Logik-Programmierung und Negation

Sebastian Marius Kirsch
skirsch@moebius.inka.de

12. Juli 2003

1 Überblick

- Motivation und Vorbetrachtungen
- Rückblick: Herbrand-Theorie, Vervollständigung, SLDNF-Resolution
- SLD-CNF-Resolution
- Dreiwertige Logik, dreiwertige Herbrandmodelle
- Wohlfundierte Semantik
- SLS-Resolution

2 Motivation

- Monotonie: $\Gamma \vdash L \Rightarrow \Gamma \cup \Gamma' \vdash L$
- klassische Logik ist monoton
- Negation führt zu Nichtmonotonie.(Bsp: $\emptyset \models \neg p$, aber $\{p\} \not\models \neg p$)
- klassische Interpretation der Semantik mit Negation führt zu einem Theorembeweiser erster Ordnung, mit den damit verbundenen unentscheidbaren Problemem.
- Warum ist Logikprogrammierung mit Negation trotzdem ein interessantes Gebiet, obwohl das Grundproblem nicht entscheidbar ist?
- Semantik mit Negation ist teilw. berechenbar. In der Vorlesung hatten wir das SLDNF-Kalkül gesehen.
- Alle vorgestellten Kalküle erfüllen immer noch die Rationalität: $\Gamma \not\vdash \neg A$ und $\Gamma \vdash L \Rightarrow \Gamma \cup \{A\} \vdash L$

- Datenbanken (negative Information implizit)

- „gesunder Menschenverstand“ ist nichtmonoton (vergl. Default-Logiken) Beispiel: „Peter hat sich verliebt, die beiden sind sehr glücklich“ – hier werden die meisten zuerst annehmen, dass Peter nun eine Freundin hat. Erfährt man später, dass Peter schwul ist, dann gilt die Folgerung, dass er eine Freundin hat, nicht mehr; stattdessen wird gefolgert, dass er einen Freund hat.

- Um das Grundproblem der KI zu lösen, also ein System zu schaffen, das den Verstand des Menschen approximiert, ist nichtmonotone Logik also unverzichtbar.

3 Abhängigkeitsgraphen

- Formalismus, der in der Vorlesung nicht benutzt wurde, jedoch teilweise zu sehr eingängingen Formulierungen der Anforderungen an Logikprogramme führt.

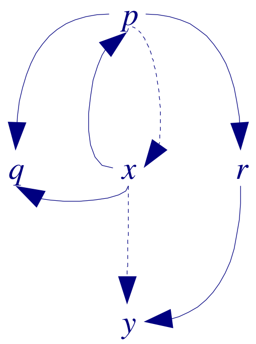
- gerichteter Graph mit positiven und negativen Kanten

- Kanten von der Relation im Kopf einer Klausel zu jeder Relation im Rumpf

- p hängt gerade (ungerade) von q ab, wenn es einen Pfad mit einer geraden (ungeraden) Anzahl negativer Kanten gibt.

- *call consistency*: Keine Relation hängt ungerade von sich selbst ab.

$$
\begin{array}{c}
p \leftarrow q \\
p \leftarrow r, \neg x \\
q \\
r \leftarrow y \\
x \leftarrow q \\
x \leftarrow p \\
x \leftarrow \neg y, r
\end{array}
$$



4 Rückblick: Herbrand-Theorie

- Idee: keine beliebigen Interpretationen, stattdessen werden Konstanten und Funktionsanwendungen durch sich selbst interpretiert.

- Herbrand-Universum $\mathcal{U}_{\mathrm{P}}$: Menge aller variablenfreien Terme, die aus Funktions- und Konstantensymbolen in P gebildet werden können

- Alternative zum Herbrand-Universum: Postuliere eine universelle Sprache, in der alle Programme und Fragen ausgedrückt werden.

- Vorteil: Löst das Problem von Sprachelementen, die in der Frage, aber nicht im Programm vorkommen.
- Herbrand-Basis $\mathcal{B}_P$: Menge aller Grundatome
- Repräsentiere Herbrand-Interpretation I als Menge aller Grundatome, die unter I wahr sind.

