TECHNISCHE UNIVERSITÄT DRESDEN

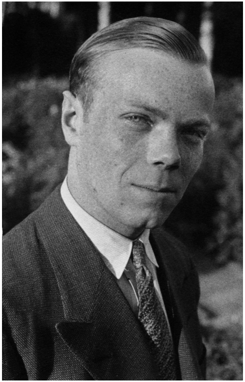
THEORETISCHE INFORMATIK UND LOGIK

20. Vorlesung: Resolution/Endliche Modelle

Markus Krötzsch

Lehrstuhl Wissensbasierte Systeme

TU Dresden, 30. Juni 2017



Jacques Herbrand

12.2.1908 – 27.7.1931

Resolution

Resolutionsregel:

$$\frac{\{A_1, \ldots, A_n, L_1, \ldots, L_k\} \quad \{\neg A'_1, \ldots, \neg A'_m, L'_1, \ldots, L'_\ell\}}{\{L_1\sigma, \ldots, L_k\sigma, L'_1\sigma, \ldots, L'_\ell\sigma\}}$$

falls σ allgemeinster Unifikator von $\{A_1, \ldots, A_n, A'_1, \ldots, A'_m\}$ ist

Algorithmus (Skizze):

(1) Bilde Klauselform

(2) Bilde systematisch Resolventen durch Resolution von Varianten bereits abgeleiteter Klauseln

(3) Wiederhole (2), bis entweder $\perp$ erzeugt wird („unerfüllbar“) oder keine neuen Klauseln mehr entstehen[1]

---

[1] Dieser Fall ist eher ungewöhnlich: Meist entstehen bei erfüllbaren Theorien immer mehr neue Klauseln ohne dass das Verfahren terminiert.

Vollständigkeit und Korrektheit

Resolutionssatz: Sei F eine prädikatenlogische Formel und $\mathcal{K}_i$ ($i \geq 0$) die vom Resolutionsalgorithmus ermittelten Klauselmengen. Dann sind die folgenden Aussagen äquivalent:

- F ist unerfüllbar
- Es gibt ein $\ell \geq 0$ mit $\bot \in \mathcal{K}_\ell$

- Korrektheit hatten wir bereits gezeigt
- Vollständigkeit steht noch aus

Korrektheit (Korrektur)

Beweis (Korrektheit Resoutionsschritt): Gegeben:

- Klauseln $K_1 = \{A_1, \ldots, A_n, L_1, \ldots, L_k\}$ und $K_2 = \{\neg A'_1, \ldots, \neg A'_m, L'_1, \ldots, L'_\ell\}$
- (allgemeinster) Unifikator σ der Menge $\{A_1, \ldots, A_n, A'_1, \ldots, A'_m\}$
- zugehörige Resolvente $K = \{L_1\sigma, \ldots, L_k\sigma, L'_1\sigma, \ldots, L'_\ell\sigma\}$

Korrektheit (Korrektur)

Beweis (Korrektheit Resoutionsschritt): Gegeben:

- Klauseln $K_1 = \{A_1, \ldots, A_n, L_1, \ldots, L_k\}$ und $K_2 = \{\neg A'_1, \ldots, \neg A'_m, L'_1, \ldots, L'_\ell\}$
- (allgemeinster) Unifikator σ der Menge $\{A_1, \ldots, A_n, A'_1, \ldots, A'_m\}$
- zugehörige Resolvente $K = \{L_1\sigma, \ldots, L_k\sigma, L'_1\sigma, \ldots, L'_\ell\sigma\}$

Sei $\mathcal{I}$ eine beliebige Interpretation.

- Falls $\mathcal{I} \models \forall K_1 \wedge \forall K_2$, dann gilt auch $\mathcal{I} \models \forall(K_1\sigma) \wedge \forall(K_2\sigma)$ (die Substitution konkretisiert eine Allaussage)
- Also gilt für alle Zuweisungen $\mathcal{Z}$: $\mathcal{I}, \mathcal{Z} \models (K_1\sigma) \wedge (K_2\sigma)$
- Fall 1: $\mathcal{I}, \mathcal{Z} \models A_1\sigma$ $(= A_2\sigma = \ldots = A'_m\sigma)$. Dann gilt $\mathcal{I}, \mathcal{Z} \models L'_1\sigma \vee \ldots \vee L'_\ell\sigma$, und damit $\mathcal{I}, \mathcal{Z} \models K$
- Fall 2: $\mathcal{I}, \mathcal{Z} \not\models A_1\sigma$ $(= A_2\sigma = \ldots = A'_m\sigma)$. Dann gilt $\mathcal{I}, \mathcal{Z} \models L_1\sigma \vee \ldots \vee L_k\sigma$, und damit $\mathcal{I}, \mathcal{Z} \models K$
- Also gilt $\mathcal{I} \models \forall K$.

Da $\mathcal{I}$ beliebig ist gilt also $\forall K_1 \wedge \forall K_2 \models \forall K$ – d.h., jede Resolvente ist logische Konsequenz der resolvierten Klauseln

Prädikatenlogisches Schließen mit Aussagenlogik

Herbrand-Expansionen

Die Herbrand-Expansion $HE(F)$ einer Formel $F = \forall x_1, \ldots, x_n.G$ in Skolemform ist die Menge:

$$HE(F) := \{G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \mid t_1, \ldots, t_n \in \Delta_F\}$$

$HE(F)$ ist also die (möglicherweise unendliche) Menge von variablen-freien Sätzen, die in Herbrandmodellen von F gelten müssten.

