Formale Grundlagen der Informatik 1
Kapitel 15
Normalformen
und
Hornformeln

Frank Heitmann
heitmann@informatik.uni-hamburg.de

30. Mai 2016

Zusammenfassung Syntax

Zusammenfassung **Syntax**:

- Motivation
- Definition der Syntax:
  - Alphabet, Junktor
  - Aussagesymbol, atomare Formel, komplexe Formel
  - Hauptoperator, Teilformel
  - Negation, Disjunktion, Konjunktion, Implikation, Biimplikation
- Strukturbäume
- strukturelle Induktion
- strukturelle Rekursion
- Grad und Tiefe einer Formel

Literaturhinweis

Siehe hierzu neben dem Buch von Schöning auch noch die ca. 10 Seiten im Skript.

Zusammenfassung Semantik (Grundlagen)

Zusammenfassung **Semantik** (Grundlagen):

- Belegung, Auswertung (einer Formel)
- Wahrheitstafeln, Wahrheitswerteverlauf
- erfüllende Belegung, falsifizierende Belegung, Modell
- kontingent, (allgemein-)gültig, unerfüllbar
- Tautologie, Kontradiktion
- $\mathcal{A} \models F$, $\mathcal{A} \not\models F$, $\models F$, $F \models$

Literaturhinweis

Siehe hierzu das Buch von Schöning!

Zusammenfassung

Zusammenfassung Folgerbarkeit und Äquivalenz

- Folgerbarkeit und Äquivalenz eingeführt
  - Nachweis mit Wahrheitstafeln
  - Nachweis ohne Wahrheitstafeln
  - Gegenbeispiel (mit und ohne Wahrheitstafeln)
- Sätze zur Folgerbarkeit und Äquivalenz
- Ersetzbarkeitstheorem (Beweis mit struktureller Induktion)

Jetzt:

- Normalformen KNF und DNF definieren
- Dazu: Literal, Klausel, duale Klausel
- Die Existenz von KNF und DNF beweisen
- und zwei Verfahren zur Herstellung kennenlernen

Folgerung und Äquivalenz (Wdh.)

Definition (Folgerung)

Eine Formel F **folgt** genau dann aus einer Formelmenge M, wenn jede Belegung, die Modell für M ist, auch Modell für F ist.
Notation: $M \models F$ bzw. $G \models F$, wenn $M = \{G\}$.

Definition (Äquivalenz)

Zwei Formeln F und G heißen **äquivalent** genau dann, wenn jede Belegung beiden Formeln den gleichen Wahrheitswert zuweist, wenn also $\mathcal{A}(F) = \mathcal{A}(G)$ für jede Belegung $\mathcal{A}$ gilt.
Notation: $F \equiv G$.

Einige Sätze

Satz (aus den Präsenzaufgaben)

1. Wenn $F \equiv G$ und $G \models$ gilt, dann gilt auch $F \models$
2. Wenn $\models F$ und $\models G$ gilt, dann gilt $F \equiv G$
3. $F \equiv G$ genau dann, wenn $\models F \Leftrightarrow G$

Beweis.

1. Sie $\mathcal{A}$ eine zu F und G passende Belegung. Da G eine Kontradiktion ist, gilt $\mathcal{A}(G) = 0$. Da ferner $F \equiv G$ gilt, gilt $\mathcal{A}(F) = \mathcal{A}(G)$, also auch $\mathcal{A}(F) = 0$. Damit ist keine Belegung Modell für F und folglich F eine Kontradiktion.
2. Wir zeigen, dass $\mathcal{A}(F) = \mathcal{A}(G)$ für jede Belegung $\mathcal{A}$ gilt. Sei dazu $\mathcal{A}$ eine zu F und G passende Belegung. Da sowohl F als auch G Tautologien sind, gilt stets $\mathcal{A}(F) = \mathcal{A}(G) = 1$ womit bereits alles gezeigt ist.
3. ...

Einige Sätze

$F \equiv G$ genau dann, wenn $\models F \Leftrightarrow G$

Beweis.

