Theoretische Informatik 2

vorgetragen von Prof. Matthias Baaz[*]
erstellt von Christoph Pickl[†]

28. Juli 2006

[*] http://www.logic.at/people/baaz/
[†] http://stud4.tuwien.ac.at/~e0525580/

Inhaltsverzeichnis

1 Grundlagen **4**
- 1.1 Bewertung von Aussagen . . . 4
- 1.2 Normalformen . . . 5
- 1.3 relevante Gesetze und Regeln . . . 6

2 Aussagenlogik **7**
- 2.1 Operatoren . . . 7
- 2.2 Syntaktische Methode . . . 8
- 2.3 Semantische Methode . . . 9
- 2.4 Horner-Schema . . . 10

3 Prädikatenlogik **13**
- 3.1 Modelle . . . 14
- 3.2 Verfahren für einstellige Prädikate . . . 15
- 3.3 Unifikation . . . 15

4 Kalküle **18**
- 4.1 Sequentialkalkül . . . 18
  - 4.1.1 Interpretation von einem Sequenten . . . 18
  - 4.1.2 Ableitungsregeln . . . 18
- 4.2 Resolution . . . 19
- 4.3 Resolution in AL . . . 20
  - 4.3.1 Unit-Resolution . . . 20
- 4.4 Resolution in PL . . . 20
  - 4.4.1 Pränex-Normalform . . . 21
  - 4.4.2 Skolem-Normalform . . . 22
  - 4.4.3 Faktorisierung . . . 23

5 Rekursionstheorie **24**
- 5.1 Grundlagen . . . 24
  - 5.1.1 Abzählbarkeit . . . 24
  - 5.1.2 Die totale Funktion f . . . 24
  - 5.1.3 Die numerische Funktion g . . . 25
- 5.2 Funktionen, Funktionen . . . 25
  - 5.2.1 μ-rekursive Funktionen . . . 25
  - 5.2.2 Grundfunktionen . . . 25
  - 5.2.3 Funktionale . . . 26
  - 5.2.4 Funktionsfamilien . . . 27
  - 5.2.5 μ-rekursive Programme . . . 27
- 5.3 Beispiel numerische Funktionen . . . 28

6 Aufgaben **30**

Vorwort

Dieses Dokument entstand im Laufe der Vorlesung "*Theoretische Informatik 2*" im SS06 von Prof. Matthias Baaz an der Technischen Universität Wien. Die LVA wurde in Form einer zweiwöchigen Blocklehrveranstaltung abgehalten und umfasste grob die Kapiteln Logik (Aussagen- als auch Prädikatenlogik), Automatische Beweisverfahren (Sequentialkalül, Resolution) und Komplexitätstheorie/Berechenbarkeit.

Ich möchte darauf hinweisen, dass dies **kein** offizielles Skriptum ist, sondern nur eine private Mitschrift und könnte somit unvollständig und/oder fehlerhaft sein. Wenn dem so ist, bitte ich herzlichst um eine kurze Mail an `e0525580@student.tuwien.ac.at`.

An dieser Stelle möchte ich auch der Mitschriften-Tausch-Börse[1], dem Informatik-Forum[2], der Fachschaft Informatik[3] als auch Paul Staroch[4] für sein Skriptum zu "*Informatik und Gesellschaft 2*" danken.

Außerdem seien noch die zwei von Prof. Baaz zur Verfügung gestellten Dokumente hervorgehoben:

- Übungsbeispiele: `http://logic.at/people/terwijn/aufgaben.ps`
- Sequentialkalkül: `http://logic.at/people/terwijn/gentzen.ps`

[1] `http://www.mtb-projekt.at.tf/`
[2] `http://www.informatik-forum.at/`
[3] `http://www.fsinf.at/`
[4] `http://stud4.tuwien.ac.at/~e0425426/IuG/Mitschrift.pdf`

1 Grundlagen

Begriffe:

Logik: diese untersucht allgemeine Prinzipien korrekten Schließens
mathematische Logik: stellt zu diesem Zweck formale Kalküle bereit und analysiert die Beziehung zwischen Syntax und Semantik von Aussagen
Syntax: Vorschriften/Regeln, welche festlegen ob Symbole/Zeichenketten zu einer Sprache gehören (oder auch nicht)
Semantik: die konkrete Interpretation, und die daraus resultierende Bedeutung von Symbole/Worte/Sprachkonstrukte
Wahrheitswert: ein boolescher[5] Wert der nur *wahr*/*falsch* annehmen kann
Konstante: ein fixer Wert, per Konvention zb: a, b, c, d
Variable: ein sich verändernder Wert, per Konvention zb: x, y, z, u, v, w
Funktion: arbeitet mit Objekten und liefert Wert zurück, zb: f, g, h
Prädikat: eine Funktion der Form $f : D \to b \in \{0, 1\}$, zb: P, Q, R
Quantor: Erweiterung der AL durch Existenz- ($\exists$) / All-Quantor ($\forall$) in PL
Operator: kann als n-stellige Funktion angesehen werden
Präfix-Notation: der Operator steht *vor* den Operanden, zb: $\circ(x, y)$
Infix-Notation: der Operator steht *zwischen* den Operanden, zb: $x \circ y$
Konjunktion: nichts weiter als eine UND-Verknüpfung: $\wedge$
Disjunktion: nichts weiter als eine ODER-Verknüpfung: $\vee$
Klausel: Form, die nur aus disjunktiv verknüpften Literalen besteht
Literal: negierte oder nicht-negierte Variablen, zb: x, $\neg x$
Interpretation: (auch "Belegung") weist einer Unbekannten einen konkreten Wert zu, erst mit der Interpretation entstehen aus Daten Informationen
Formel: drückt ein mathematisches Gesetz mit Hilfe von Variablen und Regel/Vorschriften die miteinander in Beziehung stehen aus
Atomformel: in der PL sind dies auf Variablen angewandte Prädikate
Regelsystem: bestehend aus Ableitungs-/Inferenzregeln um Formelbeweise (Ableitungen) vorzunehmen
Axiom: eine als wahr vorausgesetzte (Grund-)Formel
Theorem: von Axiomenmenge (durch Logik) abgeleitete Formel

:Begriffe

1.1 Bewertung von Aussagen

Sei $\mathcal{I}$ die Menge aller möglichen Interpretation, I eine konkrete Interpretation daraus, F die zu bewertende Formel und $\mathcal{M}$ (von *Meaning*) die resultierende Bedeutung von F anhand I, so lassen sich die möglichen Bewertungen kompakt anschreiben – wie in der nachfolgenden Tabelle ersichtlich ist.

[5] benannt nach *George Boole*, eingeführt im 19. Jhdt für algebraische Methoden in der AL

Bewertung	Beschreibung	formal angeschrieben
"**gültig**"	ergibt immer *wahr*	$\forall I \in \mathcal{I} : \mathcal{M}\big(F, I\big) = \mathbf{1}$
"**erfüllbar**"	kann *wahr* ergeben	$\exists I \in \mathcal{I} : \mathcal{M}\big(F, I\big) = \mathbf{1}$
"**widerlegbar**"	kann *falsch* ergeben	$\exists I \in \mathcal{I} : \mathcal{M}\big(F, I\big) = \mathbf{0}$
"**unerfüllbar**"	ergibt immer *falsch*	$\forall I \in \mathcal{I} : \mathcal{M}\big(F, I\big) = \mathbf{0}$

Tabelle 1: die vier möglichen Bewertungen von Aussagen

Somit sind folgende Aussagen automatisch immer wahr:

- Wenn $\forall I' \in \mathcal{I} : \mathcal{M}(F, I) = \mathbf{1} \Rightarrow F$ nicht nur erfüllbar, sondern auch gültig.
- Wenn F nicht gültig ist $\Rightarrow F$ bestimmt widerlegbar (wenn nicht sogar unerfüllbar).
- Ist $\neg F$ unerfüllbar $\Rightarrow F$ gültig ("indirekter Beweis").
- Ist F nur erfüllbar $\Rightarrow F$ auch widerlegbar – und vice versa.

1.2 Normalformen

Unter der Normalform einer Formel versteht man eine standardisierte vereinfachte Variante, die äquivalent zur ursprünglichen Formel ist.

F in **disjunktiver Normalform** besteht aus Disjunktionen von Konjunktionsterme:

$$DNF(F) = (\ldots \wedge \ldots \wedge \ldots) \bigvee (\ldots \wedge \ldots \wedge \ldots) \bigvee \ldots$$

Eine Formel in DNF (und dazugehörige Anschreibung in Mengen-Notation) wäre zb:

$$(x \wedge y) \bigvee (\neg x \wedge y) \quad = \quad \Big\{ \{x, y\}, \{\neg x, y\} \Big\}$$

F in **konjunktiver Normalform** besteht aus Kunjunktionen von Disjunktionsterme:

$$KNF(F) = (\ldots \vee \ldots \vee \ldots) \bigwedge (\ldots \vee \ldots \vee \ldots) \bigwedge \ldots$$

Eine Formel in KNF (und dazugehörige Anschreibung in Mengen-Notation) wäre zb:

$$(x \vee \neg y) \bigwedge (\neg x \vee \neg y) \quad = \quad \Big\{ \{x, \neg y\}, \{\neg x, \neg y\} \Big\}$$

Hinweis: Unter einer Klauselnormalform versteht man eine KNF.

Die Pränex- und die Skolemnormalform sind zwei sehr ähnliche Vorstufen zu einer DNF/KNF. Bei beiden bestehen die ersten drei Schritte aus dem Transformieren der Operatoren (`TRANSFORM`), dem "Nach-innen-ziehen" von Negationen (`NEGATION`) und der Umbennung von mehrfach vorkommenden Variablennamen (`SYNTAX`, "Syntaxbereinigung").

Bei der PNF werden die Quantoren – unter Berücksichtigung ihrer Anordnung – nach vorne gezogen (`SHIFT`), wo hingegen bei der SNF die Existenquantoren $\exists$ durch neue Konstanten- bzw Funktionssymbole ersetzt werden und Allquantoren $\forall$ weggelassen werden können (`SKOLEM`).

Siehe mehr dazu auf Seite 21.

Weiters sei noch die Negationsnormalform (NNF) angemerkt, welche entsteht, nachdem alle Negationen nach innen gezogen wurden (`NEGATION`).

1.3 relevante Gesetze und Regeln

De Morgan'sche Gesetze

Diese Gesetze sind an und für sich mit etwas Überlegen und Logik herleitbar.

Wenn es richtig ist, dass x oder y falsch ist (das *oder* zuerst ausgewertet), dann müssen beide falsch sein. Genauso wenn beide Variablen zusammen falsch ergeben (*zusammen/oder* zuerst ausgewertet), dann muss eine von beiden falsch sein.

$$\begin{aligned}\neg(x \vee y) &= \neg x \wedge \neg y \\ \neg(x \wedge y) &= \neg x \vee \neg y\end{aligned}$$

Folgerung:

$$\begin{aligned}x \vee y &= \neg(\neg x \wedge \neg y) \\ x \wedge y &= \neg(\neg x \vee \neg y)\end{aligned}$$

Und weil es so nett ist, existiert noch diese Umformung $\neg(A \to B) = A \wedge \neg B$, da:

$$\neg(A \to B) \quad\Rightarrow\quad \neg(\neg A \vee B) \quad\Rightarrow\quad \neg\neg A \wedge \neg B \quad\Rightarrow\quad A \wedge \neg B$$

Natürlich sollten die **Distributivgesetze** jedem noch immer im Gedächtnis sein:

$$x * (y \circ z) = (x * y) \circ (x * z)$$
$$(x \circ y) * z = (x * z) \circ (y * z)$$

Dadurch, dass die Operatoren $\wedge$ und $\vee$ assoziativ sind, kann folgendermaßen eine **Klammereinsparung** erfolgen:

$$\begin{array}{ccc} F_1 \wedge \ldots \wedge F_n & \text{steht für} & (F_1 \wedge (\ldots \wedge F_n)) \\ F_1 \vee \ldots \vee F_n & \text{steht für} & (F_1 \vee (\ldots \vee F_n)) \end{array}$$

Beispiel A: Sei $\Pi = F \to (G \to F)$, bewerten Sie Π.

F	G	$G \to F$	Π
0	0	**1**	**1**
0	**1**	0	**1**
1	0	**1**	**1**
1	**1**	**1**	**1**

Da $\forall I \in \mathcal{I} : M_{AL}(\Pi, I) = \mathbf{1}$, ist die Formel Π gültig und daher auch erfüllbar.

:Beispiel

2 Aussagenlogik – AL

Begriffe:

klassische Aussagenlogik: diese kennt – im Gegensatz zur Fuzzy-Logik – nur die zwei möglichen Wahrheitswerte *wahr* (**1**) und *falsch* (0)
aussagenlogische Konnektive: damit sind Funktionen über den Wahrheitswerten gemeint (Junktoren und Operatoren)
funktional vollständige Funktionen: ist eine Operatorenmenge, wenn alle anderen Operatoren durch diese ausgedrückt werden können, zb: $\{\overline{\wedge}\}$

:Begriffe

2.1 Operatoren

x	$\neg$
0	1
1	0

x	y	$\vee$	$\wedge$	$\rightarrow$	$\leftarrow$	$\nrightarrow$	$\nleftarrow$	$\underline{\vee}$	$\overline{\wedge}$	$\Leftrightarrow$	$\nLeftrightarrow$
0	0	0	0	**1**	**1**	0	0	**1**	**1**	**1**	0
0	**1**	**1**	0	**1**	0	0	**1**	0	**1**	0	**1**
1	0	**1**	0	0	**1**	**1**	0	0	**1**	0	**1**
1	**1**	**1**	**1**	**1**	**1**	0	0	0	0	**1**	0

Tabelle 2: Operatoren in AL – Wahrheitstabelle

$\circ$	**Operator**	**Beschreibung**
$\neg$	Negation	liefert den invertierten (umgekehrten) werd des Terms
$\vee$	OR	liefert **1**, wenn mindestens ein Term **1** ist ("Disjunktion")
$\wedge$	AND	liefert **1**, wenn beide Terme **1** sind ("Konjunktion")
$\rightarrow$	Implikation	liefert 0, wenn von **1** auf 0 geschlossen wird, ansonsten **1**
$\leftarrow$	Implikation	da Implikation nicht kommutativ ist, selbiges umgedreht
$\nrightarrow$	$\neg$ Impl R	eine negierte Implikation, anders ausgedrückt: $\neg(A \rightarrow B)$
$\nleftarrow$	$\neg$ Impl L	da Implikation nicht kommutativ ist, selbiges umgedreht
$\underline{\vee}$	NOR	ein negiertes Oder, anders ausgedrückt: $\underline{\vee} \Leftrightarrow \neg(A \vee B)$
$\overline{\wedge}$	NAND	ein negiertes Und, anders ausgedrückt: $\overline{\wedge} \Leftrightarrow \neg(A \wedge B)$
$\Leftrightarrow$	Äquivalenz	liefert **1**, wenn beide Terme gleich sind
$\nLeftrightarrow$	$\neg$ Äquivalenz	liefert **1**, wenn beide Terme verschieden sind

Tabelle 3: Operatoren in AL – Erklärung

Die sogenannte **Prezedenz** (welcher Operator zuerst ausgewertet wird) ist in dieser Reihenfolge definiert: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.

2.2 Syntaktische Methode

Die syntaktische (oder auch algebraische) Methode formt eine Formel durch folgende Transformierungen in äquivalente Gleichungen um. Die Korrektheit dieses Verfahrens beruht auf drei Tatsachen:

1. Alle angegebenen Gleichungen sind gültige Äquivalenzen.
2. Auf jede Formel, die noch nicht in DNF/KNF ist, ist eine der Äquivalenzen anwendbar.
3. Da es keine unendlich langen Umformungsketten gibt, terminiert das Verfahren.

1. `REPLACE`: Alle Operatoren werden durch $\vee$, $\wedge$ und $\neg$ ersetzt:

$$\begin{array}{rcl@{\qquad}rcl}
x \rightarrow y & = & \neg x \vee y & x \nrightarrow y & = & x \wedge \neg y \\
x \leftarrow y & = & x \vee \neg y & x \nleftarrow y & = & \neg x \wedge y \\
x \veebar y & = & \neg x \wedge \neg y & x \barwedge y & = & \neg x \vee \neg y \\
x \Leftrightarrow y & = & (x \wedge y) \vee (\neg x \wedge \neg y) & x \nLeftrightarrow y & = & (\neg x \wedge y) \vee (x \wedge \neg y)
\end{array}$$

2. `NEGATION`: Alle Negationen werden "nach innen" gezogen (De Morgan Gesetze), doppelte Negationen werden eliminiert:

$$\neg(x \wedge y) = \neg x \vee \neg y \qquad \neg(x \vee y) = \neg x \wedge \neg y \qquad \neg\neg x = x$$

3. `ELIMINATE`: Eventuell auftretende Konstanten werden eliminiert:

$$x \wedge \mathbf{1} = x \qquad x \wedge 0 = 0 \qquad x \vee \mathbf{1} = \mathbf{1} \qquad x \vee 0 = x \qquad \neg\mathbf{1} = 0 \qquad \neg 0 = \mathbf{1}$$

4. `TRANSFORM`: Durch mehrmaliges anwenden der **Distributivgesetze** gelangt man letztendlich zur gewünschten Normalform:

$$DNF : x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \qquad KNF : x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

Abbildung 1: syntaktische Vorgangsweise in AL zur Umformung in die DNF/KNF

Beispiel B: Konstruktion der Normalformen mit der syntaktischen Methode

Es sei die Formel $F = (x \barwedge (y \vee z)) \rightarrow (\neg u \wedge \neg\neg\mathbf{1})$ gegeben.

