Grundlagen Theoretischer Informatik I

SoSe 2011 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Grundlagen Theoretischer Informatik I Gesamtübersicht

- Organisatorisches; Einführung
- Logik & Beweisverfahren
- Mengenlehre
- reguläre Sprachen

Organisatorisches

Vorlesungen FR 10.10-11.50 im HS 13
FR 12.30-13.50 im HS 13

Übungsbetrieb BEGINN: FR 29.04.2011

Dozentensprechstunde DO 13-14 in meinem Büro H 410 (4. Stock)

Mitarbeitersprechstunde (Daniel Meister) DO 13-14 H 413

Tutorensprechstunde MO 13-14 H 407

aus der letzten Vorlesung

Aussagen und Aussageformen

Verknüpfung von Aussagen (Junktoren)

einfache Rechenregeln / Tautologien

Syntax und Semantik

skizzierte Anwendung: Programmverifikation

... heute ...

Normalformen

etwas Prädikatenlogik

skizzierte Anwendung: Logikprogrammierung / Datenbanken

Normalformen

siehe: Uwe Schöning: Logik für Informatiker, Spektrum Verlag

Erinnerung: Da Konjunktion und Disjunktion assoziative Operationen sind, kann man diese auch als mehrstellige Operatoren auffassen.
Ein *Literal* ist eine atomare Aussage oder deren Negation.
Eine Formel F ist in *konjunktiver Normalform* (KNF), wenn sie eine Konjunktion von Disjunktionen von Literalen $L_{ij} \in \{A_1, A_,2 \ldots\} \cup \{\neg A_1, \neg A_2, \ldots\}$ ist, d.h.:

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

Eine Formel F ist in *disjunktiver Normalform* (DNF), wenn sie eine Disjunktion von Konjunktionen von Literalen $L_{ij} \in \{A_1, A_,2 \ldots\} \cup \{\neg A_1, \neg A_2, \ldots\}$ ist, d.h.:

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

Beispiele

1. Atomare Formeln und allgemeiner Literale sind sowohl in KNF als auch in DNF.
2. $A_1 \wedge (\neg A_2 \vee A_3)$ ist in KNF, aber nicht in DNF.
3. Distributivgesetz $\rightsquigarrow (A_1 \wedge (\neg A_2 \vee A_3)) \equiv ((A_1 \wedge \neg A_2) \vee (A_1 \wedge A_3))$.

Die unter 2. angegebene Formel lässt sich also in eine äquivalente Formel in DNF überführen.

Gilt dies allgemein?

Normalformen: Der Satz

Satz: Für jede Formel F gibt es eine äquivalente Formel F' in KNF und eine äquivalente Formel F'' in DNF.

Beweis: (Induktion über den Formelaufbau on F)

Induktionsanker: Ist F atomar, so setze $F' = F'' = F$.

Induktionsannahme: Zu G (bzw. H) gibt es äquivalente Formeln G' (bzw. H') in KNF und G'' (bzw. H'') in DNF.

Induktionsschritt: Wir unterscheiden drei Fälle:

1. $F = \neg G$. Betrachte $G' \equiv G$ in KNF, d.h., $G' = \bigwedge_{i=1}^{n}(\bigvee_{j=1}^{m_i} L_{ij})$.
Ersetzbarkeitstheorem $\rightsquigarrow F \equiv \neg G' = \neg(\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{m_i} L_{ij}))$.
De Morgan liefert: $\neg(\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{m_i} L_{ij})) \equiv (\bigvee_{i=1}^{n} \neg(\bigvee_{j=1}^{m_i} L_{ij})) \equiv (\bigvee_{i=1}^{n}(\bigwedge_{j=1}^{m_i} \neg L_{ij}))$.

Setze $\overline{L_{ij}} = \begin{cases} A_k, & \text{falls } L_{ij} = \neg A_k \\ \neg A_k, & \text{falls } L_{ij} = A_k \end{cases}$

Doppelnegationsgesetz $\leadsto F \equiv (\bigvee_{i=1}^{n}(\bigwedge_{j=1}^{m_i} \overline{L_{ij}})) =: F''$.

Entsprechend erhält man aus G'' in DNF die Formel F' in KNF.

2. $F = (G \vee H)$.

F'' erhält man sofort aus G'' und H'' (Ersetzbarkeitstheorem und Assoziativität).

Ersetzbarkeitstheorem $\leadsto F \equiv (G' \vee H')$.

Dabei ist $G' = \bigwedge_{i=1}^{n} G_i$ und $H' = \bigwedge_{j=1}^{k} H_j$ für Disjunktionen von Literalen G_i bzw. H_i.

