SAT-Solving und Anwendungen

Einführung / Aussagenlogik

Prof. Dr. Wolfgang Küchlin
Dipl. Inform. Christoph Zengler

Universität Tübingen

22. April 2009

Organisatorisches

- Vorlesung: 2-stündig (Mi. 10.00 - 12.00)
  - Anrechenbar für Theorie **oder** Praxis mit 4 LP
- Übung: 2-stündig (Termin nach Absprache)
  - Benotete Übungsblätter
  - Abwechselnd Theorie- und Praxis
  - Theorie: Einüben der theoretischen Konzepte aus der VL
  - Praxis: Bauen eines kleinen SAT-Solvers in Java über das Semester
- **Klausur**
  - **Mid-Term Klausur:** 10. Juni 2009 (nach Pfingstferien)
  - **Final Klausur:** 22. Juli 2009 (letzte Vorlesungswoche)
- **Note:** 30% Mid-Term Klausur + 30% Final Klausur + 40% Übung

Logischer Formalismus

Syntax

Wie werden Formeln gebildet?

- klassisch-mathematisch: bestimmte ausgezeichnete Zeichenreihen
  - (typisch: induktive Definitionen)
- informatisch: Sprache einer Grammatik

Semantik

Was ist die Bedeutung einer Formel?

- Allgemein: Abbildung in einen (bekannten) Semantik-Bereich
- Hier: Semantik-Bereich $\mathbb{B} = \{\mathbf{T}, \mathbf{F}\}$ der Boole'schen Wahrheitswerte

Kalkül

Wie kann der Wahrheitswert einer Formel ausgerechnet werden?

- Inferenzregeln, Algorithmen

Syntax

- Bestandteile von Formeln:
  - Aussagenvariablen aus einer unendlichen Menge $\mathcal{P}$ von Variablen: Platzhalter für beliebige (atomare) Aussagen, wie z.B. "5 ist eine Primzahl", "Eine Woche hat 7 Tage", "$4 = 2$" etc.
  - Konstanten $\top$, $\bot$: Zur Repräsentation der wahren bzw. falschen Aussage.
  - Operatoren ($\wedge$ und / $\vee$ oder / $\neg$ nicht / $\Rightarrow$ Implikation / $\Leftrightarrow$ Äquivalenz / $\oplus$ xor): zur Bildung komplexer Formeln (zus.-gesetzte Aussagen)
  - Hilfssymbole: Klammern
- Menge $\mathcal{F}$ aller gültigen Formeln ist rekursiv definiert:
  - Jede Aussagenvariable $x \in \mathcal{P}$ ist in $\mathcal{F}$
  - Die Konstanten $\top$ und $\bot$ sind in $\mathcal{F}$
  - Sind P und Q in $\mathcal{F}$, so sind auch $(\neg P), (P \wedge Q), (P \vee Q), (P \Rightarrow Q), (P \Leftrightarrow Q), (P \oplus Q)$ in $\mathcal{F}$

Syntax (ctd.)

- (EBNF)-Grammatik für Formeln: Übung.
- Zur Einsparung von Klammern:
  Priorität der Operatoren (abnehmend): $\neg, \wedge, \vee, \oplus, \Rightarrow, \Leftrightarrow$
- Literal: positives oder negatives Vorkommen einer Aussagenvariable
  $\{x, \neg x\}$
- $var(P)$: Menge der Variablen, die in der Formel P vorkommen

Beispiel (Prioritäten)

$x \wedge y \vee \neg z \Leftrightarrow x \oplus \neg z \wedge \neg w$ ist klammerfreie Schreibweise für:
$((x \wedge y) \vee (\neg z)) \Leftrightarrow (x \oplus ((\neg z) \wedge (\neg w)))$

Syntax - Beispiele

Konvention:

- *Aussagenvariablen: Kleinbuchstaben* $x, y, z, ..., x_1, x_2, ...$
- *Meta-Symbole zur Bezeichnung beliebiger Formeln: Großbuchstaben* $P, Q, R, ..., P_1, P_2, ...$

