

Automatisierte Logik und Programmierung

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Sommersemester 2014

Blatt 3 — Abgabetermin: 25.6.2014

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

Aufgabe 3.1 (Globalsuch-Synthese)

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

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

wobei

perm(L, S) ≡ nodups(L) ∧ range(L) = S
 safe(L) ≡ ∀i, j < |L|. i ≠ j ⇒ |L[i] - L[j]| ≠ |i - j|

- 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 3.2 (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

- 3.2–a *G* generalisiert *spec* mit $\vartheta \Rightarrow G_{\vartheta}(spec)$ ist eine Globalsuchtheorie
 3.2–b *G* generalisiert *spec* mit $\vartheta \Rightarrow \Phi_{\vartheta}$ ist Wohlfundiertheitsfilter für $G_{\vartheta}(spec)$
 3.2–c Φ notwendiger Wohlfundiertheitsfilter für *G* und Ψ notwendig für *G*
 $\Rightarrow \Xi \equiv \lambda x, s. \Phi(x, s) \wedge \Psi(x, s)$ notwendiger Wohlfundiertheitsfilter für *G*

Aufgabe 3.3 (Vordefinierte Globalsuchtheorien und Filter)

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

- 3.3–a `gs_sequences_over_finite_set` (α) ist eine Globalsuchtheorie¹
 3.3–b $\lambda S, V. |V| \leq k$ ist ein Wohlfundiertheitsfilter für `gs_sequences_over_finite_set` (α)
 3.3–c $\lambda S, V. |V| \leq k * |S|$ ist ein Wohlfundiertheitsfilter für `gs_sequences_over_finite_set` (α)
 3.3–d $\lambda S, V. \text{nodups}(V)$ ist ein Wohlfundiertheitsfilter für `gs_sequences_over_finite_set` (α)
 3.3–e `gs_subsets_of_a_finite_set` (α) ist eine wohlfundierte Globalsuchtheorie

¹In der Vorlesung wurde diese Theorie kurz `gs_seq_set` (α) genannt.

Anhang 3.1: Globalsuchtheorien und Wohlfundiertheitsfilter

The following global search theories which we have adopted from the 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.

```

D   ⇨      Set ( $\alpha$ )
R   ⇨      Seq ( $\alpha$ )
I   ⇨       $\lambda S. \text{True}$ 
O   ⇨       $\lambda S, L. \text{range}(L) \subseteq S$ 
S   ⇨      Seq ( $\alpha$ )
J   ⇨       $\lambda S, V. \text{range}(V) \subseteq S$ 
 $s_0$  ⇨       $\lambda S. []$ 
sat  ⇨       $\lambda L, V. V \subseteq L$ 
split⇨       $\lambda S, V. \{V \cdot i \mid i \in S\}$ 
ext  ⇨       $\lambda V. \{V\}$ 

```

2. `gs_subsets_of_a_finite_set` (α) enumerates subsets of a given finite set S . A set of subsets 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.

```

D   ⇨      Set ( $\alpha$ )
R   ⇨      Set ( $\alpha$ )
I   ⇨       $\lambda S. \text{True}$ 
O   ⇨       $\lambda S, T. T \subseteq S$ 
S   ⇨      Set ( $\alpha$ )2
J   ⇨       $\lambda S, U, V. \text{empty?}(U \cap \alpha V) \wedge U \cup V \subseteq S$ 
 $s_0$  ⇨       $\lambda S. \langle \emptyset, S \rangle$ 
sat  ⇨       $\lambda T, U, V. U \subseteq T \wedge T \subseteq U \cup V$ 
split⇨       $\lambda S, U, V. \{(U+a, V-a) \mid a \in V\} \cup \{(U, V-a) \mid a \in V\}$ 
ext  ⇨       $\lambda U, V. \text{if empty?}(V) \text{ then } \{U\} \text{ else } \emptyset$ 