Distributivgesetz $\leadsto F \equiv (\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{k}(G_i \vee H_j)) =: F'$ in DNF.

3. $F = (G \wedge H)$ analog.

Algorithmische Sicht: Überführen einer Formel F in KNF:

1. Ersetze in F jedes Vorkommen einer Teilformel der Bauart

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

bis keine solche Teilformel mehr vorkommt.

2. Ersetze jedes Vorkommen einer Teilformel der Bauart

$$\begin{array}{rl} (J \vee (G \wedge H)) & \text{durch } ((J \vee G) \wedge (J \vee H)), \\ ((G \wedge H) \vee J) & \text{durch } ((J \vee G) \wedge (J \vee H)), \end{array}$$

bis keine solche Teilformel mehr vorkommt.

Nur Schritt 2 muss angepasst werden für einen DNF-Algorithmus.

Beispiel $F = (\neg B \wedge (\neg A \implies \neg C))$.

Erst einmal muss die Implikation mit Oder und Negation ausgedrückt werden:

$F \equiv (\neg B \wedge (\neg\neg A \vee \neg C))$.

Nun ist die Doppelnegation zu eliminieren:

$F \equiv (\neg B \wedge (A \vee \neg C))$.

Diese Formel ist in KNF.

Für die DNF muss man noch das Distributivgesetz nutzen:

$F \equiv (\neg B \wedge A) \vee (\neg B \wedge \neg C))$.

Entwickle Wahrheitstafel an der Tafel:

$\neg A$	$\neg B$	$\neg C$	$(\neg A \implies \neg C)$	F
w	w	w	w	w
w	w	f	f	f

Algorithmische Sicht: Überführen einer Formel F in KNF:

Liegt eine Wahrheitstafel für eine Formel F vor, so kann man daraus eine KNF bzw. DNF für F sofort ablesen.
a) DNF: Jede Zeile mit Wahrheitswert w bestimmt ein Disjunktionsglied.
Seine Literale ergeben sich wie folgt:
Ist der A_i-Eintrag w, so nimm Literal A_i, andernfalls $\neg A_i$.

b) KNF: Jede Zeile mit Wahrheitswert f bestimmt ein Konjunktionsglied.
Seine Literale ergeben sich wie folgt:
Ist der A_i-Eintrag f, so nimm Literal A_i, andernfalls $\neg A_i$.

Beispiel

A	B	C	F	DNF	KNF
f	f	f	w	$(\neg A \wedge \neg B \wedge \neg C)$	
f	f	w	f		$(A \vee B \vee \neg C)$
f	w	f	f		$\wedge(A \vee \neg B \vee C)$
f	w	w	f		$\wedge(A \vee \neg B \vee \neg C)$
w	f	f	w	$\vee(A \wedge \neg B \wedge \neg C)$	
w	f	w	w	$\vee(A \wedge \neg B \wedge C)$	
w	w	f	f		$\wedge(\neg A \vee \neg B \vee C)$
w	w	w	f		$\wedge(\neg A \vee \neg B \vee \neg C)$

Hinweise zu KNF / DNF

— Die “Normalformen” sind nicht eindeutig.
Beispielsweise wäre $(\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B)$ eine “kürzere” DNF für F auf letzter Folie. Eine noch kürzere finden Sie auf einer späteren Folie!

— Wichtige Anwendung der Aussagenlogik: Entwicklung von Schaltkreisen.
Spezielle Form: PLAs (Programmable Logic Arrays)
Normalform berücksichtigt Signallaufzeiten
Näheres erfahren Sie in der Technischen Informatik

Prädikatenlogik — Eine Einführung

Erinnerung: Aussageformen $=$ Prädikate

Beispiel: $P(x) := (x > 3)$ oder $Q(x, y) := (x = y + 3)$

(mit einem Zahluniversum, das 3 enthält auf wo $>$ und $+$ definiert sind)

$P(4)$ ist wahr, $P(2)$ ist falsch.

Für jede *Belegung* der Leerstelle x mit einem Wert aus dem gewählten Universum liefert $P(x)$ einen Wahrheitswert, denn damit wird $P(x)$ zu einer Aussage, z.B. $P(2)$ ist die Aussage $2 > 3$.

Prädikatenlogik "$=$" Aussagenlogik, angereichert mit Quantoren.

Universelle Quantifizierung / Allquantifizierung

Beispiel: Durch U.Q. wird aus $P(x) := x > 3$ die Aussage:
$\forall x P(x)$, in Worten: *Für alle x gilt* $P(x)$.
Ist diese Aussage wahr oder falsch ?