Beispiel (Gültige Formeln)

- x
- $x \vee y$
- $(x \vee y) \wedge (y \oplus z) \Rightarrow y$

Beispiel (Formel-Muster)

- $(x \vee y) \wedge (y \oplus z) \Rightarrow P$
- $var((x \vee y) \wedge (y \oplus z) \Rightarrow P) = \{x, y, z\} \cup var(P)$

Semantik

- Wie wird der Wahrheitsgehalt einer Formel bestimmt?
- Menge der Booleschen Wahrheitswerte: $\mathbb{B} = \{\mathbf{T}, \mathbf{F}\}$
- Dazu: Belegung der Aussagenvariablen mit **T** (wahr) oder **F** (falsch)
  - Variablenbelegung einer Variable x ist Funktion $\nu_0 : \mathcal{P} \to \mathbb{B}$
  - Notation $x \mapsto \mathbf{T}$ oder $x \mapsto \mathbf{F}$
- Funktion ν_0 kann rekursiv erweitert werden auf Formeln P ($\nu : \mathcal{F} \to \mathbb{B}$):
  - $\nu(x) = \nu_0(x)$ für $x \in \mathcal{P}$
  - $\nu(\top) = \mathbf{T}$
  - $\nu(\bot) = \mathbf{F}$
  - $\nu(\neg P) = \texttt{if } \nu(P) = \mathbf{T} \texttt{ then } \mathbf{F} \texttt{ else } \mathbf{T}$
  - $\nu(P \wedge Q) = \texttt{if } \nu(P) = \mathbf{T} \texttt{ then (if } \nu(Q) = \mathbf{T} \texttt{ then } \mathbf{T}\texttt{) else } \mathbf{F}$
  - $\nu(P \vee Q) = \texttt{if } \nu(P) = \mathbf{F} \texttt{ then (if } \nu(Q) = \mathbf{F} \texttt{ then } \mathbf{F}\texttt{) else } \mathbf{T}$
  - $\nu(P \Rightarrow Q) = \nu(\neg F \vee G)$
  - $\nu(P \Leftrightarrow Q) = \nu((F \Rightarrow G) \wedge (G \Rightarrow F))$
  - $\nu(P \oplus Q) = \nu(\neg(F \Leftrightarrow G))$
- $\nu(P)$ heißt *Interpretation von P*

Semantik - Beispiel

- $\nu_0 = \{x \mapsto \mathbf{F}, y \mapsto \mathbf{T}\}$

Beispiel (Evaluation von Formeln)

$$
\begin{aligned}
& \nu(x \oplus y) \\
= \; & \nu(\neg(x \Leftrightarrow y)) \\
= \; & \nu(\neg((x \Rightarrow y) \wedge (y \Rightarrow x))) \\
= \; & \nu(\neg((\neg x \vee y) \wedge (\neg y \vee x))) \\
= \; & \texttt{if } \nu((\neg x \vee y) \wedge (\neg y \vee x)) = \mathbf{F} \texttt{ then } \mathbf{T} \texttt{ else } \mathbf{F} \\
= \; & \ldots \\
= \; & \texttt{if } \mathbf{F} = \mathbf{F} \texttt{ then } \mathbf{T} \texttt{ else } \mathbf{F} \\
= \; & \mathbf{T}
\end{aligned}
$$

Semantik

Definition (Erfüllbarkeit)

Eine Formel P heißt **erfüllbar**, wenn eine Variablenbelegung $\nu_0 : var(P) \to \mathbb{B}$ *existiert*, so dass $\nu(P) = \mathbf{T}$ (ν_0 ist ein Modell von P)

- Notation: $\nu_0 \models P$

Definition (Tautologie)

Eine Formel P ist eine **Tautologie** (oder heißt **allgemeingültig**), falls *für alle* $\nu_0 : var(P) \to \mathbb{B}$ gilt, dass $\nu_0 \models P$

- Notation: $\models P$

Definition (Kontradiktion)