Herbrand-Expansionen

Die Herbrand-Expansion $HE(F)$ einer Formel $F = \forall x_1, \ldots, x_n.G$ in Skolemform ist die Menge:

$$HE(F) := \{G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \mid t_1, \ldots, t_n \in \Delta_F\}$$

$HE(F)$ ist also die (möglicherweise unendliche) Menge von variablen-freien Sätzen, die in Herbrandmodellen von F gelten müssten.

Quantorenfreie Sätze = aussagenlogische Formeln:

- $HE(F)$ enthält Formeln ohne Variablen, d.h. Boolsche Kombinationen geschlossener Atome
- Geschlossene Atome können unabhängig voneinander wahr oder falsch sein, egal wie ihre genaue Struktur aussieht
- Wir können sie also als „ungewöhnlich benannte" aussagenlogische Atome auffassen und die gesamte Formel aussagenlogisch interpretieren

$\rightsquigarrow$ $HE(F)$ als aussagenlogische Theorie

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Beweis: Wir zeigen, dass $F = \forall x_1, \ldots, x_n.G$ genau dann ein Herbrandmodell hat, wenn $HE(F)$ aussagenlogisch erfüllbar ist:

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Beweis: Wir zeigen, dass $F = \forall x_1, \ldots, x_n.G$ genau dann ein Herbrandmodell hat, wenn $HE(F)$ aussagenlogisch erfüllbar ist:

- $\mathcal{I}$ ist ein Herbrandmodell von F

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Beweis: Wir zeigen, dass $F = \forall x_1, \ldots, x_n.G$ genau dann ein Herbrandmodell hat, wenn $HE(F)$ aussagenlogisch erfüllbar ist:

- $\mathcal{I}$ ist ein Herbrandmodell von F
- gdw. $\mathcal{I}, \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \models G$ für alle $t_1, \ldots, t_n \in \Delta_F$

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Beweis: Wir zeigen, dass $F = \forall x_1, \ldots, x_n.G$ genau dann ein Herbrandmodell hat, wenn $HE(F)$ aussagenlogisch erfüllbar ist:

- $\mathcal{I}$ ist ein Herbrandmodell von F
- gdw. $\mathcal{I}, \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \models G$ für alle $t_1, \ldots, t_n \in \Delta_F$
- gdw. $\mathcal{I} \models G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ für alle $t_1, \ldots, t_n \in \Delta_F$ (Lemma)

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Beweis: Wir zeigen, dass $F = \forall x_1, \ldots, x_n.G$ genau dann ein Herbrandmodell hat, wenn $HE(F)$ aussagenlogisch erfüllbar ist:

- $\mathcal{I}$ ist ein Herbrandmodell von F
- gdw. $\mathcal{I}, \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \models G$ für alle $t_1, \ldots, t_n \in \Delta_F$
- gdw. $\mathcal{I} \models G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ für alle $t_1, \ldots, t_n \in \Delta_F$ (Lemma)
- gdw. für alle $H \in HE(F)$ gilt $\mathcal{I} \models H$

Gödel, Herbrand, Skolem

Satz von Gödel, Herbrand & Skolem: Eine Formel F in Skolemform ist genau dann erfüllbar, wenn $HE(F)$ aussagenlogisch erfüllbar ist.

Beweis: Wir zeigen, dass $F = \forall x_1, \ldots, x_n.G$ genau dann ein Herbrandmodell hat, wenn $HE(F)$ aussagenlogisch erfüllbar ist:

- $\mathcal{I}$ ist ein Herbrandmodell von F
- gdw. $\mathcal{I}, \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \models G$ für alle $t_1, \ldots, t_n \in \Delta_F$
- gdw. $\mathcal{I} \models G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ für alle $t_1, \ldots, t_n \in \Delta_F$ (Lemma)
- gdw. für alle $H \in HE(F)$ gilt $\mathcal{I} \models H$
- gdw. $\mathcal{I}$ als aussagenlogisches Modell für $HE(F)$ angesehen werden kann. $\square$

Satz von Herbrand

Als Korollar der gezeigten Ergebnisse erhalten wir ein wichtiges Resultat:

Satz: Eine Formel F in Skolemform ist genau dann unerfüllbar, wenn eine endliche Teilmenge von $E(F)$ aussagenlogisch unerfüllbar ist.

Satz von Herbrand

Als Korollar der gezeigten Ergebnisse erhalten wir ein wichtiges Resultat:

Satz: Eine Formel F in Skolemform ist genau dann unerfüllbar, wenn eine endliche Teilmenge von $E(F)$ aussagenlogisch unerfüllbar ist.

Beweis: Das Kontrapositiv des Satzes von Gödel, Herbrand & Skolem besagt:

Eine Formel F in Skolemform ist genau dann unerfüllbar, wenn $HE(F)$ aussagenlogisch unerfüllbar ist.