Formel F:	$(x \barwedge (y \vee z)) \rightarrow (\neg u \wedge \neg\neg\mathbf{1})$
`REPLACE`:	$\neg(x \barwedge (y \vee z)) \vee (\neg u \wedge \neg\neg\mathbf{1})$
`REPLACE`:	$\neg(\neg x \vee \neg(y \vee z)) \vee (\neg u \wedge \neg\neg\mathbf{1})$
`NEGATION`:	$(\neg\neg x \wedge \neg\neg(y \vee z)) \vee (\neg u \wedge \neg\neg\mathbf{1})$
`NEGATION`:	$(x \wedge (y \vee z)) \vee (\neg u \wedge \mathbf{1})$
`ELIMINATE`:	$(x \wedge (y \vee z)) \vee \neg u$
`TRANSFORM`:	Distributivgesetzte anwenden …
DNF(F):	$(x \wedge y) \vee (x \wedge z) \vee \neg u$
KNF(F):	$(x \vee \neg u) \wedge (y \vee z \vee \neg u)$

:Beispiel

2.3 Semantische Methode

Diese Methode ist nur für wenige Variablen geeignet, da ansonsten die zu verwendende Wahrheitstabelle zu groß wird.

Gegeben sei eine beliebige binäre Funktion $\circ$, welche folgende Wahrheitswerte erzeugt:

x	y	$\circ(x,y)$		.NF	
0	0	**1**		D	$\neg x \wedge \neg y$
0	**1**	0	$\Rightarrow$	K	$\neg(\neg x \wedge y) = x \vee \neg y$
1	0	**1**		D	$x \wedge \neg y$
1	**1**	**1**		D	$x \wedge y$

Abbildung 2: semantische Vorgangsweise in AL zur Umformung in die DNF/KNF

Die wahr-liefernden Konfigurationen dienen zur Erzeugung der DNF, die falsch-liefernden der KNF, welche aber zuvor noch negiert werden muss (siehe De Morgan'sche Regeln Seite 6). Nun kann man beide Normalformen in Mengen-Notation angeben:

$$DNF(\circ) = \Big\{\{\neg x, \neg y\}, \{x, \neg y\}, \{x, y\}\Big\} \text{ und } KNF(\circ) = \Big\{\{x, \neg y\}\Big\}$$

Interessant sind nun die **Extremfälle** Tautologie und Kontradiktion:

Bei einer *Tautologie* werden alle Variablenbelegungen für die DNF verwendet (also eine beliebige Interpretation), bzw keine für die KNF (da das Resultat von jeglicher Variable unabhängig – **1** – ist).

Bei einer *Kontradiktion* hingegen werden alle Variablenbelegungen für die KNF verwendet (durch duales Vorkommen der Literale ist diese Bedingung niemals erreichbar), bzw keine für die DNF (Unabhängigkeit des Resultats; es $\nexists\, I$, sodass $M_{AL}(F, I) = \mathbf{1}$).

Redundanzelimination

- **Tautologie-Elimination**: Die Klausel der Form $\{x, \neg x\}$ wird gestrichen.
- **Subsumtion**: Wenn gilt $A \subseteq B$, kann A gestrichen werden.

Beispiel C: Konstruktion der Normalformen mit der semantischen Methode

Es sei die Formel $F = (x \to y) \land (x \to z)$ gegeben.

x	y	z	$x \to y$	$x \to z$	F	.NF
0	0	0	**1**	**1**	**1**	D
0	0	**1**	**1**	**1**	**1**	D
0	**1**	0	**1**	**1**	**1**	D
0	**1**	**1**	**1**	**1**	**1**	D
1	0	0	0	0	0	K
1	0	**1**	0	**1**	0	K
1	**1**	0	**1**	0	0	K
1	**1**	**1**	**1**	**1**	**1**	D

$$\begin{aligned}
\mathrm{DNF}(F) &= (\neg x \land \neg y \land \neg z) \lor (x \land y \land z) \lor (x \land \neg y \land z) \lor \\
& (x \land y \land z) \lor (x \land y \land z) \\
&= \Big\{\{\neg x, \neg y, \neg z\}, \{x, y, \neg z\}, \{x, \neg y, z\}, \{x, \neg y, \neg z\}, \{x, y, z\}\Big\} \\
\mathrm{KNF}(F) &= \neg\Big((x \land \neg y \land \neg z) \lor (x \land \neg y \land z) \lor (x \land y \land \neg z)\Big) \\
&= \Big\{\{\neg x, y, z\}, \{\neg x, y, \neg z\}, \{\neg x, \neg y, z\}\Big\}
\end{aligned}$$

:Beispiel

2.4 Horner-Schema

Begriffe:

Hornformel: besteht aus Konjunktionen von Hornklauseln
Hornklausel: besteht aus Disjunktionen von Literalen, wobei $\exists$ höchstens ein positives Literal

:Begriffe

Es existieren bestimmte Klassen von Formeln für die es besonders schnelle Verfahren gibt um ihre Erfüllbarkeit zu beweisen. Das Horner-Verfahren arbeitet nun mit Konjunktionen von Teilformeln folgender Art:

$$\begin{gathered}
1 \to x \\
x_1 \land \ldots \land x_n \to 1 \\
x_1 \land \ldots \land x_n \to y
\end{gathered}$$

Formeln zu Klauseln

Die Umwandlung zwischen Formeln und Klauseln geschieht nach folgenden Regeln:

Hornformel:	$(a \land b \to c)$	$\land$	$(1 \to a)$	$\land$	$(1 \to b)$	$\land$	$(c \to 0)$
Hornklausel:	$\{\neg a, \neg b, c\}$		$\{a\}$		$\{b\}$		$\{\neg c\}$

Eine Hornformel ist folgendermaßen aufzufassen:

$$a \land b \land c \to d \quad = \quad \neg(a \land b \land c) \lor d \quad = \quad \neg a \lor \neg b \lor \neg c \lor d$$

Es existieren zwei mögliche Typen von Hornklauseln:

Name	Disjunktion	Implikation	Beschreibung
Zielklausel	$\neg x_1 \lor \ldots \lor \neg x_n$	$x_1 \land \ldots \land x_n \to 0$	$\nexists$ positives Literal
definite Hornkl.	$\neg x_1 \lor \ldots \lor \neg x_n \lor y$	$x_1 \land \ldots \land x_n \to y$	$\exists$ 1 positives Literal

Tabelle 4: zwei Typen von Hornklauseln: Zielklausel und definite Hornklausel

Markierungsalgorithmus

Mit diesem Algorithmus (auch "Erfüllbarkeitsalgorithmus" genannt) kann man die Erfüllbarkeit einer Formel in Polynomialzeit testen und damit feststellen, ob es eine Variablenbelegung gibt, damit die Hornformel wahr ist.

1. `INIT`: markiere jede vorkommende **1** (wenn $\nexists$ **1** $\Rightarrow$ fertig, Formel ist erfüllbar)
2. `MARK`: markiere rechte Seite, wo linke Seite schon vollkommen markiert wurde
   - `ABORT`: wenn eine **0** markiert wurde stoppen, Formel ist unerfüllbar
3. `UPDATE`: gerade rechts markiertes auch überall links markieren, gehe zu 2.

Abbildung 3: Vorgangsweise des Markierungsalgorithmus zum Horner-Schema

Variablenbelegung

Jede markierte Variable wird mit *wahr* und jede unmarkierte mit *falsch* belegt, um ein Modell einer erfüllbaren Formel anzugeben. Eine unerfüllbare Formel besitzt logischerweise kein Modell und jede Belegung ist ein Gegenbeispiel.

Beispiel D: Horner-Schema: Anwendung des Markierungsalgorithmus

$$F_{AL} = (1 \to A) \wedge (A \wedge B \to C) \wedge (A \to B) \wedge (C \to D) \wedge (C \wedge D \to 0)$$

`INIT`: alle 1er markieren
$(\underline{\mathbf{1}} \to A) \wedge (A \wedge B \to C) \wedge (A \to B) \wedge (C \to D) \wedge (C \wedge D \to 0)$

`MARK`: markiere Variable A, da linke Seite vollständig markiert
$(\underline{1} \to \underline{\mathbf{A}}) \wedge (A \wedge B \to C) \wedge (A \to B) \wedge (C \to D) \wedge (C \wedge D \to 0)$

`UPDATE`: alle As auch links markieren
$(\underline{1} \to \underline{A}) \wedge (\underline{\mathbf{A}} \wedge B \to C) \wedge (\underline{\mathbf{A}} \to B) \wedge (C \to D) \wedge (C \wedge D \to 0)$

`MARK`: markiere Variable B, da links als einziges A schon markiert
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge B \to C) \wedge (\underline{A} \to \underline{\mathbf{B}}) \wedge (C \to D) \wedge (C \wedge D \to 0)$

`UPDATE`: alle Bs auch links markieren
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge \underline{\mathbf{B}} \to C) \wedge (\underline{A} \to \underline{B}) \wedge (C \to D) \wedge (C \wedge D \to 0)$

`MARK`: markiere Variable C, da A und B schon markiert
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge \underline{B} \to \underline{\mathbf{C}}) \wedge (\underline{A} \to \underline{B}) \wedge (C \to D) \wedge (C \wedge D \to 0)$

`UPDATE`: alle Cs auch links markieren
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge \underline{B} \to \underline{C}) \wedge (\underline{A} \to \underline{B}) \wedge (\underline{\mathbf{C}} \to D) \wedge (\underline{\mathbf{C}} \wedge D \to 0)$

`MARK`: markiere Variable D, da C schon markiert
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge \underline{B} \to \underline{C}) \wedge (\underline{A} \to \underline{B}) \wedge (\underline{C} \to \underline{\mathbf{D}}) \wedge (\underline{C} \wedge D \to 0)$

`UPDATE`: alle Ds auch links markieren
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge \underline{B} \to \underline{C}) \wedge (\underline{A} \to \underline{B}) \wedge (\underline{C} \to \underline{D}) \wedge (\underline{C} \wedge \underline{\mathbf{D}} \to 0)$

`MARK`: markiere Konstante 0, da D schon markiert
$(\underline{1} \to \underline{A}) \wedge (\underline{A} \wedge \underline{B} \to \underline{C}) \wedge (\underline{A} \to \underline{B}) \wedge (\underline{C} \to \underline{D}) \wedge (\underline{C} \wedge \underline{D} \to \underline{\mathbf{0}})$

`ABORT`: Konstante 0 wurde markiert, Verfahren anhalten, Formel unerfüllbar

:Beispiel

3 Prädikatenlogik – *PL*

Umgangssprachlich wird nicht strikt zwischen Objekt und Prädikat getrennt – anders aber in der Mathematik. So kann man die Aussage

$$\textit{Jeder Schwan ist weiß.} \quad \Leftrightarrow \quad (\forall x)P(x) = \mathbf{1}$$

in zweierlei Arten auffassen/interpretieren:

	Objekt x	Prädikat P
1.	Schwan	istWeiß
2.	Weiß	istSchwan

Quanität der Urteile

Die generellen Urteile werden in der traditionellen formalen Logik, die auf Aristoteles zurückgeht, eingeteilt in universelle (allgemeine) und partikuläre (besondere). Zusammen mit den singulären (einzelnen) Urteilen bilden sie eine Trias in der Rubrik der Quantität der Urteile. Übliche Beispiele sind (1) für ein allgemeines Urteil: "Alle Griechen sind Menschen", (2) für ein besonderes Urteil: "Einige Griechen sind Philosophen", (3) für ein einzelnes Urteil: "Sokrates ist ein Mensch".

In der modernen Prädikatenlogik, die auf Frage zurückgeht und die in den Logik-Grundkursen gelehrt wird, erhalten diese grammatisch parallel gebauten Sätze freilich jeweils eine ganz verschiedene Analyse. Die grammatische Oberflächenstruktur verbirgt insofern die logische Tiefenstruktur. Die Analysen lauten:

1. "Alle Griechen sind Menschen" $\Rightarrow$
   "Für alle x gilt: Wenn x ein Grieche ist, so ist x ein Mensch."
   $\forall x(G(x) \rightarrow M(x))$

2. "Einige Griechen sind Philosophen" $\Rightarrow$
   "Es gibt mindestens ein x, für das gilt: x ist ein Grieche und x ist Philosoph."
   $\exists x(G(x) \wedge P(x))$

3. "Sokrates ist ein Mensch" $\Rightarrow$
   "Eine konkretes Objekt mit dem Namen Sokrates ist ein Mensch."
   $M(s)$, wobei s hier nun eine Konstante repräsentiert

Begriffe:

gebundene Variable: die Variable ist "untergeordneter Knecht" eines Quantors, geschrieben $x \sqsubset y$ (y ist im Bindungsbereich von x, zb: $\forall x \exists y P(x, y)$)
freie Variable: die Variable wird durch niemanden gebunden und das Ergebnis ist demnach von dieser Variable unabhängig
geschlossene Formel: wenn in der Formel $\nexists$ freie Variable

Variablensubstitution: zb $F(x/t)$ ersetzt in Formel F alle freien Vorkommen von $x \in F$ durch den Term t

:Begriffe

Beispiel E: gebundene/freie Variablen in PL

Es sei die Formel $\forall x(P(x) \wedge Q(x,y)) \vee \exists y(R(x,y))$ gegeben.

x ist links durch $\forall x$ gebunden, da x im Allquantor steht und auch noch in der Formel in $P(x)$ und auch in $Q(x,y)$ benutzt wird. y ist links jedoch nicht gebunden und ist damit frei im ersten Teil der Formel.

Im zweiten Teil der Formel ist y durch $\exists y$ gebunden und wird auch in $R(x,y)$ benutzt. x hingegen ist im zweiten Teil frei.

:Beispiel

3.1 Modelle

Ein Modell ist eine Art abstrakter Datentyp, welcher definiert ist durch das Quadrupel $\mathcal{D} = \langle D, FS, PS, KS \rangle$, wobei:

D	...	**D**omäne, Gegenstandsbereich; zb: $\mathbb{N}$, $\mathbb{Z}$
FS	...	**F**unktions**S**ymbole; zb: $f(x), g(y)$
PS	...	**P**rädikaten**S**ymbole; zb: $P(x), Q(y)$
KS	...	**K**onstanten**S**ymbole; zb: c, d

Dadurch ist die saubere Trennung von Syntax (Struktur) und Semantik (Interpretation) gewährleistet – siehe weiters dazu “Signaturen” und “Signaturinterpretation”.

Weiters ist noch die Abkürzung IVS üblich, stehend für **I**ndividuen**V**ariablen**S**ymbol.

Beispiel F: Eine Modellstruktur für die ganzen Zahlen $\mathbb{Z}$

$\mathbb{Z} = \langle Z, \{+, -, *\}, \{<, =\}, Z \rangle$

wobei $Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

:Beispiel

3.2 Verfahren für einstellige Prädikate

Ein gutes Verfahren für einstellige Prädikate ist das monadische Verfahren (es funktioniert zwar noch für zweistellige, aber alles darüber hinaus ist nicht mehr händisch rechenbar, da die Anzahl der Belegungen 2^n ist, wenn n die Stelligkeit bezeichnet). Dabei werden die Quantoren auf folgende Weise eliminiert, wobei $c, d \in KS$:

$$\begin{array}{lcl}(\forall x)P(x) & \rightarrow & P(c) \wedge P(d) \\ (\exists x)P(x) & \rightarrow & P(c) \vee P(d)\end{array}$$

Beziehungsweise für zweistellige Prädikate:

$$\begin{array}{lcl}(\forall x)P(x,y) & \rightarrow & P(c,y) \wedge P(d,y) \\ (\exists x)P(x,y) & \rightarrow & P(c,y) \vee P(d,y)\end{array}$$

Beispiel G: Elimination der Quantoren mittels monadischem Verfahren

$$(\exists x)(\forall y)\Big(P(x) \rightarrow P(y)\Big) =$$
$$(\exists x)\Big((P(x) \rightarrow P(c)) \wedge (P(x) \rightarrow P(d))\Big) =$$
$$\Big((P(c) \rightarrow P(c)) \wedge (P(c) \rightarrow P(d))\Big) \vee \Big((P(d) \rightarrow P(c)) \wedge (P(d) \rightarrow P(d))\Big)$$

:Beispiel

3.3 Unifikation

Begriffe:

Substitution: eine Abbildung $\rho : IVS \rightarrow t \in$ Terme dieser Signatur
Unifikator: wenn $\rho(A) = \rho(B)$, dann nennt man ρ Unifikator von A und B
Allgemeinheit von Unifkatoren: A ist allgemeiner als B, wenn B eine Instanz von A ist, also falls $\rho(A) \equiv B$

:Begriffe

Beispiel H: Anwenden einer Substitutionen auf eine Formel

Sei die Formel $F = P(x,y) \vee R(f(y),y)$ und die Substitution $\theta = \{x/g(a), y/z\}$, somit ergibt $\theta(F) = P(g(a),z) \vee R(f(z),z)$.

:Beispiel

Die Unifikation ist ein "string-mäßiges Parsen" zwei gegebener Terme s und t. Man stellt sich nun die Frage, ob irgendeine Substitution ρ existiert, so dass $\rho(s) \equiv \rho(t)$. Falls eine solche gefunden wurde, dann wird ρ "Unifikator" genannt.

