

Extraktion von Logikprogrammen aus Neuronalen Netzen

Jens Lehmann

Institut für Künstliche Intelligenz
Fakultät Informatik
Technische Universität Dresden

1. März 2005

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Einleitung

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

- ▶ versucht künstliche neuronale Netze und Logikprogrammierung zu vereinen
- ▶ künstliche neuronale Netze:
 - + parallel, flexibel, bekannte Lernalgorithmen
 - in neuronalen Netzen kodiertes Wissen nicht direkt lesbar
- ▶ Logikprogrammierung:
 - + können leicht verstanden werden, bekannte Beweisverfahren
 - nicht parallel, Lernproblem schwierig

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

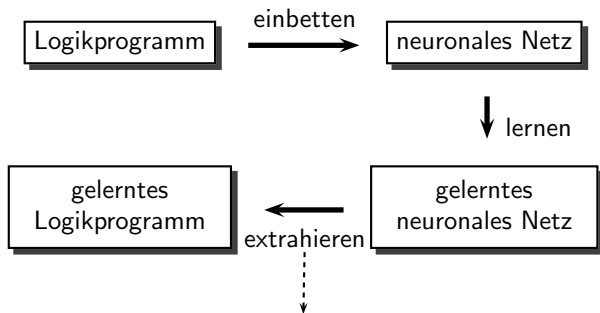
- ▶ Idee (u.a. Hölldobler, Kalinke, Störr 1999):
 - ▶ gegeben: Programm P
 - ▶ Darstellung des Konsequenzoperators T_P als neuronales Netz

Beispiel für Anwendung des T_P -Operators:

`add(X,0,X) .`

`add(X,s(Y),s(Z)) :- add(X,Y,Z) .`

$$T_P(\{add(0,0,0)\}) = \{add(s^n(0),0,s^n(0)) \mid n \geq 0\} \\ \cup \{add(0,s(0),s(0))\}$$



1. Extraktion mit ILP (induktive Logikprogrammierung)
2. exakte Programmextraktion mit speziell dafür entwickelten Algorithmen für Aussagenlogik

Extraktion mit ILP

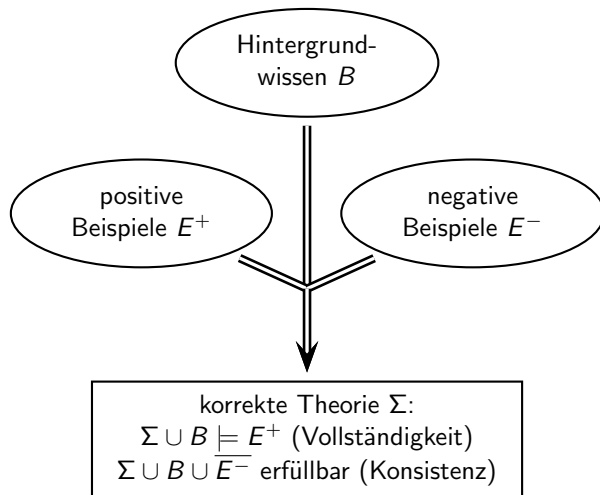


Abbildung: Aufgabe der ILP-Systeme: Erstellen einer Theorie

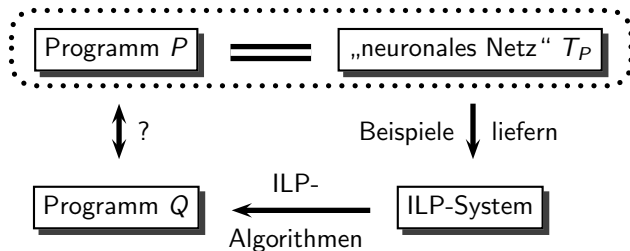


Abbildung: Setup für Tests der ILP-Systeme

Generierung von Beispielen:

der gegebene T_P -Operator sei monoton

- beginnend mit der leeren Interpretation iterative Anwendung des T_P -Operators
- man erhält eine Teilmenge des kleinsten Herbrandmodells, deren Elemente positive Beispiele sind

$$\emptyset \xrightarrow{T_P} I_1 \xrightarrow{T_P} I_2 \xrightarrow{T_P} I_3 \cdots$$

```
add(X,0,X).  
add(X,s(Y),s(Z)) :- add(X,Y,Z).
```

