

# Informatik I at a glance

Martin Burger

06.02.2006

## Inhaltsverzeichnis

<b>Allgemeines</b>	<b>2</b>
Begriffe . . . . .	2
Techniken . . . . .	2
<b>Terminierung</b>	<b>3</b>
<b>Nachweis von Eigenschaften</b>	<b>3</b>
<b>Grammatiken</b>	<b>4</b>
<b>Typisierung</b>	<b>5</b>

## Allgemeines

### Begriffe:

- Algorithmus	(systematische, „schematisch“ ausführbare Verarbeitungsvorschrift)
- terminierend [für eine Eingabe]	(Ausführung endet nach endl. Schritten [für Eingabe])
- deterministisch	(Reihenfolge der el. Verarb.schritte eindeutig bestimmt für alle Eingaben)
- determiniert	(Resultat für jede Eingabe eindeutig bestimmt)
- sequenziell	(Hintereinanderausführung der Verarb.schritte)
- parallel	(Nebeneinanderausführung gewisser Verarb.schritte)
- Funktionen: total <-> partiell	(funktionieren für kompletten Def.bereich <-> nur Teil)
- Rekursive Funktionen	(Fu's, die in ihrem Rumpf [mind.] einen Aufruf von sich selbst enthalten)
- Prä-, Post-, Infixschreibweise	
- polymorph	(keine eindeutige Festlegung der Typen, Bsp.: a' list)
- monomorph	
- partielle Korrektheit	(wenn die Fu' terminiert, tut sie das richtige)
- totale Korrektheit	(die Fu' tut für jede Eingabe das richtige)
	[„partielle Korrektheit + Terminierung = totale Korrektheit“]
- Syntax	(präzise Festlegung der textuellen Gestalt eines Progs)
- Semantik	(präzise Festlegung der Bedeutung und Wirkungsweise eines Programms)
- strikte Auswertung [Wertaufruf, call-by-value]	(alle Argumente eines Funktionsaufrufs müssen definiert sein, sonst ist dessen Wert undefiniert)
- verzögerte Auswertung [lazy evaluation, call-by-need]	(nicht benötigte Terme werden nicht ausgewertet; if...then...else, andalso und orelse sind nicht-strikte Fu's)
- repetitive Rekursion	(der rekursive Aufruf steht allein, keine Anhäufung von Berechnungsschritten, die final abgearbeitet werden müssen ... üblicherweise nur Verringerung der Platzkomplexität, aber auch Zeitkomplexitätsänderung möglich)

### Techniken:

- Einbettung
- verschränkte Rekursion



$A = \text{nat}$  ,  $M = \text{nat}$  , „<“ = < ,  $h : A \rightarrow M$  ,  $h(n) = n$

Induktionsvoraussetzung : Für alle  $n' \in A$  mit  $h(n') < h(n)$  gilt:  
 $\text{sum}(n') = \text{Summe } i=0 \text{ bis } n' \text{ von } i^2$

Induktionsschluss:

1. Fall:  $n = 0$

$$\text{sum}(0) = 0 = \text{Summe}(\text{bis } 0)$$

2. Fall:  $n > 0$

Nach I.V. gilt:  $\text{sum}(n-1) = \text{Summe } i=0 \text{ bis } (n-1) \text{ von } i^2$

$$\text{sum}(n) = n * n + \text{sum}(n-1) = n^2 + \text{Summe}(\text{bis } n-1) = \text{Summe}(\text{bis } n)$$

## Grammatiken

- Alphabet [endliche Menge von Zeichen, Bez.:  $\Sigma$ ]
- Zeichenreihe [endliche Folge von Elementen aus  $\Sigma$  ,  $\epsilon$  ist leere Zeichenkette]
- formale Sprache [Teilmenge von  $\Sigma^*$ ]

### BNF (Backus-Naur-Form):

- zugrundeliegendes Alphabet  $\Sigma$
- weiteres Alphabet  $V$
- Startzeichen  $S \in V$
- Produktionsregeln

$A ::= \beta$  [ $A \in V$  und  $\beta$  BNF-Satzform]

BNF-Satzform:

(1)  $\beta_1 \mid \beta_2 \mid \dots \mid \beta_n$  ( $n \geq 1$ )

(2)  $\beta$  ist Aneinanderreihung von Elementen aus  $A, V$   
bzw.  $\{x\}, \{x\}^*, \{x\}^+, [x]$

*Beispiel:* Unterstrichene Wörter:

$\Sigma = \text{ASCII} - \text{Zeichensatz}$

$V = \{\text{UnderWord}, \text{UnderChar}, \text{Char}\}$

$S = \text{UnderWord}$

$P =$

$\langle \text{UnderWord} \rangle ::= \{ \langle \text{UnderChar} \rangle \}^*$

$\langle \text{UnderChar} \rangle ::= \langle \text{Char} \rangle [\text{BS}] \_$

$\langle \text{Char} \rangle ::= a \mid \dots \mid z \mid A \mid \dots \mid Z$

**Typisierung****Typisierungsregeln:**

- (R1)  $\Gamma_1 \triangleright t_1 : \text{typ}_1 \quad \Gamma_2 \triangleright t_2 : \text{typ}_2$   
 $\vdash \Gamma_1 \cup \Gamma_2 \triangleright (t_1, t_2) : \text{typ}_1 \times \text{typ}_2$
- (R2)  $\Gamma_1 \triangleright t_1 : \text{int} \quad \Gamma_2 \triangleright t_2 : \text{int} \quad \Gamma_3 \triangleright \text{op} : \text{int} \times \text{int} \rightarrow \text{int}$   
 $\vdash \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \triangleright t_1 \text{ op } t_2 : \text{int}$
- (R3)  $\Gamma_1 \triangleright b : \text{bool} \quad \Gamma_2 \triangleright t_1 : \text{typ} \quad \Gamma_3 \triangleright t_2 : \text{typ}$   
 $\vdash \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \triangleright \text{if } b \text{ then } t_1 \text{ else } t_2 : \text{typ}$
- (R4)  $\Gamma \cup \{x : \text{typ}_1\} \triangleright t : \text{typ}_2$   
 $\vdash \Gamma \triangleright \text{fn } x \Rightarrow t : \text{typ}_1 \rightarrow \text{typ}_2$
- (R5)  $\Gamma \triangleright t : \text{typ}_1 \rightarrow \text{typ}_2 \quad \Gamma \triangleright s : \text{typ}_1$   
 $\vdash \Gamma \triangleright t(s) : \text{typ}_2$
- (R6)  $\Gamma \triangleright (\text{fn } x \Rightarrow s)(t) : \text{typ}$   
 $\vdash \Gamma \triangleright \text{let val } x = t \text{ in } s \text{ end} : \text{typ}$

<i>Beispiel:</i>	(1) $\emptyset \triangleright 2 : \text{int}$	Typaxiom
	(2) $\emptyset \triangleright + : \text{int} \rightarrow \text{int}$	~
	(3) $\{a : \text{int}\} \triangleright a : \text{int}$	~
	(4) $\{x : \text{int}\} \triangleright x : \text{int}$	~
	(5) $\{a : \text{int}, x : \text{int}\} \triangleright a + x + 2 : \text{int}$	2*(R2) mit Prämissen 1,2,3,4
	(6) $\{a : \text{int}\} \triangleright \text{fn } x \Rightarrow a + x + 2 : \text{int} \rightarrow \text{int}$	(R4) mit Prämisse 5