Beim Untersuchen von allen auftretenden unterschiedlichen Stellen stoßt man dabei auf einen von drei Fällen:

1. Fall: gegeben ein Term und eine Variable, wobei Variable $\notin$ Term:
   $\Rightarrow$ ersetzen aller Vorkommen der Variable durch den Term
2. Fall: gegeben ein Term und eine Variable, wobei Variabel $\in$ Term:
   $\Rightarrow \nexists$ Substition ρ, zb: ist $x \leftarrow f(x)$ nicht möglich
3. Fall: gegeben zwei Funktionen:
   $\Rightarrow \nexists$ Substition ρ, zb: ist $f(x) \leftarrow g(y)$ nicht möglich

Beispiel I: Unifikation Intro

Es seien zwei Funktionen g und g' gegeben. Man finde den MGU dieser Funktionen.

$$\begin{array}{ccll}
 & & g(x,h(x),y,h(y)) & g'(z,u,r(u),v) \\
\{x/z\} & \longmapsto & g(x,h(x),y,h(y)) & g'(\mathbf{x},u,r(u),v) \\
\{h(x)/u\} & \longmapsto & g(x,h(x),y,h(y)) & g'(x,\mathbf{h}(\mathbf{x}),r(\mathbf{h}(\mathbf{x})),v) \\
\{y/r(h(x))\} & \longmapsto & g(x,h(x),y,h(y)) & g'(x,h(x),\mathbf{y},v) \\
\{h(y)/v\} & \longmapsto & g(x,h(x),y,h(y)) & g'(x,h(x),y,\mathbf{h}(\mathbf{y}))
\end{array}$$

Somit ist $\rho = \{z \leftarrow x, u \leftarrow h(x), r(h(x)) \leftarrow y, v \leftarrow h(y)\}$.
Bzw $\rho = \{\{z/x\}, \{u/h(x)\}, \{r(h(x))/y\}, \{v/h(y)\}\}$
angeschrieben als Variablensubstitutionen.
Das Term-Gleichungssystem dazu: $\varepsilon = \{x \doteq z, h(x) \doteq u, y \doteq r(u), h(y) \doteq v\}$

:Beispiel

Inferenzregelsystem

Diese Regeln dienen zur Ableitung des Term-Gleichungssystem ε und dem Beweisen der Lösbarkeit/Unlösbarkeit. Wenn ε zu $\perp$ transformiert wird, ist ε unlösbar, ansonsten wurde die allgemeinste Lösung von ε (der MGU) gefunden.

`DELETE`	$\dfrac{\{s \doteq s\} \dot\cup \varepsilon}{\varepsilon}$	
`DECOMPOSE`	$\dfrac{\{f(s_1,\ldots,s_n) \doteq f(t_1,\ldots,t_n)\} \dot\cup \varepsilon}{\{s_1 \doteq t_1,\ldots,s_n \doteq t_n \cup \varepsilon\}}$	
`ORIENT`	$\dfrac{\{t \doteq v\} \dot\cup \varepsilon}{\{v \doteq t\} \cup \varepsilon}$	
`ELIMINATE`	$\dfrac{\{v \doteq t\} \dot\cup \varepsilon}{\{v \doteq t\} \cup (\{v \leftarrow t\}(\varepsilon))}$	$v \in IVS$, $v \in V(\varepsilon)$, $v \notin V(t)$
`CLASH`	$\dfrac{\{f(s_1,\ldots,s_m) \doteq g(t_1,\ldots,t_n)\} \dot\cup \varepsilon}{\perp}$	$f, g \in FS \cup KS$ verschieden
`OCCURS CHECK`	$\dfrac{\{v \doteq t\} \dot\cup \varepsilon}{\perp}$	$v \in IVS$, $v \neq t$, $v \in V(t)$

Abbildung 4: Inferenzregelsystem für die Unifikation

Beispiel J: Unifikation mit Hilfe des Inferenzregelsystems

Finden Sie den MGU von $A = P(g(a), x, g(x))$ und $B = P(u, f(u, v), w)$.

Term-Gleichungssystem ist $\varepsilon = \{g(a) \doteq u, x \doteq f(u, v), g(x) \doteq w\}$.

`ORIENT:`	$\{u \doteq g(a), x \doteq f(u, v), g(x) \doteq w\}$
`ELIMINATE:`	$\{u \doteq g(a), x \doteq f(g(a), v), g(x) \doteq w\}$
`ELIMINATE:`	$\{u \doteq g(a), x \doteq f(g(a), v), g(f(g(a), v)) \doteq w\}$
`ORIENT:`	$\{u \doteq g(a), x \doteq f(g(a), v), w \doteq g(f(g(a), v))\}$

Somit ist $\rho = mgu\{A, B\} = \{u/g(a), x/f(g(a), v), w/g(f(g(a), v))\}$.

:Beispiel

4 Kalküle

4.1 Sequentialkalkül

Begriffe:

Sequent: ein Ausdruck/Ableitungsversuch der Form $\mathbf{\Pi} \perp \mathbf{\Gamma}$
Multiset: eine Menge, in der Elemente mehrfach vorkommen können
Axiom: Blätter des Ableitungsbaumes mit disjunkten Formelmengen
Prämisse: die durch Regelanwendung entstandenen Kindknoten
Konklusion: die zur Prämisse gehörende Elternknoten

:Begriffe

Der Sequentialkalül – oder auch LK (Logik Kalkül Klassik) genannt – leitet neue Formeln durch das Anwenden von bestimmten Regeln ab, bis alle Äste in einem Axiom enden – oder auch nicht. Ein Axiom wird genauer definiert durch:

$$\mathbf{\Pi} \perp \mathbf{\Gamma} \text{ ist Axiom, wenn } \mathbf{\Pi} \cap \mathbf{\Gamma} \neq \{\}$$

Also bei denen ein Term links **und** rechts vom Zeichen $\perp$ steht und demnach $\mathbf{\Pi}$ und $\mathbf{\Gamma}$ nicht disjunkte Mengen sind. Zum Beispiel ist $\mathbf{\Pi}, A \perp \mathbf{\Gamma}, A$ ein Axiom, da A auf beiden Seiten vorhanden ist.

$\{\} \perp \{G\} \ldots G$ ist gültig (Tautologie)
$\{F\} \perp \{\} \ldots F$ ist unerfüllbar (Kontradiktion)

Somit kann die Bewertung einer Formel *direkt* bewiesen werden.

4.1.1 Interpretation von einem Sequenten

Der Sequent $\mathbf{\Pi} \Rightarrow \mathbf{\Gamma}$ kann aufgefasst werden als $\bigwedge \mathbf{\Pi} \rightarrow \bigvee \mathbf{\Gamma}$. Schreibt man nun die einzelnen Formeln der Multisets aus, ergibt das $A_1, \ldots, A_m \Rightarrow B_1, \ldots, B_n$ bzw:

$$\bigwedge_{j=1}^{m} A_j \rightarrow \bigvee_{i=1}^{n} B_i$$

4.1.2 Ableitungsregeln

Folgende Regeln sind nichts anderes als eine Kodierung der Wahrheitstafel und dienen zum Beweisen (ableiten aller Äste in Axiome) von Formeln mit Hilfe des Sequential-Kalküls.

Die Regeln W und C stehen für *Weakening* (neue Formel hinzufügen) bzw *Constructing* (zwei gleiche Formeln auf eine reduzieren).

$$\frac{\mathbf{\Pi} \perp A, \mathbf{\Gamma} \qquad \mathbf{\Pi}, A \perp \mathbf{\Gamma}}{\mathbf{\Pi} \perp \mathbf{\Gamma}} \text{Schnittregel}$$

$$\frac{\mathbf{\Pi} \perp \mathbf{\Gamma}, A}{\mathbf{\Pi}, \neg A \perp \mathbf{\Gamma}} \neg \mathrm{L} \qquad \frac{\mathbf{\Pi}, A \perp \mathbf{\Gamma}}{\mathbf{\Pi} \perp \mathbf{\Gamma}, \neg A} \neg \mathrm{R}$$

$$\frac{\mathbf{\Pi}, A \perp \mathbf{\Gamma} \qquad \mathbf{\Pi}, B \perp \mathbf{\Gamma}}{\mathbf{\Pi}, A \vee B \perp \mathbf{\Gamma}} \vee \mathrm{L} \qquad \frac{\mathbf{\Pi} \perp \mathbf{\Gamma}, A, B}{\mathbf{\Pi} \perp \mathbf{\Gamma}, A \vee B} \vee \mathrm{R}$$

$$\frac{A, B, \mathbf{\Pi} \perp \mathbf{\Gamma}}{A \wedge B, \mathbf{\Pi} \perp \mathbf{\Gamma}} \wedge \mathrm{L} \qquad \frac{\mathbf{\Pi} \perp \mathbf{\Gamma}, A \qquad \mathbf{\Pi} \perp \mathbf{\Gamma}, B}{\mathbf{\Pi} \perp \mathbf{\Gamma}, A \wedge B} \wedge \mathrm{R}$$

$$\frac{\mathbf{\Pi} \perp \mathbf{\Gamma}, A \qquad \mathbf{\Pi}, B \perp \mathbf{\Gamma}}{\mathbf{\Pi}, A \rightarrow B \perp \mathbf{\Gamma}} \rightarrow \mathrm{L} \qquad \frac{\mathbf{\Pi}, A \perp \mathbf{\Gamma}, B}{\mathbf{\Pi} \perp \mathbf{\Gamma}, A \rightarrow B} \rightarrow \mathrm{R}$$

Abbildung 5: Regeln des Sequential-Kalküls$_{AL}$

$$\frac{\mathbf{\Gamma}, At \perp \mathbf{\Pi}}{\mathbf{\Gamma}, \forall x A x \perp \mathbf{\Pi}} \forall \mathrm{L} \qquad \frac{\mathbf{\Gamma} \perp Ay, \mathbf{\Pi}}{\mathbf{\Gamma} \perp \forall x A[x/y], \mathbf{\Pi}} \forall \mathrm{R}$$

$$\frac{\mathbf{\Gamma}, Ay \perp \mathbf{\Pi}}{\mathbf{\Gamma}, \exists x A x \perp \mathbf{\Pi}} \exists \mathrm{L} \qquad \frac{\mathbf{\Gamma} \perp At, \mathbf{\Pi}}{\mathbf{\Gamma} \perp \exists A[x/t], \mathbf{\Pi}} \exists \mathrm{R}$$

$$\frac{\mathbf{\Pi} \perp \mathbf{\Gamma}}{\mathbf{\Pi}, A \perp \mathbf{\Gamma}} W\mathrm{L} \qquad \frac{\mathbf{\Pi} \perp \mathbf{\Gamma}}{\mathbf{\Pi} \perp A, \mathbf{\Gamma}} W\mathrm{R}$$

$$\frac{\mathbf{\Pi}, A, A \perp \mathbf{\Gamma}}{\mathbf{\Pi}, A \perp \mathbf{\Gamma}} C\mathrm{L} \qquad \frac{\mathbf{\Pi} \perp A, A, \mathbf{\Gamma}}{\mathbf{\Pi} \perp A, \mathbf{\Gamma}} C\mathrm{R}$$

Abbildung 6: sonstige Regeln des Sequential-Kalküls

4.2 Resolution

Begriffe:

Klausel: eine Menge von disjunktiv verknüpfte Teilformeln
leere Klausel: durch Anwendung der Resolutionsregeln ableitbare Klauselmenge {}
duale Literale: ein Literal das sowohl negiert als auch unnegiert auftaucht
Klauselform: äquivalent zur KNF: Konjunktionen von Disjunktionen
Subsumtion: das Prinzip der Vereinfachung, kleinere Klausel (nur eine) ersetzt größere Klausel (aber nur eine)
Resolvent: eine abgeleitete Klausel der Resolution

:Begriffe

Dieses Verfahren beweist indirekt die Gültigkeit einer Formel durch zwei Schritte:

1. Transformation von $\neg F$ (da indirekt) in **konjunktive Normalform** (KNF)
2. wiederholtes Anwenden der **Resolutionsregeln**

Ist die leere Klausel $\{\}$ am Schluß ableitbar, dann ist damit die Gültigkeit von F bewiesen. Das Ableiten geschieht durch "schneiden" von zwei duale Literale (x und $\neg x$).

Es ist erlaubt, dass

- nicht alle Klauseln benützt werden müssen um zur leeren Klausel $\{\}$ zu kommen.
- eine Klausel öfters benutzt wird.
- neu entstandene Klauseln zur Erzeugung weiterer wiederverwendet werden.

4.3 Resolution in AL

Beispiel K: Eine einfache Resolution in AL

Folgende Klauselmenge K sei gegeben: $\{p, q\}, \{\neg p\}, \{\neg q, r\}, \{\neg r\}$

1:	$\{p, q\}$	$\in K$
2:	$\{\neg p\}$	$\in K$
3:	$\{q\}$	Res(1, 2)
4:	$\{\neg q, r\}$	$\in K$
5:	$\{r\}$	Res(3, 4)
6:	$\{\neg r\}$	$\in K$
7:	$\{\}$	Res(5, 6)

:Beispiel

4.3.1 Unit-Resolution

Diese Resolution ist auf Horn-Klauseln anwendbar, da einer der beiden Klauseln/Resolventen eine Einer-Klausel sein **muss**.

4.4 Resolution in PL

Beispiel L: Transformation in Klauselform und Anwendung der Resoultion in PL

Gegeben sei folgende Formel:

$$F = (\exists x)\Big(P(x) \vee Q(x)\Big) \rightarrow \Big((\exists x)P(x) \vee (\exists x)Q(x)\Big)$$

Da das Resolutionsverfahren eine indirekte Beweismethode ist, muss F zuerst negiert werden ($\neg F$), dann kann diese Formel in die KNF gebracht werden um letztendlich die Resolutionsregeln anwenden zu können:

$$
\begin{aligned}
&\neg\Big[(\exists x)\Big(P(x) \vee Q(x)\Big) \rightarrow \Big((\exists x)P(x) \vee (\exists x)Q(x)\Big)\Big]\\
&\neg\Big[\neg(\exists x)\Big(P(x) \vee Q(x)\Big) \vee \Big((\exists x)P(x) \vee (\exists x)Q(x)\Big)\Big]\\
&\neg\neg(\exists x)\Big(P(x) \vee Q(x)\Big) \wedge \neg\Big((\exists x)P(x) \vee (\exists x)Q(x)\Big)\Big]\\
&(\exists x)\Big(P(x) \vee Q(x)\Big) \wedge \Big(\neg(\exists x)P(x) \wedge \neg(\exists x)Q(x)\Big)\Big]\\
&(\exists x)\Big(P(x) \vee Q(x)\Big) \wedge \Big((\forall \tilde{x})\neg P(\tilde{x}) \wedge (\forall \hat{x})\neg Q(\hat{x})\Big)\Big]\\
&\Big(P(c) \vee Q(c)\Big) \wedge \Big(\neg P(\tilde{x}) \wedge \neg Q(\hat{x})\Big)\Big]\\
&cl(F) = \Big\{\{P(c), Q(c)\}, \{\neg P(\tilde{x})\}, \{\neg Q(\tilde{x})\}\Big\}
\end{aligned}
$$

$$
\begin{array}{ccccc}
\{P(c), Q(c)\} & & \{\neg P(c)\} & & \\
 & \diagdown \quad \diagup & & & \\
 & \{Q(c)\} & & \{\neg Q(c)\} & \\
 & & \diagdown \quad \diagup & & \\
 & & \{\} & &
\end{array}
$$

:Beispiel

Schritt 1: Transformation in KNF

Um die Formel F in die KNF zu bringen, gibt es zweierlei Möglichkeiten, die sich jeweils nur im letzten/vierten Schritt unterscheiden: Pränex-Normalform und Skolem-Normalform (*Skolemisierung*).

4.4.1 Pränex-Normalform

Beispiel M: Transformation in die Pränex-Normalform

Gegeben sei die Formel $\forall x(A(x) \wedge \exists y B(y)) \rightarrow \forall x C(x)$.

`REPLACE:`	$\neg\forall x(A(x) \wedge \exists y B(y)) \vee \forall x C(x)$
`NEGATION:`	$\exists x \neg(A(x) \wedge \exists y B(y)) \vee \forall x C(x)$
`NEGATION:`	$\exists x(\neg A(x) \vee \neg\exists y B(y)) \vee \forall x C(x)$
`NEGATION:`	$\exists x(\neg A(x) \vee \forall y \neg B(y)) \vee \forall x C(x)$
`SYNTAX:`	$\exists x(\neg A(x) \vee \forall y \neg B(y)) \vee \forall x' C(x')$
`SHIFT:`	$\exists x \forall y \forall x'(\neg A(x) \vee \neg B(y)) \vee C(x')$

Durch die PNF lasst sich nun schnell und einfach die DNF $\Big\{\{\neg A(x)\}, \{\neg B(y)\}, \{C(x')\}\Big\}$ und die KNF $\Big\{\{\neg A(x), \neg B(y), C(x')\}\Big\}$ zur weiteren Verwendung erstellen.

:Beispiel

1. `REPLACE`: $A \rightarrow B$ durch $\neg A \vee B$ (siehe syntaktische Methode Seite 8)
2. `NEGATION`: ziehen der Negationen nach innen vor die Atomformel, einerseits durch die Gesetze von De Morgan (Seite 6) und andererseits durch die Vertauschung der Quantoren:

$$\neg\forall x P(x) \Rightarrow \exists x \neg P(x) \quad \neg\exists x \Rightarrow \forall x \neg P(x)$$

3. `SYNTAX`: Umbenennung von gebundenen Variablen, sodass diese durch genau einen Quantor gebunden werden (und verschieden von allen freien Variablen sind)
4. `SHIFT`: nach vorne ziehen aller Quantoren

Abbildung 7: Vorgangsweise zur Erstellung der PNF (Pränex-Normalform)

4.4.2 Skolem-Normalform

1. `REPLACE`: siehe Pränex-Normalform
2. `NEGATION`: siehe Pränex-Normalform
3. `SYNTAX`: siehe Pränex-Normalform
4. `SKOLEM`: ("*Skolemisierung*") Existenzquantoren durch Einführen neuer Funktions- und/oder Konstanten-Symbole eliminieren, Allquantoren fallen weg

Abbildung 8: Vorgangsweise zur Erstellung der SNF (Skolem-Normalform)

Die **Skolemisierung**[6] führt für jede gebundene Variable mit einem Existenzquantor eine neue Funktion $(f, g, h, \ldots)$ ein, deren Parameter die Variablen der bindenden Allquantoren ist. Zum Beispiel wird

$$\forall x \exists y (P(x, y)) \qquad \text{zu} \qquad R(x, f(x)).$$

Oder eine etwas längere Variante:

$$\forall u \forall v \exists x \Big(P(u, v, x) \vee \exists y Q(x, u, y) \Big) \wedge \exists z R(z, x) \qquad \text{wird zu}$$
$$\Big(P(u, v, f(u, v)) \vee Q(f(u, v), u, g(u, v)) \Big) \wedge P(c, x).$$

Die Einführung einer neuen Konstante geschieht aus dem Grund, da sie der eigentlich einzuführenden nullstelligen Funktion (keine Parameter) gleich kommt: $f_0 = c$.