Gelte $F \equiv G$ und sei $\mathcal{A}$ eine zu F und G passende Belegung. Da wegen $F \equiv G$ stets $\mathcal{A}(F) = \mathcal{A}(G)$ gilt, folgt $\mathcal{A}(F \Leftrightarrow G) = 1$ sofort aus der Semantik von $\Leftrightarrow$ und folglich gilt $\models F \Leftrightarrow G$.
Sei umgekehrt $\models F \Leftrightarrow G$ und sei $\mathcal{A}$ eine zu F und G passende Belegung. Aus $\mathcal{A}(F \Leftrightarrow G) = 1$ folgt wegen der Semantik von $\Leftrightarrow$ sofort $\mathcal{A}(F) = \mathcal{A}(G)$ und damit $F \equiv G$. □

Ersetzbarkeitstheorem

Satz (Ersetzbarkeitstheorem)

Seien F und G äquivalente Formeln und sei H eine Formel mit (mindestens) einem Vorkommen der Formel F als Teilformel. Gehe H' aus H hervor, indem ein Vorkommen von F (in H) durch G ersetzt wird. Dann sind H und H' äquivalent.

Ergebnis

Wir dürfen dank des Ersetzbarkeitstheorems Teilformeln durch andere Formeln ersetzen, sofern diese zu der gewählten Teilformel äquivalent sind.

$$\neg A \Rightarrow \neg\neg B \equiv \neg A \Rightarrow B \equiv \neg\neg A \vee B \equiv A \vee B$$

Normalformen - Motivation

Wir wollen nun eine **Normalform** für aussagenlogische Formeln einführen, d.h. eine Form

- in die wir jede aussagenlogische Formel durch Äquivalenzumformungen bringen können und
- die eine praktische Form hat (z.B. für Berechnungen)

Dazu erst ein paar Begriffe ...

Normalformen - Begriffe

Definition

1. Ein **Literal** ist eine atomare Formel oder eine negierte atomare Formel.
2. Ein **positives Literal** ist eine atomare Formel, ein **negatives Literal** eine negierte atomare Formel.
3. Zwei Literale heißen **komplementär**, wenn sie positives und negatives Literal der gleichen atomaren Formel sind. Bspw. ist A das komplementäre Literal zu $\neg A$ und umgekehrt.
4. Literale und Disjunktionen von Literalen werden als **Klauseln** bezeichnet.
5. Literale und Konjunktionen von Literalen werden als **duale Klauseln** bezeichnet.

Normalformen - Begriffe 2

Definition

1. Eine Formel F ist in **konjunktiver Normalform** (KNF), wenn sie eine Konjunktion von Klauseln ist, also eine Konjunktion von Disjunktionen von Literalen, z.B.

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

2. Eine Formel F ist in **disjunktiver Normalform** (DNF), wenn sie eine Disjunktion von dualen Klauseln ist, also eine Disjunktion von Konjunktionen von Literalen, z.B.

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

Eigenschaften der KNF und DNF

Merkhilfe

KNF: $(\neg A \vee B \vee \neg C) \wedge B \wedge (\neg C \vee \neg B)$
DNF: $(A \wedge B) \vee (\neg A \wedge \neg C) \vee (B \wedge \neg A \wedge C)$

Satz

1. *Eine KNF ist gültig gdw. alle ihre Klauseln gültig sind gdw. in allen Klauseln mindestens ein Paar komplementäre Literale vorkommt.*
2. *Eine DNF ist unerfüllbar gdw. alle ihre dualen Klauseln unerfüllbar sind gdw. in allen dualen Klauseln mindestens ein Paar komplementäre Literale vorkommt.*
3. *Ein Erfüllbarkeitstest für DNFs ist effizient implementierbar (Laufzeit ist in P). (Für KNFs gilt dies wahrscheinlich nicht! SAT ist NP-vollständig.)*

KNF und DNF

Satz

Zu jeder Formel F gibt es (mindestens) eine konjunktive Normalform und (mindestens) eine disjunktive Normalform, d.h. es gibt Formeln K in konjunktiver Normalform und D in disjunktiver Normalform mit $F \equiv K \equiv D$.