```

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) 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   ⇨      Set (α) × Set (β)
R   ⇨      Map (α, β)
I   ⇨      λS, S'. True
O   ⇨      λS, S', N. domain(N) = S ∧ range(N) ⊆ S'
S   ⇨      Set (α) × Map (α, β)
J   ⇨      λS, S', T, M. domain(M) ⊆ S ∧ rangeα(M) ⊆ S' ∧ T = S \ domain(M)
s0 ⇨      λS, S'. ⟨S, { | }⟩
sat  ⇨      λN, T, M. M ⊆ N
split ⇨    λS, S', T, M. ⋃ { {⟨T-a, extend(M, a, b)⟩ | a ∈ T} | b ∈ S' }
ext  ⇨      λT, M. if empty?(T) then {M} else ∅

```

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 are split roughly in half. Extraction selects the only element if an interval has shrunk to a singleton set.

```

D   ⇨      ℤ2
R   ⇨      ℤ
I   ⇨      λm, n. m ≤ n
O   ⇨      λm, n, k. k ∈ {m..n}
S   ⇨      ℤ2
J   ⇨      λm, n, i, j. {i..j} ⊆ {m..n}
s0 ⇨      λm, n. ⟨m, n⟩
sat  ⇨      λk, i, j. k ∈ {i..j}
split ⇨    λm, n, i, j. if i < j then {⟨i, (i+j)/2⟩, ⟨1+(i+j)/2, j⟩} else ∅
ext  ⇨      λi, j. if i = j then {i} else ∅

```

Lösung 3.1

Die Lösung besitzt eine erstaunliche Verwandtschaft zu der des Costas-Arrays Problems, obwohl die Aufgabenstellungen sehr verschieden sind.

FUNCTION queen ($n:\mathbb{Z}$) WHERE $n \geq 1$ RETURNS $\{nq:\text{Seq}(\mathbb{Z}) \mid \text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq)\}$

1. Wähle die Globalsuchtheorie $G = \text{gs_sequences_over_finite_set}(\mathbb{Z})$, da diese – wie das Problem – Folgen über ganzen Zahlen aufzählt.
2. Zu zeigen ist ‘ G generalisiert (D, R, I, O) ’
 - (a) Ausgabebereich R stimmt mit $R_G = \text{Seq}(\mathbb{Z})$ überein
 - (b) Eingabebereiche $D = \mathbb{Z}$ und $D_G = \text{Set}(\mathbb{Z})$ sind anzupassen
 - (c) Keine Eingabebedingung zu prüfen, da $I_G(S) = \text{true}$.
 - (d) Zu zeigen bleibt also

$$\forall n:\mathbb{Z}. n \geq 1 \Rightarrow \exists S:\text{Set}(\mathbb{Z}).$$

$$\forall nq:\text{Seq}(\mathbb{Z}). \text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \Rightarrow \text{range}(nq) \subseteq S$$

– Suche Folgerungen von $\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq)$, in denen $\text{range}(nq)$ vorkommt

– Auffalten von perm liefert: $\text{perm}(nq, \{1..n\}) \Rightarrow \text{range}(nq) \subseteq \{1..n\}$

– Wähle $S := \{1..n\}$

Der Beweis liefert $\vartheta(n) := \{1..n\}$. Die mit ϑ und der Ausgangsspezifikation modifizierte Globalsuchtheorie G_ϑ hat folgende Komponenten

D	\mapsto	Set(\mathbb{Z})
R	\mapsto	Seq(\mathbb{Z})
I	\mapsto	$\lambda n. n \geq 1$
O	\mapsto	$\lambda n, nq. \text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq)$
S	\mapsto	Seq(\mathbb{Z})
J	\mapsto	$\lambda n, V. \text{range}(V) \subseteq \{1..n\}$
s_0	\mapsto	$\lambda n. []$
sat	\mapsto	$\lambda nq, V. V \sqsubseteq nq$
split	\mapsto	$\lambda n, V. \{V \cdot i \mid i \in \{1..n\}\}$
ext	\mapsto	$\lambda V. \{V\}$