[6] benannt nach *Albert Thoralf Skolem*, norwegischer Mathematiker, Logiker und Philosoph im 20. Jhdt

Beispiel N: Transformation in die Skolem-Normalform

Gegeben sei die Formel $\forall x(A(x) \land \exists y B(y)) \rightarrow \forall x C(x)$.

`REPLACE:`	$\neg\forall x(A(x) \land \exists y B(y)) \lor \forall x C(x)$
`NEGATION:`	$\exists x(\neg A(x) \lor \neg\exists y B(y)) \lor \forall x C(x)$
`NEGATION:`	$\exists x(\neg A(x) \lor \forall y \neg B(y)) \lor \forall x C(x)$
`SYNTAX:`	$\exists x(\neg A(x) \lor \forall y \neg B(y)) \lor \forall x' C(x')$
`SKOLEM:`	$(\neg A(c) \lor \forall y \neg B(y)) \lor \forall x' C(x')$
`SKOLEM:`	$(\neg A(c) \lor \neg B(y)) \lor C(x')$

Die Klauselmenge ist – ähnlich dem vorherigen Bsp – $\Big\{\{\neg A(c), \neg B(y), C(x')\}\Big\}$.

:Beispiel

Schritt 2: Anwenden der Resolutionsregeln

Das Anwenden der Resolutionsregeln erfolgt analog zur AL, nur zusätzlich in Kombination mit Substitutionen.

4.4.3 Faktorisierung

Bei dem Versuch $A = \{P(x), P(y)\}$ und $B = \{\neg P(u), \neg P(v)\}$ zu unifizieren wird man scheitern, da $\nexists \rho$, so dass $\rho(A) \equiv \rho(B)$. **Lösung** des Problems ist eine *Unifikation innerhalb von Klauseln.*

Es seien die Unifikatoren $\rho_1 = \{x/y,\}$ und $\rho_2 = \{u/v,\}$ gegeben. Die daraus resultierenden Klauseln bezeichnet man als "**Faktoren**". Die leere Klausel ist nun durch Faktorisierung und Anwenden der Resolution AL ableitbar, wobei $\theta = mgu\{P(y), P(v)\} = \{y/v\}$ ist:

$$\dfrac{\dfrac{\dfrac{P(x), P(y)}{P(y)}\,\rho_1}{P(v)}\,\theta \qquad\qquad \dfrac{\dfrac{\neg P(u), \neg P(v)}{\neg P(v)}\,\rho_2}{\neg P(v)}\,\theta}{\{\}}$$

5 Rekursionstheorie

5.1 Grundlagen

Begriffe:

Algorithmus: ein Verfahren zur Lösung eines Problems (Funktionswertberechnung) für gegebene Argumente
Explikat: Beschreibung eines Algorithmus, die untereinander äquivalent sind
numerische Funktion: diese Funktionen haben einen Definitionsbereich $D = \mathbb{N} \times \ldots \times \mathbb{N}$ und einen Wertebereich von $W = \mathbb{N}$
rekursiv aufzählbar: eine Menge $\mathcal{M} \subseteq W$ ist rekursiv aufzählbar, wenn es eine berechenbare Funktion $f : D \rightarrow W$ gibt, so dass $\mathcal{M} = f(D)$, bzw der Wertebereich von f stimmt mit $\mathcal{M}$ überein, so dass $\mathcal{M} = \{f(x) \mid x \in D\}$
rekursiv entscheidbar: (oder nur "rekursiv") eine Menge $\mathcal{M} \subseteq W$ ist rekursiv entscheidbar, wenn $\mathcal{M}$ und $W - \mathcal{M}$ rekursiv aufzählbar sind

:Begriffe

5.1.1 Abzählbarkeit

Eine unendliche Menge $\mathcal{M}$ ist **effektiv abzählbar** mittels der bijektiven Funktion $f : \mathcal{M} \rightarrow \mathbb{N}$, wenn es für alle Eingaben aus $\mathbb{N}$ einen abbrechenden Algorithmus gibt, mit dessen Hilfe man für jedes $n \in \mathbb{N}$ den Funktionswert der inversen Funktion von f an der Stelle n berechnen kann, nämlich $f^{-1}(n) \in \mathcal{M}$.

Falls $\mathcal{M}$ endlich ist, und $|\mathcal{M}| = n$, dann ist natürlich $\mathcal{M}$ durch $0, 1, \ldots, n-1$ zu ersetzen.

Man kann also von einem n-ten Element $f^{-1}(n)$ von $\mathcal{M}$ sprechen. Wenn man $f^{-1}(n)$ mit x_n bezeichnet, so ist folgendes eine effektive Abzählung von $\mathcal{M}$:

$$x_0, x_1, \ldots, x_n, \ldots$$

f codiert also $\mathcal{M}$, und f^{-1} decodiert $\mathcal{M}$.

5.1.2 Die totale Funktion f

Ab nun seien die zwei effektiv abzählbaren Mengen D (Definitionsbereich) und W (Wertebereich) gegeben – D ist meistens ein Kreuzprodukt von einem Datentyp, was einer mehrstelligen Funktion gleich kommt.

Sei die totale Funktion f gegeben durch:

$$f : D \rightarrow W$$

- f ist *berechenbar* ("rekursiv")

  ... wenn es einen abbrechenden Algorithmus gibt, mit dem man für jedes Argument $x \in D$ den entsprechenden Funktionswert $f(x) \in D$ effektiv berechnen kann.

- f ist *partiell berechenbar* ("partiell rekursiv")

  ...wenn ein $x \in E$ existiert, für das $f(x)$ nicht definiert ist (zb: 1/0) und der Algorithmus bricht nicht ab.

Satz: Jeder Algorithmus definiert eine partiell berechenbare Funktion, die durch ihn berechnet wird.

5.1.3 Die numerische Funktion g

Im wesentlichen kann man sich auf die numerischen Funktion $g : \mathbb{N} \to \mathbb{N}$ beschränken. Sei nämlich $g : D \to W$ eine (partiell) berechenbare Funktion, so existiert die bijektive Funktion $f_1 : D \to \mathbb{N}$ und $f_2 : W \to \mathbb{N}$, so dass D mittels f_1 und W mittels f_2 effektiv abzählbar sind.

Sei $g' : \mathbb{N} \to \mathbb{N}$ die folgendermaßen definierte Funktion:

$$g'(n) = f_2(g(f_1^{-1}(n))) \qquad n \in \mathbb{N}$$

Natürlich ist g' (partiell) berechenbar, und es gilt $g(x) = f_2^{-1}(g'(f_1(x)))$, $x \in D$. Denn aus der Definition von g' folgt, dass $f_2^{-1}(g'(n)) = g(f_1^{-1}(n))$. Setzt man nun $x = f_1^{-1}(n)$, so gilt $n = f_1(x)$ und $f_2^{-1}(g'(f_1(x))) = g(x)$.

5.2 Funktionen, Funktionen

Sei $\mathcal{M} \subseteq W$, dann definiere die *charakteristische Funktion* $\chi_\mathcal{M} : W \to \{0, 1\}$ von $\mathcal{M}$ (bezüglich W) durch:

$$\chi_\mathcal{M}(x) = \begin{cases} 1, & \text{falls } x \in \mathcal{M}, \\ 0, & \text{falls } x \notin \mathcal{M} (\text{also} x \in D - \mathcal{M}). \end{cases}$$

$\mathcal{M} \subseteq W$ ist entscheidbar, genau dann, wenn $\chi_\mathcal{M}$ berechenbar ist (den Beweis ersparen wir uns).

5.2.1 μ-rekursive Funktionen

Wir wollen nun ein wichtiges Explikat des Begriffs der partiell berechenbaren Funktionen definieren, nämlich die *μ-rekursive Funktionen.* Diese Funktionen bilden das n-fache kartesische Produkt von Σ^*, wobei Σ ein Alphabet ist, und der Stern $*$ eine Menge von diesem Alphabet angibt (inklusive dem Leerwort ε).

Oder formal ausgedrückt: $D = (\Sigma^*)^n, n \geq 0$.

5.2.2 Grundfunktionen

1. **Nachfolgerfunktion** N_a (für jedes $a \in \Sigma$): $\Sigma^* \to \Sigma^*$

   $N_a(w) = wa, w \in \Sigma^*$

2. **Nullfunktion** C^n (für jedes $n \geq 0$): $(\Sigma^*)^n \to \Sigma^*$

   $C^n(w_1, \ldots, w_n) = \varepsilon, w_i \in \Sigma^*, 1 \leq i \leq n$

3. **Projektion** U_j^n (für jedes $n \geq 1, 1 \leq j \leq n$): $(\Sigma^*)^n \to \Sigma^*$

   $U_j^n(w_1, \dots, w_n) = w_j, w_i \in \Sigma^*, 1 \leq i \leq n$

5.2.3 Funktionale

Diese sind notwendig, um aus bereits definierten Funktionen, neu Funktionen zu gewinnen.

1. **Einsetzung**:

   Folgendes sei gegeben: $n \geq 0, \quad m \geq 1, \quad 1 \leq j \leq m$

   $$\begin{gathered} h : (\Sigma^*)^m \to \Sigma^* \\ g_j : (\Sigma^*)^n \to \Sigma^* \end{gathered}$$

   Die Funktion $f : (\Sigma^*)^n \to \Sigma^*$ entsteht aus h durch Einsetzung von $g_1, \dots, g_m$, gdw: für alle $w_i \in \Sigma^*, 1 \leq i \leq n$ gilt:

   $$f(w_1, \dots, w_n) = h(g_1(w_1, \dots, w_n), \dots, g_m(w_1, \dots, w_n))$$

2. **Rekursion**:

   Folgendes sei gegeben: $n \geq 1, a \in \Sigma$

   $$\begin{gathered} g : (\Sigma^*)^{n-1} \to \Sigma^* \\ h_a : (\Sigma^*)^{n+1} \to \Sigma^* \end{gathered}$$

   Die Funktion $f : (\Sigma^*)^n \to \Sigma^*$ entsteht aus g und h_a durch Rekursion, gdw: $\forall w_i \in \Sigma^*, 1 \leq i \leq n$ gilt:

   $$\begin{gathered} f(\varepsilon, w_2, \dots, w_n) = g(w_2, \dots, w_n) \\ f(w_1 a, \dots, w_n) = h_a(f(w_1, \dots, w_n), w_1, \dots, w_n) \end{gathered}$$

3. **Minimisierung**:

   Folgendes sei gegeben: $n \geq 0$

   $$g : (\Sigma^*)^{n+1} \to \Sigma^*$$

   Die Funktion $f : (\Sigma^*)^n \to \Sigma^*$ entsteht aus g durch Minimisierung bezüglich $a \in \Sigma^*$, gdw: $\forall w_i \in \Sigma^*, 1 \leq i \leq n$ gilt:

   $$f(w_1, \dots, w_n) = \mu_a[g(w_0, w_1, \dots, w_n)]$$

   Dabei ist $\mu_a[g(w_0, w_1, \dots, w_n)] = a^m$ wenn (andernfalls undefiniert):
   Es existiert ein $m \geq 0$, so dass $g(a^l, w_1, \dots, w_n), 0 \leq l \leq m$ definiert ist mit $g(a^l, w_1, \dots, w_n) \neq \varepsilon, 0 \leq l < m$ und $g(a^m, w_1, \dots, w_n) = \varepsilon$.

5.2.4 Funktionsfamilien

Mit Hilfe der Grundfunktionen und der Funktionale, ist es uns nun möglich primitiv rekursive und μ-rekursive Funktionen zu definieren.

Familie der primitiv rekursiven (bzw μ-rekursiven) Funktionen über Σ

Dies ist die kleinste Familie von Funktionen $f : (\Sigma^*)^n \to \Sigma^*, n \geq 0$, die ...

- $N_a, a \in \Sigma, C^0, U_1^n, n \geq 1$ enthält
- und abgeschlossen gegenüber Einsetzung/Rekursion(/Minimisierung) ist.

Familie der numerisch primitiv rekursiven (bzw μ-rekursiven) Funktionen

Dies ist die kleinste Familie von Funktionen $f : \mathbb{N}^n \to \mathbb{N}, n \geq 0$, die ...

- $N(= N_a), C^0, U_1^n, n \geq 1$ enthält
- und abgeschlossen gegenüber Einsetzung/Rekursion(/Minimisierung) ist.

Aus obiger Definition gewinnt man numerische Funktionen von $\mathbb{N}^n$ in $\mathbb{N}$, indem ...

- Σ zu einem einelementigen Alphabet wird: $\Sigma = \{a\}$, und a^m die Zahl m identifiziert
- das Leerwort ε der Zahl 0 entspricht.

Satz: Eine μ-rekursive Funktion ist iA eine partielle Funktion, während primitiv rekursive Funktionen immer total μ-rekursiv sind.

5.2.5 μ-rekursive Programme

μ-rekursive Programme sind eine kompakte Beschreibung von μ-rekursiven Funktionen über Σ. Im Folgenden werden μ-rekursive Funktionen mit Großbuchstaben angeschrieben, und das μ-rekursive Programm, das dieses berechnet, mit den entsprechenden Kleinbuchstaben.

1. Programm $\mathbf{n}_{a_i}$: einstellig, stellt die Nachfolgerfunktion $N_{a_i}, 1 \leq i \leq k$ dar
2. Programm $\mathbf{c}$: nullstellig, stellt die Nullfunktion C^0 dar
3. Programm $\mathbf{u}$: n-stellig, $n \geq 1$, stellt die Projektion U_1^n dar
4. Programm $\mathbf{h}(g_1, \ldots, g_m)$: n-stellig, es werden die μ-rekursive Funktionen H und G_j durch das m-stellige Programm $\mathbf{h}$ und das n-stellige Programm $\mathbf{g}_j$ dargestellt:

$$H : (\Sigma^*)^m \to \Sigma^*, m \geq 1$$
$$G_j : (\Sigma^*)^n \to \Sigma^*, 1 \leq j \leq m, n \geq 0$$

Die Funktion F (welche durch das Einsetzen von $G_1, \ldots, G_m$ in H entsteht) wird durch $\mathbf{h}(g_1, \ldots, g_m)$ dargestellt.

5. Programm $[gh_{a_j} \ldots h_{a_k}]$: n-stellig, es werden die μ-rekursive Funktionen G und H_{a_i} durch das $(n-1)$-stellige Programm $\mathbf{g}$ und das $(n+1)$-stellige Programm $\mathbf{h}_{a_i}$ dargestellt:

$$G : (\Sigma^*)^{n-1} \to \Sigma^*, n \geq 1$$
$$H_{a_i} : (\Sigma^*)^{n+1} \to \Sigma^*, 1 \leq i \leq k$$

Die Funktion F (welche durch Rekursion aus G und H_{a_i} entsteht) wird durch $[gh_{a_j} \ldots h_{a_k}]$ dargestellt.

6. Programm $[[\mathbf{g}]]_{a_i}$: n-stellig, es wird die μ-rekursive Funktion $G : (\Sigma^*)^{n+1} \to \Sigma^*, n \geq 1$ dargestellt durch das $(n+1)$-stellige Programm $\mathbf{g}$. Die Funktion F (welche durch Minimisierung bezüglich a_i entsteht) wird durch das n-stellige Programm $[[\mathbf{g}]]_{a_i}$ dargestellt.

Die Syntax der Sprache der μ-rekursiven Programme (ohne die Stelligkeit zu berücksichtigen) ist also durch folgende Produktionen gegeben:

$P \to \mathbf{n}_{a_1} \mid \ldots \mid \mathbf{n}_{a_k} \mid \mathbf{c} \mid \mathbf{u} \mid P(L) \mid [P^{k+1}] \mid [[P]]_{a_1} \mid \ldots \mid [[P]]_{a_k}$
$L \to PL \mid P$

Satz: Ein μ-rekursives Programm, in dem die Symbole [[und]] nicht vorkommen, heißt *primitiv rekursives Programm.*

5.3 Beispiel numerische Funktionen

Beispiel O: numerische Funktionen durch Programme darstellen

Es sei $x, y \in \mathbb{N}$:

1. **Nachfolgefunktion**:
   Sei $N_t : \mathbb{N} \to \mathbb{N}, t \geq 1$, die Funktion $N_t(x) = x + t$, also $N_1(x) = N(x)$ und $N_{t+1}(x) = N(N_t(x))$.
   $$n_1 = 1 \quad n_{t+1} = n(n_t) \quad t \geq 1$$
2. **Konstante**:
   Sei $K_i^n : \mathbb{N}^n \to \mathbb{N}$ die n-stellige Konstantenfunktion $K_i^n(x_1, \ldots, x_n) = i$.
   - **Nullstellige Konstante**:
     $$k_0^0 = c \quad k_i^0 = n_i(c)$$
   - **Einstellige Konstante**:

$$k_0^1 = [cu] \quad k_i^1 = n_i(k_0^1)$$

- **n-stellige Konstante**:

$$k_0 = k_0^1(u) \quad k_i = n_i(k_0)$$

3. **Summe**:
$SUM(0, y) = y = U_1^1(y)$
$SUM(x + 1, y)$
$= SUM(x, y) + 1$
$= N(SUM(x, y))$
$= N(U_1^3(SUM(x, y)), x, y)$
4. **Produkt**:
$PROD(0, y) = 0 = K_0^1(y)$
$PROD(x + 1, y)$
$= SUM(PROD(x, y), y)$
$= SUM(U_1^3(PROD(x, y), x, y), U_3^3(PROD(x, y), x, y))$
5. **Potenzieren**:
$EXP(0, y) = 1 = K_1^1(y)$
$EXP(x + 1, y)$
$= PROD(EXP(x, y), y)$
$= PROD(U_1^3(EXP(x, y), x, y), U_3^3(EXP(x, y), x, y))$
6. **Vorgänger**:
$V(0) = 0 = C^0$
$V(x + 1) = x = U_2^2(V(x), x)$
7. **Differenz**:
$DIFF1(0, y) = y = U_1^1(y)$
$DIFF1(x + 1, y)$
$= V(DIFF1(x, y))$
$= V(U_1^3(DIFF1(x, y), x, y))$
$DIFF(x, y)$
$= DIFF1(y, x)$
$= DIFF1(U_2^2(x, y), U_1^2(x, y))$
8. **Fakultät**:
$FAK(0) = 1 = K_1^0$
$FAK(x + 1)$
$= PROD(FAK(x), N(X))$
$= PROD(U_1^2(FAK(x), x), N(U_2^2(FAK(x), x))))$
9. **Signum**:
$SGN(0) = 0 = C^0$
$SGN(x + 1) = 1 = K_1^2(SGN(x), x)$

:Beispiel

6 Aufgaben

Dies ist die Ausarbeitung der offiziellen Aufgaben zur LVA Theoretische Informatik 2, zur Verfügung gestellt von Prof. Baaz unter `http://www.logic.at/people/terwijn/aufgaben.ps`.

