Theoretische Informatik 2

Sabine Kuske
Linzer Str. 9a, OAS 3005
Tel.: 2335, 8794
kuske@informatik.uni-bremen.de
www.informatik.uni-bremen.de/theorie

7. April 2008

CE-S- Datentypen

Typen	Grundoperationen
Wahrheitswerte $BOOL$	$T, F: \ \to BOOL$ $\neg: BOOL \to BOOL$ $\wedge, \vee, \ldots : \ BOOL \times BOOL$ $\to BOOL$
Natürliche und ganze Zahlen $\mathbb{N}, \mathbb{Z}$	übliche Konstanten, arithm. Operationen und Vergleiche
Alphabete $A, B, \ldots$	Elemente $a: \ \to A, \ \ldots$ $\equiv: A \times A \to BOOL$
Sprachen $A^*, B^*, \ldots, \ \mathbb{N}^*, \ \ldots, (A^*)^*, \ \ldots$	λ, Konkatenation, $length$, $\equiv, \ \ldots$

Syntaxschema

spec	Name
opns: $decl_1, \ldots, decl_k$	Operationsdeklarationen
vars: $tv_1, \ldots tv_p$	Variablendeklarationen
eqns: $ce_1, \ldots, ce_l$	Bedingte Gleichungen

Operationsdeklaration

$$f\colon D_1 \times \cdots \times D_m \to D$$

- f: Name
- $D_1, \ldots, D_m$: Argumenttypen
- D: Wertetyp

Spezialfall Konstantendeklaration: $c\colon \;\to D$

Beispiele

- $count\colon A \times A^* \to \mathbb{N}$
- $5\colon \;\to \mathbb{N}$

Variablendeklaration

$$x \in D$$

- x: Name
- D: Typ

Beispiel

$$x \in A^*$$

Bedingte Gleichung

$$L = R \text{ falls } b$$

- L, R: Terme desselben Typs
- b: Term vom Typ $BOOL$

Term

Sei $DECL$ eine Menge von Operationsdeklarationen und X eine Menge von Variablendeklarationen.

Menge T_D aller Terme des Typs D

1. $(c\colon \to D) \in DECL$ impliziert $c \in T_D$;
2. $(x \in D) \in X$ impliziert $x \in T_D$;
3. $(f : D_1 \times \cdots \times D_k \to D) \in DECL$ und $t_i \in T_{D_i}$ für $i = 1, \ldots, k$ implizieren $f(t_1, \ldots, t_k) \in T_D$.

Wertzuweisung und Substitution

Variablen sind Platzhalter für Terme desselben Typs

Wertzuweisung: $a(x) \in T_D$ für alle $(x \in D) \in X$

Substitution (von x durch $a(x)$ in Term)

1. $c[a] = c$ für $(c\colon \ \to D) \in DECL$
2. $x[a] = a(x)$ für $(x \in D) \in X$
3. $f(t_1, \ldots, t_k)[a] = f(t_1[a], \ldots, t_k[a])$ für $(f\colon D_1 \times \cdots \times D_k \to D) \in DECL$, $t_i \in T_{D_i}$, $i = 1, \ldots, k$

Auswertungsschritt (Gleichungsanwendung)

1. Zuweisung aktueller Werte $a(x) \in T_D$ an die Variablen $x \in D$ und Substitution in Gleichung $L = R$:

$$L[a] \rightarrow R[a]$$

2. Dasselbe im Argumentterm:[1]

$$f(t_1, \ldots t_{i-1}, t_i, t_{i+1}, \ldots, t_k) \rightarrow f(t_1, \ldots, t_{i-1}, t'_i, t_{i+1}, \ldots, t_k)$$

falls $t_i \rightarrow t'_i$

[1]Wegen Rekursion in 2. kann direkte Gleichungsanwendung gemäß 1. beliebig weit innen im Term stattfinden.