Eine Formel P ist eine **Kontradiktion** (oder heißt **unerfüllbar**), falls *kein* $\nu_0 : var(P) \to \mathbb{B}$ *existiert*, so dass $\nu_0 \models P$

- Notation: $\not\models P$

Wahrheitstabellen

$$P = ((x \wedge \neg(y \Rightarrow z)) \oplus x)$$

x	y	z	$y \Rightarrow z$	$\neg(y \Rightarrow z)$	$x \wedge \neg(y \Rightarrow z)$	$((x \wedge \neg(y \Rightarrow z)) \oplus x)$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0
1	0	0	1	0	0	1
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	1	1	0	0	1

- P ist **erfüllbar** aber keine **Tautologie**
- Problem: n Variablen benötigen 2^n Tabellenzeilen, daher nicht für große Formeln geeignet

Äquivalenzumformungen

- Distributivgesetze

$$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$$

$$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$$

- Absorptionsgesetze

$$P \vee (P \wedge Q) = P$$

$$P \wedge (P \vee Q) = P$$

- DeMorgansche Gesetze

$$\neg(P \wedge Q) = \neg P \vee \neg Q$$

$$\neg(P \vee Q) = \neg P \wedge \neg Q$$

- Kommutativität und Assoziativität von $\wedge, \vee$
- ... und viele weitere

Basen von Operatoren

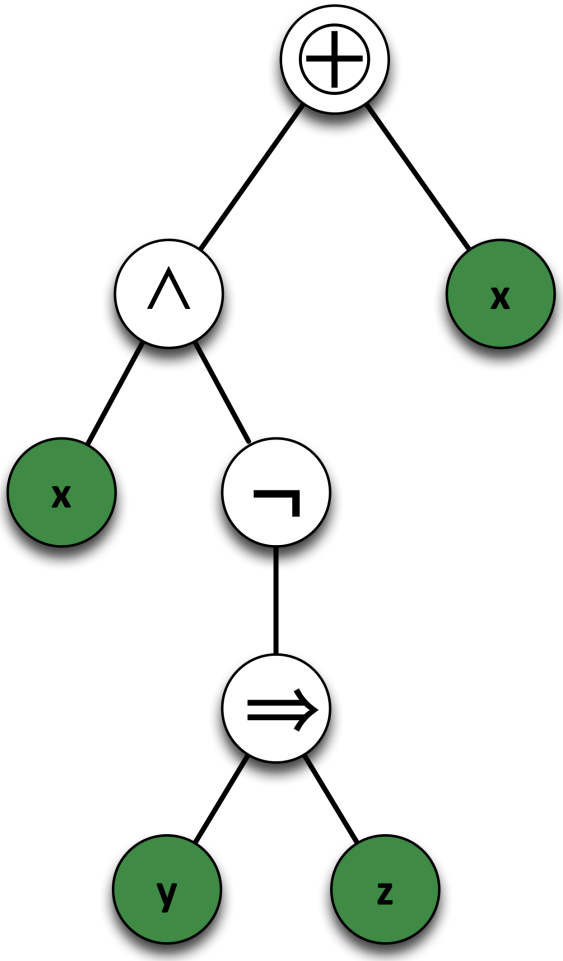
- Operatoren lassen sich durch andere ausdrücken
- Frage: Welche sind ausreichend?
- Was wir bereits wissen: $\Rightarrow, \Leftrightarrow$ und $\oplus$ lassen sich durch $\neg, \wedge, \vee$ ausdrücken:
  - $P \Rightarrow Q = \neg P \vee Q$
  - $P \Leftrightarrow Q = P \Rightarrow Q \wedge Q \Rightarrow P = (\neg P \vee Q) \wedge (\neg Q \vee P)$
  - $P \oplus Q = \neg(P \Leftrightarrow Q) = \neg((\neg P \vee Q) \wedge (\neg Q \vee P))$
- $\vee$ lässt sich mit der DeMorgan Regel eliminieren