Der Satz folgt nun, weil jede unerfüllbare aussagenlogische Formelmenge eine endliche Teilmenge hat, die unerfüllbar ist (Kompaktheit der Aussagenlogik, siehe Formale Systeme, WS 2016/2017, Vorlesung 24) □

Prädikatenlogik semi-entscheiden

Das Ergebnis Herbrands ermöglicht bereits einen naiven Algorithmus zur Semi-Entscheidung von Unerfüllbarkeit in der Prädikatenlogik:

Gegeben: Eine Formel F

- Wandle F in Skolemform F' um
- Definiere eine Reihenfolge der Formeln in $HE(F')$: $F_1, F_2, F_3, \ldots$
- Für alle $i \geq 1$:
  - Prüfe ob die endliche Menge $\{F_1, \ldots, F_i\}$ aussagenlogisch unerfüllbar ist
  - Falls ja, dann gib „unerfüllbar" aus; andernfalls fahre fort

Offenbar ist das kein praktischer Algorithmus, aber er zeigt Semi-Entscheidbarkeit

Vollständigkeit der Resolution

Ansatz

Herbrands Satz liefert uns auch eine Strategie zum Beweis der Vollständigkeit des Resolutionsalgorithmus

Wir wissen bereits:

- Unerfüllbarkeit einer Klauselmenge zeigt sich in der Unerfüllbarkeit ihrer Herbrand-Expansion
- Die Unerfüllbarkeit der Herbrand-Expansion kann man mit aussagenlogischer Resolution beweisen
- Prädikatenlogische Resolution verallgemeinert aussagenlogische Resolution indem wir direkt mit Klauseln arbeiten, die noch Variablen enthalten

Ansatz

Herbrands Satz liefert uns auch eine Strategie zum Beweis der Vollständigkeit des Resolutionsalgorithmus

Wir wissen bereits:

- Unerfüllbarkeit einer Klauselmenge zeigt sich in der Unerfüllbarkeit ihrer Herbrand-Expansion
- Die Unerfüllbarkeit der Herbrand-Expansion kann man mit aussagenlogischer Resolution beweisen
- Prädikatenlogische Resolution verallgemeinert aussagenlogische Resolution indem wir direkt mit Klauseln arbeiten, die noch Variablen enthalten

Frage: Kann man alle Schlüsse, die man auf expandierten Formeln aussagenlogisch erzeugen kann, auch direkt prädikatenlogisch (mit Variablen) erhalten?

Lifting-Lemma

Wir zeigen: Ja, jeder aussagenlogische Schluss (auf der Expansion) kann auf einen prädikatenlogischen Schluss (auf den Klauseln mit Variablen) „angehoben“ werden

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.[1]

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

[1] Die Verwendung der selben Substitution für K_1' und K_2' ist keine Einschränkung, da wir durch Variantenbildung sicherstellen können, dass K_1 und K_2 keine Variablen gemein haben.

Lifting-Lemma: Beweis

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

Beweis: Sei $A' \in K_1'$ das (geschlossene) Atom, über das resolviert wurde, d.h. $\neg A' \in K_2'$.

Lifting-Lemma: Beweis

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

Beweis: Sei $A' \in K_1'$ das (geschlossene) Atom, über das resolviert wurde, d.h. $\neg A' \in K_2'$.

Sei $\mathcal{A}_1 := \{A \mid A \in K_1,\ A\sigma = A'\}$ und $\mathcal{A}_2 := \{A \mid \neg A \in K_2, A\sigma = A'\}$.

Lifting-Lemma: Beweis

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

Beweis: Sei $A' \in K_1'$ das (geschlossene) Atom, über das resolviert wurde, d.h. $\neg A' \in K_2'$.

Sei $\mathcal{A}_1 := \{A \mid A \in K_1,\ A\sigma = A'\}$ und $\mathcal{A}_2 := \{A \mid \neg A \in K_2, A\sigma = A'\}$.

Dann ist σ ein Unifikator für $\mathcal{A}_1 \cup \mathcal{A}_2$.

Also hat $\mathcal{A}_1 \cup \mathcal{A}_2$ einen allgemeinsten Unifikator θ.

Lifting-Lemma: Beweis

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

Beweis: Sei $A' \in K_1'$ das (geschlossene) Atom, über das resolviert wurde, d.h. $\neg A' \in K_2'$.

Sei $\mathcal{A}_1 := \{A \mid A \in K_1,\ A\sigma = A'\}$ und $\mathcal{A}_2 := \{A \mid \neg A \in K_2, A\sigma = A'\}$.

Dann ist σ ein Unifikator für $\mathcal{A}_1 \cup \mathcal{A}_2$.

Also hat $\mathcal{A}_1 \cup \mathcal{A}_2$ einen allgemeinsten Unifikator θ.

Sei R die Resolvente von K_1 und K_2 bzgl. θ.

Lifting-Lemma: Beweis

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

Beweis: Sei $A' \in K_1'$ das (geschlossene) Atom, über das resolviert wurde, d.h. $\neg A' \in K_2'$.

Sei $\mathcal{A}_1 := \{A \mid A \in K_1,\ A\sigma = A'\}$ und $\mathcal{A}_2 := \{A \mid \neg A \in K_2, A\sigma = A'\}$.

Dann ist σ ein Unifikator für $\mathcal{A}_1 \cup \mathcal{A}_2$.
Also hat $\mathcal{A}_1 \cup \mathcal{A}_2$ einen allgemeinsten Unifikator θ.

Sei R die Resolvente von K_1 und K_2 bzgl. θ.

Dann enthalten R' und R Instanzen der gleichen Literale, d.h. sie sind von der Form $R' = \{L_1\sigma, \ldots, L_n\sigma\}$ und $R = \{L_1\theta, \ldots, L_n\theta\}$

Lifting-Lemma: Beweis

Satz (Lifting-Lemma): Seien K_1 und K_2 prädikatenlogische Klauseln mit Grundinstanzen $K_1' = K_1\sigma$ und $K_2' = K_2\sigma$.

Wenn R' eine (aussagenlogische) Resolvente von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R, welche R' als Grundinstanz hat.