Abhängig vom zugrunde liegenden Universum X !
X: alle ganzen Zahlen $\rightsquigarrow$ $\forall x P(x)$ ist falsch.
X: alle natürlichen Zahlen $\rightsquigarrow$ $\forall x P(x)$ ist falsch.
X: alle Quadratzahlen größer Eins $\rightsquigarrow$ $\forall x P(x)$ ist wahr.
X: alle komplexen Zahlen $\rightsquigarrow$ $\forall x P(x)$ ist undefiniert.

Universelle Quantifizierung Jargon:

$\forall$: *Allquantor*
$\forall x$: x ist die *quantifizierte Variable*

Ist das Universum X endlich, d.h., ist $X = \{x_1, \ldots, x_r\}$ (Hinweis: Mengenlehre in den folgenden Vorlesungen), so gilt:

$$\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_r).$$

Dann schreibt man auch: $\bigwedge_x P(x)$.

Existentielle Quantifizierung

Beispiel: Durch E.Q. wird aus $P(x) := x > 3$ die Aussage:
$\exists x P(x)$, in Worten: *Es gibt ein* x *mit* $P(x)$.
Ist diese Aussage wahr oder falsch ?

Abhängig vom zugrunde liegenden Universum X !
X: alle ganzen Zahlen $\rightsquigarrow$ $\exists x P(x)$ ist wahr.
X: alle natürlichen Zahlen $\rightsquigarrow$ $\exists x P(x)$ ist wahr.
X: alle negativen Zahlen $\rightsquigarrow$ $\exists x P(x)$ ist falsch.
X: alle komplexen Zahlen $\rightsquigarrow$ $\exists x P(x)$ ist undefiniert (?).

Existentielle Quantifizierung Jargon:

$\exists$: *Existenzquantor*
$\exists x$: x ist die *quantifizierte Variable*

Ist das Universum X endlich, d.h., ist $X = \{x_1, \ldots, x_r\}$ (Hinweis: Mengenlehre in den folgenden Vorlesungen), so gilt:

$$\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee \cdots \vee P(x_r).$$

Dann schreibt man auch: $\bigvee_x P(x)$.

Wann liefern quantifizierte Aussageformen den Wahrheitswert “falsch” ?

Beispiel: Wir haben gesehen:
Für das Universum X aller natürlichen Zahlen gilt:
$\forall x (x > 3)$ ist falsch.

WARUM ?

Wir können ein *Gegenbeispiel* liefern, z.B. $x = 2$.

Allgemein gilt: $\forall x P(x)$ ist falsch gdw. $\exists x \neg P(x)$ ist wahr.
Satz: $\neg(\forall x P(x)) \equiv \exists x \neg P(x)$. (de Morgan)

Wann liefern quantifizierte Aussageformen den Wahrheitswert "falsch" ?

Beispiel: Wir haben gesehen:
Für das Universum X aller negativen ganzen Zahlen gilt:
$\exists x(x > 3)$ ist falsch.

WARUM ?

Wir können kein Beispiel finden, d.h.: $\forall x(x \leq 2)$ ist wahr.

Allgemein gilt: $\exists x P(x)$ ist falsch gdw. $\forall x \neg P(x)$ ist wahr.
Satz: $\neg(\exists x P(x)) \equiv \forall x \neg P(x)$. (de Morgan)

Aussageformen mit mehreren Leerstellen

Wir können dann über einzelne Variablen quantifizieren (*gebundene Variable*); die nicht quantifizierten Variablen heißen auch *frei*.

Beispiel: $P(x, y) := (x^2 + y = 1)$.

$\exists x P(x, y)$ ist Aussageform mit Leerstelle y

$\forall y P(x, y)$

$\forall x \forall y P(x, y)$

$\forall x \exists y P(x, y)$ oder $\forall y \exists x P(x, y)$

$\exists x \forall y P(x, y)$ oder $\exists y \forall x P(x, y)$

$\exists x \exists y P(x, y)$

Wann ist eine jede der Aussagen wahr, wann falsch ?

Achtung: **“$\forall x \exists y \neq \exists y \forall x$”**

Beispiel: Eine reelle Zahl a heißt *Grenzwert* einer Folge (a_n) reeller Zahlen gdw.
$\forall \epsilon > 0 \exists n_0 \forall n \geq n_0 (|a_n - a| < \epsilon)$.
In Worten: Zu jedem $\epsilon > 0$ gibt es ein n_0, so dass für alle n . . .