$\neg, \wedge$ ist eine minimale Basis

- d.h. sämtliche aussagenlogische Formeln lassen sich unter Verwendung von $\neg$ und $\wedge$ darstellen
- weitere minimale Basen: $\{\neg, \vee\}, \{\wedge, \oplus\}$

Darstellung von Formeln

- Als Zeichenreihen (wie auf den letzten Folien)
- Als Bäume

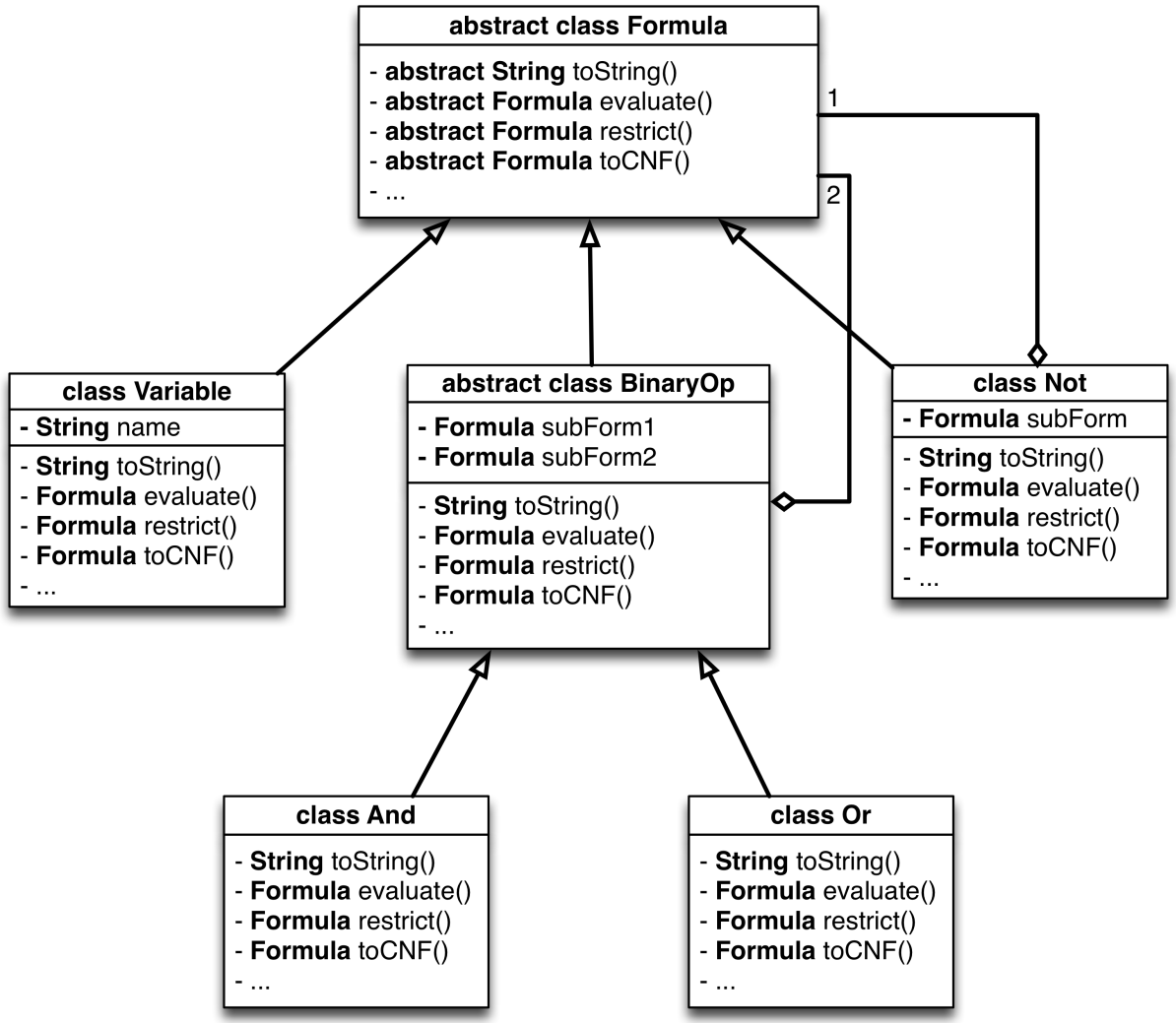


$$P = ((x \wedge \neg(y \Rightarrow z)) \oplus x)$$

- Innere Knoten: Operatoren
- Blätter: Aussagenvariablen oder Konstanten

Formeln in Java

- Abstrakte Basisklasse: `Formula`
- Abstrakte Basisklasse für 2-stellige Operationen: `BinaryOp`
- Für jeden Knotentyp eine abgeleitete Klasse: `Variable`, `Not`, `And`,...



Das SAT-Problem

Fragestellung: Ist eine gegebene Formel P erfüllbar oder nicht **(SATisfiability)**.

Beispiel (SAT Probleme)

- $(x \vee y) \wedge (\neg x \vee \neg y)$ ist erfüllbar (z.B. $\{x \mapsto \mathbf{T}, y \mapsto \mathbf{F}\}$)
- $(x \vee y) \wedge (\neg x \vee \neg y) \wedge y \wedge x$ ist nicht erfüllbar
- $x \wedge y \vee \neg z \Leftrightarrow x \oplus \neg z \wedge \neg y$ ist nicht mehr durch einfaches Hinschauen lösbar...

Vorschau (mehr dazu in zwei Wochen): Das SAT-Problem ist (mit großer Wahrscheinlichkeit) nicht in polynomialer Zeit entscheidbar.

Wer einen polynomiellen Algorithmus für SAT findet, bekommt **1 Mio. $**[1]

[1] http://www.claymath.org/millennium/P_vs_NP/

Ein kleines Anwendungsbeispiel - 1

3 Lehrer, 3 Fächer

- Albrecht (a) gibt Französisch (F) und Geschichte (G)
- Bert (b) gibt Englisch (E) und Französisch (F)
