]) &\overset{\text{Lemma A}}{=} \mathrm{Var}_{a'}([f, f'g]) \\
&= 1 \\
&= 1 + \mathrm{Var}_{b'}([f, f'g]) \\
&\overset{\text{Lemma A}}{=} 1 + \mathrm{Var}_{b}([f, f'g])
\end{aligned}$$

Beweis des Satzes von Tarski und Sturm:

Seien $f, g \in \mathbb{R}[x]$, $\mathrm{ggT}(f, g) = \mathrm{ggT}(f, f') = 1$, $a, b \in \mathbb{R}$, $a < b$, $f(a) \neq 0 \neq f(b)$.

Sei $N = \{c \in (a, b) \mid f(c) = 0\}$ (endlich).

Falls $N = \emptyset$ gilt wegen Lemma A:

$$(\#x : f(x) = 0 \wedge g(x) > 0)_a^b - (\#x : f(x) = 0 \wedge g(x) < 0)_a^b = 0 = \mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]).$$

Sei nun $N = \{c_1, c_2, \ldots, c_n\}$ mit $n \geq 1$.

Wähle Punkte $a = a_0 < c_1 < a_1 < c_2 < a_2 < \cdots < a_{n-1} < c_n < a_n = b$.

Dann gilt wegen Lemma B für alle $1 \leq i \leq n$:

$$\mathrm{Var}_{a_{i-1}}([f, f'g]) - \mathrm{Var}_{a_i}([f, f'g]) = \begin{cases} 1 & \text{falls } g(c_i) > 0 \\ -1 & \text{falls } g(c_i) < 0 \end{cases}$$

Aufsummieren ergibt

$$\mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]) =$$
$$(\#x : f(x) = 0 \wedge g(x) > 0)_a^b - (\#x : f(x) = 0 \wedge g(x) < 0)_a^b$$

Dies beendet den Beweis des Satzes von Tarski und Sturm. □

Korollar aus dem Satz von Tarski und Sturm

Seien $f, g \in \mathbb{R}[x]$, $\mathrm{ggT}(f,g) = \mathrm{ggT}(f,f') = 1$, $a, b \in \mathbb{R}$, $a < b$, $f(a) \neq 0 \neq f(b)$ (insbedondere $f \neq 0$). Dann gilt

$$\begin{aligned}
&\#x(f(x) = 0 \wedge g(x) > 0)_a^b \\
&= \frac{1}{2}(\mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]) + \mathrm{Var}_a([f, f']) - \mathrm{Var}_b([f, f'])).
\end{aligned}$$

Beweis: Nach dem Satz von Tarski und Sturm gilt:

$$\begin{aligned}
&(\#x : f(x) = 0 \wedge g(x) > 0)_a^b - (\#x : f(x) = 0 \wedge g(x) < 0)_a^b \\
&= \mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]).
\end{aligned}$$

sowie (da f und g keine gemeinsame Nullstelle haben)

$$\begin{aligned}
&(\#x : f(x) = 0 \wedge g(x) > 0)_a^b + (\#x : f(x) = 0 \wedge g(x) < 0)_a^b \\
&= (\#x : f(x) = 0)_a^b = \\
&= (\#x : f(x) = 0 \wedge 1 > 0)_a^b - (\#x : f(x) = 0 \wedge 1 < 0)_a^b \\
&= \mathrm{Var}_a([f, f']) - \mathrm{Var}_b([f, f']).
\end{aligned}$$

Addieren der beiden Gleichungen ergibt:

$$\begin{aligned}
&2 \cdot \#x(f(x) = 0 \wedge g(x) > 0)_a^b \\
&= \mathrm{Var}_a([f, f'g]) - \mathrm{Var}_b([f, f'g]) + \mathrm{Var}_a([f, f']) - \mathrm{Var}_b([f, f'])
\end{aligned}$$

□

Entscheidbarkeit der reellen Arithmetik

Wir kommen nun zurück zu Fall 1.

