Wiederholung zur SLD-Resolution (1)

SLD-Resolution:

- Selektionsfunktion: Wählt Atom der Anfrage.
  Nächstes zu bearbeitendes Atom. Normalerweise einfach das erste.
- Lineare Resolution: Beweisverfahren.
  Linear, weil jeder Schritt das Ergebnis des letzten verwendet.
- Definite Hornklauseln: $A \leftarrow B_1 \wedge \cdots \wedge B_m$.
  Dabei sind A und B_i Atome der Form $p(t_1, \ldots, t_n)$.

Resolutions-Schritt:

- Anfrage: ?- $A_1, A_2, \ldots, A_n$.
- Selektionsfunktion wählt z.B. A_1.
- Regel: B_0 :- $B_1, \ldots, B_m$.
- Nenne Variablen der Regel so um, daß von allen bisher verwendeten verschieden: B_0' :- $B_1', \ldots, B_m'$.
  Möglich, weil Variablennamen nur Regel-lokal von Bedeutung.
  Nötig, sonst nicht vollständig: $\Phi := \{p(a) \leftarrow q(X),\ q(b)\}$, $\psi := p(X)$.
- Allgemeinster Unifikator von A_1 und B_0': θ.
- Neue Anfrage: ?- $B_1'\theta, \ldots, B_m'\theta, A_2\theta, \ldots, A_n\theta$.
  Dies heißt auch Resolvente von $A_1 \wedge \cdots \wedge A_n$ und $B_0 \leftarrow B_1 \wedge \cdots \wedge B_m$.

Wiederholung zur SLD-Resolution (2)

SLD-Ableitung für ψ:

- Folge von Anfragen $\psi_0, \psi_1, \ldots$, wobei
- ψ_0 die vom Benutzer gegebene Anfrage ψ ist,
- ψ_i ist Resolvente von ψ_{i-1} und einer Regel aus Φ.

Klassifizierung von Ableitungen:

- Erfolgreich: endet in $\top$.
- Gescheitert: endet in ψ_m ohne anwendbare Regel.

  Eine Regel ist auf ψ_m anwendbar, wenn ihr Kopf mit dem in ψ_m ausgewählten Atom A unifizierbar ist (nach Umbenennung der Variablen).
- Unendlich: endet nicht.

Berechnete Antwort:

- Seien $X_1, \ldots, X_n$ die Variablen der Anfrage ψ.
- Sei $\psi_0, \ldots, \psi_m$ eine erfolgreiche Ableitung für ψ, und $\theta_1, \ldots, \theta_m$ die verwendeten Unifikatoren.
- Dann ist $\theta_1 \circ \cdots \circ \theta_m|_{X_1,\ldots,X_n}$ die von dieser Ableitung berechnete Antwort auf ψ.

  Komposition der Unifikatoren und Restriktion auf Antwortvariablen.
- Die leere Substitution $\{\}$ wird „yes“ geschrieben.

Wiederholung zur SLD-Resolution (3)

Beachte:

- Es gibt mehrere Ableitungen für eine Anfrage.
  Jede erfolgreiche Ableitung berechnet ja auch nur eine Antwort.
- Verzweigung, wenn mehrere Regeln anwendbar.

Ableitungsbaum für ψ:

- Jeder Knoten ist mit einer Anfrage markiert.
- Wurzel mit gegebener Anfrage ψ markiert.
- Ein mit ψ_i markierter Knoten hat einen Nachfolgeknoten für jede auf ψ_i anwendbare Regel.
  Der Nachfolgeknoten ist mit der entsprechenden Resolvente markiert.

Prolog:

- Die Prolog-Selektionsfunktion wählt immer das erste Literal der Anfrage.
- Prolog sucht den Ableitungsbaum „Tiefe zuerst“ ab und verfängt sich in unendlichen Zweigen.
  „Breite zuerst“ wäre theoretisch besser, läßt sich aber nicht effizient implementieren. Kompromiß: stufenweise erhöhte Tiefenschranken.
- Kein Occur-Check: falsche Resolventen möglich.

Korrektheit der SLD-Resolution

Satz (Korrektheit):

Ist θ eine berechnete Antwortsubstitution auf die Anfrage $A_1 \wedge \cdots \wedge A_n$, so gilt $\Phi \vdash \forall((A_1 \wedge \cdots \wedge A_n)\theta)$.

Beweis mit Induktion über der Länge der Ableitung.

Beispiele:

- Φ: $p(f(X)) \leftarrow q(X).$
  $q(a).$
  ψ: $p(Y).$
  θ: $\{Y/f(a)\}.$
  Es gilt: $\Phi \vdash p(f(a)).$
- Φ: $p(X) \leftarrow q(X).$
  $q(Y).$
  ψ: $p(Z).$
  θ: $\{\}.$
  Es gilt: $\Phi \vdash \forall Z\, p(Z).$

Beachte:

- Die Korrektheit gilt nur mit dem „Occur-Check“.

  Übliche Prolog-Systeme können also inkorrekte Antworten liefern, z.B.:

  Φ: $p \leftarrow q(X, X).$
  $q(X, f(X)).$
  ψ: $p.$
  θ: $\{\}.$
  Es gilt: $\Phi \not\vdash p.$

Vollständigkeit der SLD-Resolution (1)

Satz (Vollständigkeit):

Sei θ eine Substitution mit $\Phi \vdash \forall(\psi\theta)$. Dann gibt es eine berechnete Antwortsubstitution θ_1 und eine Substitution θ_2 mit $\theta = \theta_1 \circ \theta_2$.