Aufgabe 1: Formel$_{AL}$ in KNF/DNF umwandeln

Angabe: Geben Sie für folgende Formeln eine DNF und eine KNF an.

$$\begin{array}{lr} \text{(a) } F = & p \wedge q \rightarrow \big(r \leftrightarrow (p \vee q)\big) \\ \text{(b) } G = & p \vee \big((q \rightarrow r \vee r \rightarrow s) \wedge \neg p \rightarrow s\big) \\ \text{(c) } H = & p \vee \neg r \wedge (\neg p \vee s \leftrightarrow r) \end{array}$$

Hinweis: $\neg$ bindet stärker als $\wedge$, $\wedge$ bindet stärker als $\vee$, $\rightarrow$, $\leftrightarrow$.

Lösung: (a)

p	q	r	$p \wedge q$	$p \vee q$	$r \leftrightarrow (p \vee q)$	$p \wedge q \rightarrow \big(r \leftrightarrow (p \vee q)\big)$
0	0	0	0	0	**1**	**1**
0	0	**1**	0	0	0	**1**
0	**1**	0	0	**1**	0	**1**
0	**1**	**1**	0	**1**	**1**	**1**
1	0	0	0	**1**	0	**1**
1	0	**1**	0	**1**	**1**	**1**
1	**1**	0	**1**	**1**	0	0
1	**1**	**1**	**1**	**1**	**1**	**1**

Die Formel in **disjunktiver** Normalform als Menge angeschrieben:
$\Big\{\{\neg p, \neg q, \neg r\}, \{\neg p, \neg q, r\}, \{\neg p, q, \neg r\}, \{\neg p, q, r\}, \{p, \neg q, \neg r\}, \{p, \neg q, r\}, \{p, q, r\}\Big\}$
Selbige Formel in **konjunktiver** Normalform, oder formal KNF(F):
$\Big\{\{\neg q, \neg q, r\}\Big\}$
Anmerkung: Die $DNF(F)$ kann gekürzt werden auf $\neg p \vee \neg q \vee r$

Lösung: (b)

$G = p \vee ((q \rightarrow r \vee r \rightarrow s) \wedge \neg p \rightarrow s)$ ... Implikation auflösen
$G = p \vee ((\neg q \vee r \vee \neg r \vee s) \wedge p \vee s)$ ... Tautologie durch $r \vee \neg r$
$G = p \vee (\mathbf{1} \wedge p \vee s)$ ... da $p \wedge \mathbf{1} = p$ und $p \vee p = p$
$G = p \vee s$ ... die äquivalente, optimierte Formel G' zu G
Hinweis: Assoziativität/Kommutativität von $\vee$, $\wedge$ nutzen!

p	q	r	s	$A : (q \to r)$	$B : (r \to s)$	$C : (A \vee B)$	$D : (C \wedge \neg p)$	$(D \to s) \equiv G$
0	0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0
0	1	0	1	0	1	1	1	1
0	1	1	0	1	0	1	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0	1
1	0	0	1	1	1	1	0	1
1	0	1	0	1	0	1	0	1
1	0	1	1	1	1	1	0	1
1	1	0	0	0	1	1	0	1
1	1	0	1	0	1	1	0	1
1	1	1	0	1	0	1	0	1
1	1	1	1	1	1	1	0	1

$$\begin{aligned}
\mathrm{DNF}(G) = \; & \Big\{\{\neg p, \neg q, \neg r, s\}, \{\neg p, \neg q, r, s\}, \{\neg p, q, \neg r, s\}, \{\neg p, q, r, s\}, \\
& \{p, \neg q, \neg r, \neg s\}, \{p, \neg q, \neg r, s\}, \{p, \neg q, r, \neg s\}, \{p, \neg q, r, s\}, \\
& \{p, q, \neg r, \neg s\}, \{p, q, \neg r, s\}, \{p, q, r, \neg s\}, \{p, q, r, s\}\Big\} \\
\mathrm{KNF}(G) = \; & \Big\{\{p, q, r, s\}, \{p, q, \neg r, s\}, \{p, \neg q, r, s\}, \{p, \neg q, \neg r, s\}\Big\}
\end{aligned}$$

Lösung: (c)

p	r	s	$\neg p$	$A : (\neg p \vee s)$	$B : (A \leftrightarrow r)$	$\neg r$	$C : (\neg r \wedge B)$	$(p \vee C) \equiv H$
0	0	0	1	1	0	1	0	0
0	0	1	1	1	0	1	0	0
0	1	0	1	1	1	0	0	0
0	1	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1	1
1	0	1	0	1	0	1	0	1
1	1	0	0	0	0	0	0	1
1	1	1	0	1	1	0	0	1

$$\begin{aligned}
\mathrm{DNF}(H) = \; & \Big\{\{p, \neg r, \neg s\}, \{p, \neg r, s\}, \{p, r, \neg s\}, \{p, r, s\}\Big\} = \Big\{\{p\}\Big\} \\
\mathrm{KNF}(H) = \; & \Big\{\{p, r, s\}, \{p, r, \neg s\}, \{p, \neg r, s\}, \{p, \neg r, \neg s\}\Big\} = \Big\{\{p\}\Big\}
\end{aligned}$$

:Aufgabe

Aufgabe 2: Horner-Schema anwenden

Angabe: Beweisen oder widerlegen Sie, dass folgende Formeln erfüllbar sind.

(a) $(B \to E) \wedge (C \to D) \wedge (1 \to A) \wedge (1 \to B) \wedge (E \wedge A \to C) \wedge$
$(E \wedge A \wedge C \to F) \wedge (F \to G) \wedge (A \wedge G \to H) \wedge (I \to 0) \wedge (H \wedge C \to I)$

(b) $(E \to 0) \wedge (1 \to G) \wedge (H \to I) \wedge (I \to J) \wedge (H \wedge I \to J) \wedge$
$(G \to H) \wedge (H \wedge G \to A)$

(c) $(B \to E) \wedge (C \to D) \wedge (1 \to B) \wedge (E \wedge A \to C) \wedge (E \wedge A \wedge C \to F) \wedge$
$(F \to G) \wedge (A \wedge G \to H) \wedge (I \to 0) \wedge (H \wedge C \to I)$

Lösung: (a)

$(\underline{B}_3 \to \underline{E}_4) \wedge (\underline{C}_7 \to \underline{D}_8) \wedge (\underline{1}_1 \to \underline{A}_2) \wedge (\underline{1}_1 \to \underline{B}_2) \wedge (\underline{E}_5 \wedge \underline{A}_3 \to \underline{C}_6) \wedge (\underline{E}_5 \wedge \underline{A}_3 \wedge \underline{C}_7 \to \underline{F}_8) \wedge (\underline{F}_9 \to \underline{G}_{10}) \wedge (\underline{A}_3 \wedge \underline{G}_{11} \to \underline{H}_{12}) \wedge (\underline{I}_{15} \to \underline{0}_{16}) \wedge (\underline{H}_{13} \wedge \underline{C}_7 \to \underline{I}_{14})$

Der Index gibt an, in welchem Schritt der Ausdruck unterstrichen wurde, und da *falsch* (0) unterstrichen wurde, ist die Formel **unerfüllbar**.

Sei $\mathcal{TF}$ die die Menge von Teilformeln $\{A, B, C, \dots\}$, dann ist trivialerweise jede Interpretation der $\mathcal{TF}$ ein Gegenbeispiel, formal:
$\forall x \in \mathcal{TF} : I(x) = \mathbf{1} \vee 0$ (es $\nexists$ Modell, da die Hornerformel unerfüllbar ist)

Lösung: (b)

$(E \to 0) \wedge (\underline{1}_1 \to \underline{G}_2) \wedge (\underline{H}_5 \to \underline{I}_6) \wedge (\underline{I}_7 \to \underline{J}_8) \wedge (\underline{H}_5 \wedge \underline{I}_7 \to \underline{J}_8) \wedge (\underline{G}_3 \to \underline{H}_4) \wedge (\underline{H}_5 \wedge \underline{G}_3 \to \underline{A}_6)$

Da E der einzige Buchstabe ist, der 0 impliziert, und E nirgends impliziert wird (nicht auf der rechten Seite steht), ist die Formel **erfüllbar**.

Die Modelle wären $I(E) = 0, \forall x \in (\mathcal{TF}/\{E\}) : I(x) = 1$ und die Gegenbeispiele dazu $I(E) = \mathbf{1}, \forall x \in (\mathcal{TF}/\{E\}) : I(x) = \mathbf{1} \vee 0$.

Lösung: (c)

$(\underline{B}_3 \to \underline{E}_4) \wedge (C \to D) \wedge (\underline{1}_1 \to \underline{B}_2) \wedge (\underline{E}_5 \wedge A \to C) \wedge (\underline{E}_5 \wedge A \wedge C \to F) \wedge (F \to G) \wedge (A \wedge G \to H) \wedge (I \to 0) \wedge (H \wedge C \to I)$

Die Formel ist **erfüllbar**.

Ein Modell anhand des Horner-Schemas wäre
$I(B) = \mathbf{1}, I(E) = \mathbf{1}, \forall x \in (\mathcal{TF}/\{B, E\}) : I(x) = 0$ und das Gegenbeispiel dazu $I(I) = 1, \forall x \in (\mathcal{TF}/\{I\}) : I(x) = \mathbf{1} \vee 0$.

:Aufgabe

Aufgabe 3: erfüllende Variablenbelegungen

Angabe: Wieviele Belegungen gibt es, die Aufgabe 2 (b) erfüllen?

Lösung:

Es existiert eine Belegung, sodass die Hornerformel aus 2 (b) erfüllt ist, nämlich wie schon zuvor angegeben: $I(E) = 0$ und für alle anderen $\mathbf{1}$. Mit dieser Begründung, dass alle anderen Buchstaben durch die Beschaffenheit der Formel auch markiert werden würden, und durch die Interpretation einer Teilformel außer E würde dies zu einer Markierung von 0 führen und damit wäre die Formel nicht mehr erfüllbar.

Außerdem darf auf keinen Fall E als $\mathbf{1}$ interpretiert werden, da sonst sofort das Verfahren halten würde und die Formel als nicht erfüllbar ausgewertet werden würde.

:Aufgabe

Aufgabe 4: notwendige Konjunkte

Angabe: Welches Konjunkt kann bei Aufgabe 2 (a) weggelassen werden, ohne an der Unerfüllbarkeit etwas zu ändern?

Lösung:

$\Big(C \rightarrow D\Big)$, da … D auf keiner linken Seite vorkommt.
$H \wedge C$ benötigt wird um auf 0 zu kommen, H und C werden jedoch schon über einen anderen "Weg" erreicht.

:Aufgabe

Aufgabe 5: Hornformeln identifizieren

Angabe: Welche der folgenden Klauselmengen entsprechen Hornformeln?

(a) $\Big\{\{p, \neg q, \neg r\}, \{\neg p, \neg q, r\}, \{p\}, \{\neg q\}\Big\}$

(b) $\Big\{\{p, q, \neg r\}, \{\neg p, p\}, \{q\}\Big\}$

(c) $\Big\{\{\neg p, \neg q\}, \{\neg q, \neg r\}, \{\neg p, \neg r\}\Big\}$

Lösung:

Die Klauselmengen (a) und (c) entsprechen Hornformeln, da bei (b) eine Klause existiert, die mehr als ein einziges positives Literal besitzt, nämlich $\{p, q, \neg r\}$.

:Aufgabe

Aufgabe 6: Resolutionsverfahren AL

Angabe: Widerlegen Sie mit Hilfe des Resolutionsverfahren oder geben Sie eine Belegung der Variablen an, die die Klauselmenge wahr macht.

(a) $F = \Big\{\{p, q\}, \{p, \neg q\}, \{\neg q, s\}, \{\neg p, \neg s\}\Big\}$

(b) $G = \Big\{\{p, q\}, \{\neg p, r\}, \{\neg r, s\}, \{\neg s\}\Big\}$

(c) $H = \Big\{\{p, q, r\}, \{\neg p, q\}, \{\neg r, q\}, \{\neg q\}\Big\}$

Lösung: (a)

1:	$\{p, q\}$	$\in F$
2:	$\{p, \neg q\}$	$\in F$
3:	$\{p\}$	Res(1, 2)
4:	$\{\neg p, s\}$	$\in F$
5:	$\{s\}$	Res(3, 4)
6:	$\{\neg p, \neg s\}$	$\in F$
7:	$\{\neg s\}$	Res(3, 6)
8:	$\{\}$	Res(5, 7)

Lösung: (b)

1:	$\{\neg p, r\}$	$\in G$
2:	$\{\neg r, s\}$	$\in G$
3:	$\{\neg p, s\}$	Res(1, 2)
4:	$\{p, q\}$	$\in G$
5:	$\{q, s\}$	Res(3, 4)
6:	$\{\neg s\}$	$\in G$
7:	$\{q\}$	`ERROR:` $\nexists(\neg q)$

Dadurch ist die Formel erfüllbar und ein mögliches Modell wäre:
$I(p) = 0$, $I(q) = \mathbf{1}$, $I(r) = 0$, $I(s) = 0$

Lösung: (c)

1:	$\{\neg q\}$	$\in H$
2:	$\{\neg r, q\}$	$\in H$
3:	$\{\neg r\}$	Res(1, 2)
4:	$\{\neg p, q\}$	$\in H$
5:	$\{\neg p\}$	Res(1, 4)
6:	$\{p, q, r\}$	$\in H$
7:	$\{p, r\}$	Res(1, 6)
8:	$\{p\}$	Res(3, 7)
9:	$\{\}$	Res(5, 8)

:Aufgabe

Aufgabe 7: Unit-Resolution

Angabe: Welche der folgenden widerlegbaren Klauselmengen ist mit Unit-Resolution widerlegbar?

(a) $F = \Big\{\{p, q, r\}, \{p, \neg q, r\}, \{\neg p, q, r\}, \{\neg p, \neg q, r\}, \{\neg r\}\Big\}$

(b) $G = \Big\{\{p\}, \{q\}, \{\neg p, \neg q, s\}, \{\neg p, \neg s\}\Big\}$

Lösung:

Unit-Resolution: Einer der verwendeten Resolventen beim Ableiten muss immer einelementig sein.

Die Klauselmenge aus (b) ist widerlegbar, da:

1:	$\{q\}$	$\in G$
2:	$\{\neg p, \neg q, s\}$	$\in G$
3:	$\{\neg p, s\}$	Res(1, 2)
4:	$\{p\}$	$\in G$
5:	$\{s\}$	Res(3, 4)
6:	$\{\neg p, \neg s\}$	$\in G$
7:	$\{p\}$	$\in G$
8:	$\{\neg s\}$	Res(6, 7)
9:	$\{\}$	Res(5, 8)

:Aufgabe

Aufgabe 8: Beweisen (Induktion)

Angabe: Sei $x^0 = \neg x$ und $x^1 = x$. Zeigen Sie, dass folgendes Widerlegbar ist:

$$\Big\{\{p_1^{i_1} \dots p_n^{i_n}\} : \langle i_1 \dots i_n\rangle \in \{0, \mathbf{1}\}^n\Big\}$$

Lösung:

Beweis durch **Induktion** nach n.

Wenn $n = 1$ wird ein duales Literal – und damit eine Tautologie – erzeugt. Dies geschieht durch "hin- und herschalten" des Exponenten (i) zwischen 0 und 1, welches eine Negation bei 0 ergibt. Somit resultiert immer eine Tautologie.

$$\Big\{\{p^i\} : \langle i\rangle \in \{0, 1\}^1\Big\} = \Big\{\{p\}, \{\neg p\}\Big\}$$

Wenn $n > 1 \dots$

$$\Big\{\{p_i^{i_1} \dots p_n^{i_n}\} : \langle i_1 \dots i_n \rangle \in \{0,1\}^n\Big\}$$

... "schneidet" man das letzte Element n heraus (also nicht bis n, sondern bis $n-1$ angeben und n "händisch" angeben) ...

$$\Big\{\{p_i^{i_1} \dots p_{n-1}^{i_{n-1}}\} \cup \{p_n\}, \{p_i^{i_1} \dots p_{n-1}^{i_{n-1}}\} \cup \{\neg p_n\} : \langle i_1 \dots i_{n-1} \rangle \in \{0,1\}^{n-1}\Big\}$$

... und stützt sich auf die Formel ohne diesem n-ten Element.

$$\Big\{\{p_i^{i_1} \dots p_{n-1}^{i_{n-1}}\} : \langle i_1 \dots i_{n-1} \rangle \in \{0,1\}^{n-1}\Big\}$$

Wenn vorheriges widerlegbar ist, dann ist somit auch $n > 1$ widerlegbar.