3. Wähle WF-Filter Φ für G so, daß ‘ Φ_ϑ notwendig für G_ϑ ’ beweisbar. Wir testen die bekannten Filter für G und prüfen $\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge V \sqsubseteq nq \Rightarrow \Phi(\{1..n\}, V)$
 - (a) $\Phi_1(S, V) = |V| \leq k$ für ein festes k :

$$\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge V \sqsubseteq nq \Rightarrow |V| \leq k$$
 Dieser Beweis ist nicht so einfach möglich.

(b) $\Phi_2(S, V) = |V| \leq k * |S|$ für ein festes k :

$$\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge \forall \sqsubseteq nq \Rightarrow |V| \leq k * |\{1..n\}|$$

Reduktion mit Lemma B.1.16.5 liefert als Bedingung

$$\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge \forall \sqsubseteq nq \Rightarrow |V| \leq k * n$$

Auffalten von perm liefert:

$$\text{perm}(nq, \{1..n\}) \Rightarrow \text{range}(nq) \subseteq \{1..n\} \wedge \text{nodups}(nq)$$

Mit Lemma B.2.22.4, B.2.24.3, B.2.24.5 und B.1.13.12 folgt

$$\text{range}(nq) \subseteq \{1..n\} \wedge \text{nodups}(nq) \wedge \forall \sqsubseteq nq \Rightarrow \text{range}(V) \subseteq \{1..n\} \wedge |V| = |\text{range}(V)|$$

Mit Lemma B.1.16.4/5 folgt tatsächlich

$$\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge \forall \sqsubseteq nq \Rightarrow |V| \leq n$$

Damit ist die Behauptung beweisen: $\Phi_{2,\vartheta}$ ist notwendig für G_{ϑ} . Der Beweis ist automatisch allerdings recht schwer zu finden.

(c) $\Phi_3(S, V) = \text{nodups}(V) : \text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge \forall \sqsubseteq nq \Rightarrow \text{nodups}(V)$

Auffalten von perm liefert:

$$\text{perm}(nq, \{1..n\}) \Rightarrow \text{nodups}(nq)$$

Mit Lemma B.2.24.3 folgt

$$\text{nodups}(nq) \wedge \forall \sqsubseteq nq \Rightarrow \text{nodups}(V)$$

Damit ist die Behauptung beweisen: $\Phi_{3,\vartheta}$ ist notwendig für G_{ϑ} .

Wir wählen also Φ_3 und verfeinern diesen Filter zu $\Phi_{3,\vartheta}(n, V) = \text{nodups}(V)$

4. Verfeinerung des Filters durch weitere Folgerungen aus der Ausgabebedingung.

$$\text{perm}(nq, \{1..n\}) \wedge \text{safe}(nq) \wedge \forall \sqsubseteq nq \Rightarrow \Psi(n, V)$$

Die Bedingung $\text{perm}(nq, \{1..n\})$ ist mit der Generalisierungseigenschaft und dem Filter bereits ausgenutzt worden und braucht nicht mehr betrachtet zu werden. Wir ziehen daher Schlußfolgerungen aus $\text{safe}(nq) \wedge \forall \sqsubseteq nq$, falten die Definitionen auf und erhalten.

$$\forall \sqsubseteq nq \wedge \forall i < j < |nq|. |nq[i] - nq[j]| \neq j - i$$

Das Lemma über begrenzte All-Quantoren und Präfix ($\hat{=}$ Lemma B.1.11.4) führt zu

$$\forall \sqsubseteq nq \wedge \forall i < j < |V|. |nq[i] - nq[j]| \neq j - i$$

Das Lemma über Präfix und Elementselektion B.2.10.16 führt zu

$$\forall \sqsubseteq nq \wedge \forall i < j < |V|. |V[i] - V[j]| \neq j - i$$

Dies erlaubt ein Rückfalten der Definitionen und liefert $\Psi(n, V) \equiv \text{safe}(V)$

5. Die Instantiierung des Standard-Globalsuchalgorithmus liefert nun