Beweis.

Der Beweis ist wieder mittels struktureller Induktion möglich. Siehe *Logik für Informatiker* von Schöning.
Wir verfahren hier anders und geben eine Konstruktion an... □

KNF und DNF - Existenzbeweis

Wir konstruieren zu F die KNF K und die DNF D mittels Äquivalenzumformungen wie folgt:

1. Forme F so um, dass nur die Junktoren $\neg, \wedge$ und $\vee$ vorkommen.
2. Forme weiter so um, dass Negationen nur vor atomaren Formeln vorkommen.
3. Forme mittels der Distributivgesetze weiter so um, dass eine DNF bzw. KNF entsteht.

KNF und DNF - Existenzbeweis

Die Schritte im einzelnen:
Schritt 1. Forme F so um, dass nur die Junktoren $\neg, \wedge$ und $\vee$ vorkommen, indem die anderen Junktoren durch sie ausgedrückt werden. Z.B. kann $F \Rightarrow G$ durch $\neg F \vee G$ ersetzt werden und $F \Leftrightarrow G$ durch $(\neg F \vee G) \wedge (\neg G \vee F)$. Wir erhalten die Formel F_1.
Schritt 2. Wir formen F_1 durch wiederholte Anwendung von 'doppelte Negation' $(\neg\neg F \equiv F)$ und 'de Morgan' $(\neg(F \wedge G) \equiv \neg F \vee \neg G$ und $\neg(F \vee G) \equiv \neg F \wedge \neg G)$ so um, dass Negationen nur noch vor den Atomen vorkommen. Wir erhalten die Formel F_2.

KNF und DNF - Existenzbeweis

Schritt 3b. KNF. Wir formen F_2 durch wiederholte Anwendung des Distributivgesetzes in eine KNF um:

$$\begin{aligned}(F \vee (G \wedge H)) &\equiv ((F \vee G) \wedge (F \vee H)) \\ ((F \wedge G) \vee H) &\equiv ((F \vee H) \wedge (G \vee H))\end{aligned}$$

Schritt 3b. DNF. Wir formen F_2 durch wiederholte Anwendung des Distributivgesetzes in eine DNF um:

$$\begin{aligned}(F \wedge (G \vee H)) &\equiv ((F \wedge G) \vee (F \wedge H)) \\ ((F \vee G) \wedge H) &\equiv ((F \wedge H) \vee (G \wedge H))\end{aligned}$$

Verfahren für die Erstellung von KNF und DNF

1. Ersetze alle Teilformeln der Form
   - $(G \Leftrightarrow H)$ durch $(\neg G \vee H) \wedge (\neg H \vee G)$
     bzw. $(G \wedge H) \vee (\neg G \wedge \neg H)$ [Elimination von $\Leftrightarrow$]
   - $(G \Rightarrow H)$ durch $(\neg G \vee H)$ [Elimination von $\Rightarrow$]
2. Ersetze alle Teilformeln der Form
   - $\neg\neg G$ durch G [Doppelte Negation]
   - $\neg(G \wedge H)$ durch $(\neg G \vee \neg H)$ [de Morgan]
   - $\neg(G \vee H)$ durch $(\neg G \wedge \neg H)$ [de Morgan]
3. Um die KNF zu bilden ersetze alle Teilformeln der Form
   - $(F \vee (G \wedge H))$ durch $((F \vee G) \wedge (F \vee H))$ [Distributivität]
   - $((F \wedge G) \vee H)$ durch $((F \vee H) \wedge (G \vee H))$ [Distributivität]
4. Um die DNF zu bilden ersetze alle Teilformeln der Form
   - $(F \wedge (G \vee H))$ durch $((F \wedge G) \vee (F \wedge H))$ [Distributivität]
   - $((F \vee G) \wedge H)$ durch $((F \wedge H) \vee (G \wedge H))$ [Distributivität]