Beispiel/Bemerkung:

- Es werden nur allgemeinste Antworten berechnet.
- Φ: $p(f(X))$.
  $q(a)$.
  ψ: $p(Y)$.
  θ: $\{Y/f(a)\}$.
  Es gilt: $\Phi \vdash \psi\theta$.
- Die SLD-Resolution berechnet $\theta_1 := \{Y/f(X)\}$. Es gilt $\theta = \theta_1 \circ \{X/a\}$.

Beachte:

- Prolog-Interpreter finden wegen der Tiefensuche nicht alle erfolgreichen Ableitungen.
- Φ: $p(X) \leftarrow p(X)$.
  $p(a)$.
  ψ: $p(Y)$.
  θ: $\{X/a\}$.
  Prolog gerät in eine Endlosschleife, bevor es θ findet.
- Das Vollständigkeitsresultat gilt also nur bei Breitensuche oder unter der Voraussetzung, daß der Prolog-Interpreter anhält.

Vollständigkeit der SLD-Resolution (2)

Beachte (Forts.):

Auch der Cut kann die Vollständigkeit zerstören.

Lemma (Komposition von SLD-Ableitungen):

Sind $A_1, \ldots, A_n$ variablenfreie Atome und gibt es für jedes A_i eine erfolgreiche SLD-Ableitung, dann gibt es eine erfolgreiche Ableitung für $A_1 \land \cdots \land A_n$.

Beweis: Einfach hintereinanderhängen (plus Rest an jede Anfrage).

Lemma (Vollständigkeit ohne Variablen):

Sei A ein variablenfreies Atom, das im minimalen Herbrandmodell von Φ gilt. Dann gibt es eine erfolgreiche SLD-Ableitung von A bezüglich $G(\Phi)$ (Grundinstanzen von Φ).

Beweis: Induktion über den Iterationsstufen der Fixpunktsemantik.

Verallgemeinerte Ableitung:

Ableitung, bei der beiliebige Unifikatoren benutzt werden können, nicht nur allgemeinste.

Vollständigkeit der SLD-Resolution (3)

Lemma (MGU Lemma):

- Habe $A_1 \wedge \cdots \wedge A_n$ eine erfolgreiche verallgemeinerte Ableitung mit Unifikatoren $\theta_1, \ldots, \theta_l$. Dann gibt es auch eine Ableitung mit allgemeinsten Unifikatoren $\theta'_1, \ldots, \theta'_l$ und eine Substitution θ' mit $\theta_1 \circ \cdots \circ \theta_l = \theta'_1 \circ \cdots \circ \theta'_l \circ \theta'$.
- Beweis: Induktion über der Länge der Ableitung.

Lemma (Vollständigkeit bzgl. Antwort „yes“):

- Gelte $\Phi \vdash \forall(A_1 \wedge \cdots \wedge A_n)$. Dann gibt es eine erfolgreiche Ableitung für $A_1 \wedge \cdots \wedge A_n$ mit der berechneten Antwort $\{\}$.
- Beweis: Ersetze Variablen durch neue Konstanten. Vollständigkeit bzgl. $G(\Phi)$. MGU-Lemma: Ableitung bzgl. $G(\Phi) \longrightarrow$ Ableitung bzgl. Φ.

Lemma (Lifting Lemma):

- Habe $(A_1 \wedge \cdots \wedge A_n)\theta$ eine erfolgreiche Ableitung mit Unifikatoren $\theta_1, \ldots, \theta_l$. Dann gibt es auch eine erfolgreiche Ableitung von $A_1 \wedge \cdots \wedge A_n$ mit Unifikatoren $\theta'_1, \ldots, \theta'_l$ und eine Substitution θ' mit $\theta \circ \theta_1 \circ \cdots \circ \theta_l = \theta'_1 \circ \cdots \circ \theta'_l \circ \theta'$.

Die Antwort „No“

Berechnung:

- Die Antwort „no“ wird ausgegeben, wenn es keine erfolgreichen oder unendlichen Ableitungen gibt („finite failure“).

  Dabei ist die Selektionsfunktion fest gewählt.

Korrektheit:

- Wenn für ψ die Antwort „no“ berechnet wird, so gilt $\mathit{comp}(\Phi) \vdash \forall(\neg\psi)$.

Vollständigkeit:

- Eine Ableitung heißt fair, wenn sie endlich ist, oder jedes vorkommende Atom (bzw. eine weiter instanziierte Version) einmal ausgewählt wird.
- Wenn $\mathit{comp}(\Phi) \vdash \forall(\neg\psi)$ gilt, so gibt es keine erfolgreichen oder fairen unendlichen Ableitungen.
- Ohne Fairness würde die Vollständigkeit nicht gelten:

  $$\Phi: \begin{array}{l} p \leftarrow q \wedge r(a). \\ q \leftarrow q. \\ r(b). \end{array}$$

  Es gilt $\mathit{comp}(\Phi) \vdash \neg p$, aber Prolog (unfair) würde nicht terminieren.