Beweis: Sei $A' \in K_1'$ das (geschlossene) Atom, über das resolviert wurde, d.h. $\neg A' \in K_2'$.

Sei $\mathcal{A}_1 := \{A \mid A \in K_1,\ A\sigma = A'\}$ und $\mathcal{A}_2 := \{A \mid \neg A \in K_2, A\sigma = A'\}$.

Dann ist σ ein Unifikator für $\mathcal{A}_1 \cup \mathcal{A}_2$.
Also hat $\mathcal{A}_1 \cup \mathcal{A}_2$ einen allgemeinsten Unifikator θ.

Sei R die Resolvente von K_1 und K_2 bzgl. θ.

Dann enthalten R' und R Instanzen der gleichen Literale, d.h. sie sind von der Form $R' = \{L_1\sigma, \ldots, L_n\sigma\}$ und $R = \{L_1\theta, \ldots, L_n\theta\}$

Da θ allgemeinster Unifikator ist gibt es λ mit $\sigma = \theta \circ \lambda$ und es gilt:
$R\lambda = \{L_1\theta\lambda, \ldots, L_n\theta\lambda\} = \{L_1\sigma, \ldots, L_n\sigma\} = R'$ □

Vollständigkeit der Resolution (1)

Resolutionssatz: Sei F eine prädikatenlogische Formel und $\mathcal{K}_i$ ($i \geq 0$) die vom Resolutionsalgorithmus ermittelten Klauselmengen. Dann sind die folgenden Aussagen äquivalent:

- F ist unerfüllbar
- Es gibt ein $\ell \geq 0$ mit $\perp \in \mathcal{K}_\ell$

Vollständigkeit der Resolution (1)

Resolutionssatz: Sei F eine prädikatenlogische Formel und $\mathcal{K}_i$ ($i \geq 0$) die vom Resolutionsalgorithmus ermittelten Klauselmengen. Dann sind die folgenden Aussagen äquivalent:

- F ist unerfüllbar
- Es gibt ein $\ell \geq 0$ mit $\bot \in \mathcal{K}_\ell$

Beweis (Vollständigkeit): Sei F unerfüllbar

- Dann ist $HE(F)$ unerfüllbar
- Dann gibt es eine (endliche) aussagenlogische Resolutionsableitung von $\bot$ aus $HE(F)$
- Die Ableitung erzeugt eine endliche Folge von Klauseln: $K'_1, K'_2, \ldots, K'_{m-1}, K'_m = \bot$
- Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.
- Für $i = m$ folgt daraus der Satz, denn $K'_m = \bot$ kann nur Grundinstanz von $\bot$ sein, d.h. $\bot \in \mathcal{K}_\ell$ für ein $\ell \geq 0$.

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Aussage klar für $K'_i \in HE(F)$: in diesem Fall ist K'_i ist Grundinstanz einer Klausel K_i in der Klauselform von F und in $\mathcal{K}_0$

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Aussage klar für $K'_i \in HE(F)$: in diesem Fall ist K'_i ist Grundinstanz einer Klausel K_i in der Klauselform von F und in $\mathcal{K}_0$

Restlicher Beweis durch Induktion über i:

- Indukstionsannahme: Die Aussage gilt für alle $j < i$

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K_i' ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Aussage klar für $K_i' \in HE(F)$: in diesem Fall ist K_i' ist Grundinstanz einer Klausel K_i in der Klauselform von F und in $\mathcal{K}_0$

Restlicher Beweis durch Induktion über i:

- Indukstionsannahme: Die Aussage gilt für alle $j < i$
- K_i' ist Resolvente zweier Klauseln K_a' und K_b' mit $a, b < i$

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Aussage klar für $K'_i \in HE(F)$: in diesem Fall ist K'_i ist Grundinstanz einer Klausel K_i in der Klauselform von F und in $\mathcal{K}_0$

Restlicher Beweis durch Induktion über i:

- Indukstionsannahme: Die Aussage gilt für alle $j < i$
- K'_i ist Resolvente zweier Klauseln K'_a und K'_b mit $a, b < i$
- Laut Hypothese sind K'_a und K'_b also Instanzen von Klauseln K_a und K_b in einer Menge $\mathcal{K}_\ell$

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Aussage klar für $K'_i \in HE(F)$: in diesem Fall ist K'_i ist Grundinstanz einer Klausel K_i in der Klauselform von F und in $\mathcal{K}_0$

Restlicher Beweis durch Induktion über i:

- Indukstionsannahme: Die Aussage gilt für alle $j < i$
- K'_i ist Resolvente zweier Klauseln K'_a und K'_b mit $a, b < i$
- Laut Hypothese sind K'_a und K'_b also Instanzen von Klauseln K_a und K_b in einer Menge $\mathcal{K}_\ell$
- Laut Lifting-Lemma ist demnach K'_i ebenfalls die Instanz einer Klausel K_i, die durch Resolution aus K_a und K_b entsteht

Vollständigkeit der Resolution (2)

Beweis (Vollständigkeit): Behauptung: Jede Klausel K'_i ist Grundinstanz einer Klausel K_i die in $\mathcal{K}_\ell$ vorkommt für ein $\ell \geq 0$.