- Christine (c) gibt alle drei Fächer (E, F, G)

Wir wollen die Vorlieben der Lehrer beachten:

- Wenn Bert Französisch gibt, will Christine Englisch geben
- Wenn Christine Englisch gibt, will Albrecht kein Geschichte geben

Jeder Lehrer darf nur ein Fach geben und jedes Fach muss unterrichtet werden

Codierung

Variable a_F bedeutet, dass Lehrer A das Fach F gibt

- Anhand der Fächerkombinationen gibt es 7 Variablen:

$$a_F, a_G, b_E, b_F, c_E, c_F, c_G$$

Ein kleines Anwendungsbeispiel - 2

Codierung Teil 1: Jeder Lehrer darf nur ein Fach geben...

- $P_1 = (a_F \oplus a_G)$ Lehrer a gibt entweder F oder G
- $P_2 = (b_E \oplus b_F)$ Lehrer b gibt entweder E oder F
- $P_3 = (c_E \Rightarrow \neg c_F \wedge \neg c_G) \wedge (c_F \Rightarrow \neg c_E \wedge \neg c_G) \wedge (c_G \Rightarrow \neg c_E \wedge \neg c_F)$

Codierung Teil 2: Jedes Fach muss unterrichtet werden...

- $P_4 = a_F \vee b_F \vee c_F$ Französisch (F) wird unterrichtet
- $P_5 = a_G \vee c_G$ Geschichte (G) wird unterrichtet
- $P_6 = b_E \vee c_E$ Englisch (E) wird unterrichtet

Codierung der Vorlieben:

- $P_7 = b_F \Rightarrow c_E$ Wenn Bert Französisch, dann Christine Englisch
- $P_8 = c_E \Rightarrow \neg a_G$ Wenn Christine Englisch, dann Albrecht kein Geschichte

Gesamte Formel: $P = P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5 \wedge P_6 \wedge P_7 \wedge P_8$

- Erfüllbar! (z.B mit $\{b_E \mapsto \mathbf{T}, a_F \mapsto \mathbf{T}, c_G \mapsto \mathbf{T}\}$, alle anderen Variablen $\mathbf{F}$)