KNF und DNF - Existenzbeweis

Man beachte, dass

1. In jedem Schritt Äquivalenzumformungen vorgenommen werden. Die entstehenden Formeln sind also zur ursprünglichen äquivalent (Ersetzbarkeitstheorem).
2. Jeder Schritt terminiert:
   1. Im ersten Schritt gibt es nur endlich viele $\Rightarrow$ und $\Leftrightarrow$ zu ersetzen.
   2. Im zweiten Schritt 'rutschen' die Negationen stets eine Ebene tiefer, so dass nur endlich oft umgeformt werden kann.
   3. Im dritten Schritt 'rutschen' die Disjunktionen (bei der DNF) bzw. die Konjunktionen (bei der KNF) eine Ebene höher, so dass nur endlich oft umgeformt werden kann.
3. Am Ende nach Konstruktion eine DNF bzw. eine KNF steht.

Damit haben wir ein Verfahren, das *terminiert*, eine *äquivalente* Formel liefert, die zudem in DNF bzw. KNF ist. Damit ist das Verfahren korrekt.

Verallgemeinerte Äquivalenzen

Anmerkung

Oft ist es hilfreich mit *Verallgemeinerungen* der Distributivgesetze und der Regel von de Morgan zu arbeiten. Diese können nämlich auf größere Formeln verallgemeinert werden. Für de Morgan:

- $\neg(G \wedge H \wedge I) \equiv (\neg G \vee \neg H \vee \neg I)$
- $\neg(G \vee H \vee I) \equiv (\neg G \wedge \neg H \wedge \neg I)$

Und bei den Distributivgesetzen z.B.

- $(F \wedge I) \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H) \wedge (I \vee G) \wedge (I \vee H)$
- $(F \vee I) \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H) \vee (I \wedge G) \vee (I \wedge H)$

Dies ist dann weiter auf beliebig viele Teilformeln erweiterbar.

Beispiel

Wir erstellen eine KNF zu $(A \wedge (B \Rightarrow C)) \Rightarrow D$:

$$
\begin{aligned}
& (A \wedge (B \Rightarrow C)) \Rightarrow D && \text{Elimination von } \Rightarrow \\
\equiv\ & (A \wedge (\neg B \vee C)) \Rightarrow D && \text{Elimination von } \Rightarrow \\
\equiv\ & \neg(A \wedge (\neg B \vee C)) \vee D && \text{de Morgan} \\
\equiv\ & (\neg A \vee \neg(\neg B \vee C)) \vee D && \text{de Morgan} \\
\equiv\ & (\neg A \vee (\neg\neg B \wedge \neg C)) \vee D && \text{Doppelte Negation} \\
\equiv\ & (\neg A \vee (B \wedge \neg C)) \vee D && \text{Distributivität} \\
\equiv\ & ((\neg A \vee B) \wedge (\neg A \vee \neg C)) \vee D && \text{Distributivität} \\
\equiv\ & ((\neg A \vee B) \vee D) \wedge ((\neg A \vee \neg C) \vee D) && \text{Klammern} \\
\equiv\ & (\neg A \vee B \vee D) \wedge (\neg A \vee \neg C \vee D)
\end{aligned}
$$

Bemerkung

Zum Schluss (und auch zwischendurch) können Kommutativgesetze, Assoziativgesetze und Regeln wie Absorption, Idempotenz, Tautologieregeln, Kontradiktionsregeln und Komplementregeln angewendet werden, um die Formel zu vereinfachen.

Wahrheitstafelmethode

Mit obigen Verfahren kann **durch Äquivalenzumformungen** zu jeder gegebenen aussagenlogischen Formel F eine zu F äquivalente Formel in KNF bzw. DNF erstellt werden.

Eine weitere Möglichkeit eine DNF und eine KNF zu einer Formel zu konstruieren, geht mit **Wahrheitstafeln**.