:Aufgabe

Aufgabe 9: Beweisen

Angabe: Beweisen Sie, dass C erfüllbar ist, wenn

$$C \subsetneq \Big\{\{p_1^{i_1} \dots p_n^{i_n}\} : \langle i_1 \dots i_n \rangle \in \{0,\mathbf{1}\}^n\Big\}$$

Lösung:

Erweitere C zu ...

$$C' = \Big\{\{p_1^{i_1} \dots p_n^{i_n}\} : \langle i_1 \dots i_n \rangle \in \{0,1\}^n\Big\} - \Big\{\{p_1^{v_1} \dots p_n^{v_n}\}\Big\}$$

... für gewisse $v_1 \dots v_n \in \{0,1\}^n$. C' ist erfüllbar (und damit auch C), da jede Klausel aus C' mindestens ein $p_i^{1-v_i}$ enthält. Setze $I(p_i) = 1 - v_i$.

:Aufgabe

Aufgabe 10: Beweisen (Resolution AL)

Angabe: Sei $\mathcal{C}$ eine Klauselmenge, $C \in \mathcal{C}$, und V die Menge der Variablen in C. Sei

COMP$(C,\mathcal{C})$= $\{C' : C \subseteq C' \wedge C'$ enthält jede Variable in V genau einmal $\}$,

$$\text{COMP}(\mathcal{C}) = \bigcup_{C \in \mathcal{C}} \text{COMP}(C,\mathcal{C}).$$

Zeigen Sie, dass $\mathcal{C}$ mit Resolution$_{AL}$ widerlegbar ist, und nur dann, wenn COMP($\mathcal{C}$) mit Resolution$_{AL}$ widerlegbar ist.

Lösung:

$\mathcal{C}$ ist widerlegbar, genau dann, wenn COMP($\mathcal{C}$) widerlegbar ist, da COMP($C, \mathcal{C}$) die Klausel C herleitet.

COMP($\mathcal{C}$) ist widerlegbar, genau dann, wenn $\mathcal{C}$ widerlegbar ist, da es für jede Klausel $D \in$ COMP($\mathcal{C}$) eine Klausel $D' \in \mathcal{C}$ gibt, sodass $D' \subseteq D$. Kürze die Resolutionswiderlegung von COMP($\mathcal{C}$) dementsprechend.

:Aufgabe

Aufgabe 11: Vollständigkeit beweisen (Resolution AL)

Angabe: (Vollständigkeit der Aussagenlogische Resolution) Sei $\mathcal{C}$ eine Klauselmenge. Zeige

$$\mathcal{C} \text{ mit Resolution widerlegbar} \Leftrightarrow \mathcal{C} \text{ unerfüllbar.}$$

Lösung:

$\mathcal{C}$ mit Resolution widerlegbar $\Leftrightarrow$
COMP($\mathcal{C}$) mit Resolution widerlegbar $\Leftrightarrow$
COMP($\mathcal{C}$) $= \Big\{ \{p_1^{i_1} \dots p_n^{i_n}\} : \langle i_1 \dots i_n \rangle \in \{0,1\}^n \Big\}$.

:Aufgabe

Aufgabe 12: Formeln PL in PNF/SNF/KNF

Angabe: Seien folgende prädikatenlogischen Formeln gegeben:
(a) $\Big[\forall x \exists y P(x,y) \rightarrow \Big(\forall x B(x) \vee \exists x C(x)\Big)\Big] \rightarrow \exists x C(x)$
(b) $\forall x \exists y \forall z \Big[P(x,y) \vee Q(y,z)\Big] \vee \neg \forall z \Big[P(z,z) \vee Q(z,z)\Big]$
(c) $\forall x \Big[P(x) \rightarrow \exists y \Big(Q(y) \rightarrow \forall z \Big(R(z) \rightarrow \exists u \Big(W(u) \rightarrow \forall v S(v)\Big)\Big)\Big)\Big]$

Geben Sie jeweils die (i) Pränex-/ (ii) Skolem-/ (ii) Klausel-Normalform an.

Lösung: (a)

(i) Die Pränex-Normalform ergibt sich durch folgende Schritte:

1. $\Big[\forall x \exists y P(x,y) \rightarrow \Big(\forall x B(x) \vee \exists x C(x)\Big)\Big] \rightarrow \exists x C(x)$

$$\left.\begin{array}{ll} 2. & \Big[\neg\forall x\exists y P(x,y) \lor \Big(\forall x B(x) \lor \exists x C(x)\Big)\Big] \to \exists x C(x) \\ 3. & \neg\Big[\neg\forall x\exists y P(x,y) \lor \Big(\forall x B(x) \lor \exists x C(x)\Big)\Big] \lor \exists x C(x) \end{array}\right\} \texttt{REPLACE}$$

$$\left.\begin{array}{ll} 4. & \Big[\neg\neg\forall x\exists y P(x,y) \land \neg\Big(\forall x B(x) \lor \exists x C(x)\Big)\Big] \lor \exists x C(x) \\ 5. & \Big[\forall x\exists y P(x,y) \land \Big(\neg\forall x B(x) \land \neg\exists x C(x)\Big)\Big] \lor \exists x C(x) \\ 6. & \Big[\forall x\exists y P(x,y) \land \Big(\exists x \neg B(x) \land \forall x \neg C(x)\Big)\Big] \lor \exists x C(x) \end{array}\right\} \texttt{NEGATION}$$

$$\left.\begin{array}{ll} 7. & \Big[\forall x\exists y P(x,y) \land \Big(\exists x' \neg B(x') \land \forall x'' \neg C(x'')\Big)\Big] \lor \exists \tilde{x} C(\tilde{x}) \end{array}\right\} \texttt{SYNTAX}$$

$$\left.\begin{array}{ll} 8. & \forall x\exists y\exists x'\forall x''\exists\tilde{x}\Big[P(x,y) \land \neg B(x') \land \neg C(x'') \lor C(\tilde{x})\Big] \end{array}\right\} \texttt{SHIFT}$$

(ii) Die Skolem-Normalform ergibt sich durch folgende Schritte:

$$\left.\begin{array}{ll} 1. & \Big[\forall x\exists y P(x,y) \land \Big(\exists x' \neg B(x') \land \forall x'' \neg C(x'')\Big)\Big] \lor \exists \tilde{x} C(\tilde{x}) \end{array}\right\} \text{siehe (i) 7. Schritt}$$

$$\left.\begin{array}{ll} 2. & \Big[\forall x P(x,f(x)) \land \Big(\exists x' \neg B(x') \land \forall x'' \neg C(x'')\Big)\Big] \lor \exists \tilde{x} C(\tilde{x}) \\ 3. & \Big[\forall x P(x,f(x)) \land \Big(\neg B(c) \land \forall x'' \neg C(x'')\Big)\Big] \lor \exists \tilde{x} C(\tilde{x}) \\ 4. & \Big[\forall x P(x,f(x)) \land \Big(\neg B(c) \land \forall x'' \neg C(x'')\Big)\Big] \lor C(d) \\ 5. & \Big[P(x,f(x)) \land \Big(\neg B(c) \land \neg C(x'')\Big)\Big] \lor C(d) \end{array}\right\} \texttt{SKOLEM}$$

$$\begin{array}{ll} 6. & P(x,f(x)) \land \neg B(c) \land \neg C(x'') \lor C(d) \end{array}$$

(iii) Die Klausel-Normalform ergibt sich durch folgende Schritte:

$$\left.\begin{array}{ll} 1. & \Big[P(x,f(x)) \land \neg B(c) \land \neg C(x'')\Big] \lor C(d) \end{array}\right\} \text{siehe (ii) 6. Schritt}$$

$$\left.\begin{array}{ll} 2. & \Big(P(x,f(x)) \lor C(d)\Big) \land \Big(\neg B(c) \lor C(d)\Big) \land \Big(\neg C(x'') \lor C(d)\Big) \end{array}\right\} \text{Distribution}$$

$$\left.\begin{array}{ll} 3. & \Big\{\{P(x,f(x)), C(d)\}, \{\neg B(c), C(d)\}, \{\neg C(x''), C(d)\}\Big\} \end{array}\right\} \text{als Menge}$$

Lösung: (b)

(i) Die Pränex-Normalform ergibt sich durch folgende Schritte:

$$\begin{array}{ll} 1. & \forall x\exists y\forall z\Big[P(x,y) \lor Q(y,z)\Big] \lor \neg\forall z\Big[P(z,z) \lor Q(z,z)\Big] \end{array}$$

$$\left.\begin{array}{ll} 2. & \forall x\exists y\forall z\Big[P(x,y) \lor Q(y,z)\Big] \lor \exists z\neg\Big[P(z,z) \lor Q(z,z)\Big] \\ 3. & \forall x\exists y\forall z\Big[P(x,y) \lor Q(y,z)\Big] \lor \exists z\Big[\neg P(z,z) \land \neg Q(z,z)\Big] \end{array}\right\} \texttt{NEGATION}$$

$$\left.\begin{array}{ll} 4. & \forall x\exists y\forall z\Big[P(x,y) \lor Q(y,z)\Big] \lor \exists z'\Big[\neg P(z',') \land \neg Q(z',z')\Big] \end{array}\right\} \texttt{SYNTAX}$$

$$\left.\begin{array}{ll} 5. & \forall x\exists y\forall z\exists z'\Big[P(x,y) \lor Q(y,z) \lor \neg P(z',z') \land \neg Q(z',z')\Big] \end{array}\right\} \texttt{SHIFT}$$

(ii) Die Skolem-Normalform ergibt sich durch folgende Schritte:

$$\left.\begin{array}{ll} 1. & \forall x\exists y\forall z\Big[P(x,y) \lor Q(y,z)\Big] \lor \exists z'\Big[\neg P(z',z') \land \neg Q(z',z')\Big] \end{array}\right\} \text{siehe (i) 4. Schritt}$$

$$\left.\begin{array}{ll} 2. & \forall x\forall z\Big[P(x,f(x)) \lor Q(f(x),z)\Big] \lor \exists z'\Big[\neg P(z',z') \land \neg Q(z',z')\Big] \\ 3. & \forall x\forall z\Big[P(x,f(x)) \lor Q(f(x),z)\Big] \lor \Big[\neg P(c,c) \land \neg Q(c,c)\Big] \\ 4. & \Big[P(x,f(x)) \lor Q(f(x),z)\Big] \lor \neg P(c,c) \land \neg Q(c,c) \end{array}\right\} \texttt{SKOLEM}$$

(iii) Die Klausel-Normalform ergibt sich durch folgende Schritte:

1. $\Big[P(x, f(x)) \vee Q(f(x), z)\Big] \vee \Big[\neg P(c, c) \wedge \neg Q(c, c)\Big]$ } siehe (ii) 4. Schritt
2. $\Big(P(x, f(x)) \vee Q(f(x), z) \vee \neg P(c, c)\Big) \wedge \Big(P(x, f(x)) \vee Q(f(x), z) \vee \neg Q(c, c)\Big)$
2. $\Big\{\{P(x, f(x)), Q(f(x), z), \neg P(c, c)\}, \{P(x, f(x)), Q(f(x), z), \neg Q(c, c)\}\Big\}$

Lösung: (c)

(i) Die Pränex-Normalform ergibt sich durch folgende Schritte:

1. $\forall x\Big[P(x) \to \exists y\Big(Q(y) \to \forall z\Big(R(z) \to \exists u\Big(W(u) \to \forall v S(v)\Big)\Big)\Big)\Big]$
2. $\forall x\Big[\neg P(x) \vee \exists y\Big(Q(y) \to \forall z\Big(R(z) \to \exists u\Big(W(u) \to \forall v S(v)\Big)\Big)\Big)\Big]$
3. $\forall x\Big[\neg P(x) \vee \exists y\Big(\neg Q(y) \vee \forall z\Big(R(z) \to \exists u\Big(W(u) \to \forall v S(v)\Big)\Big)\Big)\Big]$
4. $\forall x\Big[\neg P(x) \vee \exists y\Big(\neg Q(y) \vee \forall z\Big(\neg R(z) \vee \exists u\Big(W(u) \to \forall v S(v)\Big)\Big)\Big)\Big]$
5. $\forall x\Big[\neg P(x) \vee \exists y\Big(\neg Q(y) \vee \forall z\Big(\neg R(z) \vee \exists u\Big(\neg W(u) \vee \forall v S(v)\Big)\Big)\Big)\Big]$ } `REPLACE`
6. $\forall x \exists y \forall z \exists u \forall v\Big[\neg P(x) \vee \neg Q(y) \vee \neg R(z) \vee \neg W(u) \vee S(v)\Big]$ } `SHIFT`

(ii) Die Skolem-Normalform ergibt sich durch folgende Schritte:

1. $\forall x\Big[\neg P(x) \vee \exists y\Big(\neg Q(y) \vee \forall z\Big(\neg R(z) \vee \exists u\Big(\neg W(u) \vee \forall v S(v)\Big)\Big)\Big)\Big]$
2. $\forall x\Big[\neg P(x) \vee \Big(\neg Q(f(x)) \vee \forall z\Big(\neg R(z) \vee \exists u\Big(\neg W(u) \vee \forall v S(v)\Big)\Big)\Big)\Big]$
3. $\forall x\Big[\neg P(x) \vee \Big(\neg Q(f(x)) \vee \forall z\Big(\neg R(z) \vee \Big(\neg W(g(x, z)) \vee \forall v S(v)\Big)\Big)\Big)\Big]$
4. $\neg P(x) \vee \neg Q(f(x)) \vee \neg R(z) \vee \neg W(g(x, z)) \vee S(v)$ } `SKOLEM`

(iii) Die Klausel-Normalform ergibt sich durch folgende Schritte:

1. $\neg P(x) \vee \neg Q(f(x)) \vee \neg R(z) \vee \neg W(g(x, z)) \vee S(v)$ } siehe (ii) 4. Schritt
2. $\Big\{\{\neg P(x), \neg Q(f(x)), \neg R(z), \neg W(g(x, z)), S(v)\}\Big\}$ } als Menge

:Aufgabe

Aufgabe 13: Modell/Gegenbeispiel finden zu Formel PL

Angabe: Sei $A = \forall x \exists y(P(x) \to \neg P(y)) \to \forall x P(x) \vee \forall x \neg P(x)$.

(a) Geben Sie ein Modell für A an.
(b) Geben Sie ein Gegenbeispiel für A an (d.h. ein Modell für $\neg A$).

Lösung:

Da A nur das einstellige Prädikat P enthält, genügt es zweielementige Modelle zu betrachten. A übersetzt sich demnach zu der Form A^*:

$$\forall x\exists y\Big(P(x)\to\neg P(y)\Big)\to\underbrace{\forall xP(x)}_{\Omega}\vee\forall x\neg P(x)$$

$$\forall x\exists y\Big(P(x)\to\neg P(y)\Big)\to\Big(\underbrace{P(a)}_{\Omega(x/a)}\wedge\underbrace{P(b)}_{\Omega(x/b)}\Big)\vee\underbrace{\forall x\neg P(x)}_{\Sigma}$$

$$\forall x\underbrace{\exists y\Big(P(x)\to\neg P(y)\Big)}_{\Phi}\to\Big(P(a)\wedge P(b)\Big)\vee\Big(\underbrace{\neg P(a)}_{\Sigma(x/a)}\wedge\underbrace{\neg P(b)}_{\Sigma(x/b)}\Big)$$

$$\underbrace{\forall x\Big[\underbrace{\Big(P(x)\to\neg P(a)\Big)}_{\Phi(y/a)}\vee\underbrace{\Big(P(x)\to\neg P(b)\Big)}_{\Phi(y/b)}\Big]}_{\Delta}\to\dots$$

$$\Big[\underbrace{\Big(P(a)\to\neg P(a)\Big)\vee\Big(P(a)\to\neg P(b)\Big)}_{\Delta(x/a)}\Big]\wedge\Big[\underbrace{\Big(P(b)\to\neg P(a)\Big)\vee\Big(P(b)\to\neg P(b)\Big)}_{\Delta(x/b)}\Big]\to\dots$$

$P(a)$	$P(b)$	$\neg P(a)$	$\neg P(b)$	$P(a)\to\neg P(a)$	$P(a)\to\neg P(b)$	$\dots$	A^*
0	0	**1**	**1**	**1**	**1**	$\dots$	**1**
0	**1**	**1**	0	**1**	**1**	$\dots$	0
1	0	0	**1**	0	**1**	$\dots$	0
1	**1**	0	0	0	0	$\dots$	**1**

(a) Ein Modell für A wäre: $\langle\{a,b\},P\rangle, P=\{\}$
(b) Ein Modell für $\neg A$ wäre: $\langle\{a,b\},P\rangle, P=\{\langle b\rangle\}$

:Aufgabe

Aufgabe 14: Unifikation

Angabe: Sei c eine Konstante. Geben Sie von jede der folgenden Mengen Terme an, ob sie unifizierbar sind, und wenn ja, gebe einen MGU an.