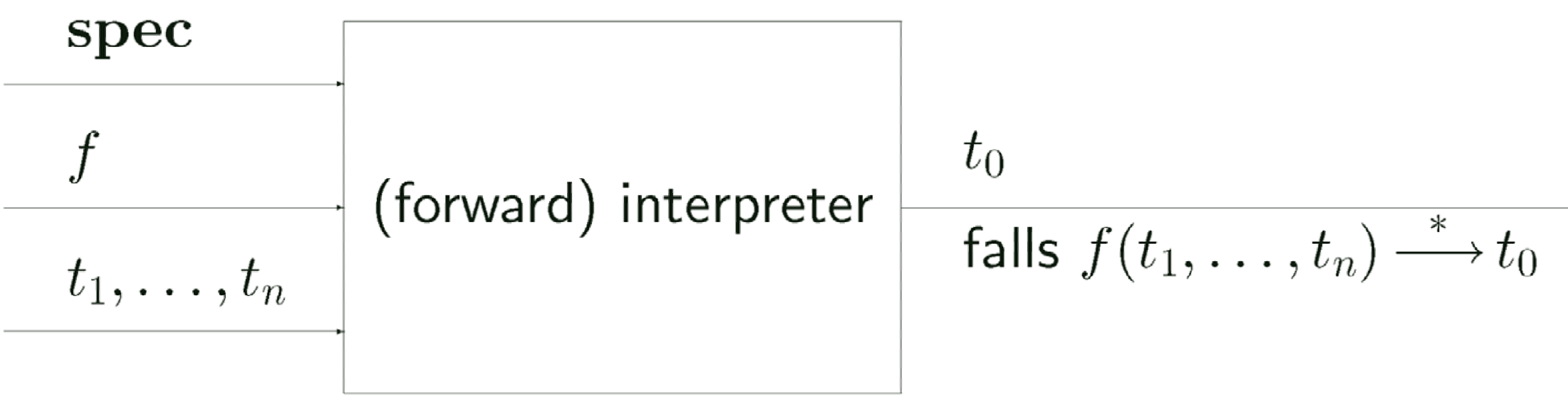
Berechnung

- Berechnung als Auswertungsschrittfolge:

$$\boxed{t \equiv t_0 \longrightarrow t_1 \longrightarrow \cdots \longrightarrow t_n \equiv t' \qquad\qquad (n = 0 : t = t')}$$

dafür kurz: $t \stackrel{*}{\longrightarrow} t'$

(Vorwärts-)Interpreter



- $t_1, \ldots t_n$: Werteterme, d.h. Terme, die nur aus Grundoperationen und ohne Variablen aufgebaut sind

Gleichwertigkeit

- Gleichwertigkeit wie Berechnung mit Rückwärtsschritt ($R = L$ statt $L = R$)

$$\boxed{t \longleftrightarrow t' \text{ falls } t \longrightarrow t' \text{ oder } t \leftarrow t'}$$

$\overset{*}{\longleftrightarrow}$ analog zu $\overset{*}{\longrightarrow}$

- für $t \overset{*}{\longleftrightarrow} t'$ auch $t = t'$
- Beachte: $\longrightarrow \subseteq \overset{*}{\longrightarrow} \subseteq \overset{*}{\longleftrightarrow} \supseteq \longleftrightarrow$