Generierte Beispiele:

```
add(0,0,0).  
add(s(0),0,s(0)).  
add(s(s(0)),0,s(s(0))).  
add(s(s(s(0))),0,s(s(s(0)))).  
  
add(0,s(0),s(0)).  
add(s(0),s(0),s(s(0))).  
add(s(s(0)),s(0),s(s(s(0)))).  
add(s(s(s(0))),s(0),s(s(s(s(0))))).  
  
add(0,s(s(0)),s(s(0))).  
add(s(0),s(s(0)),s(s(s(0)))).  
add(s(s(0)),s(s(0)),s(s(s(s(0))))).  
add(s(s(s(0))),s(s(0)),s(s(s(s(s(0)))))).
```

- ▶ ein ILP-System (CProgol) findet mit diesen Beispielen die korrekte Theorie

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Fazit zur Verwendung von ILP zur Extraktion

- ▶ negativ:
 - ▶ außer CProgol war keines der fünf getesteten ILP-Systeme für die Aufgabe geeignet
 - ▶ Systeme können meistens keine nicht-definiten Programme lernen
 - ▶ Korrektheit, d.h. Erhalt des T_P -Operators, spielt für ILP-Systeme keine Rolle
 - ▶ Paare $(I, T_P(I))$ liefern unendlich viele Beispiele, von denen man intelligent gute Beispiele wählen muss
- ▶ positiv:
 - ▶ ILP konnte für Aufgabenstellung angewandt werden
 - ▶ mit CProgol konnte erfolgreich eine Extraktionsaufgabe gelöst werden

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Exakte Programmextraktion

- ▶ Ziele:
 - ▶ Extraktion, bei der Konsequenzoperator des extrahierten Programmes genau dem Konsequenzoperator entspricht, den das neuronale Netz repräsentiert
 - ▶ Schwerpunkt: extrahierte Programme sollen möglichst klein sein (im Idealfall minimal)
 - ▶ Größe = Anzahl der Literale eines Programmes
- ▶ entwickelte Algorithmen:
 - ▶ Extraktion definiter Programme
 - ▶ Extraktion normaler Programme
 - ▶ α -, β -, γ -Reduktion
 - ▶ Greedy-Algorithmus
 - ▶ Intelligente Programmsuche

Es seien C_1 und C_2 beliebige Klauseln mit:

- ▶ $C_1: h \leftarrow p_1, \dots, p_a, \neg q_1, \dots, \neg q_b$
- ▶ $C_2: h \leftarrow r_1, \dots, r_c, \neg s_1, \dots, \neg s_d$
- ▶ $\{p_1, \dots, p_a\} \subseteq \{r_1, \dots, r_c\}$
- ▶ $\{q_1, \dots, q_b\} \subseteq \{s_1, \dots, s_d\}$

Dann *subsummiert* die Klausel C_1 die Klausel C_2 .

Konstruktion eines korrekten Programms Q :

- ▶ alle Interpretationen werden nach Anzahl der Elemente geordnet durchlaufen
- ▶ für jede solche Interpretation I mit $I = \{p_1, \dots, p_n\}$ und jedes Element $q \in T_P(I)$ wird $q \leftarrow p_1, \dots, p_n$ zu Q hinzugefügt, falls diese Klausel nicht bereits von einer Klausel in Q subsummiert wird.

[Einleitung](#)[Extraktion mit ILP](#)[Überblick](#)[Aufgabenanalyse](#)[Beispiel: add](#)[Fazit](#)[Exakte Programmextraktion](#)[Übersicht](#)[Definite Programme](#) [\$\alpha\$ -Reduktion](#)[Greedy-Algorithmus](#)[Fazit](#)

Extraktion definiter Programme

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Extraktion definiter Programme

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

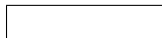
I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

$r \leftarrow p$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$



I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$



I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$r \leftarrow p, q$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$r \leftarrow p, r$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$



I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

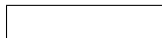
Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$



I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$

$p \leftarrow p, q, r$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$

$q \leftarrow p, q, r$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

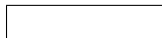
$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$

$q \leftarrow p, q, r$



I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm:

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$

$q \leftarrow p, q, r$

$r \leftarrow p, q, r$

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{r\}$
$\{q\}$	\emptyset
$\{r\}$	\emptyset
$\{p, q\}$	$\{p, r\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r\}$

Programm (vollständig):

$r \leftarrow p$

$p \leftarrow p, q$

$p \leftarrow q, r$

$r \leftarrow q, r$

$q \leftarrow p, q, r$

Konstruktion eines korrekten Programms:

- ▶ für jede Interpretation I mit $I = \{r_1, \dots, r_a\}$ und $B_P \setminus I = \{s_1, \dots, s_b\}$ und jedes Element $p \in T_P(I)$ wird eine Klausel $p \leftarrow r_1, \dots, r_a, \neg s_b, \dots, \neg s_m$ generiert (top-down Ansatz)
- ▶ für jedes Mapping von (endlichen) Interpretationen auf (endliche) Interpretationen lässt sich Programm konstruieren, dessen Konsequenzoperator dieses Mapping ist

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

 α -Reduktion

Greedy-Algorithmus

Fazit

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{p, r\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{p\}$
$\{p, r\}$	$\{r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	\emptyset

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

 α -Reduktion

Greedy-Algorithmus

Fazit

I	$T_P(I)$
\emptyset	\emptyset
$\{p\}$	$\{p, r\}$ $\rightarrow p \leftarrow p, \neg q, \neg r$
$\{q\}$	\emptyset $\rightarrow r \leftarrow p, \neg q, \neg r$
$\{r\}$	$\{r\}$ $\rightarrow r \leftarrow \neg p, \neg q, r$
$\{p, q\}$	$\{p\}$ $\rightarrow p \leftarrow p, q, \neg r$
$\{p, r\}$	$\{r\}$ $\rightarrow r \leftarrow p, \neg q, r$
$\{q, r\}$	$\{p\}$ $\rightarrow p \leftarrow \neg p, q, r$
$\{p, q, r\}$	\emptyset

α -Reduktion:

- ▶ Angenommen es existieren Klauseln C_1 und C_2 , so dass C_2 von C_1 subsummiert wird. Dann entferne C_2 .
- ▶ Angenommen es existieren Klauseln C_1 und C_2 mit:
 - ▶ $C_1 : p \leftarrow q, r_1, \dots, r_a, \neg s_1, \dots, \neg s_b$
 - ▶ $C_2 : p \leftarrow \neg q, t_1, \dots, t_c, \neg u_1, \dots, \neg u_d$
 - ▶ $\{r_1, \dots, r_a\} \subseteq \{t_1, \dots, t_c\}$
 - ▶ $\{s_1, \dots, s_b\} \subseteq \{u_1, \dots, u_d\}$

Dann entferne $\neg q$ aus C_2 .

- ▶ analog auch für den Fall, dass q und $\neg q$ in den Klauseln vertauscht sind, wobei dann q entfernt wird

1. Reduktionsschritt:

$$p \leftarrow p, \neg q, \neg r$$

$$p \leftarrow p, q, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q, \neg r$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

24 Literale

$$p \leftarrow p, \neg r$$

$$p \leftarrow p, q, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q, \neg r$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

23 Literale

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

2. Reduktionsschritt:

$$p \leftarrow p, \neg r$$

$$p \leftarrow p, q, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q, \neg r$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

23 Literale

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q, \neg r$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

19 Literale

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

3. Reduktionsschritt:

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q, \neg r$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

19 Literale

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

18 Literale

4. Reduktionsschritt:

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q$$

$$r \leftarrow \neg p, \neg q, r$$

$$r \leftarrow p, \neg q, r$$

18 Literale

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q$$

$$r \leftarrow \neg p, \neg q, r$$

14 Literale

5. Reduktionsschritt:

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q$$

$$r \leftarrow \neg p, \neg q, r$$

14 Literale

$$p \leftarrow p, \neg r$$

$$p \leftarrow \neg p, q, r$$

$$r \leftarrow p, \neg q$$

$$r \leftarrow \neg q, r$$

13 Literale

Funktionsweise (grob):

- ▶ Unterteilung in Unterprogramme für jedes Prädikat
- ▶ Berechnung von erlaubten Klauselkörpern für jedes Prädikat
- ▶ schrittweise Konstruktion eines Unterprogramms mittels der score-Heuristik

→ diese Schritte werden informell anhand eines Beispiels erklärt

Erlaubte Klauselkörper für p

I	$T_P(I)$
\emptyset	$\{p\}$
$\{p\}$	$\{p\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{r\}$
$\{p, r\}$	$\{p, r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	$\{p, r\}$

Kl.körper	Erlaubt für p ?
-----------	-------------------

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$
\emptyset	$\{p\}$
$\{p\}$	$\{p\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{r\}$
$\{p, r\}$	$\{p, r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	$\{p, r\}$

Kl.körper	Erlaubt für p ?
\square	$p \notin T_P(\{q\})$

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$
\emptyset	$\{p\}$
$\{p\}$	$\{p\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{r\}$
$\{p, r\}$	$\{p, r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	$\{p, r\}$

Kl.körper	Erlaubt für p ?
\square	$p \notin T_P(\{q\})$
p	$p \notin T_P(\{p, q\})$