```

FUNCTION queens(n:ℤ) WHERE n ≥ 1 RETURNS {nq:Seq(ℤ) | perm(nq, {1..n}) ∧ safe(nq)}
≡ if nodups([]) ∧ safe([]) then queensgs(n, []) else ∅
FUNCTION queensgs(n:ℤ, V:Seq(ℤ)) WHERE n ≥ 1 ∧ range(V) ⊆ {1..n} ∧ nodups(V) ∧ safe(V)
RETURNS {nq:Seq(ℤ) | perm(nq, {1..n}) ∧ ∀ ⊆ nq ∧ safe(nq)}
≡ {nq | nq ∈ {V} ∧ perm(nq, {1..n}) ∧ safe(nq)}
  ∪ ⋃ {queensgs(n, W) | W ∈ {V·i | i ∈ {1..n}} ∧ nodups(W) ∧ safe(W)}

```

In diesem Algorithmus sind noch einige Vereinfachungen möglich, die im nächsten Übungsblatt besprochen werden.

Lösung 3.2 Ziel dieser Aufgabe ist es, die Grundlagen der Globalsuchstrategie abzusichern.

Es sei $spec = (D, R, I, O)$ und $G = ((D', R', I', O'), S, J, s_0, sat, split, ext)$. Nach Voraussetzung erfüllen G und Φ die folgenden 5 Axiome.

1. $\forall x' : D' . I' (x') \Rightarrow J(x', s_0(x'))$
2. $\forall x' : D' . \forall s : S . I(x') \wedge J(x', s) \Rightarrow \forall t \in split(x', s) . J(x', t)$
3. $\forall x' : D' . \forall z : R' . I' (x') \wedge O(x', z) \Rightarrow sat(z, s_0(x'))$
4. $\forall x' : D' . \forall s : S . \forall z : R' . I' (x') \wedge J(x', s) \wedge O(x', z) \Rightarrow sat(z, s) \Leftrightarrow \exists k : \mathbb{N} . \exists t \in split^k(x', s) . z \in ext(t)$
5. $\forall x' : D' . \forall s : S . I' (x') \wedge J(x', s) \Rightarrow \exists k : \mathbb{N} . split_{\Phi}^k(x', s) = \emptyset$

3.2–a Es gelte “ G generalisiert $spec$ mit ϑ ”. Dann ist R eine Teilmenge von R' und es gilt

$$\forall x : D . I(x) \Rightarrow I'(\vartheta(x)) \wedge \forall z : R . O(x, z) \Rightarrow O'(\vartheta(x), z)$$

Per Definition ist $G_{\vartheta}(spec) =$

$$((D, R, I, O), S, \lambda x, s . J(\vartheta(x), s), \lambda x, s . s_0(\vartheta(x)), sat, \lambda x, s . split(\vartheta(x), s), ext)$$

Wir zeigen, daß $G_{\vartheta}(spec)$ die vier Axiome einer Globalsuchtheorie erfüllt.

1. $\forall x : D . I(x) \Rightarrow J(\vartheta(x), s_0(\vartheta(x)))$
Für $x \in D$ mit $I(x)$ gilt $I'(\vartheta(x))$ und nach Axiom 1 für G dann $J(\vartheta(x), s_0(\vartheta(x)))$
2. $\forall x : D . \forall s : S . I(x) \wedge J(\vartheta(x), s) \Rightarrow \forall t \in split(\vartheta(x), s) . J(\vartheta(x), t)$
Für $x \in D, s \in S$ mit $I(x)$ und $J(\vartheta(x), s)$ gilt $I'(\vartheta(x))$ und nach Axiom 2 für G dann $\forall t \in split(\vartheta(x), s) . J(\vartheta(x), t)$
3. $\forall x : D . \forall z : R . I(x) \wedge O(x, z) \Rightarrow sat(z, s_0(\vartheta(x)))$
Für $x \in D, z \in R$ mit $I(x)$ und $O(x, z)$ gilt $I'(\vartheta(x))$ und $O'(\vartheta(x), z)$ und nach Axiom 3 für G dann $sat(z, s_0(\vartheta(x)))$
4. $\forall x : D . \forall s : S . \forall z : R . I(x) \wedge J(\vartheta(x), s) \wedge O(x, z) \Rightarrow sat(z, s) \Leftrightarrow \exists k : \mathbb{N} . \exists t \in (\lambda x, s . split(\vartheta(x), s))^k(x, s) . z \in ext(t)$