Anmerkung

Hier ist das Problem, dass die Wahrheitstafeln sehr groß werden können und das Verfahren daher ineffizient ist. (Aber auch das Verfahren eben ist nicht stets effizient, z.B. nicht, wenn man DNFs in KNFs umwandelt!)

Wahrheitstafelmethode - DNF

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

Idee für die DNF?

Wahrheitstafelmethode - DNF

A	B	C	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\neg A \wedge B \wedge C$
1	0	0	1	$A \wedge \neg B \wedge \neg C$
1	0	1	0	
1	1	0	0	
1	1	1	1	$A \wedge B \wedge C$

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

Wahrheitstafelmethode - KNF

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

KNF ... ?

Wahrheitstafelmethode - KNF

A	B	C	F	
0	0	0	0	$A \vee B \vee C$
0	0	1	0	$A \vee B \vee \neg C$
0	1	0	0	$A \vee \neg B \vee C$
0	1	1	1	
1	0	0	1	
1	0	1	0	$\neg A \vee B \vee \neg C$
1	1	0	0	$\neg A \vee \neg B \vee C$
1	1	1	1	

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

Wahrheitstafelmethode - DNF

Herstellung der DNF:

1. Erstelle pro Zeile, die zu 1 ausgewertet wird, eine duale Klausel:
   1. Wird ein Aussagesymbole A zu 1 ausgewertet, nehme A in die Klausel
   2. Wird ein Aussagesymbole A zu 0 ausgewertet, nehme $\neg A$ in die Klausel
2. Verknüpfe alle so gewonnenen dualen Klauseln mit $\vee$
3. Gibt es keine Zeile, die zu 1 ausgewertet wird, nimm die Formel $A \wedge \neg A$

Wahrheitstafelmethode - KNF

Herstellung der KNF:

1. Erstelle pro Zeile, die zu 0 ausgewertet wird, eine Klausel:
   1. Wird ein Aussagesymbole A zu 1 ausgewertet, nehme $\neg A$ in die Klausel
   2. Wird ein Aussagesymbole A zu 0 ausgewertet, nehme A in die Klausel
2. Verknüpfe alle so gewonnenen Klauseln mit $\wedge$
3. Gibt es keine Zeile, die zu 0 ausgewertet wird, nimm die Formel $A \vee \neg A$

Wahrheitstafelmethode

Korrektheit

Bei der Wahrheitstafelmethode werden offensichtlich DNFs bzw. KNFs erstellt.
Diese sind zudem äquivalent zur ursprünglichen Formel, wie man sich schnell überlegt. So erzeugt man bei der DNF mit den einzelnen dualen Klauseln gerade Formeln, die genau an der Stelle der Zeile zu 1 (sonst 0en) ausgewertet werden. Durch die $\vee$-Verknüpfung hat man dann gerade eine zur ursprünglichen Formel äquivalente Formel. Bei der Erstellung der KNF hat zunächst die einzelne Klausel gerade an der Stelle der Zeile eine 0 (sonst 1en). Durch die $\wedge$-Verknüpfung hat man dann gerade wieder eine zur ursprünglichen Formel äquivalente Formel.

Motivation

Erfüllbarkeit zu prüfen ist bei

- KNFs schwierig (*NP*-vollständig (sehen wir noch))
- aussagenlogischen Formeln allgemein schwierig (folgt aus obigem)
- DNFs einfach

Aber DNFs herzustellen ist (wegen obigem) schwierig bzw. dauert lange.

Wichtige Anmerkung

Zusammengefasst ist Erfüllbarkeit zu testen bereits in der Aussagenlogik im Allgemeinen schwierig.

Motivation

Zwei Möglichkeiten:

Erste Möglichkeit

Man sucht nach **Einschränkungen der Aussagenlogik**, so dass die noch zur Verfügung stehenden Formeln weiterhin möglichst aussagekräftig sind, aber **Erfüllbarkeit schnell zu testen ist**.

Zweite Möglichkeit

Man such **Verfahren**, die zumindest **oft schnell** arbeiten.