Aussage klar für $K'_i \in HE(F)$: in diesem Fall ist K'_i ist Grundinstanz einer Klausel K_i in der Klauselform von F und in $\mathcal{K}_0$

Restlicher Beweis durch Induktion über i:

- Indukstionsannahme: Die Aussage gilt für alle $j < i$
- K'_i ist Resolvente zweier Klauseln K'_a und K'_b mit $a, b < i$
- Laut Hypothese sind K'_a und K'_b also Instanzen von Klauseln K_a und K_b in einer Menge $\mathcal{K}_\ell$
- Laut Lifting-Lemma ist demnach K'_i ebenfalls die Instanz einer Klausel K_i, die durch Resolution aus K_a und K_b entsteht
- Diese Resolvente K_i ist also ebenfalls in einer Menge der Form $\mathcal{K}_{\ell'}$ $\square$

Kompaktheit

Die Existenz von vollständigen und korrekten logischen Schließverfahren wie Resolution ist eng verwandt mit einer grundsätzlichen Eigenschaft der Prädikatenlogik:

Satz (Endlichkeitssatz, Kompaktheitssatz): Falls eine unendliche Menge prädikatenlogischer Sätze $\mathcal{T}$ eine logische Konsequenz F hat, so ist F auch Konsequenz einer endlichen Teilmenge von $\mathcal{T}$.

Kompaktheit

Die Existenz von vollständigen und korrekten logischen Schließverfahren wie Resolution ist eng verwandt mit einer grundsätzlichen Eigenschaft der Prädikatenlogik:

Satz (Endlichkeitssatz, Kompaktheitssatz): Falls eine unendliche Menge prädikatenlogischer Sätze $\mathcal{T}$ eine logische Konsequenz F hat, so ist F auch Konsequenz einer endlichen Teilmenge von $\mathcal{T}$.

Beweis: Die gegebene logische Konsequenz ist gleichbedeutend damit, dass $\mathcal{T} \cup \{\neg F\}$ unerfüllbar ist.

Laut Resolutionssatz (Vollständigkeit) kann die Unerfüllbarkeit von $\mathcal{T} \cup \{\neg F\}$ nach endlich vielen Schritten durch Ableitung der leeren Klausel nachgewiesen werden.

Dabei können nur endlich viele Klauseln aus der Klauselform von $\mathcal{T} \cup \{\neg F\}$ verwendet worden sein. Laut Resolutionssatz (Korrektheit) folgt die Konsequenz also bereits aus einer endlichen Teilmenge von $\mathcal{T}$. $\square$

Die Grenzen der Prädikatenlogik

Kompaktheit zeigt uns auch erste Grenzen der Prädikatenlogik auf.

Eine logische Formel F mit zwei freien Variablen x und y drückt den transitiven Abschluss einer binären Relation r aus, wenn in jeder Interpretation $\mathcal{I}$ und für alle $\delta_1, \delta_2 \in \Delta^{\mathcal{I}}$ gilt:

$$\mathcal{I}, \{x \mapsto \delta_1, y \mapsto \delta_2\} \models F \quad \text{gdw.} \quad \langle \delta_1, \delta_2 \rangle \in (r^{\mathcal{I}})^*$$

Die Grenzen der Prädikatenlogik

Kompaktheit zeigt uns auch erste Grenzen der Prädikatenlogik auf.

Eine logische Formel F mit zwei freien Variablen x und y drückt den transitiven Abschluss einer binären Relation r aus, wenn in jeder Interpretation $\mathcal{I}$ und für alle $\delta_1, \delta_2 \in \Delta^{\mathcal{I}}$ gilt:

$$\mathcal{I}, \{x \mapsto \delta_1, y \mapsto \delta_2\} \models F \quad \text{gdw.} \quad \langle \delta_1, \delta_2 \rangle \in (r^{\mathcal{I}})^*$$

Satz: Es gibt keine prädikatenlogische Formel, die den transitiven Abschluss einer binären Relation ausdrückt.

Die Grenzen der Prädikatenlogik

Kompaktheit zeigt uns auch erste Grenzen der Prädikatenlogik auf.

Eine logische Formel F mit zwei freien Variablen x und y drückt den transitiven Abschluss einer binären Relation r aus, wenn in jeder Interpretation $\mathcal{I}$ und für alle $\delta_1, \delta_2 \in \Delta^{\mathcal{I}}$ gilt:

$$\mathcal{I}, \{x \mapsto \delta_1, y \mapsto \delta_2\} \models F \quad \text{gdw.} \quad \langle \delta_1, \delta_2 \rangle \in (r^{\mathcal{I}})^*$$

Satz: Es gibt keine prädikatenlogische Formel, die den transitiven Abschluss einer binären Relation ausdrückt.

Beweis: Angenommen es gäbe so eine Formel F.

Dann ist die folgende unendliche Theorie unerfüllbar:

$$\{ \quad F\{x \mapsto a, y \mapsto b\}, \neg r(a,b), \neg\exists x_1.(r(a,x_1) \wedge r(x_1,b)),$$
$$\neg\exists x_1, x_2.(r(a,x_1) \wedge r(x_1,x_2) \wedge r(x_2,b)), \ldots \quad \}$$

Aber jede endliche Teilmenge der Theorie ist erfüllbar. Die Existenz der Theorie würde also dem Endlichkeitssatz widersprechen. □

Endliche Modelle

Endlichkeit von Modellen

Löwenheim-Skolem: Jede erfüllbare Formel hat ein abzählbar großes Modell

Kann man dies noch verstärken? Hat jede erfüllbare Formel vielleicht sogar ein endliches Modell?

Endlichkeit von Modellen

Löwenheim-Skolem: Jede erfüllbare Formel hat ein abzählbar großes Modell

Kann man dies noch verstärken? Hat jede erfüllbare Formel vielleicht sogar ein endliches Modell?

