Übung zur Vorlesung

Automatisierte Logik und Programmierung

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Sommersemester 2004

Blatt 5 — Abgabetermin: —

Das Übungsblatt soll die Vorteile von Algorithmentheorien, speziell der Globalsuchtheorien für die Programmsynthese deutlich machen und einen Teil des theoretischen Fundaments aufdecken.

Aufgabe 5.1 (Globalsuch-Synthese)

Erzeugen Sie einen Globalsuch-Algorithmus für das n-Damen Problem.

`FUNCTION queens(n:ℤ):Set(Seq(ℤ)) WHERE n≥1 RETURNS {nq | perm(nq,{1..n}) ∧ safe(nq)}`

- Wählen Sie eine der vorgegebenen Globalsuch-Theorien.
- Spezialisieren sie diese auf das Problem.
- Wählen Sie einen Wohlfundiertheitsfilter und spezialisieren Sie diesen mit der eben festgelegten Substitution, so daß er notwendig ist.
- Verfeinern sie den Filter, wenn möglich.
- Erzeugen Sie den schematischen Algorithmus
- Vereinfachen Sie den Algorithmus (von Hand) nachträglich

Aufgabe 5.2 (Vordefinierte Globalsuchtheorien und Filter)

Zeigen Sie die Korrektheit der folgenden im Anhang definierten Einträge einer Wissensbank.

5.2–a `gs_sequences_over_finite_set`(α) ist eine Globalsuchtheorie[1]

5.2–b λ`S,V.|V|≤k` ist ein Wohlfundiertheitsfilter für `gs_sequences_over_finite_set`(α)

5.2–c λ`S,V.|V|≤k*|S|` ist ein Wohlfundiertheitsfilter für `gs_sequences_over_finite_set`(α)

5.2–d λ`S,V.nodups(V)` ist ein Wohlfundiertheitsfilter für `gs_sequences_over_finite_set`(α)

5.2–e `gs_subsets_of_a_finite_set`(α) ist eine wohlfundierte Globalsuchtheorie

Aufgabe 5.3 (Verfeinerung und Spezialisierung von Globalsuchtheorien)

Es sei *spec* eine Spezifikation, G eine Globalsuchtheorie, Φ ein Wohlfundiertheitsfilter für G und Ψ ein beliebiger Filter für G. Zeigen Sie

5.3–a G generalisiert *spec* mit $\theta \Rightarrow G_\theta$(spec) ist eine Globalsuchtheorie

5.3–b G generalisiert *spec* mit $\theta \Rightarrow \Phi_\theta$ ist Wohlfundiertheitsfilter für $G_\theta(spec)$

5.3–c Φ notwendig für G und Ψ notwendig für G
$\Rightarrow \Xi \equiv \lambda \mathtt{x,s}.\Phi(\mathtt{x,s}) \wedge \Psi(\mathtt{x,s})$ notwendiger Wohlfundiertheitsfilter für G

---

[1] In der Vorlesung hies diese Theorie kurz `gs_seq_set`(α)

Anhang 5.1: Globalsuchtheorien und Wohlfundiertheitsfilter

The following global search theories which we have adopted from the current KIDS system are sufficient to support the synthesis of many global search algorithms on sets, sequences, mappings, and integers.

1. `gs_sequences_over_finite_set(`α`)` enumerates all sequences over a given finite set `S`. A set of sequences is represented by their greatest common prefix `V`. Splitting is performed by appending an element from `S` onto the end of `V`. Extracting selects the greatest common prefix `V` itself.

$$\begin{array}{lll}
\mathtt{D} & \mapsto & \mathtt{Set(}\alpha\mathtt{)} \\
\mathtt{R} & \mapsto & \mathtt{Seq(}\alpha\mathtt{)} \\
\mathtt{I} & \mapsto & \lambda\mathtt{S.True} \\
\mathtt{O} & \mapsto & \lambda\mathtt{S,L.range(L)}\subseteq\mathtt{S} \\
\mathtt{S} & \mapsto & \mathtt{Seq(}\alpha\mathtt{)} \\
\mathtt{J} & \mapsto & \lambda\mathtt{S,V.range(V)}\subseteq\mathtt{S} \\
\mathtt{s}_0 & \mapsto & \lambda\mathtt{S.[]} \\
\mathtt{sat} & \mapsto & \lambda\mathtt{L,V.V}\sqsubseteq\mathtt{L} \\
\mathtt{split} & \mapsto & \lambda\mathtt{S,V.}\{\mathtt{V}\cdot\mathtt{i}\,|\,\mathtt{i}\in\mathtt{S}\} \\
\mathtt{ext} & \mapsto & \lambda\mathtt{V.}\{\mathtt{V}\}
\end{array}$$