Erinnerung: Wir müssen eine quantorenfreie arithmetische Formel für $(\#x : s = 0 \wedge t > 0) = k$ finden, wobei $s = p_m x^m + \cdots + p_1 x + p_0$, $m \geq 1$, $p_i \in \mathbb{Z}[y_1, \ldots, y_n]$

Wir können uns auf den Fall beschränken, dass $s = y_m x^m + \cdots y_1 x + y_0$ und $t = z_l x^l + \cdots z_1 x + z_0$ gilt.

Wir müssen dann für $(\#x : s = 0 \wedge t > 0) = k$ eine quantorenfreie arithmetische Formel in den freien Variablen $y_0, \ldots, y_m, z_0, \ldots, z_l$ finden.

Lemma (Schranke für Nullstellen eines Polynoms)

Sei $f(x) = a_m x^m + \cdots a_1 x + a_0 \in \mathbb{R}[x]$, $a_m \neq 0$. Dann liegen alle Nullstellen von f im Intervall $(-c, c)$ mit

$$c = \frac{|a_m| + \max\{|a_0|, \ldots, |a_{m-1}|\}}{|a_m|}.$$

Entscheidbarkeit der reellen Arithmetik

Es genügt daher für $(\#x : s = 0 \wedge t > 0)_y^z = k$ eine quantorenfreie arithmetische Formel in den freien Variablen $y, z, y_0, \ldots, y_m, z_0, \ldots, z_l$ zu finden.

In dieser Formel können wir dann y durch $-\frac{|y_m|+\max\{|y_0|,\ldots,|y_{m-1}|\}}{|y_m|}$ und z durch $\frac{|y_m|+\max\{|y_0|,\ldots,|y_{m-1}|\}}{|y_m|}$ ersetzen.

Anwendungen von $|\cdot|$ und max können durch eine große Fallunterscheidung ersetzt werden.

Ebenso muss der Fall $y_m = 0$ durch eine Fallunterscheidung abgefangen werden.

Anwendungen von $\frac{\cdot}{|y_m|}$ (im Fall $y_m \neq 0$) können durch Multiplizieren mit genügend großen Potenzen von y_m eliminiert werden.

Entscheidbarkeit der reellen Arithmetik

Wir konstruieren eine quantorenfreie arithmetische Formel in den freien Variablen $y, z, y_0, \ldots, y_m, z_0, \ldots, z_l$ für $(\#x : s = 0 \wedge t > 0)_y^z = k$ mittels des Satzes von Sturm und Tarski (Korollar hiervon).

Problem: Wie stellen wir die Vorbedingung $\mathrm{ggT}(s, t) = \mathrm{ggT}(s, s')$ sicher?

Lösung: Wir lassen den Euklidischen Algorithmus symbolisch mit $s = y_m x^m + \cdots y_1 x + y_0$ und $t = z_l x^l + \cdots z_1 x + z_0$ (bzw. s und s') laufen.

Dabei werden die Koeffizienten $y_m, \ldots, y_1, y_0$ durch z_l geteilt. Wir müssen daher den Fall $z_l = 0$ abfangen.

Entscheidbarkeit der reellen Arithmetik

Besipiel: $m = 2$, $l = 1$, d.h.

$$s(x) = y_2x^2 + y_1x + y_0 \text{ und } t(x) = z_1x + z_0$$

Dann gilt $(\#x : s = 0 \wedge t > 0)_y^z = k$ g.d.w.

$$(z_1 \neq 0 \wedge (\#x : z_1^2 \cdot s = 0 \wedge t > 0)_y^z = k) \vee$$
$$(z_1 = 0 \wedge (\#x : s = 0 \wedge z_0 > 0)_y^z = k).$$

Division mit Rest:

$$\begin{array}{l} (y_2z_1^2x^2 + y_1z_1^2x + y_0z_1^2) : (z_1x + z_0) = y_2z_1x + (y_1z_1 - y_2z_0) \\ \underline{-(y_2z_1^2x^2 + y_2z_1z_0x)} \\ ((y_1z_1^2 - y_2z_1z_0)x + y_0z_1^2) \\ \underline{-((y_1z_1^2 - y_2z_1z_0)x + (y_1z_1z_0 - y_2z_0^2))} \\ y_0z_1^2 - y_1z_1z_0 + y_2z_0^2 \text{ (Rest)} \end{array}$$