Hornformeln sind eine Ausführung der ersten Möglichkeit

Hornformeln

Definition (Hornklauseln und Hornformeln)

1. Eine Klausel K ist genau dann eine **Hornklausel**, wenn K höchstens ein positives Literal enthält.
2. Eine Formel in KNF, bei der jede Klausel eine Hornklausel ist, ist eine **Hornformel**.

Beispiel

1. Hornklauseln: $A, \neg A \vee B, \neg A \vee B \vee \neg C, \neg A \vee \neg B$
2. Keine Hornklausel: $A \vee B, \neg A \vee B \vee C$
3. Hornformel: $A \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B \vee \neg C)$

Typen von Hornklauseln

Es gibt drei Typen von Hornklauseln. Für alle kann eine **Implikationsschreibweise** benutzt werden:

1. K enthält negative Literale und genau ein positives Literal (Regeln):

$$\begin{aligned} K &= \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i \vee A_k \equiv (\neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i) \vee A_k \\ &\equiv \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \vee A_k \equiv (A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow A_k \end{aligned}$$

2. K enthält nur negative Literale (Beschränkungen):

$$\begin{aligned} K &= \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i \equiv (\neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i) \vee \bot \\ &\equiv \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \vee \bot \equiv (A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow \bot \end{aligned}$$

3. K enthält nur genau ein positives Literal (Fakten):

$$K = A_1 \equiv \neg\top \vee A_1 \equiv \top \Rightarrow A_1$$

Zu Hornformeln

Definition

1. Eine Hornklausel, in der kein Aussagensymbol mehrfach aufritt, heißt reduziert.
2. Eine Hornformel heißt reduziert genau dann, wenn alle Klauseln reduziert und verschieden sind.

Satz

Für jede (nicht allgemeingültige) Hornformel existiert eine äquivalente reduzierte Hornformel.

Satz

Es gibt keine zu $A \vee B$ äquivalente Hornformel.

Zu Hornformeln

- Hornformeln sind also in gewisser Weise recht eingeschränkt.
- Dennoch geht mit ihnen ziemlich viel ($\Rightarrow$ Logik-Programmierung).
- Daher lohnt es sich, sich Gedanken über eine effizienten Erfüllbarkeitstest zu machen.

Ideen?

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

$$(A \Rightarrow C) \wedge (\top \Rightarrow A) \wedge ((A \wedge C) \Rightarrow \bot)$$

