Anforderungsanalyse für ein generisches Visualisierungssystem für verschiedene Deduktionsverfahren

- Christian Braun -

15.04.1999

Übersicht über den Vortrag

- Einleitung und Motivation
- Vorstellung des begrifflichen Rahmens
- Besprechung einiger Verfahren
- Grenzen

Einleitung und Motivation

In dieser Projektarbeit sollen die Anforderungsbeschreibung und ein detailliertes Konzept für das Design der Anwendung *SNARKS II* vorgestellt werden.

Ziele von *SNARKS II*:

- Erweiterung auf weitere Verfahren
  Denkbar wären hier zum Beispiel andere *Tableaux*- oder *SLD*-Verfahren.

- Verbesserung der Darstellungsmöglichkeiten

Vorstellung des begrifflichen Rahmens

Wir unterscheiden zwei Ebenen: eine Deduktions- und eine Darstellungsebene.

- *Deduktionsebene*
  Die Deduktionsebene beschäftigt sich mit möglichen Bestandteilen von Ableitungen, möglichen Beziehungen zwischen diesen Bestandteilen und den vorkommenden Ableitungsstrukturen.

- *Darstellungsebene*
  Die Darstellungsebene beschäftigt sich mit den wünschenswerten Möglichkeiten der Visualisierung dieser Ableitungen.

Darstellungsebene
Deduktionsebene Deduktionssystem

Oder auch:

Darstellungsebene
Deduktionsebene
Deduktionssystem

Benutzer

Es gibt zwei Arten von Benutzern:

- Konfigurator

  Ein Konfigurator schließt ein Deduktionssystem an die Deduktionsebene bzw. die Darstellungsebene an. Er hat eine Reihe von Entscheidungen zu treffen, wie das zu visualisierende Verfahren in den begrifflichen Rahmen eingepasst wird.

- Endanwender

  Ein Endanwender benutzt das konfigurierte System zur Visualisierung und Analyse von Ableitungen in (möglicherweise) verschiedenen Verfahren.

Diese Projektarbeit richtet sich im besonderen Maße an einen Konfigurator.

Die Deduktionsebene

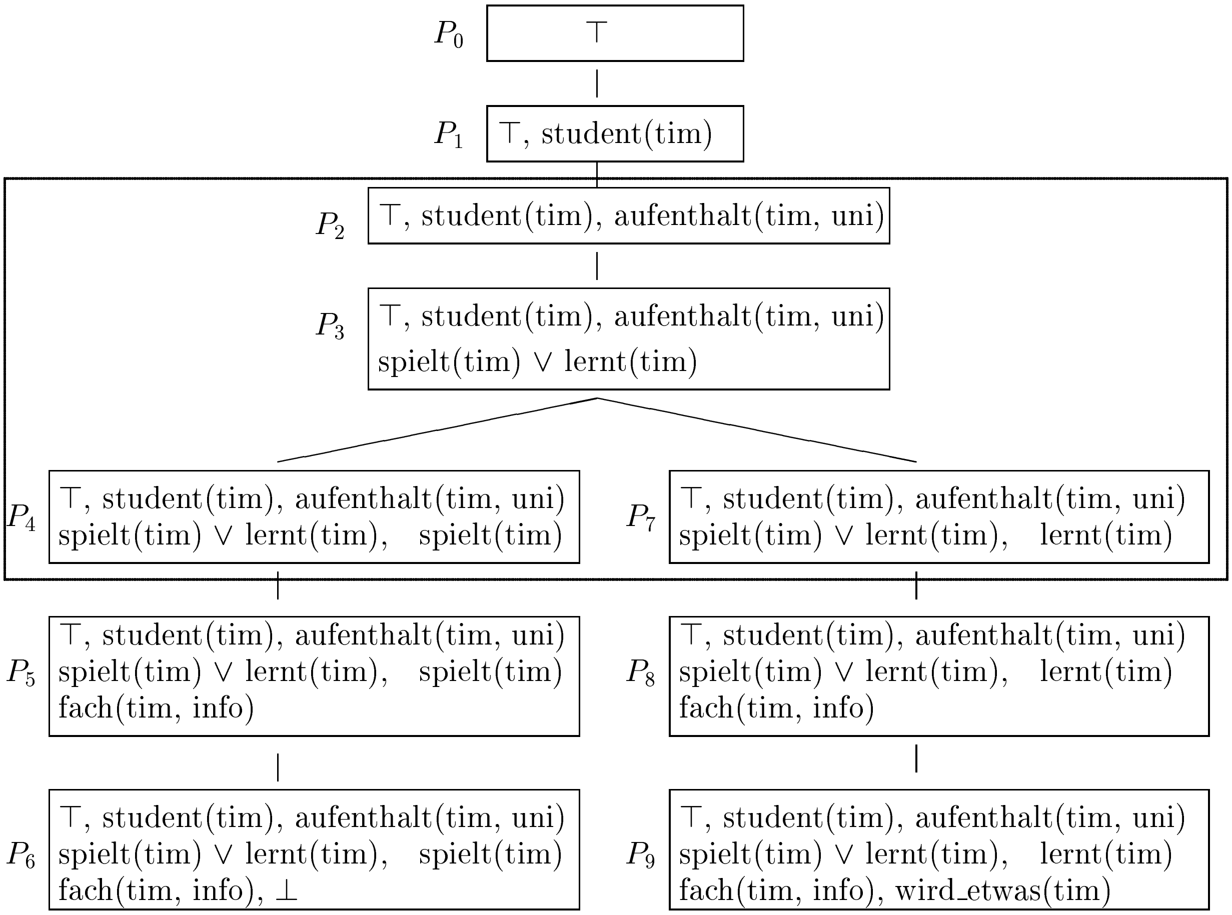
- Partikel, Partikelliste, initiale Partikelliste
- Eigenschaften
- Transformator und Transformatorenliste
- Regel
- Substitution und Substitutionsliste
- Expansion, Teilexpansion, potentielle Teilexpansion
- Strategie