Nein! Prädikatenlogik kann unendliche Modelle erzwingen:

Beispiel:

$$\forall x.(\mathsf{Mensch}(x) \to \exists y.(\mathsf{hatMutter}(x,y) \wedge \mathsf{Mensch}(y)))$$

$$\forall x,y.(\mathsf{hatMutter}(x,y) \to \mathsf{hatVorfahre}(x,y))$$

$$\forall x,y,z.((\mathsf{hatVorfahre}(x,y) \wedge \mathsf{hatVorfahre}(y,z)) \to \mathsf{hatVorfahre}(x,z))$$

$$\forall x.\neg\mathsf{hatVorfahre}(x,x)$$

Diese Theorie ist erfüllbar, aber hat nur unendliche Modelle.
(Kontrollfrage: Warum?)

Logik über endlichen Modellen

Sind unendliche Modelle in der Praxis überhaupt wünschenswert?

Geht es auch endlich?

Logik über endlichen Modellen

Sind unendliche Modelle in der Praxis überhaupt wünschenswert?
Geht es auch endlich?

Prädikatenlogik mit endlichen Modellen verwendet die gleiche Syntax und Semantik wie Prädikatenlogik allgemein, aber mit der zusätzlichen Bedingung, dass die Domäne von Interpretationen endlich sein muss.

Logik über endlichen Modellen

Sind unendliche Modelle in der Praxis überhaupt wünschenswert? Geht es auch endlich?

Prädikatenlogik mit endlichen Modellen verwendet die gleiche Syntax und Semantik wie Prädikatenlogik allgemein, aber mit der zusätzlichen Bedingung, dass die Domäne von Interpretationen endlich sein muss.

Monotonie (Rückblick): weniger Modelle = mehr Konsequenzen

Beispiel:

$$\forall x.(\mathsf{Mensch}(x) \to \exists y.(\mathsf{hatVorfahre}(x, y) \land \mathsf{Mensch}(y)))$$

$$\forall x, y, z.((\mathsf{hatVorfahre}(x, y) \land \mathsf{hatVorfahre}(y, z)) \to \mathsf{hatVorfahre}(x, z))$$

Diese Theorie ist in der Prädikatenlogik mit endlichen Modellen erfüllbar, aber jedes endliche Modell muss einen hatVorfahre-Zyklus enthalten. Daher folgt $\exists x.\mathsf{hatVorfahre}(x, x)$, obwohl dies in der allgemeinen Prädikatenlogik keine Konsequenz wäre.

Erfüllbarkeit wird semi-entscheidbar

Die Bezeichnung der Elemente einer Interpretationsdomäne ist irrelevant – für die Wahrheit von Sätzen kommt es nur darauf an, wie Konstanten und Prädikatsymbole interpretiert werden.

Erfüllbarkeit wird semi-entscheidbar

Die Bezeichnung der Elemente einer Interpretationsdomäne ist irrelevant – für die Wahrheit von Sätzen kommt es nur darauf an, wie Konstanten und Prädikatsymbole interpretiert werden.

Erfüllbarkeitstest

Gegeben: Ein Satz F

- Betrachte systematisch alle endlichen Interpretationen der Symbole in F (z.B. geordnet nach aufsteigender Größe der Domäne)
- Prüfe für jedes Modell $\mathcal{I}$, ob $\mathcal{I} \models F$ gilt:
  - Falls ja, dann gib aus „erfüllbar"
  - Falls nein, dann fahre mit nächster Interpretation fort

Es ist leicht zu sehen, dass dieser Algorithmus die Erfüllbarkeit in endlichen Modellen semi-entscheidet.

Endlich = einfach?

Endlich = einfach?

Trotzdem bleibt logisches Schließen schwer:

Satz von Trakhtenbrot: Logisches Schließen (Erfüllbarkeit, Allgemeingültigkeit, logische Konsequenz) in der Prädikatenlogik mit endlichen Modellen ist unentscheidbar.

(ohne Beweis)

Endlich = einfach?

Trotzdem bleibt logisches Schließen schwer:

> Satz von Trakhtenbrot: Logisches Schließen (Erfüllbarkeit, Allgemeingültigkeit, logische Konsequenz) in der Prädikatenlogik mit endlichen Modellen ist unentscheidbar.

(ohne Beweis)

> Korollar: Es gibt kein vollständiges und korrektes Beweissystem für Prädikatenlogik mit endlichen Modellen.

Beweis: Angenommen es gäbe ein solches System.

Endlich = einfach?

Trotzdem bleibt logisches Schließen schwer:

Satz von Trakhtenbrot: Logisches Schließen (Erfüllbarkeit, Allgemeingültigkeit, logische Konsequenz) in der Prädikatenlogik mit endlichen Modellen ist unentscheidbar.

(ohne Beweis)

Korollar: Es gibt kein vollständiges und korrektes Beweissystem für Prädikatenlogik mit endlichen Modellen.

Beweis: Angenommen es gäbe ein solches System. Dann wäre logische Konsequenz und speziell auch Unerfüllbarkeit semi-entscheidbar.

Endlich = einfach?

Trotzdem bleibt logisches Schließen schwer:

> Satz von Trakhtenbrot: Logisches Schließen (Erfüllbarkeit, Allgemeingültigkeit, logische Konsequenz) in der Prädikatenlogik mit endlichen Modellen ist unentscheidbar.

(ohne Beweis)

> Korollar: Es gibt kein vollständiges und korrektes Beweissystem für Prädikatenlogik mit endlichen Modellen.

