Logik für Informatiker
Logic for Computer Scientists

Till Mossakowski

Wintersemester 2014/15

Vollständigkeit der Aussagenlogik

Objekt- und Metatheorie

Objekttheorie = Beweise innerhalb eines formalen Beweissystems (z. B. Fitch)

Metatheorie = Beweise über ein formales Beweissystem

Vollständigkeit und Korrektheit von $\mathcal{F}_T$

Korrektheits-Satz. Das Beweissystem $\mathcal{F}_T$ ist korrekt, d. h., wenn

$$\mathcal{T} \vdash_T S$$

gilt, dann auch

$$\mathcal{T} \models_T S.$$

Vollständigkeit-Satz (Bernays, Post). Das Beweissystem $\mathcal{F}_T$ ist vollständig, d. h. wenn

$$\mathcal{T} \models_T S$$

gilt, dann auch

$$\mathcal{T} \vdash_T S.$$

Kompaktheits-Theorem

Kompaktheits-Theorem. Es sei $\mathcal{T}$ eine Menge von Sätzen. Wenn jede endliche Teilmenge von $\mathcal{T}$ wt-erfüllbar ist, so ist auch $\mathcal{T}$ wt-erfüllbar.

Resolution

Wiederholung: Konjunktive Normalform (KNF)

Für jeden aussagenlogischen Satz gibt es einen äquivalenten Satz in konjuntiver Normalform (KNF), d. h. einen Satz der Form

$$(\varphi_{1,1} \vee \cdots \vee \varphi_{1,m_1}) \wedge \cdots \wedge (\varphi_{n,1} \vee \cdots \vee \varphi_{n,m_n}) \quad (n \geq 1, m_i \geq 1),$$

wobei $\varphi_{i,j}$ Literale sind, d. h. atomare Sätze oder negierte atomare Sätze.

Man beachte, dass n auch 1 sein kann, z. B. ist $A \vee B$ in KNF.

Man beachte, dass m_i auch 1 sein kann, z. B. sind sowohl A als auch $A \wedge B$ in KNF.

Hornformeln

Ein Satz in KNF heißt Hornformel, wenn in dem Satz jede Disjunktion von Literalen höchstens ein positives Literal, d. h. einen nichtnegierten atomaren Satz, enthält.

Beispiele für Hornformeln

$\neg Home(claire) \wedge (\neg Home(max) \vee Happy(carl))$

$Home(claire) \wedge Home(max) \wedge \neg Home(carl)$

$Home(claire) \vee \neg Home(max) \vee \neg Home(carl)$

$Home(claire) \wedge Home(max) \wedge (\neg Home(max) \vee \neg Home(max))$

Beispiele für Nicht-Hornformeln

$\neg Home(claire) \wedge (Home(max) \vee Happy(carl))$

$(Home(claire) \vee Home(max) \vee \neg Happy(claire)) \wedge Happy(carl)$

$Home(claire) \vee (Home(max) \vee \neg Home(carl))$

Alternative Darstellung der Disjunktionen in Hornformeln

$$\neg A_1 \vee \cdots \vee \neg A_n \vee B \quad \Leftrightarrow \quad (A_1 \wedge \cdots \wedge A_n) \rightarrow B$$

$$\neg A_1 \vee \cdots \vee \neg A_n \quad \Leftrightarrow \quad (A_1 \wedge \cdots \wedge A_n) \rightarrow \bot$$

$$B \quad \Leftrightarrow \quad \top \rightarrow B$$

Jede Hornformel ist äquivalent zu einer Konjunktion von Implikationen der obigen drei Formen.

Erfüllbarkeitsalgorithmus für Hornformeln

Es sei S eine Hornformel in konditionaler Form, die aus atomaren Sätzen sowie aus $\top$ und $\bot$ besteht.

1. Für jede Implikation der Form $\top \to B$ belege B mit WAHR.
2. Falls für eine Implikation der Form $(A_1 \wedge \cdots \wedge A_n) \to B$ die atomaren Sätze $A_1, \ldots, A_n$ bereits mit WAHR belegt sind, so belege auch B mit WAHR.
3. Wiederhole Schritt 2 so lange als möglich.
4. Falls es eine Implikation der Form $(A_1 \wedge \cdots \wedge A_n) \to \bot$ gibt, wobei alle atomaren Sätze $A_1, \ldots, A_n$ bereits mit WAHR belegt sind, so ist S nicht wt-erfüllbar.
   Andernfalls belege alle noch nicht belegten atomaren Sätze von S mit FALSCH. Die so erhaltene Belegung macht S WAHR.