Für $x \in D, s \in S, z \in R$ mit $I(x), J(\vartheta(x), s)$ und $O(x, z)$ gilt $I'(\vartheta(x))$ und $O'(\vartheta(x), z)$ und nach Axiom 4 für G dann

$$sat(z, s) \Leftrightarrow \exists k : \mathbb{N} . \exists t \in split^k(\vartheta(x), s) . z \in ext(t)$$

Durch Induktion kann man zeigen, daß für alle $k \in \mathbb{N}$ gilt:

$$(\lambda x, s . split(\vartheta(x), s))^k(x, s) = split^k(\vartheta(x), s)$$

Hieraus folgt dann die Behauptung.

3.2–b Es gelte “ G generalisiert $spec$ mit ϑ ” wie oben. Zu zeigen ist

$$\forall x : D . \forall s : S . I(x) \wedge J(\vartheta(x), s) \Rightarrow \exists k : \mathbb{N} . split_{\vartheta, \Phi}^k(x, s) = \emptyset$$

Für $x \in D, s \in S$ mit $I(x), J(\vartheta(x), s)$ gilt $I'(\vartheta(x))$ und nach Axiom 5 für G und Φ dann

$$\exists k : \mathbb{N} . split_{\Phi}^k(\vartheta(x), s) = \emptyset$$

Es ist $split_{\vartheta, \Phi}^k \equiv \lambda x, s . \{t \mid t \in split(\vartheta(x), s) \wedge \Phi(\vartheta(x), t)\} = \lambda x, s . split_{\Phi}(\vartheta(x))$

Durch Induktion kann man wie in Teil (a).4 zeigen, daß für alle $k \in \mathbb{N}$ gilt:

$$split_{\vartheta, \Phi}^k(x, s) = (\lambda x, s . split_{\Phi}(\vartheta(x), s))^k(x, s) = split_{\Phi}^k(\vartheta(x), s)$$

Hieraus folgt dann die Behauptung.

3.2–c Da Φ und Ψ notwendig für G sind, gilt

$$\forall x' : D. \forall s : S. \exists z' : R'. \text{sat}(z', s) \wedge O(x', z) \Rightarrow \Phi(x', s) \wedge \Psi(x', s)$$

Wegen $\Xi(x', s) = \Phi(x', s) \wedge \Psi(x', s)$ folgt hieraus, daß Ξ notwendig für G ist.

Da Φ ein Wohlfundiertheitsfilter für G ist und für alle $x' \in D'$, $s \in S$ gilt

$$\text{split}_{\Xi}(x', s) = \{t \mid t \in \text{split}(x', s) \wedge \Phi(x, t) \wedge \Psi(x, t)\} \subseteq \text{split}_{\Phi}(x', s)$$

folgt $\forall x' : D'. \forall s : S. I'(x') \wedge J(x', s) \Rightarrow \exists k : \mathbb{N}. \text{split}_{\Xi}^k(x', s) = \emptyset$.

Damit ist Ξ ein Wohlfundiertheitsfilter für G .

Lösung 3.3 Ziel dieser Aufgabe ist es, zu erkennen, wie die Korrektheit von Einträge einer Wissensbank sichergestellt werden kann.

Eine eine wohlfundierte Globalsuchtheorie muß die folgenden 5 Axiome erfüllen.

1. Initialdeskriptor ist sinnvoll
 - $\forall x : D. I(x) \Rightarrow J(x, s_0(x))$
2. Splitting sinnvoller Deskriptoren liefert sinnvolle Deskriptoren
 - $\forall x : D. \forall s : S. I(x) \wedge J(x, s) \Rightarrow \forall t \in \text{split}(x, s). J(x, t)$
3. Initialdeskriptor enthält alle Lösungen
 - $\forall x : D. \forall z : R. I(x) \wedge O(x, z) \Rightarrow \text{sat}(z, s_0(x))$
4. Alle Lösungen in endlich vielen Schritten extrahierbar