Beispiel: PUHR-Tableau

- *Partikel*:
  Grundatom oder Disjunktion von Grundatomen
- *Partikelliste*:
  Jede Partikelliste definiert eine Interpretation.
- *Initiale Partikelliste*:
  Die initiale Partikelliste besteht nur aus $\top$.
- *Eigenschaft*:
  Partikelliste, die $\perp$ enthält, ist „geschlossen“. Nicht-expandierbare Partikelliste, die $\perp$ *nicht* enthält, ist „offen“.
- *Transformatoren*:
  Bereichsbeschränkte Implikationen $A_1 \wedge \ldots \wedge A_n \Rightarrow B_1 \vee \ldots \vee B_k$.
- *Regeln*:
  PUHR- und Splitting-Regel.
- *Substitutionen*:
  Es gibt nur die Substitutionen, die eine Teilexpansion betreffen. Jede Substitutionsliste ist einelementig.
- *Expansion*:
  Eine Teilexpansion modelliert eine Anwendung der PUHR-Regel oder eine Anwendung der Splitting-Regel.

Beispiel: PUHR-Tableau II

$$
\begin{aligned}
lernt(Person) &\Rightarrow wird\_etwas(Person). && (1)\\
student(Person) \wedge aufenthalt(Person, uni) &\Rightarrow spielt(Person) \vee lernt(Person). && (2)\\
assi(Person) \wedge aufenthalt(Person, uni) &\Rightarrow lehrt(Person) \vee forscht(Person). && (3)\\
fach(Person, informatik) \wedge spielt(Person) &\Rightarrow \bot. && (4)\\
\top &\Rightarrow student(tim). && (5)\\
\top &\Rightarrow fach(tim, info). && (6)\\
\top &\Rightarrow aufenthalt(tim, uni). && (7)
\end{aligned}
$$



Partikelliste	P_2
Eigenschaften	ε
Teilexpansionen	$< P_2 \rightarrow P_3 >$
Potentielle Teilexpansionen	$< P_2 \rightarrow X >$

Teilexpansion	$P_2 \rightarrow P_3$
Vorgänger	P_2
Regel	PUHR
Transformator	$T[2]$
Substitutionsliste	$< [tim/Person] >$
Verwendete Partikel	$< P_2[2], P_2[3] >$
Nachfolger	P_3
korrespondiert	$(P_2[i], P_3[i]), \quad 1 \leq i \leq 3$
abhängig	$(P_2[2], P_3[4])$
	$(P_2[3], P_3[4])$

Potentielle Teilexpansion	$P_2 \rightarrow X$
Vorgänger	P_2
Regel	PUHR
Transformator	$T[6]$
Substitutionsliste	$< [] >$
Verwendete Partikel	$< P_2[1] >$

Partikelliste	P_3
Eigenschaften	ε
Teilexpansionen	$< P_3 \rightarrow P_4 P_7 >$
Potentielle Teilexpansionen	$< P_3 \rightarrow Y >$

Teilexpansion	$P_3 \rightarrow P_4 P_7$
Vorgänger	P_3
Regel	Splitting
Transformator	—
Substitutionsliste	$< [] >$
Verwendete Partikel	$< P_3[4] >$
Nachfolger	P_4
korrespondiert	$(P_3[i], P_4[i]), \quad 1 \leq i \leq 4$
abhängig	$(P_3[4], P_4[5])$
Nachfolger	P_7
korrespondiert	$(P_3[i], P_7[i]), \quad 1 \leq i \leq 4$
abhängig	$(P_3[4], P_7[5])$

Potentielle Teilexpansion	$P_3 \rightarrow Y$
Vorgänger	P_3
Regel	PUHR
Transformator	$T[6]$
Substitutionsliste	$< [] >$
Verwendete Partikel	$< P_3[1] >$