Entscheidbarkeit der reellen Arithmetik

Also:

- Wenn $z_1 \neq 0 \neq y_0 z_1^2 - y_1 z_1 z_0 + y_2 z_0^2$, dann gilt $\mathrm{ggT}(z_1^2 s, t) = 1$ und wir können Tarski-Sturm anwenden.
- Wenn $z_1 \neq 0 = y_0 z_1^2 - y_1 z_1 z_0 + y_2 z_0^2$, dann gilt
  $\mathrm{ggT}(z_1^2 s, t) = t = (z_1 x + z_0)$ und $\frac{z_1^2 s}{t} = y_2 z_1 x + (y_1 z_1 - y_2 z_0)$.
  Ausserdem gilt $(z_1 \neq 0 \wedge (\#x : z_1^2 \cdot s = 0 \wedge t > 0)_y^z = k)$ g.d.w.

  $$(z_1 \neq 0 \wedge (\#x : y_2 z_1 x + y_1 z_1 - y_2 z_0 = 0 \wedge t > 0)_y^z = k).$$

  Es gilt jetzt $\mathrm{ggT}(y_2 z_1 x + y_1 z_1 - y_2 z_0, t) = 1$.

Auf die gleiche Weise kann die Voraussetzung $\mathrm{ggT}(s, s') = 1$ sichergestellt werden.

Beachte hierzu: $(\#x : s = 0 \wedge t > 0)_y^z = k$ gdw.
$(\#x : s/\mathrm{ggT}(s, s') = 0 \wedge t > 0)_y^z = k$.

Entscheidbarkeit der reellen Arithmetik

Unter der Voraussetzung $\mathrm{ggT}(s,t) = \mathrm{ggT}(s,s') = 1$ (und $s(y) \neq 0 \neq s(z)$) ist $(\#x : s = 0 \wedge t > 0)_y^z = k$ äquivalent zu

$$\mathrm{Var}_y([s, s't]) - \mathrm{Var}_z([s, s't]) + \mathrm{Var}_y([s, s']) - \mathrm{Var}_z([s, s']) = 2k$$

Dies kann als eine Boolesche Kombination von Aussagen der Gestalt $\mathrm{Var}_y([s, s't]) = i$, $\mathrm{Var}_z([s, s't]) = i$, $\mathrm{Var}_y([s, s']) = j$ und $\mathrm{Var}_z([s, s']) = j$ geschrieben werden.

Eine Aussage $\mathrm{Var}_y([s, s't]) = i$ (analog für die anderen Polynome) kann schließlich durch eine quantorenfreie Formel ausgedrückt werden.

Hierzu lassen wir wieder symbolisch den Euklidischen Algorithmus für s und $s't$ laufen und berechnen so die Sturmfolge symbolisch.

Dies beendet die Behandlung von Fall 1.

Entscheidbarkeit der reellen Arithmetik

Fall 2: $G = (s = 0 \wedge \bigwedge_{i=1}^{m} t_i > 0)$, $m \geq 1$, und x kommt in s vor.

Induktion über m:

IA: $m = 1$. Siehe Fall 1.

IS: $m \geq 2$.

Sei $G' = (s = 0 \wedge \bigwedge_{i=1}^{m-2} t_i > 0)$.

$$\begin{aligned} &\#x(G' \wedge t_{m-1} > 0 \wedge t_m > 0) + \\ &\#x(G' \wedge t_{m-1} > 0 \wedge t_m < 0) \quad = \quad \#x(G' \wedge t_{m-1} t_m^2 > 0) \end{aligned} \tag{13}$$

$$\begin{aligned} &\#x(G' \wedge t_{m-1} > 0 \wedge t_m > 0) + \\ &\#x(G' \wedge t_{m-1} < 0 \wedge t_m > 0) \quad = \quad \#x(G' \wedge t_{m-1}^2 t_m > 0) \end{aligned} \tag{14}$$

$$\begin{aligned} &\#x(G' \wedge t_{m-1} > 0 \wedge t_m < 0) + \\ &\#x(G' \wedge t_{m-1} < 0 \wedge t_m > 0) \quad = \quad \#x(G' \wedge t_{m-1} t_m < 0) \end{aligned} \tag{15}$$