 - $\forall x : D. \forall s : S. \forall z : R. I(x) \wedge J(x, s) \wedge O(x, z) \Rightarrow \text{sat}(z, s) \Leftrightarrow \exists k : \mathbb{N}. \exists t \in \text{split}^k(x, s). z \in \text{ext}(t)$
5. Splitting nur endlich oft durchführbar (**Wohlfundiertheit**)
 - $\forall x : D. \forall s : S. I(x) \wedge J(x, s) \Rightarrow \exists k : \mathbb{N}. \text{split}^k(x, s) = \emptyset$

Die ersten 4 Axiome kennzeichnen eine Globalsuchtheorie. Ein Filter ist ein Wohlfundiertheitsfilter, wenn $\text{split}_{\Phi} \equiv \lambda x, s. \{t \mid t \in \text{split}(x, s) \wedge \Phi(x, t)\}$ das Axiom 5 erfüllt.

3.3–a Wir zeigen, daß $\text{gs_sequences_over_finite_set}(\alpha)$ ist eine Globalsuchtheorie ist, indem wir die ersten 4 Axiome überprüfen.

1. $\forall S : \text{Set}(\alpha). \text{true} \Rightarrow \text{range}([]) \subseteq S$
Dies folgt unmittelbar aus Lemma B.2.22.1 und B.1.13.1
2. $\forall S, V : \text{Set}(\alpha). \text{true} \wedge \text{range}(V) \subseteq S \Rightarrow \forall W \in \{V \cdot i \mid i \in S\}. W \subseteq S$
Umwandlung mit Lemma B.1.11.5 ergibt
... $\text{range}(V) \subseteq S \Rightarrow \forall i \in S. \text{range}(V \cdot i) \subseteq S$
Dies folgt mit Lemma B.2.22.5 und B.1.13.3
3. $\forall S, L : \text{Set}(\alpha). \text{true} \wedge \text{range}(L) \subseteq S \Rightarrow [] \in L$
Dies folgt mit Lemma B.2.10.1
4. $\forall S, V, L : \text{Set}(\alpha). \text{true} \wedge \text{range}(V) \subseteq S \wedge \text{range}(L) \subseteq S \Rightarrow \forall L \in L \Leftrightarrow \exists k : \mathbb{N}. \exists W \in (\lambda S, V. \{V \cdot i \mid i \in S\})^k(S, V). L \in \{W\}$

Durch Induktion kann man zeigen:

$$(\lambda S, V. \{V \cdot i \mid i \in S\})^k(S, V) = \{V \circ U \mid \text{range}(U) \subseteq S \wedge |U| = k\}$$

Damit bleibt zu zeigen

$$\dots \forall L \in L \Leftrightarrow \exists k : \mathbb{N}. \exists U : \text{Seq}(\alpha). \text{range}(U) \subseteq S \wedge |U| = k \wedge L = V \circ U$$

Dies folgt nun mit Lemma B.2.10.8.

3.3-b $\Phi_1 := \lambda S, V. |V| \leq k$: Es ist $\text{split}_\Phi(S, V) = \{V \cdot i \mid i \in S \wedge |V \cdot i| \leq k\}$

Zu zeigen ist $\forall S, V: \text{Set}(\alpha). \text{true} \wedge \text{range}(V) \subseteq S \Rightarrow \exists k: \mathbb{N}. \text{split}_\Phi^k(S, V) = \emptyset$

Durch Induktion zeige man:

$$\forall j: \mathbb{N}. \text{split}_\Phi^j(S, V) = \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge |V| \leq k - j\}$$

Wählt man nun $j := k - |V| + 1$ so ist

$$\text{split}_\Phi^j(S, V) = \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge |V| \leq |V| - 1\} = \emptyset$$

3.3-c $\Phi_2 := \lambda S, V. |V| \leq k * |S|$: Es ist $\text{split}_\Phi(S, V) = \{V \cdot i \mid i \in S \wedge |V \cdot i| \leq k * |S|\}$