Beispiel: Smullyan Tableaux

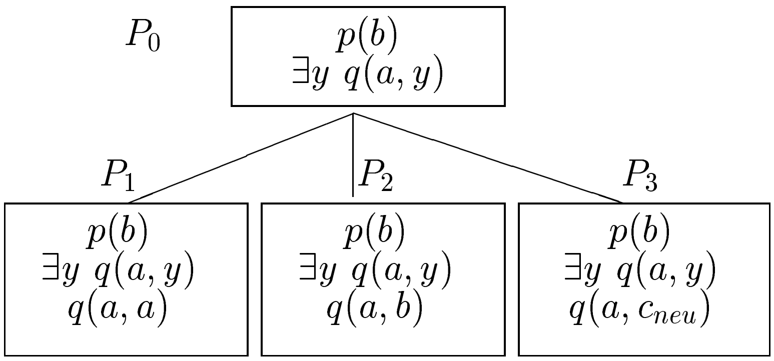
- *Partikel*:
  Partikel sind geschlossene Formeln der PL1S.
- *Partikelliste*:
  Reihenfolge der Partikel ist irrelevant.
- *Initiale Partikelliste*:
  Besteht aus der vom Benutzer vorgegebenen Formel.
- *Eigenschaft*:
  Partikelliste P ist „geschlossen“ gdw. es ψ in P gibt mit $\neg\psi \in P$.
  Nicht-expandierbare Partikelliste P ist „offen“ gdw. für jede Formel $\psi \in P$ gilt: $\neg\psi \notin P$.
- *Transformatoren*:
  —
- *Regeln*:
  $\alpha-, \beta-, \gamma-, \delta-$Regeln.
- *Substitutionen*:
  Es gibt nur die Substitutionen, die eine Teilexpansion betreffen. Jede Substitutionsliste ist einelementig.
- *Expansion*:
  Jede Teilexpansion hat zwei (β-Regel) oder eine Nachfolger-Partikellisten (sonst).
  Das Verfahren ist monoton.

Beispiel: EP-Tableaux

- *Partikel*:
  Partikel sind PRQ-Formeln.
- *Partikelliste*:
  Reihenfolge spielt keine Rolle.
- *Initiale Partikelliste*:
  Besteht aus der vom Benutzer vorgegebenen Formel.
- *Eigenschaft*:
  Partikelliste P ist „geschlossen“ gdw. es ψ gibt, mit $\psi \in P$ und $\neg\psi \in P$ oder wenn $\top \in P$.

  Partikelliste P ist „offen“ gdw. für jedes $\psi \in P$ gilt $\neg\psi \notin P$ und $\top \notin P$.
- *Transformatoren*:
  —
- *Regeln*:
  Erweiterte PUHR-Regel, sowie Regeln für $\exists, \vee$ und $\wedge$.
- *Substitutionen*:
  Substitutionen betreffen stets Teilexpansion. Substitutionsliste kann mehrelementig sein.
- *Expansion*:
  Eine Teilexpansion kann mehrere Nachfolger Partikellisten haben.

Beispiel: EP-Tableaux II



Partikelliste	P_0
Eigenschaften Teilexpansionen Potentielle Teilexpansionen	ε $< P_0 \rightarrow P_1 P_2 P_3 >$ ε

Teilexpansion	$P_0 \rightarrow P_1 P_2 P_3$
Vorgänger	P_0
Regel Transformator Substitutionsliste Verwendete Partikel	$\exists$-Regel — $< [a/y],\; [b/y],\; [c_{neu}/y] >$ $< P_0[2] >$
Nachfolger korrespondiert abhängig	P_1 $(P_0[i], P_1[i]),\; 1 \leq i \leq 2$ $(P_0[2], P_1[3])$
Nachfolger korrespondiert abhängig	P_2 $(P_0[i], P_2[i]),\; 1 \leq i \leq 2$ $(P_0[2], P_2[3])$
Nachfolger korrespondiert abhängig	P_3 $(P_0[i], P_3[i]),\; 1 \leq i \leq 2$ $(P_0[2], P_3[3])$

Beispiel: SLD-Bäume

- *Partikel*:
  Partikel sind atomare Anfragen.
- *Partikelliste*:
  Reihenfolge der Partikel ist irrelevant. Liste repräsentiert eine Konjunktion.
- *Initiale Partikelliste*:
  Besteht aus der vom Benutzer bestimmten Anfrage.
- *Eigenschaft*:
  Eine nicht-leere, nicht-expandierbare Partikelliste hat die Eigenschaft „gescheitert". Eine leere Liste hat die Eigenschaft „erfolgreich".
- *Transformatoren*:
  Programmklauseln.
- *Regeln*:
  SLD-Resolution.
- *Substitutionen*:
  Es gibt nur die Substitutionen, die eine Teilexpansion betreffen. Jede Substitutionsliste ist einelementig.
- *Expansion*:
  Eine Teilexpansion beschreibt die Ersetzung eines Partikels π durch $n \in \mathbb{N}$ Partikel.

Übersicht über die betrachteten Verfahren

1. PUHR-Tableaux
2. Smullyan-Tableaux
3. EP-Tableaux
4. SLD-Bäumen
5. Proof-trees
6. Allgemeine Abhängigkeitsbäume
7. DDB trees in diesem Rahmen
8. Syntaxbäume

Grenzen des begrifflichen Rahmens