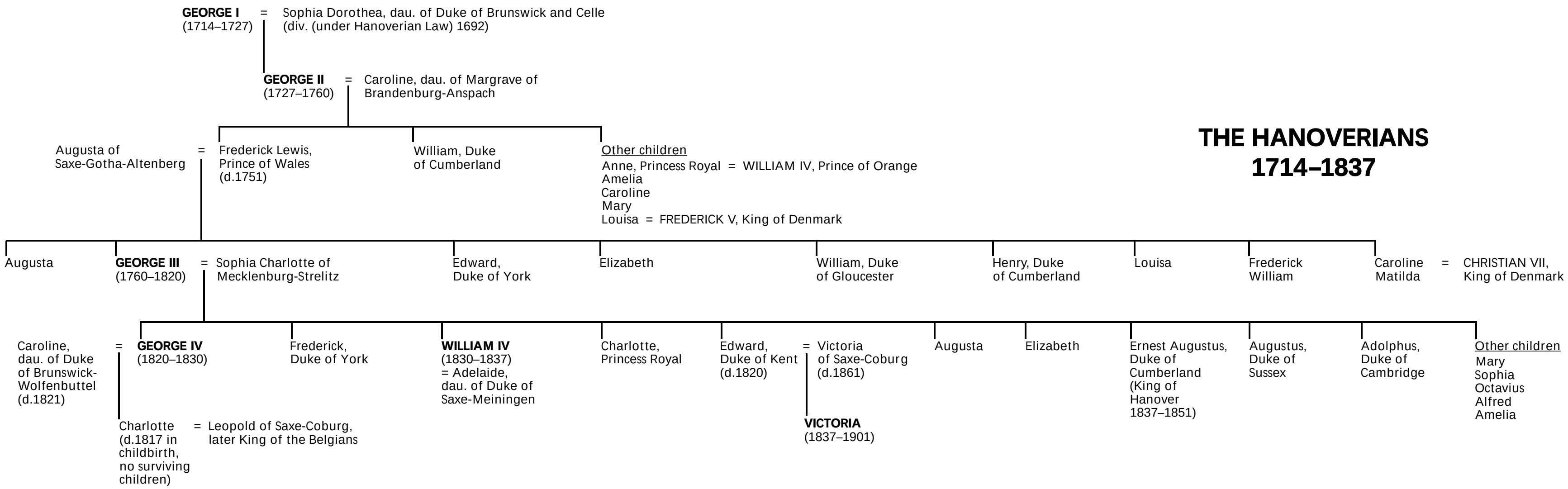
Vertauschen der Quantoren liefert die Aussage:
$\exists n_0 \forall \epsilon > 0 \forall n \geq n_0 (|a_n - a| < \epsilon)$.
In Worten: Es gibt ein n_0, so dass für alle $\epsilon > 0$ und alle n . . .
Was bedeutet das ?!

Hinweis: Mit der zweiten Aussage gilt auch die erste !

Anwendung: Wissensbasierte Systeme

Genealogie der englischen Königsfamilie ab George 1

eine mögliche Darstellung: ein “Stammbaum”



Anwendung: Wissensbasierte Systeme

Genealogie der englischen Königsfamilie ab Georg 1 als *Faktenbasis*

```
Elternteil(george1,george2)
Elternteil(george3,george4)
Elternteil(george3,william4)
Elternteil(edward,victoria)
Elternteil(victoria,edward7)
Elternteil(edward7,george5)
Elternteil(george5,edward8)
Elternteil(george5,george6)
Elternteil(george6,elizabeth2)
Elternteil(victoria,alice)
Elternteil(alice,victoriaalberta)
Elternteil(victoriaalberta,alicemountbatten)
Elternteil(alicemountbatten,philip)
```

Anwendung: Genealogie der englischen Königsfamilie: Wer mit wem ?

```
Gattin(sophie,george1)
Gattin(wilhelmina,george2)
Gattin(charlotte,george3)
Gattin(caroline,george4)
Gattin(adelaide,william4)
Gattin(victoria,albert)
Gattin(alexandra,edward7)
Gattin(victoriamaria,george5)
Gattin(elizabethQM,george6)
Gattin(elizabeth2,philip)
```

Anfragen in Anlehnung an PROLOG (Programming in Logic)

(a) `?- Gattin(elizabeth2,philip)`
liefert JA, denn der Eintrag wird in der Faktenbasis gefunden.