Durch Induktion zeige man:

$$\forall j: \mathbb{N}. \text{split}_\Phi^j(S, V) = \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge |V| \leq k * |S| - j\}$$

Wählt man nun $j := k * |S| - |V| + 1$ so ist

$$\text{split}_\Phi^j(S, V) = \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge |V| \leq |V| - 1\} = \emptyset$$

3.3-d $\Phi_3 := \lambda S, V. \text{nodups}(V)$: Es ist $\text{split}_\Phi(S, V) = \{V \cdot i \mid i \in S \wedge \text{nodups}(V \cdot i)\}$

Durch Induktion zeige man:

$$\forall j: \mathbb{N}. \text{split}_\Phi^j(S, V) = \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge \text{nodups}(V \circ W)\}$$

Aus den Eigenschaften von nodups , \circ , range und \subseteq folgt (da $\text{range}(V) \cup \text{range}(W) \subseteq S$):

$$|V| + |W| = |V \circ W| = |\text{range}(V \circ W)| = |\text{range}(V) \cup \text{range}(W)| \leq |S|,$$

und damit

$$\forall j: \mathbb{N}. \text{split}_\Phi^j(S, V) = \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge \text{nodups}(V \circ W) \wedge |V| \leq |S| - j\}$$

Wählt man nun $j := |S| - |V| + 1$ so ist

$$\begin{aligned} \text{split}_\Phi^j(S, V) &= \{V \circ W \mid \text{range}(W) \subseteq S \wedge |W| = j \wedge \text{nodups}(V \circ W) \wedge |V| \leq |V| - 1\} \\ &= \emptyset \end{aligned}$$

3.3-e Den Beweis, daß $\text{gs_subsets_of_a_finite_set}(\alpha)$ eine wohlfundierte Globalsuchtheorie ist, wollen wir nur kurz skizzieren.

1. Zu zeigen ist $\text{empty?}(\emptyset \cap \alpha S) \wedge \emptyset \cup S \subseteq S$, was elementare Mengeneigenschaften sind.

2. $\text{empty?}(U \cap \alpha V) \wedge U \cup V \subseteq S$

$$\Rightarrow \forall a \in V. \text{empty?}(U + a \cap \alpha V - a) \wedge U + a \cup V - a \subseteq S \wedge \text{empty?}(U \cap \alpha V - a) \wedge U \cup V - a \subseteq S$$

Auch dies ist eine Standardeigenschaft von Vereinigung, Durchschnitt, Mengenaddition und Elemententfernung.

3. $T \subseteq S \Rightarrow \emptyset \subseteq T \wedge T \subseteq \emptyset \cup S$ ist trivial

4. Es ist $\text{split}^j(S, U, V) = \{\langle U \cup X, V \setminus W \rangle \mid W \subseteq V \wedge X \subseteq W \wedge |W| = j\}$

Es gilt $\text{empty?}(V \setminus W)$ genau dann, wenn $W = V$ ist. Zu zeigen bleibt

$$\text{empty?}(U \cap \alpha V) \wedge U \cup V \subseteq S \wedge T \subseteq S \Rightarrow U \subseteq T \wedge T \subseteq U \cup V \Leftrightarrow \exists k: \mathbb{N}. T \in \{U \cup X \mid X \subseteq V\}$$

Auch dies ist nun eine Standardeigenschaft.

5. Es ist $\text{split}^k(S, U, V) = \emptyset$ genau dann, wenn $j > |V|$ ist. In diesem Fall ist nämlich die Bedingung $W \subseteq V$ unerfüllbar.