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$
\emptyset	$\{p\}$
$\{p\}$	$\{p\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{r\}$
$\{p, r\}$	$\{p, r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	$\{p, r\}$

Kl.körper	Erlaubt für p ?
\square	$p \notin T_P(\{q\})$
p	$p \notin T_P(\{p, q\})$
q	$p \notin T_P(\{q\})$

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$	Kl.körper	Erlaubt für p ?
\emptyset	$\{p\}$	\square	$p \notin T_P(\{q\})$
$\{p\}$	$\{p\}$	p	$p \notin T_P(\{p, q\})$
$\{q\}$	\emptyset	q	$p \notin T_P(\{q\})$
$\{r\}$	$\{r\}$	\vdots	\vdots
$\{p, q\}$	$\{r\}$	\vdots	\vdots
$\{p, r\}$	$\{p, r\}$	p, r	OK.
$\{q, r\}$	$\{p\}$		
$\{p, q, r\}$	$\{p, r\}$		

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$	Kl.körper	Erlaubt für p ?
\emptyset	$\{p\}$	\square	$p \notin T_P(\{q\})$
$\{p\}$	$\{p\}$	p	$p \notin T_P(\{p, q\})$
$\{q\}$	\emptyset	q	$p \notin T_P(\{q\})$
$\{r\}$	$\{r\}$	\vdots	\vdots
$\{p, q\}$	$\{r\}$	\vdots	\vdots
$\{p, r\}$	$\{p, r\}$	p, r	OK.
$\{q, r\}$	$\{p\}$	q, r	OK.
$\{p, q, r\}$	$\{p, r\}$	$p, \neg q$	OK.
		$\neg q, \neg r$	OK.

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$	Kl.körper	Erlaubt für p ?
\emptyset	$\{p\}$	\square	$p \notin T_P(\{q\})$
$\{p\}$	$\{p\}$	p	$p \notin T_P(\{p, q\})$
$\{q\}$	\emptyset	q	$p \notin T_P(\{q\})$
$\{r\}$	$\{r\}$	\vdots	\vdots
$\{p, q\}$	$\{r\}$	\vdots	\vdots
$\{p, r\}$	$\{p, r\}$	p, r	OK.
$\{q, r\}$	$\{p\}$	q, r	OK.
$\{p, q, r\}$	$\{p, r\}$	$p, \neg q$	OK.
		$\neg q, \neg r$	OK.
		\vdots	\vdots
		p, q, r	p, r ist kleiner.

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

I	$T_P(I)$	Kl.körper	Erlaubt für p ?
\emptyset	$\{p\}$	\square	$p \notin T_P(\{q\})$
$\{p\}$	$\{p\}$	p	$p \notin T_P(\{p, q\})$
$\{q\}$	\emptyset	q	$p \notin T_P(\{q\})$
$\{r\}$	$\{r\}$	\vdots	\vdots
$\{p, q\}$	$\{r\}$	\vdots	\vdots
$\{p, r\}$	$\{p, r\}$	p, r	OK.
$\{q, r\}$	$\{p\}$	q, r	OK.
$\{p, q, r\}$	$\{p, r\}$	$p, \neg q$	OK.
		$\neg q, \neg r$	OK.
		\vdots	\vdots
		p, q, r	p, r ist kleiner.
		\vdots	\vdots
		$\neg p, \neg q, \neg r$	$\neg q, \neg r$ ist kleiner.

$$S_p = \{p, r; \quad q, r; \quad p, \neg q; \quad \neg q, \neg r\}$$

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Erlaubte Klauselkörper für p

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

I	$T_P(I)$	Interpretation	Schritt
\emptyset	$\{p\}$	\emptyset	
$\{p\}$	$\{p\}$	$\{p\}$	
$\{q\}$	\emptyset	$\{p, r\}$	
$\{r\}$	$\{r\}$	$\{q, r\}$	
$\{p, q\}$	$\{r\}$	$\{p, q, r\}$	
$\{p, r\}$	$\{p, r\}$		
$\{q, r\}$	$\{p\}$		
$\{p, q, r\}$	$\{p, r\}$		

Berechnung des Unterprogramms Q_p

Kl.körper	Interpretation	Schritt
$\neg q, \neg r$	\emptyset	
$p, \neg q$	$\{p\}$	
p, r	$\{p, r\}$	
q, r	$\{q, r\}$	
	$\{p, q, r\}$	

Berechnung des Unterprogramms Q_p

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Kl.körper	I	Interpretation	Schritt
$\neg q, \neg r$	2	\emptyset	
$p, \neg q$	2	$\{p\}$	
p, r	2	$\{p, r\}$	
q, r	2	$\{q, r\}$	
		$\{p, q, r\}$	