Korrektheit des Erfüllbarkeitsalgorithmus für Hornformeln

Theorem. Der Erfüllbarkeitsalgorithmus für Hornformeln ist korrekt, d. h. er klassifizert genau die wt-erfüllbaren Hornformeln als wt-erfüllbar.

Aussagenlogisches Prolog

$$
\begin{aligned}
&AncestorOf(a,b) :- MotherOf(a,b).\\
&AncestorOf(b,c) :- MotherOf(b,c).\\
&AncestorOf(a,b) :- FatherOf(a,b).\\
&AncestorOf(b,c) :- FatherOf(b,c).\\
&AncestorOf(a,c) :- AncestorOf(a,b), AncestorOf(b,c).\\
&MotherOf(a,b). \qquad FatherOf(b,c). \qquad FatherOf(b,d).
\end{aligned}
$$

Für die Frage, ob diese Menge von Sätzen zu B führt, fügt Prolog den Satz $\bot \leftarrow B$ hinzu und führt den Erfüllbarkeits-Algorithmus von Horn aus.
Falls der Algorithmus "unerfüllbar" ergibt, antwortet Prolog mit "ja", andernfalls mit "nein"

Klauseln

Eine Klausel ist eine endliche Menge von Literalen.

Beispiele:

$$C_1 = \{Small(a), Cube(a), BackOf(b, a)\}$$
$$C_2 = \{Small(a), Cube(b)\}$$
$$C_3 = \emptyset \quad \text{(auch bezeichnet mit } \square\text{)}$$

Jede Menge $\mathcal{T}$ von Sätzen in KNF kann durch eine äquivalente Menge $\mathcal{S}$ von Klauseln ersetzt werden, dabei entspricht jede Disjunktion eines Satzes aus $\mathcal{T}$ einer Klausel.

Resolution

Eine Klausel R heißt Resolvente der Klauseln C_1 und C_2, wenn es einen atomaren Satz A gibt, mit $A \in C_1$ and $(\neg A) \in C_2$ und

$$R = (C_1 \setminus \{A\}) \cup (C_2 \setminus \{\neg A\}).$$

Resolutionsalgorithmus:

Gegeben sei eine Menge $\mathcal{S}$ von Klauseln.

1. Bilde (falls möglich) aus zwei Klauseln von $\mathcal{S}$ eine Resolvente und füge sie zu $\mathcal{S}$ hinzu.
2. Wiederhole Schritt 1 solange als möglich.

Falls die leere Klausel $\square$ zu $\mathcal{S}$ hinzugefügt wurde, ist die ursprüngliche Satzmenge $\mathcal{S}$ nicht wt-erfüllbar, andernfalls wt-erfüllbar.

Beispiel einer Resolution

Wir starten mit folgendem Satz in KNF.

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

Wir erhalten folgende Menge von Klauseln:

$$\{\{\neg A\}, \{B, C\}, \{\neg C, \neg D\}, \{A, D\}, \{\neg B, \neg D\}\}$$

Anwenden des Resolutionsverfahrens:

$$\dfrac{\dfrac{\{A, D\} \quad \{\neg A\}}{\{D\}} \qquad \dfrac{\dfrac{\{B, C\} \quad \{\neg C, \neg D\}}{\{B, \neg D\}} \qquad \{\neg B, \neg D\}}{\{\neg D\}}}{\square}$$

Korrektheit und Vollständigkeit

Theorem.
Der Resolutionsalgorithmus ist korrekt und vollständig, d. h. für eine gegebene Menge $\mathcal{S}$ von Klauseln lässt sich durch den Resolutionsalgorithmus genau dann die leere Klausel $\square$ herleiten, wenn $\mathcal{S}$ nicht wt-erfüllbar ist.

Damit erhalten wir ein alternatives korrektes und vollständiges Beweisverfahren für die tautologische Folgerung:

$$\mathcal{T} \models_T S$$

gilt genau dann, wenn mittels Resolution die leere Klausel $\square$ aus der Klauselmenge für $\mathcal{T} \cup \{\neg S\}$ hergeleitet werden kann.