$A=\{f(g(x),y),g(f(x),y)\}$
$B=\{Q(x,y),Q(f(c),f(x,z)),Q(x,f(f(c),x))\}$
$C=\{g(P(x,u),Q(y)),g(P(f(y),c),Q(f(z))),g(P(x,u),Q(f(f(x))))\}$

Lösung: (a)

Das Termgleichungssystem $\varepsilon=\{f(g(x),y)\doteq g(f(x),y)\}$ zeigt schon, dass der Substitutionsversuch $f(\dots)\doteq g(\dots)$ an einem `CLASH` scheitert – zwei verschiedene Funktionen unifizieren, mit nicht notwendigerweise verschiedenen Parameter.

Lösung: (b)

$\varepsilon = \{Q(x,y) \doteq Q(f(c), f(x,z)), Q(f(c), f(x,z)) \doteq Q(x, f(f(c), x))\}$

$$\begin{array}{lclcl} Q(x,y) & x/f(c) & Q(f(c),y) & y/f(f(c) & Q(f(c), f(f(c), f(c))) \\ Q(f(c), f(x,z)) & \Longrightarrow & Q(f(c), f(f(c), z)) & \Longrightarrow & Q(f(c), f(f(c), f(c))) \\ Q(x, f(f(c), x)) & & Q(f(c), f(f(c), f(c))) & z/f(c) & Q(f(c), f(f(c), f(c))) \end{array}$$

$\rho = mgu(B) = \Big\{x/f(c), y/f(f(c)), z/f(c)\Big\}$

Lösung: (c)

$\varepsilon = \{P(x,u) \doteq P(f(y), c), Q(y) \doteq Q(f(z)), P(f(y), c) \doteq P(x,u), Q(f(z)) \doteq Q(f(f(x)))\}$

$$\begin{array}{lclc} g(P(x,u), Q(y)) & u/c & g(P(x,c), Q(y)) & y/f(z) \\ g(P(f(y),c), Q(f(z))) & \Longrightarrow & g(P(f(y),c), Q(f(z))) & \Longrightarrow \\ g(P(x,u), Q(f(f(x)))) & & g(P(x,c), Q(f(f(x)))) & \end{array}$$

$$\begin{array}{lcl} g(P(x,c), Q(f(z))) & x/f(f(z)) & g(P(f(f(z)),c), Q(f(z))) \\ g(P(f(f(z)),c), Q(f(z))) & \Longrightarrow & g(P(f(f(z)),c), Q(f(z))) \\ g(P(x,c), Q(f(f(x)))) & & g(P(f(f(z)),c), Q(f(f(f(f(z)))))) \end{array}$$

Die Unifikation von $Q(f(z))$ mit $Q(f(f(f(f(z)))))$ ist nicht möglich, daher ist C nicht unifizierbar.

:Aufgabe

Aufgabe 15: Unifikation

Angabe: Sei c eine Konstante. Geben Sie von jede der folgenden Mengen Terme an, ob sie unifizierbar sind, und wenn ja, gebe einen MGU an.

$A = \{P(x, f(z), g(c,z)), P(c, g(z), g(f(x), c))\}$
$B = \{S(f(x), v), S(u, f(u)), S(f(c), f(f(c)))\}$
$C = \{R(x, f(z), z), R(f(y), f(z), z), R(f(y), y, f(x))\}$

Lösung: (a)

Schon das Termgleichungssystem $\varepsilon = \{x \doteq c, f(z) \doteq g(z), g(c,z) \doteq c\}$ zeigt, dass A nicht unifizierbar ist, da ein `CLASH` sowohl für $f \doteq g$ als auch für $g \doteq c$ auftritt und somit zu $\perp$ führt.

Lösung: (b)

$\varepsilon = \{f(x) \doteq u, v \doteq f(u), u \doteq f(c), f(u) \doteq f(f(c))\}$

$$
\begin{array}{llllll}
S(f(x),v) & x/c & S(f(c),v) & u/f(c) & S(f(c),v) \\
S(u,f(u)) & \Longrightarrow & S(u,f(u)) & \Longrightarrow & S(f(c),f(f(c))) \\
S(f(c),f(f(c))) & & S(f(c),f(f(c))) & & S(f(c),f(f(c)))
\end{array}
$$

$$
\begin{array}{ll}
v/f(f(c)) & S(f(c),f(f(c))) \\
\Longrightarrow & S(f(c),f(f(c))) \\
 & S(f(c),f(f(c)))
\end{array}
$$

$$\rho = mgu(B) = \Big\{x/c, u/f(c), v/f(f(c))\Big\}$$

Lösung: (c)

$\varepsilon = \{x \doteq f(y), f(z) \doteq f(z), z \doteq z, f(y) \doteq f(y), f(z) \doteq y, z \doteq f(x)\}$

Nach anwenden von `DELETE`:
$\varepsilon = \{x \doteq f(y), f(z) \doteq y, z \doteq f(x)\}$

$$
\begin{array}{lllll}
R(x,f(z),z) & x/f(y) & R(f(y),f(z),z) & y/f(z) & R(f(f(z)),f(z),z) \\
R(f(y),f(z),z) & \Longrightarrow & R(f(y),f(z),z) & \Longrightarrow & R(f(f(z)),f(z),z) \\
R(f(y),y,f(x)) & & R(f(y),y,f(f(y))) & & R(f(f(z)),f(z),f(f(f(z))))
\end{array}
$$

Da $z \in f(f(f(z)))$ ist, tritt ein `OCCURS CHECK` auf und somit ist C nicht unifizierbar.

:Aufgabe

Aufgabe 16: Resolution PL

Angabe: Beweisen Sie mit Hilfe der Resolutionsmethode:

$$F = \exists x \forall y (P(x) \vee Q(y)) \rightarrow \forall y \exists x (P(x) \vee Q(y))$$

Lösung:

$$
\begin{array}{lll}
\neg F = & \neg\Big[\exists x \forall y (P(x) \vee Q(y)) \rightarrow \forall y \exists x (P(x) \vee Q(y))\Big] & \\
 & \neg\Big[\neg\exists x \forall y (P(x) \vee Q(y)) \vee \forall y \exists x (P(x) \vee Q(y))\Big] & \\
 & \neg\neg\exists x \forall y (P(x) \vee Q(y)) \wedge \neg\forall y \exists x (P(x) \vee Q(y)) & \\
 & \exists x \forall y (P(x) \vee Q(y)) \wedge \exists y \neg\exists x (P(x) \vee Q(y)) & \\
 & \exists x \forall y (P(x) \vee Q(y)) \wedge \exists y \forall x \neg(P(x) \vee Q(y)) & \\
 & \exists x \forall y (P(x) \vee Q(y)) \wedge \exists y \forall x (\neg P(x) \wedge \neg Q(y)) & \mathrm{NNF}(F) \\
 & \exists x \forall y (P(x) \vee Q(y)) \wedge \exists y' \forall x' (\neg P(x') \wedge \neg Q(y')) & \\
 & \forall y (P(c) \vee Q(y)) \wedge \forall x' (\neg P(x') \wedge \neg Q(d)) & \\
 & (P(c) \vee Q(y)) \wedge \neg P(x') \wedge \neg Q(d) & \mathrm{SNF}(F) \\
 & \Big\{\{P(c), Q(y)\}, \{\neg P(x')\}, \{\neg Q(d)\}\Big\} & \mathrm{KNF}(F)
\end{array}
$$

Durch Anwendung der Resolution$_{AL}$ in Kombination mit Substitution gelangt man nun von der Klauselmenge zur leeren Klausel.

1:	$\{P(c), Q(y)\}$	$\in F$
2:	$\{\neg Q(d)\}$	$\in F$
3:	$\{P(c)\}$	Res(1, 2)
4:	$\{\neg P(x')\}$	$\in F$
5:	$\{\}$	Res(3, 4)

:Aufgabe

Aufgabe 17: Resolution (Widerlegbarkeit)

Angabe: Ist folgende Klauselmenge F widerlegbar mittels Resolution?

$$\left\{ \begin{array}{ll} \{P(x), P(f(x)), P(f(f(x)))\}, & \{P(x), P(f(x)), \neg P(f(f(x)))\} \\ \{P(x), \neg P(f(x)), P(f(f(x)))\}, & \{P(x), \neg P(f(x)), \neg P(f(f(x)))\} \\ \{\neg P(x), P(f(x)), P(f(f(x)))\}, & \{\neg P(x), P(f(x)), \neg P(f(f(x)))\} \\ \{\neg P(x), \neg P(f(x)), P(f(f(x)))\}, & \{\neg P(x), \neg P(f(x)), \neg P(f(f(x)))\} \end{array} \right\}$$

Lösung:

Man wende folgende Substitution an:
$\rho = \Big\{P(x)/P, P(f(x))/Q, P(f(f(x)))/R\Big\}$

1:	$\{P, Q, R\}$	$\in F$
2:	$\{P, Q, \neg R\}$	$\in F$
3:	$\{P, \neg Q, R\}$	$\in F$
4:	$\{P, \neg Q, \neg R\}$	$\in F$
5:	$\{\neg P, Q, R\}$	$\in F$
6:	$\{\neg P, Q, \neg R\}$	$\in F$
7:	$\{\neg P, \neg Q, R\}$	$\in F$
8:	$\{\neg P, \neg Q, \neg R\}$	$\in F$
9:	$\{P, Q\}$	Res(1, 2)
10:	$\{P, \neg Q\}$	Res(3, 4)
11:	$\{P\}$	Res(9, 10)
12:	$\{\neg P, Q\}$	Res(5, 6)
13:	$\{\neg P, \neg Q\}$	Res(7, 8)
14:	$\{\neg P\}$	Res(12, 13)
15:	$\{\}$	Res(11, 14)

Demnach ist die Klauselmenge F mit Hilfe der Resolution widerlegbar.

:Aufgabe

Aufgabe 18: Resolution

Angabe: Widerlegen Sie mit Hilfe des Resolutionsverfahrens:

$$\Big\{\{P(a,b,c)\}, \{\neg P(x,y,z), P(y,x,z)\}, \{\neg P(x,y,z), P(z,y,x)\}, \{\neg P(x,y,z), P(x,z,y)\}, \{\neg P(c,a,b)\}\Big\}.$$

Lösung:

1:	$\{P(a,b,c)\}$	$\in F$
2:	$\{\neg P(x,y,z), P(z,y,x)\}$	$\in F$
ρ_1:	$\{\neg P(a,b,c), P(c,b,a)\}$	$\rho_1 = \{\ x/a, y/b, z/c\ \}$
3:	$\{P(c,b,a)\}$	Res(1, 2)
4:	$\{\neg P(x,y,z), P(x,z,y)\}$	$\in F$
ρ_2:	$\{\neg P(c,b,a), P(c,a,b)\}$	$\rho_2 = \{\ x/c, y/b, z/a\ \}$
5:	$\{P(c,a,b)\}$	Res(3, 4)
6:	$\{\neg P(c,a,b)\}$	$\in F$
7:	$\{\}$	Res(5, 6)

:Aufgabe

Aufgabe 19: Resolution PL

Angabe: Beweisen Sie folgende Formeln mit Hilfe des Resolutionsverfahrens oder zeigen Sie, dass sie nicht gültig sind.

(a) $F = \forall x \exists y \forall u \exists v P(x,y,u,v) \to \forall x \forall u \exists y \exists v P(x,y,u,v)$

(b) $G = \forall x \forall u \exists y \exists v P(x,y,u,v) \to \forall x \exists y \forall u \exists v P(x,y,u,v)$

Lösung: (a)

$$\begin{aligned}
\neg F = \quad & \neg\Big[\forall x \exists y \forall u \exists v P(x,y,u,v) \to \forall x \forall u \exists y \exists v P(x,y,u,v)\Big] \\
& \neg\Big[\neg\forall x \exists y \forall u \exists v P(x,y,u,v) \lor \forall x \forall u \exists y \exists v P(x,y,u,v)\Big] \\
& \neg\neg\forall x \exists y \forall u \exists v P(x,y,u,v) \land \neg\forall x \forall u \exists y \exists v P(x,y,u,v) \\
& \forall x \exists y \forall u \exists v P(x,y,u,v) \land \exists x \exists u \forall y \forall v \neg P(x,y,u,v) \\
& \forall x \exists y \forall u \exists v P(x,y,u,v) \land \exists x' \exists u' \forall y' \forall v' \neg P(x',y',u',v') \\
& \forall x \forall u P(x, f(x), u, g(x,u)) \land \forall y \forall v \neg P(c, y', d, v') \\
& P(x, f(x), u, g(x,u)) \land \neg P(c, y', d, v') \\
& \Big\{\{P(x, f(x), u, g(x,u))\}, \{\neg P(c, y', d, v')\}\Big\}
\end{aligned}$$

Nun unifiziert man mit $\rho = mgu(F) = \{x/c, y'/f(c), u/d, v'/g(c,d)\}$.

1:	$\{P(c, f(c), d, g(c,d)\}$	$\in F$
2:	$\{\neg P(c, f(c), d, g(c,d)\}$	$\in F$
3:	$\{\}$	Res(1, 2)

Dadurch, dass $\neg F$ widerlegt wurde, ist somit F bewiesen.

Lösung: (b)

$$
\begin{aligned}
\neg G = \quad & \neg\Big[\forall x \forall u \exists y \exists v P(x,y,u,v) \rightarrow \forall x \exists y \forall u \exists v P(x,y,u,v)\Big] \\
& \neg\Big[\neg\forall x \forall u \exists y \exists v P(x,y,u,v) \vee \forall x \exists y \forall u \exists v P(x,y,u,v)\Big] \\
& \neg\neg\forall x \forall u \exists y \exists v P(x,y,u,v) \wedge \neg\forall x \exists y \forall u \exists v P(x,y,u,v) \\
& \forall x \forall u \exists y \exists v P(x,y,u,v) \wedge \exists x \forall y \exists u \forall v \neg P(x,y,u,v) \\
& \forall x \forall u \exists y \exists v P(x,y,u,v) \wedge \exists x' \forall y' \exists u' \forall v' \neg P(x',y',u',v') \\
& \forall x \forall u P(x, f(x,u), u, g(x,u)) \wedge \forall y' \forall v' \neg P(c, y', h(y'), v') \\
& P(x, f(x,u), u, g(x,u)) \wedge \neg P(c, y', h(y'), v') \\
& \Big\{\{P(x, f(x,u), u, g(x,u))\}, \{\neg P(c, y', h(y'), v')\}\Big\}
\end{aligned}
$$

$$\varepsilon = \{x \doteq c, f(x,u) \doteq y', u \doteq h(y'), g(x,u) \doteq v'\}$$

Die erste Ersetzung x/c ist valid, genauso wie die dritte $u/h(y')$. Die vierte Ersetzung $g(x,u)/v'$ wird durch Anwenden von `ORIENT` zur validen Ersetzung $v'/g(x,u)$. Nachdem $f(x,u)/y'$ orientiert wurde zu $y'/f(x,u)$ geschieht bei der Rückersetzung folgendes: $y'/f(c, h(f(c, h(\ldots$ es kommt also zu einem `OCCURS CHECK` und G ist damit nicht widerlegt und $\neg G$ nur erfüllbar (nicht gültig).

:Aufgabe

Aufgabe 20: Resolution

Angabe: Sei die Klauselmenge $\Delta = \{\{P(u), P(v)\}, \{\neg P(x), \neg P(y)\}\}$ gegeben. Zeige mit Hilfe der Resolutionsmethode, dass Δ widerlegbar ist.

Lösung:

1:	$\{P(u), P(v)\}$	$\in \Delta$
2:	$\{P(u)\}$	FA(1)
3:	$\{\neg P(x), \neg P(y)\}$	$\in \Delta$
4:	$\{\neg P(x)\}$	Res(2, 3)
5:	$\{\}$	Res(2, 4)

FA(x) bezeichnet die Faktorisierung der Klausel x, mit der erst die leere Klausel $\{\}$ erreichbar und damit die Formel Δ widerlegbar ist.

:Aufgabe

Aufgabe 21: Resolution

Angabe: Seien folgende prädikatenlogische Formeln gegeben:

$$
\begin{array}{rl}
\text{(I)} & \Big(\forall x(A(x) \rightarrow B(x))\Big) \rightarrow \exists y(C(y) \wedge \not B(y)) \\
\text{(II)} & \forall x\Big(C(x) \rightarrow B(x)\Big) \\
\text{(III)} & \exists x\Big(A(x) \wedge \neg B(x)\Big)
\end{array}
$$

Beweise mit Hilfe der Resolutionsmethode, dass: $\text{(I)}, \text{(II)} \models \text{(III)}$.

Lösung:

Die logische Konsequenz $\models$ kann auch gesehen werden als
$\text{(I)} \rightarrow (\text{(II)} \rightarrow \text{(III)})$, welche gültig ist, gdw $(\text{(I)} \wedge \text{(II)}) \rightarrow \text{(III)}$ gültig ist.
Widerlegt man nun $\text{(I)} \wedge \text{(II)} \wedge \neg\text{(III)}$, so ist $(\text{(I)} \wedge \text{(II)}) \rightarrow \text{(III)}$ bewiesen.