Markierungsalgorithmus für Hornformeln

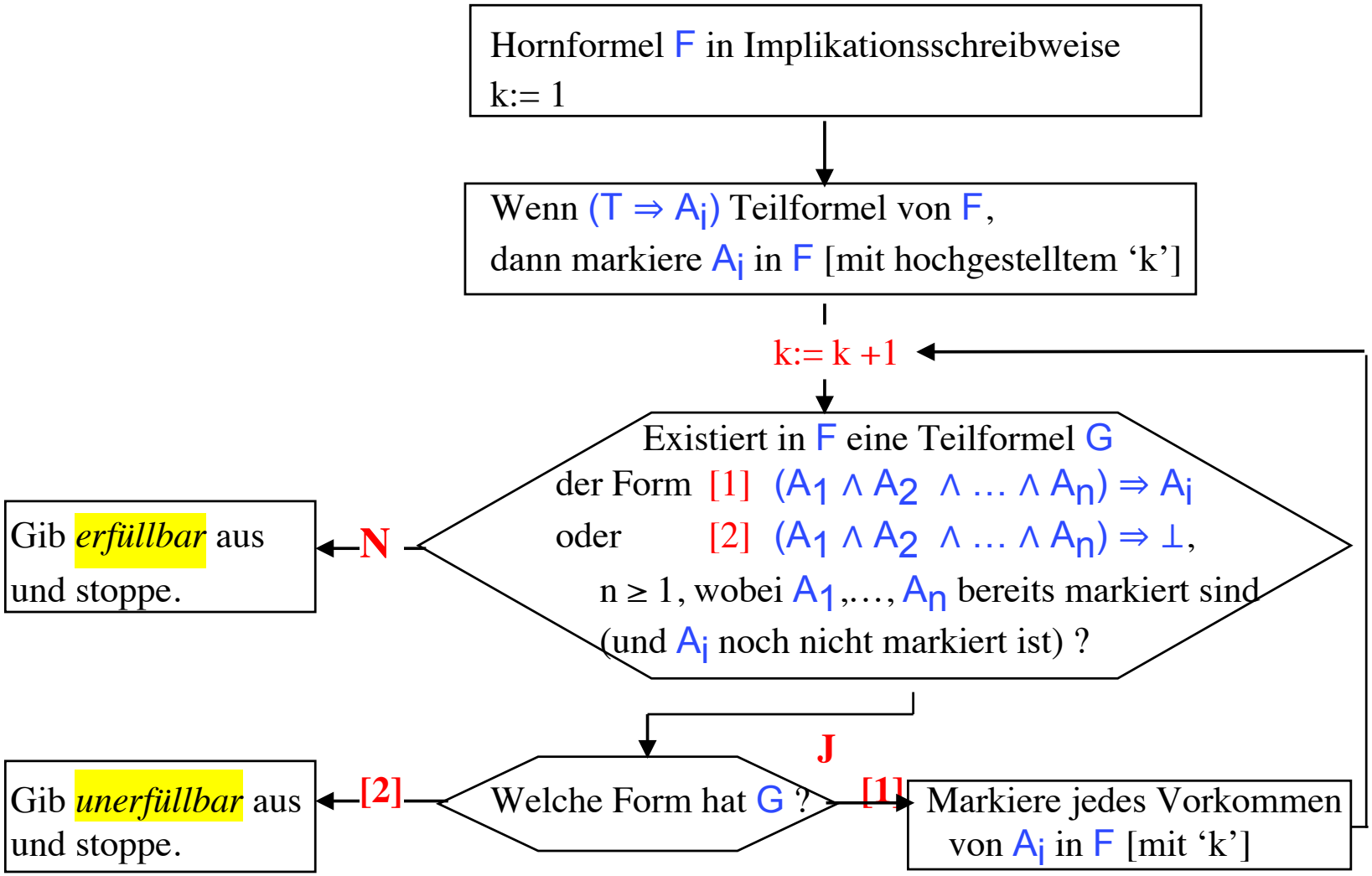
Grundidee: Markiere Aussagesymbole, die in jeder erfüllenden Belegung wahr sein müssen.

1. Versehe jedes Vorkommen eine Aussagensymbols A_i in F mit einer Markierung, falls es in F eine Teilformel der Form $(\top \Rightarrow A_i)$ gibt.
2. WHILE es gibt in F eine Teilformel G der Form
   [1] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ oder der Form
   [2] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \bot$,
   wobei $A_1, \ldots, A_n, n \geq 1$ bereits markiert sind (und A_i noch nicht)
   DO
   IF G hat die Form [1]
   THEN markiere jedes Vorkommen von A_i in F
   ELSE gib "unerfüllbar" aus und stoppe.
3. Gib "erfüllbar" aus und stoppe.

Die gefundene erfüllende Belegung ist:

$$\mathcal{A}(A_i) = \begin{cases} 1, & \text{falls } A_i \text{ eine Markierung besitzt} \\ 0, & \text{sonst} \end{cases}$$

Markierungsalgorithmus (Ablaufdiagramm)



Beispiel

$$
\begin{aligned}
& (\neg A \vee \neg B \vee E) \wedge C \wedge A \wedge (\neg C \vee B) \wedge (\neg A \vee D) \\
& \wedge (\neg C \vee \neg D \vee A) \wedge \neg E \wedge \neg G \\
\equiv \quad & ((A \wedge B) \Rightarrow E) \wedge (\top \Rightarrow C) \\
& \wedge (\top \Rightarrow A) \wedge (C \Rightarrow B) \\
& \wedge (A \Rightarrow D) \wedge ((C \wedge D) \Rightarrow A) \\
& \wedge (E \Rightarrow \bot) \wedge (G \Rightarrow \bot)
\end{aligned}
$$