Berechnung des Unterprogramms Q_p

Kl.körper	I	Interpretation	Schritt
$\neg q, \neg r$	2	\emptyset	1
$p, \neg q$	2	$\{p\}$	1
p, r	2	$\{p, r\}$	
q, r	2	$\{q, r\}$	
		$\{p, q, r\}$	

Unterprogramm Q_p :

$$p \leftarrow \neg q, \neg r$$

Berechnung des Unterprogramms Q_p

Kl.körper	I	II	Interpretation	Schritt
$\neg q, \neg r$	2		\emptyset	I
$p, \neg q$	2	1	$\{p\}$	I
p, r	2	2	$\{p, r\}$	
q, r	2	2	$\{q, r\}$	
			$\{p, q, r\}$	

Unterprogramm Q_p :

$$p \leftarrow \neg q, \neg r$$

Berechnung des Unterprogramms Q_p

Kl.körper	I	II	Interpretation	Schritt
$\neg q, \neg r$	2		\emptyset	I
$p, \neg q$	2	1	$\{p\}$	I
p, r	2	2	$\{p, r\}$	II
q, r	2	2	$\{q, r\}$	
			$\{p, q, r\}$	II

Unterprogramm Q_p :

$$p \leftarrow \neg q, \neg r$$

$$p \leftarrow p, r$$

Berechnung des Unterprogramms Q_p

Kl.körper	I	II	III	Interpretation	Schritt
$\neg q, \neg r$	2			\emptyset	I
$p, \neg q$	2	1	0	$\{p\}$	I
p, r	2	2		$\{p, r\}$	II
q, r	2	2	1	$\{q, r\}$	
				$\{p, q, r\}$	II

Unterprogramm Q_p :

$$p \leftarrow \neg q, \neg r$$

$$p \leftarrow p, r$$

Berechnung des Unterprogramms Q_p

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Kl.körper	I	II	III	Interpretation	Schritt
$\neg q, \neg r$	2			\emptyset	I
$p, \neg q$	2	1	0	$\{p\}$	I
p, r	2	2		$\{p, r\}$	II
q, r	2	2	1	$\{q, r\}$	III
				$\{p, q, r\}$	II

Unterprogramm Q_p :

$$p \leftarrow \neg q, \neg r$$

$$p \leftarrow p, r$$

$$p \leftarrow q, r$$

Berechnung des Unterprogramms Q_r

I	$T_P(I)$
\emptyset	$\{p\}$
$\{p\}$	$\{p\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{r\}$
$\{p, r\}$	$\{p, r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	$\{p, r\}$

erlaubte Klauselkörper:

$$S_r = \{r, \neg q; \quad p, q; \quad p, r\}$$

Unterprogramm Q_r :

$$r \leftarrow r, \neg q$$

$$r \leftarrow p, q$$

I	$T_P(I)$
\emptyset	$\{p\}$
$\{p\}$	$\{p\}$
$\{q\}$	\emptyset
$\{r\}$	$\{r\}$
$\{p, q\}$	$\{r\}$
$\{p, r\}$	$\{p, r\}$
$\{q, r\}$	$\{p\}$
$\{p, q, r\}$	$\{p, r\}$

extrahiertes Programm:

$$p \leftarrow \neg q, \neg r$$

$$p \leftarrow p, r$$

$$p \leftarrow q, r$$

$$r \leftarrow r, \neg q$$

$$r \leftarrow p, q$$

- ▶ Extraktion definiter Programme:
 - ▶ korrekter Algorithmus, der minimales Programm liefert
 - ▶ weiteres Resultat: es gibt zu einem monotonen Operator genau ein minimales definites Programm
- ▶ Extraktion normaler Programme:
 - ▶ für jedes Mapping $M : 2^{B_P} \rightarrow 2^{B_P}$ existiert Programm P , so dass $M = T_P$ (B_P endlich)
 - ▶ α -, β -, γ -Reduktion: korrekt, top-down, Resultat nicht minimal
 - ▶ Greedy-Algorithmus: korrekt, bottom-up, Resultat nicht minimal
 - ▶ Programmsuche: korrekt, Resultat minimal
 - ▶ weiteres Resultat: es kann zu einem Operator mehrere verschiedene minimale normale Programme geben

Einleitung

Extraktion mit ILP

Überblick

Aufgabenanalyse

Beispiel: add

Fazit

Exakte Programmextraktion

Übersicht

Definite Programme

α -Reduktion

Greedy-Algorithmus

Fazit

Danke für Ihre Aufmerksamkeit