Zuerst setzen wir (I), (II) und ($\neg$III) mittels Skolemisierung in Klausel-Normalform.

$$
\begin{array}{ll}
\text{NNF(I):} & \exists x\Big(A(x) \wedge \neg B(x)\Big) \vee \exists y\Big(C(y) \wedge \neg B(y)\Big) \\
\text{SNF(I):} & A(c) \wedge \neg B(c) \vee C(d) \wedge \neg B(d) \\
\text{KNF(I):} & \Big\{\{A(c), C(d)\}, \{A(c), \neg B(d)\}, \{\neg B(c), C(d)\}, \{\neg B(c), \neg B(d)\}\Big\}
\end{array}
$$

$$
\begin{array}{ll}
\text{NNF(II):} & \forall x(\neg C(x) \vee B(x)) \\
\text{SNF(II):} & \neg C(x) \vee B(x) \\
\text{KNF(II):} & \Big\{\{\neg C(x), B(x)\}\Big\}
\end{array}
$$

$$
\begin{array}{ll}
\neg\text{III:} & \neg\Big[\exists x(A(x) \wedge \neg B(x))\Big] \\
\text{NNF}(\neg\text{III}): & \forall x(\neg A(x) \vee B(x)) \\
\text{SNF}(\neg\text{III}): & \neg A(x) \vee B(x) \\
\text{KNF}(\neg\text{III}): & \Big\{\{\neg A(x), B(x)\}\Big\}
\end{array}
$$

Nun folgt die Resolution mit den vorhandenen Klauseln.

$$
\begin{array}{lll}
1: & \{A(c), C(d)\} & \in \text{(I)} \\
2: & \{A(c), \neg B(d)\} & \in \text{(I)} \\
3: & \{\neg B(c), C(d)\} & \in \text{(I)} \\
4: & \{\neg B(c), \neg B(d)\} & \in \text{(I)} \\
5: & \{\neg C(x), B(x)\} & \in \text{(II)} \\
6: & \{\neg A(x), B(x)\} & \in (\neg\text{III}) \\
7: & \{\neg A(c), C(d)\} & \text{Res(3, 6)} \\
8: & \{C(d)\} & \text{Res(1, 7)} \\
9: & \{\neg A(c), \neg B(d)\} & \text{Res(4, 6)} \\
10: & \{\neg B(d)\} & \text{Res(2, 9)} \\
11: & \{\neg C(d)\} & \text{Res(5, 10)} \\
12: & \{\} & \text{Res(8, 11)}
\end{array}
$$

Dadurch, dass das Gegenteil widerlegt wurde, wurde die Gültigkeit der ursprünglichen Aussage bewiesen.

:Aufgabe

Aufgabe 22: Refutierung einer Klauselmenge

Angabe: Seien c, d und e Konstanten. Geben Sie ein Refutierung der folgende Klauselmenge K:

$$\left\{ \begin{array}{l} \{r(d,y), \neg p(x)\} \\ \{\neg t(y,d), p(y)\}, \\ \{\neg r(y,f(e)), \neg s(e)\}, \\ \{t(c,d), \neg s(d)\}, \\ \{s(x)\} \end{array} \right\}$$

Lösung:

1:	$\{s(x)\}$	$\in K$
ρ_1:	$\{s(d)\}$	$\rho_1 = \{\ x/d\ \}$
2:	$\{t(c,d), \neg s(d)\}$	$\in K$
3:	$\{t(c,d)\}$	Res(1, 2)
4:	$\{\neg t(y,d), p(y)\}$	$\in K$
ρ_2:	$\{\neg t(c,d), p(c)\}$	$\rho_2 = \{\ y/c\ \}$
5:	$\{p(c)\}$	Res(3, 4)
6:	$\{r(d,y), \neg p(x)\}$	$\in K$
ρ_3:	$\{r(d,y), \neg p(c)\}$	$\rho_3 = \{\ x/c\ \}$
7:	$\{r(d,y)\}$	Res(5, 6)
8:	$\{r(d,y')\}$	Umbennen von 7
ρ_4:	$\{r(d,f(e))\}$	$\rho_4 = \{\ y'/f(e)\ \}$
9:	$\{\neg r(y,f(e)), \neg s(e)\}$	$\in K$
ρ_5:	$\{\neg r(d,f(e)), \neg s(e)\}$	$\rho_5 = \{\ y/d\ \}$
10:	$\{\neg s(e)\}$	Res(8, 9)
11:	$\{\}$	Res(1, 10)

:Aufgabe

Aufgabe 23: Refutierung (Unit-Resolution)

Angabe: Geben Sie ein ein Refutierung mit Hilfe der Unit-Resolution von der Klauselmenge $K = \{\{p,q,\neg r\}, \{\neg u\}, \{t, \neg q\}, \{r,u\}, \{\neg t, \neg s, u\}, \{\neg p\}, \{s\}\}$.

Lösung:

1:	$\{s\}$	$\in K$
2:	$\{\neg t, \neg s, u\}$	$\in K$
3:	$\{\neg t, u\}$	Res(1, 2)
4:	$\{\neg u\}$	$\in K$
5:	$\{\neg t\}$	Res(3, 4)
6:	$\{t, \neg q\}$	$\in K$
7:	$\{\neg q\}$	Res(5, 6)
8:	$\{\neg p\}$	$\in K$
9:	$\{p, q, \neg r\}$	$\in K$
10:	$\{p, \neg r\}$	Res(7, 9)
11:	$\{\neg r\}$	Res(8, 10)
12:	$\{r, u\}$	$\in K$
13:	$\{u\}$	Res(11, 12)
14:	$\{\}$	Res(4, 13)

:Aufgabe

Aufgabe 24: Rekursion (Fibonacci-Zahlen)

Angabe: Die Fibonacci-Zahlen $1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$ sind rekursiv definiert durch:

$$\begin{aligned} F(0) &= \mathbf{1}, \\ F(1) &= \mathbf{1} \\ F(n+2) &= F(n+1) + F(n) \end{aligned}$$

Sei $G(n) = \left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}$.

(a) Drücken Sie $G(n)$ aus mit Hilfe der Lösung der Gleich $x^2 = x + 1$.
(b) Benutze **1** um zu zeigen, dass $\frac{\sqrt{5}}{5}G(n) = F(n)$.

Lösung:

Sei $x_0 = \frac{1+\sqrt{5}}{2}$ und $x_1 = \frac{1-\sqrt{5}}{2}$, so dass $G(n) = x_0^{n+1} - x_1^{n+1}$. Weil die Rekursionsgleichungen F eindeutig deinieren, genügt es zu zeigen, dass diese von $\frac{\sqrt{5}}{5}G(n)$ erfüllt werden.

Tatsächlich haben wir

$$\frac{\sqrt{5}}{5}G(0) = \frac{\sqrt{5}}{5}(x_0 - x_1) = 1 \text{ und}$$
$$\frac{\sqrt{5}}{5}G(1) = \frac{\sqrt{5}}{5}(x_0^2 - x_1^2) = \frac{\sqrt{5}}{5}((x_0 + 1) - (x_1 + 1)) = 1$$

und weil $x_0^{n+3} = x_0^{n+2} + x_0^{n+1}$ (und das gleiche für x_1) haben wir auch $\frac{\sqrt{5}}{5}G(n+2) = \frac{\sqrt{5}}{5}G(n+1) + \frac{\sqrt{5}}{5}G(n)$.

:Aufgabe

Aufgabe 25: primitive Rekursivität

Zeigen Sie, dass folgende Funktionen und Relationen primitiv rekursiv sind:

Angabe: (1)

Konstante Funktion: Für eine Konstante c ist die Funktion $\lambda x.c$ primitiv rekursiv.

Lösung:

Die Nachfolgefunktion S ist als primitiv rekursive Basisfunktion gegeben. Durch c-fache Zusammensetzung erhalten wir $\lambda x.S(\dots(S(0))\dots)$.

Angabe: (2)

Die Signum Funktion:

$$\mathrm{sg}(x) = \begin{cases} \mathbf{1} & \text{falls } x > 0, \\ 0 & \text{falls } x = 0. \end{cases}$$

Lösung:

Definiere: $\mathrm{sg}(0) = 0 \quad$ und $\quad \mathrm{sg}(x+1) = 1$.

Angabe: (3)

Die Fallunterschied Funktion:

$$\mathrm{cases}(x, y, z) = \begin{cases} x & \text{falls } z = 0, \\ y & \text{falls } z > 0, \end{cases}$$

Lösung:

Definiere: $\mathrm{cases}(x, y, 0) = x \quad$ und $\quad \mathrm{cases}(x, y, z+1) = y$.

Angabe: (4)

Abgeschnittene Subtraktion:

$$x \dot{-} y = \begin{cases} 0 & \text{falls } y > x, \\ x - y & \text{sonst.} \end{cases}$$

Lösung:

Definiere zuerst die Vorgängerfunktion: $\mathrm{pd}(0) = 0 \quad$ und $\quad \mathrm{pd}(x+1) = x$.

Dann definiere: $x \dot{-} 0 = x \quad$ und $\quad x \dot{-} (y+1) = \mathrm{pd}(x \dot{-} y)$.

Angabe: (5)

Die charakteristische Funktion $\chi_<$ der Ungleichheitsrelation:

$$\chi_<(x,y) = \begin{cases} 1 & \text{falls } x < y, \\ 0 & \text{sonst.} \end{cases}$$

Lösung:

Man definiere $\chi_<(x,y) = \mathrm{sg}(y \dot{-} x)$.

Angabe: (6)

Die charackteristische Funktion $\chi_=$ der Gleichheitsrelation:

$$\chi_=(x,y) = \begin{cases} 1 & \text{falls } x = y, \\ 0 & \text{sonst.} \end{cases}$$

Lösung:

Unter Benutzung der vorher definierten Funktionen sg, $\dot{-}$, und $\chi_<$ können wir nun definieren:

$$\chi_=(x,y) = 1 \dot{-} \mathrm{sg}\Big(\chi_<(x,y) + \chi_<(y,x)\Big)$$

Angabe: (7)

Zeige, dass die Summe $\lambda x, y.x + y$ und das Produkt $\lambda x, y.x * y$ primitiv rekursiv sind.

Lösung:

...

Angabe: (8)

Sei f primitiv rekursiv. Zeige, dass auch die begrenzte Summe $g(x) = \sum_{y \leq x} f(y)$ primitiv rekursiv ist.

Lösung:

Definiere: $g(0) = f(0)$ und $g(x+1) = g(x) + f(x)$

Angabe: (9)

Sei f primitiv rekursiv. Zeige, dass auch das begrenzte Produkt $h(x) = \prod_{y \leq x} f(y)$ primitiv rekursiv ist.

Lösung:

Definiere: $h(0) = f(0)$ und $h(x+1) = h(x) * f(x)$

Angabe: (10)

Zeige, dass die aussagenlogischen Operatoren primitiv rekursiv sind. Dh, für gegebene primitive rekursive Relationen P und Q sind auch die Funktionen $\lambda x.\neg P(x)$, $\lambda x, y.P(x) \land Q(y)$, $\lambda x, y.P(x) \lor Q(y)$ und $\lambda x, y.P(x) \to Q(y)$ primitiv rekursiv.

Lösung:

Operator:	definiert durch:
$\lambda x.\neg P(x)$	$1 \dot{-} P(x)$
$\lambda x, y.P(x) \land Q(y)$	$1 \dot{-} sg((1 \dot{-} P(x)) + (1 \dot{-} Q(y)))$
$\lambda x, y.P(x) \lor Q(y)$	$\mathrm{sg}(P(x) + Q(y))$ und $\lambda x.\neg P(x)$ von $1 \dot{-} P(x)$
$\lambda x, y.P(x) \to Q(y)$	$sg((1 \dot{-} P(x)) + Q(y))$

Angabe: (11)

Begrenzte Quantifikation: Gegeben sei eine primitiv rekursive Relation R, und die Relationen $P(x) = \exists y \leq xR(y)$ und $Q(x) = \forall y \leq xR(y)$ primitiv rekursiv.

Lösung:

Man definiere $P(x) = \mathrm{sg}\Big(\sum_{y \leq x} R(y)\Big)$ und $Q(x) = \prod_{y \leq x} R(y)$.

Angabe: (12)

Begrenzte Suche: Gegeben sei die primitiv rekursive Funktion f. Definiere die Funktion

$$\mu y \leq z(f(\vec{x}, y) = 0)$$

die den kleinsten Wert $y \leq z$ ausgibt, so dass $f(\vec{x}, y) = 0$, und die 0 ausgibt, falls so ein y nicht existiert.

Lösung:

Definiere zuerst die Relation R durch:

$$R(\vec{x}, y) \equiv f(\vec{x}, y) = 0 \land \forall w < y(f(\vec{x}, w) \neq 0)$$

Dann kann es höchstens ein y geben mit $R(\vec{x}, y)$ und $\mu y \leq z(f(\vec{x}, y) = 0)$ wird gegeben durch $\sum_{y \leq z} y * R(\vec{x}, y)$.

Angabe: (13)

Division: Die Relation $x \mid y$ (“x teilt y”) ist primitiv rekursiv.

Lösung:

Mit Hilfe von begrenzter Quantifikation und $\chi_=$ (siehe oben) kann man $x \mid y$ definieren durch:

$$\exists z \leq y(x * z = y)$$

Angabe: (14)

Primzahlen: Der Funktion $n \mapsto p_n$, wo p_n die n-te Primzahl ist.

Lösung:

Man definiere zuerst ein Prädikat Prim(x) durch:

$$\text{Prim}(x) \Leftrightarrow x \geq 2 \wedge \forall y \leq x(y \mid x \rightarrow y = 1 \vee y = x)$$

Also drückt die Relation Prim(x) aus, dass x eine Primzahl ist. Wir benutzen weiter, dass für jede gegebene Zahl n immer eine Primzahl $\leq 2n$ vorhanden ist. Man könnte auch eine höhere obere Schranke benutzen, welche leichter zu beweisen ist[7]. Dann definiert man mit Hilfe begrenzter Quantifikation:

$$\begin{aligned} p_1 &= 2 \\ p_{n+1} &= \mu z \leq 2p_n(z > p_n \wedge \text{Prim}(z)) \end{aligned}$$

:Aufgabe

Aufgabe 26: Berechenbarkeit/Totalität beweisen/widerlegen

Angabe: Sei $f \cdot g(x) = f(x) \cdot g(x)$. Beweisen oder widerlegen Sie:

(a) f, g nicht berechenbar $\Rightarrow f \cdot g$ nicht berechenbar.
(b) f, g nicht total $\Rightarrow f \cdot g$ nicht total.

Lösung: (a)

Falsch. Gegenbeispiel: Sei A eine unentscheidbare Menge, und sei χ_A die charakteristische Funktion von A, also

$$\chi_A(x) = \begin{cases} 0 & \text{falls } x \notin A, \\ 1 & \text{falls } x \in A. \end{cases}$$

Dann sind sowohl χ_A als auch $1 - \chi_A$ nicht berechenbar, aber $\chi_A(x) * (1 - \chi_A(x)) = 0$, für alle x und damit berechenbar.

[7] Bertrands Postulat, zuerst bewiesen von Chebyshev

Lösung: (b)

Richtig. Das Produkt zweier Werte ist nicht definiert, wenn einer dieser Werte nicht definiert ist.

:Aufgabe

Abbildungsverzeichnis

1 syntaktische Vorgangsweise in AL zur Umformung in die DNF/KNF . . . 8
2 semantische Vorgangsweise in AL zur Umformung in die DNF/KNF . . . 9
3 Vorgangsweise des Markierungsalgorithmus zum Horner-Schema 11
4 Inferenzregelsystem für die Unifikation 17
5 Regeln des Sequential-Kalküls$_{AL}$ 19
6 sonstige Regeln des Sequential-Kalküls 19
7 Vorgangsweise zur Erstellung der PNF (Pränex-Normalform) 22
8 Vorgangsweise zur Erstellung der SNF (Skolem-Normalform) 22

Tabellenverzeichnis

1 die vier möglichen Bewertungen von Aussagen 5
2 Operatoren in AL – Wahrheitstabelle 7
3 Operatoren in AL – Erklärung 7
4 zwei Typen von Hornklauseln: Zielklausel und definite Hornklausel 11

Quellenverzeichnis

- TU Wien, G. Salzer und C. Fermüller: “Theoretische Informatik 1, SS06” `http://www.logic.at/lvas/185167/`
- TU Wien, W. Kuich: “Einführung in die Theorie der Informatik, WS93” `http://dmg.tuwien.ac.at/kuich/`
- TU Wien, M. Pickelbauer: “Mitschrift Baaz” `http://einstein2000.oldsch00l.com/uni/theoinf2_baaz/theoinf2_Zusammenfassung_baaz.pdf`
- Universität Tübingen, Autor unbekannt: “Wahrheit, Wahrheitsansprüche” `http://www.uni-tuebingen.de/philosophie/download/afkoch-grundlagen-1.pdf`
- Universität Karlsruhe, S. Abeck: “Kursbuch Informatik 1” `http://www.uvka.de/univerlag/volltexte/2005/73/pdf/Abeck_Sebastian.pdf`
- Universität Düsseldorf, G. Schurz: “Einführung in die Aussagen- und Prädikatenlogik” `http://thphil.phil-fak.uni-duesseldorf.de/index.php/filemanager/download/226/LogikEinfuehrung.pdf`

Index

Algorithmus, 24
Aussagenlogik
  klassisch, 7
Axiom, 4, 18

Disjunktion, 4
Distributivgesetz, 6

Explikat, 24

Faktoren, 23
Faktorisierung, 23
Formel, 4
  Atom-, 4
  geschlossen, 13
  Horn-, 10
Funktion, 4
  funktional vollständig, 7
  numerisch, 24

Interpretation, 4

Klausel, 4, 19
  -form, 19
  Horn-, 10
  leere, 19
Konjunktion, 4
Konklusion, 18
Konnektiv
  aussagenlogisch, 7
Konstante, 4

Literal, 4
  dual, 19
Logik, 4
  mathematische, 4

Multiset, 18

Normalform, 5
  disjunktive, 5
  konjunktive, 5
  Negations-, 6
  Pränex-, 21
  Skolem-, 21
Notation
  Infix-, 4
  Präfix-, 4

Operator, 4

Prädikat, 4
Prämisse, 18

Quantor, 4

Redundanzelimination, 9
Regelsystem, 4
rekursiv aufzählbar, 24
rekursiv entscheidbar, 24
Resolvent, 19, 20

Semantik, 4
Sequent, 18
Skolemisierung, 21, 22
Substitution, 15
Subsumtion, 19
Syntax, 4

Theorem, 4

Unifikator, 15
  allgemeiner, 15

Variable, 4
  -substitution, 14
  frei, 13
  gebunden, 13

Wahrheitswert, 4