

# Unentscheidbarkeit des Gültigkeitsproblems

Wir gehen in zwei Schritten vor

- Das Terminierungsproblem ist unentscheidbar  
Es gibt kein Programm, welches als Eingabe ein Programm  $P$  und eine Belegung  $\beta$  der Variablen von  $P$  akzeptiert und entscheidet, ob  $P$  mit  $\beta$  als Anfangsbelegung terminiert.
- Wenn das Terminierungsproblem entscheidbar wäre, dann wäre auch das Gültigkeitsproblem entscheidbar

# Unentscheidbarkeit von Terminierung

Wir betrachten nur Programme mit Variablen über  $\mathbb{N}$

Wir nehmen an, dass die Variablen des Programms  $x_1, x_2, \dots$  heißen

**Fakt:** Programme können als Zahlen kodiert werden.

**Fakt:** Es gibt zwei Programme, die andere Programme kodieren und dekodieren

- Der Kodierer. Eingabe: eine Zahl  $n$ . Ausgabe: das Programm  $\Pi_n$  mit dem Code  $n$ , falls  $n$  ein Programm kodiert, sonst KP
- Der Dekodierer. Eingabe: Ein Programm  $P$ . Ausgabe: der Code von  $P$ , d.h., die Zahl  $n$  mit  $P = \Pi_n$

**Annahme:** Es gibt ein Programm  $T(n, m)$ , welches für jedes Paar  $n, m \in \mathbb{N}$  terminiert mit

KP	falls $n$ kein Programm kodiert
JA	falls $n$ ein Programm kodiert und $\Pi_n(m)$ terminiert
NEIN	sonst

Wir zeigen, dass diese **Annahme** zu einem Widerspruch führt

# Der Widerspruch

**Fakt:** Aus der **Annahme** folgt, dass es ein Programm  $T'(n)$  gibt, welches

(mit JA) terminiert	falls $n$ ein Programm kodiert und
	$\Pi_n(n)$ nicht terminiert
nicht terminiert	sonst

Sei nun  $k$  der Code des Programms  $T'$ , d.h.  $\Pi_k = T'$ .

Wir haben:

$T'(k)$ terminiert	$T'(k)$ terminiert nicht
$\Rightarrow \Pi_k(k)$ terminiert nicht	$\Rightarrow \Pi_k(k)$ terminiert
$\Rightarrow T'(k)$ terminiert nicht	$\Rightarrow T'(k)$ terminiert

Damit ist die **Annahme falsch**.

# Unentscheidbarkeit des Gültigkeitsproblems

Wir ordnen jedem Programm  $P$  und Variablenbelegung  $\beta$  eine Formel  $\phi_{P\beta}$  der Prädikatenlogik zu mit

$\phi_{P\beta}$  ist gültig

genau dann, wenn

das Programm  $P$  mit der Anfangsbelegung  $\beta$  terminiert

Die Formel  $\phi_{P,\beta}$  kann von einem Programm konstruiert werden.

Daraus folgt, dass kein Programm das Gültigkeitsproblem der Prädikatenlogik lösen kann!

# if-goto-Programme

$Prog ::= l : Zuw$	(Zuweisung)
$l : \mathbf{goto} \ell'$	(unbedingter Sprung)
$l : \mathbf{if} x_i \neq 0 \mathbf{then goto} \ell'$	(bedingter Sprung)
$l : \mathbf{halt}$	(Terminierung)
$Prog ; Prog$	(Hintereinanderausführung)
$Zuw ::= x_i := 0 \quad   \quad x_i := x_j$	
$x_i := x_j + 1 \quad   \quad x_i := x_j - 1$	
$\ell ::= 1 \quad   \quad 2 \quad   \quad 3 \quad   \quad \dots$	

# Beispiel

```
1:  if  $x_1 = 0$  then goto 4;  
2:   $x_1 := x_1 - 1$ ;  
3:  goto 1;  
4:  halt
```

**Behauptung:** if-goto-Programme können alle anderen Programme simulieren.

**Konsequenz:** Es gibt kein Programm, egal in welcher Sprache, für die Terminierung von if-goto-Programmen

# Notationen und Definitionen

Mit  $k$  bezeichnen wir die Anzahl der Anweisungen von  $P$   
(Die letzte Anweisung ist immer **halt**)

Mit  $n$  bezeichnen wir die Anzahl der Variablen von  $P$   
(D.h. die Variablen von  $P$  sind  $x_1, \dots, x_n$ )