5 Rückblick: Programmvervollständigung

- Ausgangspunkt für die Semantik der Negation in der Vorlesung war die *closed world assumption* (CWA): $\neg A$ wird angenommen, wenn A nicht bewiesen werden kann.
- i. A. unentscheidbar, ob ein Literal A aus den Voraussetzungen bewiesen werden kann oder nicht bewiesen werden kann.
- Vervollständigung ist Formalisierung der *closed world assumption*
- Verstärkung der Semantik von Logikprogrammen
- Implikation wird ersetzt durch Äquivalenz
- Weitere Schritte zur Behandlung von Funktionsanwendungen, Gleichheitsaxiome, etc., sollen hier nicht aufgeführt werden.
- Beispiel: $P = \{p \leftarrow q; p \leftarrow r\}$, $\text{Comp}(P) = \{p \leftrightarrow q \vee r\}$
- $\text{Comp}(P)$ kann inkonsistent sein: $P = \{p \leftarrow \neg p\}$, $\text{Comp}(P) = \{p \leftrightarrow \neg p\}$
- wenn P *call-consistent* ist, hat $\text{Comp}(P)$ ein Herbrand-Modell
- Ergebnis aus der Vorlesung: Für stratifizierte Logikprogramme ist $\text{Comp}(P)$ erfüllbar.
- *call-consistency* ist eine wesentlich schwächere Anforderung
- Beispiel für den Wettlauf in der Logikprogrammierung: Das Grundproblem ist unentscheidbar; eine der aktuellen Aufgabenstellungen ist stattdessen, Anforderungen an Logikprogramme möglichst allgemein zu fassen, so dass man immer noch zu Ergebnissen kommt.

6 Rückblick: SLDNF

- endliche, berechenbare Alternative zur CWA: *negation by finite failure*
- nur variablenfreie negative Literale können ausgewählt werden

9 Dreiwertige Herbrand-Interpretationen

- Dreiwertige Herbrand-Interpretation $I = (I^+, I^-), I^+, I^- \subseteq \mathcal{B}_P$
- I ist total, wenn $I^+ \cup I^- = \mathcal{B}_P$
- I ist konsistent, wenn $I^+ \cap I^- = \emptyset$
- zweiwertige Interpretation I entspricht dreiwertiger $(I, \mathcal{B}_P \backslash I)$
- Informationsordnung: $I \subseteq J$ gdw. $I^+ \subseteq J^+$ und $I^- \subseteq J^-$
- $\subseteq$ ist eine induktive Ordnung auf der Menge der konsistenten dreiwertigen Herbrand-Interpretationen
- konsistente dreiwertige Herbrand-Interpretationen bilden damit einen Verband
- Wahrheit und Falschheit von Grundatomen verhalten sich monoton bzgl. $\subseteq$

10 Bottom-up-Charakterisierung von Herbrand-Modellen

- $H_{i+1}(P) = H_i(P) \cup$ Menge aller Grundatome $A = p(u_1, \ldots, u_n)$, für die es eine Grundinstanz $A \leftarrow A_1, \ldots, A_l$ einer Regel von P gibt, so daß $A_j \in H_i(P), j = 1, \ldots, l$
- Andere Formulierung: Operator $T_p(I) = \{A | A \leftarrow A_1, \ldots, A_l \in \text{ground}(P), I \models A_1, \ldots, A_l\}$
- H_{min} ist der kleinste Fixpunkt von T_P und kann mit endlich vielen Schritten von $T_P(\emptyset)$ erreicht werden.