(13) + (14) - (15) ergibt:

$$\begin{aligned} 2 \cdot \#xG &= 2 \cdot \#x \cdot (G' \wedge t_{m-1} > 0 \wedge t_m > 0) \\ &= \#x(G' \wedge t_{m-1} t_m^2 > 0) \; + \\ &\quad \#x(G' \wedge t_{m-1}^2 t_m > 0) \; - \\ &\quad \#x(G' \wedge -t_{m-1} t_m > 0) \end{aligned}$$

Fall 3: $G = (s = 0 \wedge \bigwedge_{i=1}^{m} t_i > 0)$, $m \geq 1$, und x kommt in s nicht vor.

Dann ist $\exists x \; G$ äquivalent zu

$$s = 0 \wedge \exists x \bigwedge_{i=1}^{m} t_i > 0.$$

Sei $H = \bigwedge_{i=1}^{m} t_i > 0$.

Entscheidbarkeit der reellen Arithmetik

Sei $t = t_1 t_2 \cdots t_m$.

Behauptung: $\exists x\, H$ is äquivalent in $\mathbb{R}$ zu

$$\exists x_0 \forall x \leq x_0 : H \;\vee\; \exists x_0 \forall x \geq x_0 : H \;\vee\; \exists x\, (t'(x) = 0 \wedge H). \tag{16}$$

Die Implikation (16) $\Rightarrow \exists x\, H$ is klar.

Gelte nun $\mathbb{R} \models \exists x\, H$.

Hieraus folgt:

$$\begin{aligned}\mathbb{R} \;\models\; & \exists x_0 \forall x \leq x_0 : H \;\vee\; \exists x_0 \forall x \geq x_0 : H \;\vee \\ & \exists x_1\, \exists x\, \exists x_2\, (x_1 < x < x_2 \;\wedge\; \neg H[x/x_1] \wedge H \wedge \neg H[x/x_2])\end{aligned}$$

Entscheidbarkeit der reellen Arithmetik

Angenommen es gilt

$$\mathbb{R} \models \exists x_1\, \exists x\, \exists x_2\, (x_1 < x < x_2 \ \wedge\ \neg H[x/x_1] \wedge H \wedge \neg H[x/x_2])$$

Dann gibt es $x_1' < x < x_2'$ mit $t(x_1') = 0 = t(x_2')$ und $t_i(y) > 0$ für alle $y \in (x_1', x_2')$ und $1 \leq i \leq m$ (insbesondere $t(y) > 0$ für alle $y \in (x_1', x_2')$).

Aus dem Satz von Rolle folgt, dass ein x existiert mit $t'(x) = 0$ und $t_i(x) > 0$ für alle $1 \leq i \leq m$, d.h.

$$\mathbb{R} \models \exists x_0 \forall x \leq x_0 : H \ \vee\ \exists x_0 \forall x \geq x_0 : H \ \vee\ \exists x\, (t'(x) = 0 \wedge H).$$

Dies zeigt die Behauptung.

Es genügt also, eine quantorenfreie Formel für

$$\exists x_0 \forall x \leq x_0 : H \ \vee\ \exists x_0 \forall x \geq x_0 : H \ \vee\ \exists x\, (t'(x) = 0 \wedge H)$$

anzugeben.

Entscheidbarkeit der reellen Arithmetik

Für die Formeln $\exists x_0 \forall x \leq x\ H$ und $\exists x_0 \forall x \geq x\ H$ kann man leicht quantorenfreie Formeln angeben.

Beachte hierzu: Für ein Polynom $a_n x^n + \cdots a_1 x + a_0$ mit $a_n \neq 0$ gilt $\exists x_0 \forall x \leq x\ (a_n x^n + \cdots a_1 x + a_0 > 0)$ genau dann, wenn einer der beiden folgenden Fälle gilt:

- n gerade und $a_n > 0$
- n ungerade und $a_n < 0$