Eine **Konfiguration** von  $P$  ist eine Tupel  $(Z, m_1, \dots, m_n) \in \mathbb{N}^{n+1}$ .  
 $Z$  bezeichnet die aktuelle Anweisung und  $m_1, \dots, m_n$  die aktuelle  
Belegung der Variablen

**Konvention:** die Nachfolgekongfiguration einer Konfiguration der Gestalt  
 $(\mathbf{k}, m_1, \dots, m_n)$  ist wieder  $(\mathbf{k}, m_1, \dots, m_n)$

# Symbole der Formel $\phi_{P,\beta}$

- $R$ , Prädikatensymbol,  $(n + 2)$ -stellig;
- $<$ , Prädikatensymbol, 2-stellig;
- $f$ , Funktionssymbol, 1-stellig;
- $0$ , Konstante.

# Eine Interpretation $\mathcal{A}$

- Universum:  $\mathbb{N}$
- $<^{\mathcal{A}}$  ist die gewöhnliche Ordnung auf  $\mathbb{N}$
- $0^{\mathcal{A}} = 0$
- $f^{\mathcal{A}}$  ist die Nachfolgerfunktion, i.e.,  $f^{\mathcal{A}}(n) = n + 1$
- $R^{\mathcal{A}}(s, Z, m_1, \dots, m_n) = 1$  wenn  $(Z, m_1, \dots, m_n)$  die Konfiguration von  $P$  nach  $s$  Schritten ist (für die Anfangsbelegung  $\beta$ ).

# Die Hilfsformel $\psi_{P,\beta}$

$$\psi_{P,\beta} = \psi_0 \wedge R(\mathbf{0}, \beta) \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$$

Unter der Interpretation  $\mathcal{A}$  besagt  $R(\mathbf{0}, \beta)$ , dass  $\beta$  die Anfangsbelegung des Programms  $P$  ist

Unter der Interpretation  $\mathcal{A}$  beschreibt  $\psi_i$  mit  $\mathbf{i} \in \{1 \dots, k-1\}$  die Wirkungsweise der  $\mathbf{i}$ -ten Zeile von  $P$ . Zum Beispiel:

- Wenn die  $\mathbf{i}$ -te Zeile der Gestalt  $\mathbf{i}: x_j := x_j + 1$  ist, dann

$$\begin{aligned} \psi_i = & \forall x \forall y_1 \dots \forall y_n ( \\ & R(x, \mathbf{i}, y_1, \dots, y_n) \rightarrow \\ & R(f(x), f(\mathbf{i}), y_1, \dots, y_{j-1}, f(y_j), y_{j+1}, \dots, y_n) \\ & ) \end{aligned}$$

- Wenn die  $i$ -te Zeile der Gestalt  $\mathbf{i}$ : **if**  $x_j = 0$  **then goto**  $\mathbf{j}$  ist, dann

$$\begin{aligned} \psi_i = & \forall x \forall y_1 \dots \forall y_n ( \\ & R(x, \mathbf{i}, y_1, \dots, y_n) \rightarrow \\ & ( \quad x_j = \mathbf{0} \quad \wedge \quad R(f(x), \mathbf{j}, y_1, \dots, y_n) \\ & \quad \vee \\ & \quad \neg(x_j = \mathbf{0}) \quad \wedge \quad R(f(x), f(\mathbf{i}), y_1, \dots, y_n) \\ & ) \\ & ) \end{aligned}$$

$\psi_0$  garantiert, dass unter allen Interpretationen  $<$  eine Ordnung ist mit  $\mathbf{0}$  als kleinstem Element, dass stets  $x \leq f(x)$  gilt, und dass  $f(x)$  der unmittelbare  $\leq$ -Nachfolger von  $x$  ist

$$\begin{aligned}\psi_0 = & \forall x \forall y (x < y \rightarrow \neg(y < x)) \quad \wedge \\ & \forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z) \quad \wedge \\ & \forall x (\mathbf{0} < x \vee \mathbf{0} = x) \quad \wedge \\ & \forall x (x < f(x)) \quad \wedge \\ & \forall x \forall z (x < z \rightarrow (f(x) < z \vee f(x) = z))\end{aligned}$$

# Die Formel $\phi_{P,\beta}$

Wir setzen

$$\phi_{P,\beta} = \psi_{P,\beta} \rightarrow \exists x \exists y_1 \dots \exists y_n R(x, \mathbf{k}, y_1, \dots, y_n)$$

Wir haben

$\phi_{P,\beta}$  ist gültig

genau dann, wenn

das Programm  $P$  mit der Anfangsbelegung  $\beta$  terminiert