Fallunterscheidung

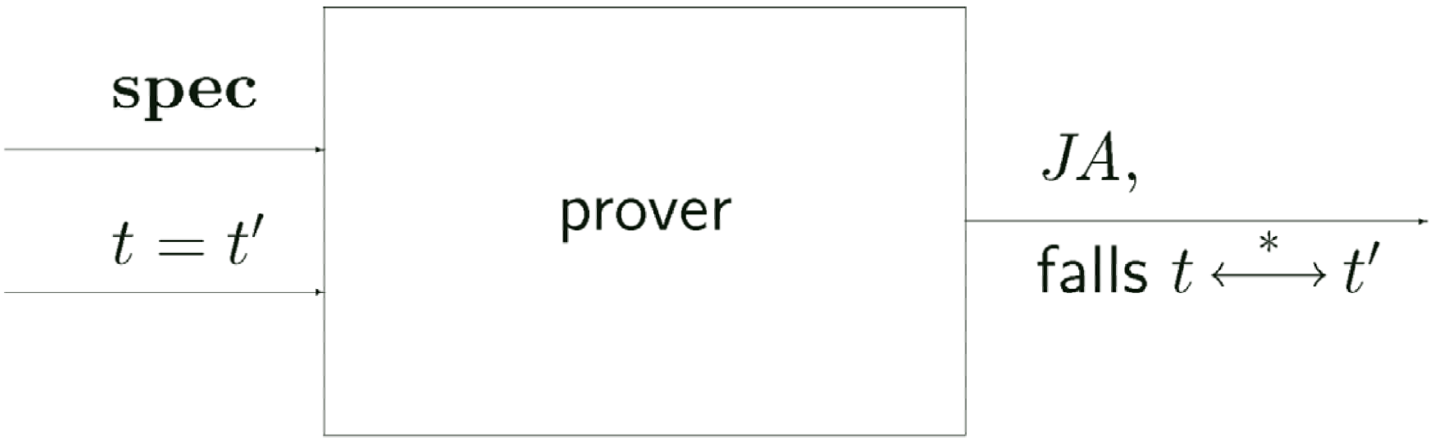
- if-then-else- als Abkürzung für 2 bedingte Gleichungen:

$$\boxed{t = \text{ if } b \text{ then } t_1 \text{ else } t_2 \text{ kurz für } \begin{cases} t = t_1 & \text{falls } b \\ t = t_2 & \text{falls } \neg b \end{cases}}$$

- Berechnungsschritte wie in Gleichungen aber bedingt:

$$\begin{array}{ll} t[a] \longrightarrow t_1[a] & \text{falls } b[a] \overset{*}{\longleftrightarrow} T \\ t[a] \longrightarrow t_2[a] & \text{falls } b[a] \overset{*}{\longleftrightarrow} F \end{array}$$

Gleichungsbeweiser



Aufwand

- Begrenzender Schlüsselfaktor
- Es macht kaum einen Unterschied, ob eine Berechnung zu lange dauert oder gar nicht endet
- Gesucht sind Methoden zur Aufwandsermittlung und -analyse
- Versuch in CE-S: Gleichungsanwendungen brauchen Zeit

CE-S-Ansatz zur Aufwandsermittlung

Zeitaufwand als maximale Zahl von Gleichungsanwendungen, die nötig sind, um eine Operation für bestimmte Argumente auszuwerten in Abhängigkeit von der Länge der eingegebenen Zeichenketten und der Größe der eingegebenen Zahlen.

Schreibweise: $T^{op}(n)$ bzw. $T^{op}(n_1, \ldots, n_k)$

k: Anzahl der Zeichenkettenargumente und der Zahlenargumente von op

Vorgehensweise (1)

- Auswirkung der spezifischen Gleichungen auf die Aufwandsfunktion

Beispiel

$$\boxed{insort(x,\lambda) = x} \quad \rightsquigarrow \quad \boxed{T^{insort}(0) = 1}$$

$$\boxed{insort(x, yv) = \left.\begin{array}{l} \text{if } x \leq y \text{ then } xyv \\ \text{else } y\,insort(x, v) \end{array}\right\}} \quad \rightsquigarrow$$

$$\boxed{\begin{array}{rl} T^{insort}(n+1) & =^{?} 1 + \left\{\begin{array}{l} 0 \text{ falls } x \leq y \\ T^{insort}(n) \text{ sonst} \end{array}\right. \\ & =_{\textit{Schlechtester Fall}} 1 + T^{insort}(n) \end{array}}$$

Vorgehensweise (3)

- Beim Beweis von Aufwandsabschätzungen werden oft Eigenschaften der Operationen benötigt (Beweis mit Hilfe von Gleichwertigkeit oft mit vollständiger Induktion über Wörter).

Zum Beispiel:

$$T^{sort}(n) = \frac{(n+2)(n+1)}{2}$$

falls $T^{insort}(n) = n + 1$ und $length(sort(u)) = length(u)$.