Markierungen:

- 1. Runde: A und C
- 2. Runde: B (wegen $C \Rightarrow B$) und D (wegen $A \Rightarrow D$)
- 3. Runde: E (wegen $(A \wedge B) \Rightarrow E$))
- 4. Runde: **unerfüllbar** (wegen $(E \Rightarrow \bot)$)

Beispiel

$$(\neg A \vee B) \wedge A \wedge (\neg B \vee \neg A \vee C) \wedge (\neg D \vee C)$$
$$\equiv \quad (A \Rightarrow B) \wedge (\top \Rightarrow A) \wedge ((A \wedge B) \Rightarrow C) \wedge (D \Rightarrow C)$$

Markierungen:

- 1. Runde: A
- 2. Runde: B (wegen $(A \Rightarrow B)$)
- 3. Runde: C (wegen $((A \wedge B) \Rightarrow C)$)
- 4. Runde: Keine weiteren Markierungen.

Erfüllende Belegung $\mathcal{A}$ mit $\mathcal{A}(A) = \mathcal{A}(B) = \mathcal{A}(C) = 1$ und $\mathcal{A}(D) = 0$.

Korrektheit

Satz

Der Markierungsalgorithmus ist korrekt, d.h. wenn F eine unerfüllbare Hornformel ist, dann gibt der Algorithmus unerfüllbar aus und wenn F erfüllbar ist, so gibt der Algorithmus eine erfüllende Belegung aus.

Markierungsalgorithmus für Hornformeln

Grundidee: Markiere Aussagesymbole, die in jeder erfüllenden Belegung wahr sein müssen.

1. Versehe jedes Vorkommen eine Aussagensymbols A_i in F mit einer Markierung, falls es in F eine Teilformel der Form $(\top \Rightarrow A_i)$ gibt.
2. WHILE es gibt in F eine Teilformel G der Form
   [1] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ oder der Form
   [2] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \bot$,
   wobei $A_1, \ldots, A_n, n \geq 1$ bereits markiert sind (und A_i noch nicht)
   DO
   IF G hat die Form [1]
   THEN markiere jedes Vorkommen von A_i in F
   ELSE gib "unerfüllbar" aus und stoppe.
3. Gib "erfüllbar" aus und stoppe.

Die gefundene erfüllende Belegung ist:

$$\mathcal{A}(A_i) = \begin{cases} 1, & \text{falls } A_i \text{ eine Markierung besitzt} \\ 0, & \text{sonst} \end{cases}$$

Korrektheit

Satz

Der Markierungsalgorithmus ist korrekt, d.h. wenn F eine unerfüllbare Hornformel ist, dann gibt der Algorithmus unerfüllbar aus und wenn F erfüllbar ist, so gibt der Algorithmus eine erfüllende Belegung aus.

Beweis.

Eben mündlich; zum Nachlesen im Buch von Schöning. □

Ergebnis

Der Markierungsalgorithmus ist sehr effizient (gemessen an der Anzahl der atomaren Formeln in Linearzeit), aber nicht jede Formel ist als Hornformel darstellbar.

Nächstes Mal lernen wir eine syntaxbasierte Möglichkeit kennen, Formeln auf Unerfüllbarkeit zu testen: Die Resolution.

Zusammenfassung

- Folgerung
- Äquivalenzen
- Nachweis und Widerlegung mit und ohne Wahrheitstafeln
- Sätze über Folgerung und Äquivalenzen
- Ersetzbarkeitstheorem
- Literal, Klauseln, duale Klauseln, KNF, DNF
- Herstellung von DNF und KNF
  - durch Äquivalenzumformungen
  - mit Wahrheitstafeln
- Hornformeln
- Markierungsalgorithmus für Hornformeln