(b) `?- Elternteil(sophie,george2)`
liefert NEIN, denn der Eintrag wird NICHT in der Faktenbasis gefunden.
ACHTUNG: Die angefragte Aussage ist wahr, d.h., die Antwort "stimmt eigentlich nicht."
Die *Annahme einer geschlossenen Welt* (closed world assumption CWA) ist eines der Probleme "intelligenter Systeme".

(c) `?- Weiblich(caroline)`
liefert NEIN, denn das Prädikat `Weiblich` ist dem System nicht bekannt.

(d) `?- Gattin(philip,elizabeth2)`
liefert NEIN, da die Argumente nicht vertauschbar sind.

Regeln machen unser System “schlauer”:

Zu (c): Wir können ein Prädikat `Weiblich` einführen und dann sagen:

Wenn es x, y gibt mit $\texttt{Gattin}(x, y)$, **dann** gilt $\texttt{Weiblich}(x)$.

Zu (d) Wir können ein Prädikat `Gatte` einführen:

Wenn es x, y gibt mit $\texttt{Gattin}(x, y)$, **dann** gilt $\texttt{Gatte}(y, x)$.

Alternativ möchten wir vielleicht ein Prädikat `Verheiratet`, gegeben durch:

Wenn es x, y gibt mit $\texttt{Gattin}(x, y)$ oder mit $\texttt{Gatte}(x, y)$, **dann** gilt $\texttt{Verheiratet}(x, y)$.

Damit könnten wir (aber stimmt dies wirklich ?) auch (b) “lösen”:

Wenn es x, y, z gibt mit $\texttt{Verheiratet}(x, y)$ und mit $\texttt{Elternteil}(y, z)$, **dann** gilt $\texttt{Elternteil}(x, z)$.

Zum Abschluss: Noch ein Rätsel ...

Ist der folgende Schluss logisch korrekt ?!

(A) Weder Samson noch Freunde von Samson schießen ein Tor.
(B) Entweder schießt Samson ein Tor, oder Johannes schießt ein Tor.
(C) Also ist Johannes kein Freund von Samson.

Formalisierung:
Das Universum X ist eine Fußballmannschaft, aus der namentlich $S =$Samson und $J =$Johannes bekannt sind.
$T(x)$ ist eine Aussageform mit der Bedeutung: x schießt ein Tor.
$F(x, y)$ ist eine Aussageform mit der Bedeutung: x und y sind Freunde.

(A) $\neg(T(S)) \wedge \forall x(F(x, S) \Rightarrow \neg T(x))$
(B) $T(S) \vee T(J)$ (alternative Interpretation: ausschließendes Oder)
(C) $\neg F(J, S)$

Wir haben zu zeigen: Aus (A) und (B) folgt (C).

Zum Abschluss: Noch ein Rätsel . . .
(A) $\neg(T(S)) \wedge \forall x(F(x, S) \Rightarrow \neg T(x))$
(B) $T(S) \vee T(J)$ (alternative Interpretation: ausschließendes Oder)
(C) $\neg F(J, S)$

Als Hilfsaussage zeigen wir: Aus (A) und (B) folgt $T(J)$.
Aus (A) folgt durch Vereinfachung: $\neg T(S)$.
Zusammen mit (B) gilt daher: $\neg T(S) \wedge (T(S) \vee T(J))$.
Das Distributivgesetz liefert: $(\neg T(S) \wedge T(S)) \vee (\neg T(S) \wedge T(J))$.
Identitätsgesetz für Oder $\rightsquigarrow \neg T(S) \wedge T(J)$.
Durch Vereinfachung folgt $T(J)$.

Aus dem zweiten Teil der Aussage (A) folgt durch *Spezialisierung*:
$F(J, S) \Rightarrow \neg T(J)$.
Umkehrschluss (und doppelte Negation) liefert: $T(J) \Rightarrow \neg F(J, S)$.
Mit der Hilfsaussage folgt durch modus ponens die Behauptung.

Noch eine Knobelaufgabe: Gespensterknobelei in der Eisdiele 

Das kleine Gespenst Green Spuk hat heute Geburtstag und will mit seinen Freunden Red Spuk und Yellow Spuk in der Eisdiele feiern.
Jedes Gespenst hält ein Eishörnchen in der Hand und wartet nun auf sein Lieblingseis.
Green Spuk ruft: „Wir haben alle unterschiedliches Lieblingseis!“
Red Spuk wispert: „Ich mag Erdbeereis!“
Yellow Spuk piepst: „Ich mag kein Erdbeereis und kein Apfeleis!“

Die entsprechenden Dateien finden Sie original hier.

Aufgabe: Wie können wir das Problem modellieren und lösen ?