11 Analogon für dreiwertige Logik

$T3_P(I) = (T, F)$

$$T = \{A | \exists A_1, \ldots, A_l (A \leftarrow A_1, \ldots, A_l \in \text{ground}(P), \\ A_1, \ldots, A_l \text{ ist wahr in } I)\}$$

$$F = \{A | \exists A_1, \ldots, A_l (A \leftarrow A_1, \ldots, A_l \in \text{ground}(P), \\ A_1, \ldots, A_l \text{ impliziert, dass } A \text{ in I falsch ist})\}$$

12 T_P und Vervollständigung

- für Herbrandinterpretationen gilt $I \models \text{Comp}(P)$ gdw. $T_P(I) = I$
- wenn T_P einen Fixpunkt hat, ist dieser ein Modell für $\text{Comp}(P)$
- bei dreiwertiger Logik gilt analog $I \models_3 \text{Comp}(P)$ gdw. $T3_P(I) = I$
- $T3_P$ ist im Gegensatz zu T_P monoton
- Wenn I konsistent ist (z. B. $I = (\emptyset, \emptyset)$), so ist $T3_P(I)$ konsistent.
- $T3_P$ hat einen kleinsten Fixpunkt auf dem Verband der konsistenten dreiwertigen Herbrand-Interpretationen (Knaster-Tarski-Theorem)
- Knaster-Tarski-Kleene-Theorem: Eine stetige Funktion f auf einem Verband $(D, \subseteq)$, wobei $\subseteq$ eine vollständige Halbordnung ist, hat einen kleinsten Fixpunkt $\text{lfp}(f)$.
- Dieser kleinste Fixpunkt von $T3_P$ ist ein konsistentes Modell von $\text{Comp}(P)$
- Das bedeutet: $\text{Comp}(P)$ kann nach zweiwertiger Logik inkonsistent sein, hat aber trotzdem ein konsistentes dreiwertiges Modell.
- Im Zweifelsfall ist das dreiwertige Modell $(\emptyset, \emptyset)$, man kann also keine weiteren Folgerungen aus diesem Modell ziehen – aber das ist immer noch ein Vorteil gegenüber einer inkonsistenten Vervollständigung, aus der man beliebige Folgerungen ziehen kann.

13 Wohlfundierte Semantik

- definite Logikprogramme haben ein eindeutig bestimmtes kleinstes Herbrandmodell
- normale Logikprogramme können mehrere kleinste Modelle haben, z. B. $P = \{p \leftarrow \neg q; q \leftarrow \neg p\}$ hat zwei kleineste Herbrandmodelle $H_1 = \{p\}$ und $H_2 = \{q\}$
- Mehrere Ansätze existieren, welches dieser kleinsten Modelle vorzuziehen ist, z. B. stabile Modelle, perfekte Modelle etc.
- Anderer Anstatz: *ein* dreiwertiges Modell $\text{WFM}(P)$ statt mehreren zweiwertigen
- dreiwertige Logik zur Erzeugung nicht nötig.
- Idee: Manche Atome *müssen* wahr sein, unabhängig von der Semantik negativer Literale (Fakten), und manche Atome *müssen* falsch sein (weil sie nicht mit dem Kopf einer Klausel unfizieren.)
- benutze diese Information, um das Programm zu vereinfachen

- wenn der Wahrheitswert von allen Atomen so zu entscheiden ist, ist das Programm *effektiv stratifizierbar*
- wenn nicht: versuche nicht, den Wahrheitswert von Atomen zu raten, sondern gebe einfach ein dreiwertiges Modell zurück
- Nicht alle erwarteten Atome werden inferiert
- $P = \{p \leftarrow \neg q; q \leftarrow \neg p; r \leftarrow p; r \leftarrow q\}, p \vee q \Rightarrow r$, in allen kleinsten Modellen gilt entweder p oder q, und wegen $r \leftarrow p \vee q$ gilt auch r, aber $r \notin \mathsf{WFM}(P)$