Beweis: Angenommen es gäbe ein solches System. Dann wäre logische Konsequenz und speziell auch Unerfüllbarkeit semi-entscheidbar.

Wir wissen, dass Erfüllbarkeit ebenfalls semi-entscheidbar ist.

Endlich = einfach?

Trotzdem bleibt logisches Schließen schwer:

Satz von Trakhtenbrot: Logisches Schließen (Erfüllbarkeit, Allgemeingültigkeit, logische Konsequenz) in der Prädikatenlogik mit endlichen Modellen ist unentscheidbar.

(ohne Beweis)

Korollar: Es gibt kein vollständiges und korrektes Beweissystem für Prädikatenlogik mit endlichen Modellen.

Beweis: Angenommen es gäbe ein solches System. Dann wäre logische Konsequenz und speziell auch Unerfüllbarkeit semi-entscheidbar.

Wir wissen, dass Erfüllbarkeit ebenfalls semi-entscheidbar ist.

Zusammen ergäbe sich also ein Entscheidungsverfahren für logisches Schließen – Widerspruch zu Trakhtenbrot. □

Endliche Modelle in der Praxis

Wozu endliche Modelle

In gewisser Weise ist Schließen mit endlichen Modellen also schwerer als mit unendlichen, weil man statt logischer Konsequenz nunmehr nur Nicht-Konsequenz semi-entscheiden kann

Trotzdem sind endliche Interpretationen in der Informatik praktisch relevant:

Eine endliche Interpretation $\mathcal{I}$ ist (im Wesentlichen) das gleiche wie eine relationale Datenbankinstanz.

Intuition:

- Prädikatsymbole p bezeichnen Tabellen
- Relationen $p^{\mathcal{I}}$ entsprechen den in der Datenbank gespeicherten Tabelleninhalten

Benannte Parameter

Relationale Datenbanken verwenden Namen für die Parameter (Spalten) in Relationen, anstatt sie mittels Reihenfolge zu addressieren:

linien:

Linie	**Typ**
85	Bus
3	Tram
F1	Fähre
...	...

haltestellen:

SID	**Name**	**Rollstuhl**
17	Hauptbahnhof	true
42	Helmholtzstr.	true
57	Stadtgutstr.	true
123	Gustav-Freytag-Str.	false
...	...	...

verbindung:

Von	**Zu**	**Linie**
57	42	85
17	789	3
...	...	...

Die einfache Arität der Prädikatenlogik wird durch ein Schema mit Namen (und oft auch Datentypen) ersetzt:

- linien[Linie:string, Typ:string]
- haltestellen[SID:int, Halt:string, Rollstuhl:bool]
- verbindung[Von:int, Zu:int, Linie:string]

Formeln = Anfragen

Benannt oder nicht – sofern die Parameter eine definierte Reihenfolge haben, kann man sie mit normalen prädikatenlogischen Atomen addressieren.

Beispiel: Die Formel

$$Q = \exists z_{\mathsf{Linie}}.(\mathsf{verbindung}(x_{\mathsf{Von}}, x_{\mathsf{Zu}}, z_{\mathsf{Linie}}) \wedge \mathsf{linien}(z_{\mathsf{Linie}}, x_{\mathsf{Typ}}))$$

hat drei freie Variablen. Für eine gegebene Datenbankinstanz (eindliche Interpretation) $\mathcal{I}$ bedeutet $\mathcal{I}, \{x_{\mathsf{Von}} \mapsto \delta_1, x_{\mathsf{Zu}} \mapsto \delta_2, x_{\mathsf{Typ}} \mapsto \delta_3\} \models Q$, dass es in der Datenbank eine Verbindung von δ_1 nach δ_2 vom Typ δ_3 gibt.

Formeln = Anfragen

Benannt oder nicht – sofern die Parameter eine definierte Reihenfolge haben, kann man sie mit normalen prädikatenlogischen Atomen addressieren.

Beispiel: Die Formel

$$Q = \exists z_{\text{Linie}}.(\text{verbindung}(x_{\text{Von}}, x_{\text{Zu}}, z_{\text{Linie}}) \wedge \text{linien}(z_{\text{Linie}}, x_{\text{Typ}}))$$

hat drei freie Variablen. Für eine gegebene Datenbankinstanz (eindliche Interpretation) $\mathcal{I}$ bedeutet $\mathcal{I}, \{x_{\text{Von}} \mapsto \delta_1, x_{\text{Zu}} \mapsto \delta_2, x_{\text{Typ}} \mapsto \delta_3\} \models Q$, dass es in der Datenbank eine Verbindung von δ_1 nach δ_2 vom Typ δ_3 gibt.

Das Beispiel illustriert:

Formeln (ev. mit freien Variablen) = Datenbankanfragen

Erfüllende Zuweisungen = Anfrage-Ergebnisse

Prädikatenlogik ≈ SQL

Was ist eine Datenbank-Anfrage?

- Syntax: Eine Anfrage Q ist ein Wort aus einer Anfragesprache
- Semantik: Jede Anfrage Q definiert eine Anfragefunktion f_Q, die für jede Datenbankinstanz $\mathcal{I}$ eine Ergebnisrelation $f_Q(\mathcal{I})$ liefert

Prädikatenlogik $\approx$ SQL

Was ist eine Datenbank-Anfrage?