2. `gs_subsets_of_a_finite_set(`α`)` enumerates subsets of a given finite set `S`. A set of subsets is is represented by their greatest common subset `U` and a variable `V` holds the set of uncommitted elements so far. Splitting involves selecting an arbitrary element of `V` and alternatively adding or not adding it to `U` (yielding two new space descriptors). Extracting selects `U` if `V` does not hold any uncommitted elements.

$$\begin{array}{lll}
\mathtt{D} & \mapsto & \mathtt{Set(}\alpha\mathtt{)} \\
\mathtt{R} & \mapsto & \mathtt{Set(}\alpha\mathtt{)} \\
\mathtt{I} & \mapsto & \lambda\mathtt{S.True} \\
\mathtt{O} & \mapsto & \lambda\mathtt{S,T.T}\subseteq\mathtt{S} \\
\mathtt{S} & \mapsto & \mathtt{Set(}\alpha\mathtt{)}^2 \\
\mathtt{J} & \mapsto & \lambda\mathtt{S,U,V.empty?(U}\cap\alpha\mathtt{V)} \ \wedge\ \mathtt{U}\cup\mathtt{V}\subseteq\mathtt{S} \\
\mathtt{s}_0 & \mapsto & \lambda\mathtt{S.}\langle\emptyset\mathtt{,S}\rangle \\
\mathtt{sat} & \mapsto & \lambda\mathtt{T,U,V.U}\subseteq\mathtt{T} \ \wedge\ \mathtt{T}\subseteq\mathtt{U}\cup\mathtt{V} \\
\mathtt{split} & \mapsto & \lambda\mathtt{S,U,V.}\{\langle\mathtt{U+a,\ V-a}\rangle\,|\,\mathtt{a}\in\mathtt{V}\} \ \cup\ \{\langle\mathtt{U,V-a}\rangle\,|\,\mathtt{a}\in\mathtt{V}\} \\
\mathtt{ext} & \mapsto & \lambda\mathtt{U,V.if\ empty?(V)\ then}\ \{\mathtt{U}\}\ \mathtt{else}\ \emptyset
\end{array}$$

This theory has the property that subsets are extracted only from the leaves. The depth of the search tree is exactly `|S|` and its size is $2^{|S|+1} - 1$.

3. `gs_finite_mappings(`α`,`β`)` is enumerates all mappings from a finite set `S` to a finite set `S'`. The space structure incrementally builds up a partial map `M` whose domain is a subset of `S` and holds uncommitted elements of `S` in a set `T`. Splitting thus extends `M` by a new pair and reduces `T` accordingly. Extracting selects `M` if all elements are committed.

D	$\mapsto$	Set(α)$\times$Set(β)
R	$\mapsto$	Map(α,β)
I	$\mapsto$	λS,S'.True
O	$\mapsto$	λS,S',N.domain(N)=S $\wedge$ range(N)$\subseteq$S'
S	$\mapsto$	Set(α) $\times$ Map(α,β)
J	$\mapsto$	λS,S',T,M.domain(M)$\subseteq$S $\wedge$ range$_\alpha$(M)$\subseteq$S' $\wedge$ T=S$\setminus$domain(M)
s_0	$\mapsto$	λS,S'.$\langle$S,$\{\|\ \|\}\rangle$
sat	$\mapsto$	λN,T,M.M$\sqsubseteq$N
split	$\mapsto$	λS,S',T,M.$\bigcup\{\ \{\langle$T-a, extend(M,a,b)$\rangle\|$a$\in$T$\}$ | b$\in$S'$\}$
ext	$\mapsto$	λT,M.if empty?(T) then $\{$M$\}$ else $\emptyset$

4. `gs_binary_split_of_integer_subrange` encodes the binary split paradigm. It enumerates all elements of an interval $\{$`m..n`$\}$ of integers. Space descriptors correspond to subintervals and and are split roughly in half. Extraction selects the only element if an interval has shrunk to a singleton set.

D	$\mapsto$	$\mathbb{Z}^2$
R	$\mapsto$	$\mathbb{Z}$
I	$\mapsto$	λm,n. m$\leq$n
O	$\mapsto$	λm,n,k. k$\in\{$m..n$\}$
S	$\mapsto$	$\mathbb{Z}^2$
J	$\mapsto$	λm,n,i,j. $\{$i..j$\}\subseteq\{$m..n$\}$
s_0	$\mapsto$	λm,n. $\langle$m,n$\rangle$
sat	$\mapsto$	λk,i,j. k$\in\{$i..j$\}$
split	$\mapsto$	λm,n,i,j. if i<j then $\{\langle$i, (i+j)/2$\rangle$,$\langle$1+(i+j)/2, j$\rangle\}$ else $\emptyset$
ext	$\mapsto$	λi,j. if i=j then $\{$i$\}$ else $\emptyset$