14 Fixpunkt-Charakterisierung von WFM

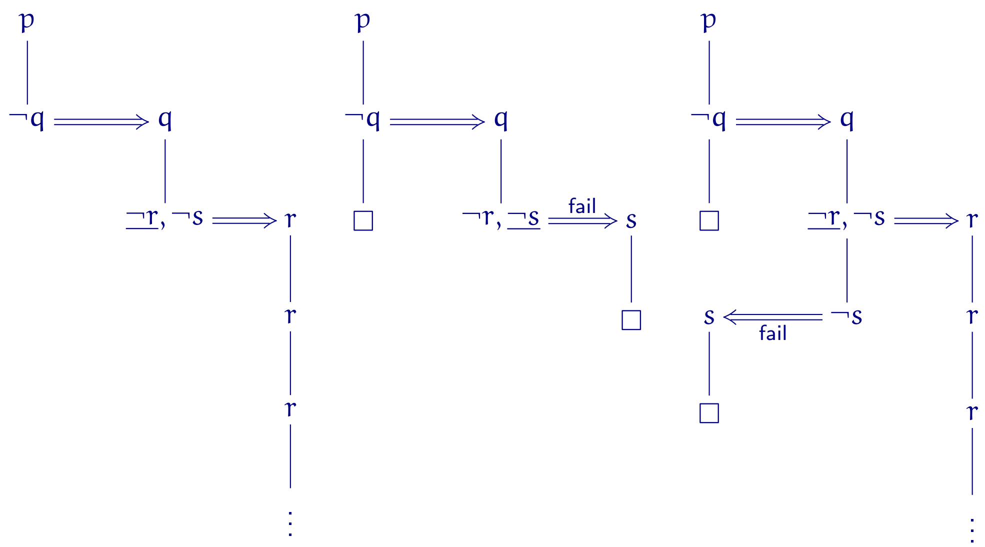
- $I_3(P) = (H_{min}(P^+), \overline{H_{min}(P^-)})$
  P^+: P ohne Klauseln mit negativen Literalen
  P^-: P ohne negative Literale
  P^+ und P^- sind definite Logikprogramme, besitzen also ein kleinstes Herbrandmodell.
- $\Phi_P(I) = I_3(P \backslash I)$
  $P \backslash I$ aus ground(P), indem man alle Klauseln mit Literalen löscht, die falsch sind in I, sowie alle Literale, die wahr sind in I
- $\Phi_P(I)$ ist monoton bzgl. der Informationsordnung $\subseteq$
- also existiert ein kleinster Fixpunkt von Φ_P (gleiches Argument wie bei $T3_P$)
- kann von $(\emptyset, \emptyset)$ erreicht werden
- dieser kleinste Fixpunkt ist das *wohlfundierte Modell* WFM(P)

15 SLS-Resolution

- Resolution für stratifizierte Logikprogramme (*linear resolution for stratified clauses*)
- auch für normale Logikprogramme anwendbar
- Idee: ersetze Negation per *finite failure* durch Negation per *failure*
- damit nicht mehr berechenbar, Implementationen können nur approximieren
- Beispiel: Implementiere Zyklentest für unendliche missglückte Äste
- korrekt bzgl. *well-founded semantics*

16 SLS-Resolution: Beispiel

$P = \{p \leftarrow \neg q; q \leftarrow \neg r, \neg s; r \leftarrow r; s\}$



SLDNF-Baum SLS-Baum SLS-Baum

- Der linke SLDNF-Baum mit Auswahlregel „linkes Literal“ schlägt fehlt, da man in einen unendlichen Ast gerät.
- Der mittlere Baum ist ein SLDNF-Baum mit Auswahlregel „rechtes Literal“ und gleichzeitig ein SLS-Baum.
- Beim rechten SLS-Baum werden die Vorteile der SLS-Resolution deutlich: Obwohl man in einen unendlichen Aste gerät, kann das Literal wegresolviert werden, da der Ast missglückt ist.

17 Schlussstrich

- SLDNF-Resolution ist nicht der Weisheit letzter Schluss
- Dreiwertige Logik führt zu konsequenteren Beschreibungsmöglichkeiten
- Logikprogrammierung mit Negation ist ein weites und sehr aktives Forschungsfeld...

Literatur

[1] Krysztof R. Apt, Roland N. Bol: *Logic Programming and Negation: A Survey.* In: Journal of Logic Programming 20, 9-71, 1994.