- Syntax: Eine Anfrage Q ist ein Wort aus einer Anfragesprache
- Semantik: Jede Anfrage Q definiert eine Anfragefunktion f_Q, die für jede Datenbankinstanz $\mathcal{I}$ eine Ergebnisrelation $f_Q(\mathcal{I})$ liefert

Beispiel: für eine prädikatenlogische Formel Q mit freien Variablen $x_1, \ldots, x_n$ ist f_Q die Funktion, die $\mathcal{I}$ auf die Relation $f_Q(\mathcal{I}) = \{\langle \delta_1, \ldots, \delta_n \rangle \mid \mathcal{I}, \{x_1 \mapsto \delta_1, \ldots, x_n \mapsto \delta_n\} \models Q\}$ abbildet.

Prädikatenlogik ≈ SQL

Was ist eine Datenbank-Anfrage?

- Syntax: Eine Anfrage Q ist ein Wort aus einer Anfragesprache
- Semantik: Jede Anfrage Q definiert eine Anfragefunktion f_Q, die für jede Datenbankinstanz $\mathcal{I}$ eine Ergebnisrelation $f_Q(\mathcal{I})$ liefert

Beispiel: für eine prädikatenlogische Formel Q mit freien Variablen $x_1, \ldots, x_n$ ist f_Q die Funktion, die $\mathcal{I}$ auf die Relation $f_Q(\mathcal{I}) = \{\langle \delta_1, \ldots, \delta_n \rangle \mid \mathcal{I}, \{x_1 \mapsto \delta_1, \ldots, x_n \mapsto \delta_n\} \models Q\}$ abbildet.

Mit so einer allgemeinen Definition kann man sehr unterschiedliche Anfragesprachen über ihre Anfragefunktion vergleichen

Satz: Die Menge der durch prädikatenlogische Formeln Q darstellbaren Anfragefunktionen f_Q ist genau die Menge der Anfragefunktionen, die durch einfache SQL-Anfragen darstellbar sind.

„einfaches SQL“: Relationale Algebra, der Kern von SQL; SELECT, JOIN, UNION, MINUS, aber keine komplexeren Features wie WITH RECURSIVE etc. Außerdem keine Datentypen, da wir diese in Logik nicht eingeführt haben.

Relationale Algebren

Datenbankanfragen werden oft in relationaler Algebra dargestellt, bei der man Relationen mit Operationen zu einem Anfrageergebnis kombiniert

Beispiel: Die Anfrage

$$Q = \exists z_{\text{Linie}}.(\text{verbindung}(x_{\text{Von}}, x_{\text{Zu}}, z_{\text{Linie}}) \wedge \text{linien}(z_{\text{Linie}}, x_{\text{Typ}}))$$

entspricht einer (natürlichen) Join-Operation ($\wedge$) mit anschließender Projektion ($\exists$):

$$\pi_{\text{Von,Zu,Linie}}(\text{verbindung} \bowtie \text{linien})$$

Anmerkung: SQL hat noch einen leicht anderen Stil. Variablen stehen dort für ganze Tabellenzeilen und man verwendet Notation der Form „linien.Typ“, um auf deren Einträge zuzugreifen („Tuple-Relational Calculus“). Das ändert an der Ausdrucksstärke nichts.

Anfragebeantwortung als Modell Checking

Erkenntnis: Die wesentliche Berechnungsaufgabe bei der Beantwortung von Datenbankabfragen ist das folgende Entscheidungsproblem:

Das Auswertungsproblem (Model Checking) der Prädikatenlogik lautet wie folgt:

Gegeben: Eine Formel Q mit freien Variablen $x_1, \ldots, x_n$; eine endliche Interpretation $\mathcal{I}$; Elemente $\delta_1, \ldots, \delta_n \in \Delta^{\mathcal{I}}$
Frage: Gilt $\mathcal{I}, \{x_1 \mapsto \delta_1, \ldots, x_n \mapsto \delta_n\} \models Q$?

Anfragebeantwortung als Modell Checking

Erkenntnis: Die wesentliche Berechnungsaufgabe bei der Beantwortung von Datenbankabfragen ist das folgende Entscheidungsproblem:

> Das Auswertungsproblem (Model Checking) der Prädikatenlogik lautet wie folgt:
>
> **Gegeben:** Eine Formel Q mit freien Variablen $x_1, \ldots, x_n$; eine endliche Interpretation $\mathcal{I}$; Elemente $\delta_1, \ldots, \delta_n \in \Delta^{\mathcal{I}}$
> **Frage:** Gilt $\mathcal{I}, \{x_1 \mapsto \delta_1, \ldots, x_n \mapsto \delta_n\} \models Q$?

Naive Methode der Anfragebeantwortung:

- Betrachte alle $(\Delta^{\mathcal{I}})^n$ möglichen Ergebnisse
- Entscheide jeweils das Auswertungsproblem

Praktisch relevante Frage:

Wie schwer ist das Auswertungsproblem?

Zusammenfassung und Ausblick

Die prädikatenlogische Resolution ist ein vollständiges und korrektes Verfahren für die Unerfüllbarkeit logischer Formeln

In gewissem Sinne ist Prädikatenlogik eine Kurzschreibweise für möglicherweise unendliche aussagenlogische Theorien

Bei Beschränkung auf endliche Modelle gibt es kein vollständiges und korrektes Verfahren zum logischen Schließen – dafür wird Erfüllbarkeit semi-entscheidbar

Auswertungsproblem auf endlichen Modellen = Anfragebeantwortung in Datenbanken

Was erwartet uns als nächstes?

- Komplexität des Auswertungsproblems
- Gödel
- Probeklausur und 3. Repetitorium

Bildrechte