

## Verbesserung:

- Nicht alle Funktionen aus  $\mathbb{D}_f \rightarrow \mathbb{D}_f$  kommen wirklich vor :-)
- Alle vorkommenden Funktionen  $\lambda D. \perp \neq M$  sind von der Form:

$$\begin{aligned} M &= \{x \mapsto (b_x \sqcup \bigsqcup_{y \in I_x} y) \mid x \in \text{Vars}\} && \text{wobei:} \\ M D &= \{x \mapsto (b_x \sqcup \bigsqcup_{y \in I_x} D y) \mid x \in \text{Vars}\} && \text{für } D \neq \perp \end{aligned}$$

- Sei  $\mathbb{M}$  die Menge aller dieser Funktionen. Dann gilt für  $M_1, M_2 \in \mathbb{M}$  ( $M_1 \neq \lambda D. \perp \neq M_2$ ):

$$(M_1 \sqcup M_2) x = (M_1 x) \sqcup (M_2 x)$$

- Für  $k = \#\text{Vars}$  hat  $\mathbb{M}$  die Höhe  $(k+1) \cdot k + 1$  :-)

## Verbesserung (Forts.):

→ Auch die Komposition lässt sich direkt implementieren:

$$(M_1 \circ M_2) x = b' \sqcup \bigsqcup_{y \in I'} y \quad \text{mit}$$

$$b' = b \sqcup \bigsqcup_{z \in I} b_z$$

$$I' = \bigcup_{z \in I} I_z \quad \text{sofern}$$

$$M_1 x = b \sqcup \bigsqcup_{y \in I} y$$

$$M_2 z = b_z \sqcup \bigsqcup_{y \in I_z} y$$

→ Die Effekte von Zuweisungen sehen dann so aus:

$$\llbracket x = e; \rrbracket^\# = \begin{cases} \text{Id}_{Vars} \oplus \{x \mapsto c\} & \text{falls } e = c \in \mathbb{Z} \\ \text{Id}_{Vars} \oplus \{x \mapsto y\} & \text{falls } e = y \in Vars \\ \text{Id}_{Vars} \oplus \{x \mapsto \top\} & \text{sonst} \end{cases}$$

... im Beispiel:

$$\begin{aligned} \llbracket t = 0; \rrbracket^\# &= \{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto \text{ret}, \boxed{t \mapsto 0}\} \\ \llbracket b_1 = a_1; \rrbracket^\# &= \{a_1 \mapsto a_1, \boxed{b_1 \mapsto a_1}, \text{ret} \mapsto \text{ret}, t \mapsto t\} \end{aligned}$$

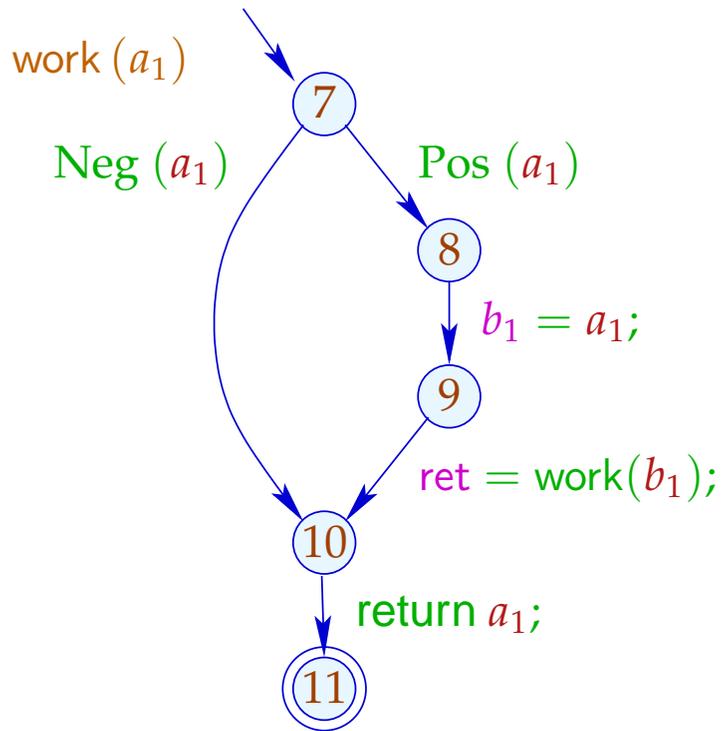
Um die Analyse zu implementieren, müssen wir nur noch den Effekt eines Aufrufs  $k = (\_, \text{ret} = f(b_1, \dots, b_k); \_)$  aus dem Effekt der Funktion  $f$  ermitteln:

$$\begin{aligned} \llbracket k \rrbracket^\# &= H_f(\llbracket f \rrbracket^\#) && \text{wobei:} \\ H_f(M) &= \text{Id} \oplus \{\text{ret} \mapsto ((M \circ E_f) \text{ret})\} \\ E_f x &= \begin{cases} b_i & \text{falls } x = a_i \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

... im Beispiel:

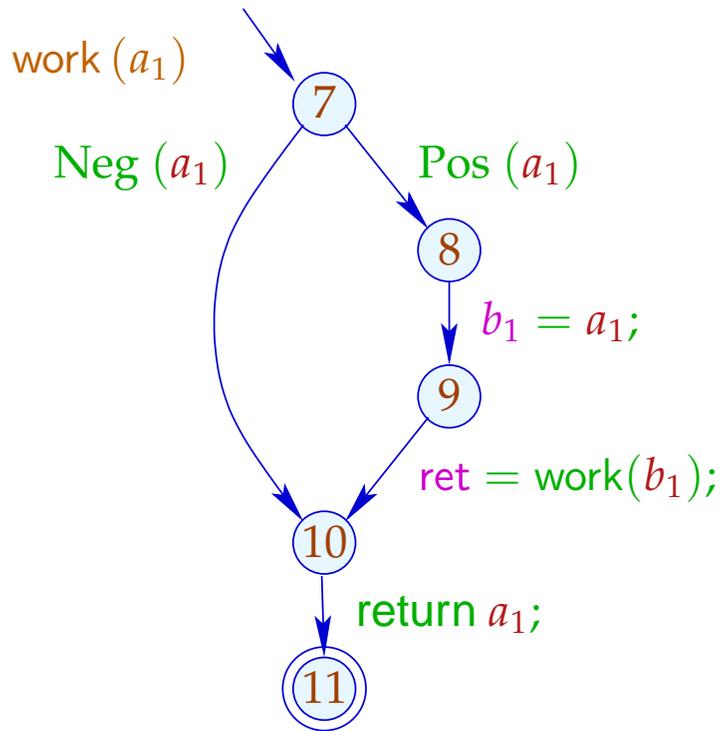
$$\begin{aligned} \text{Falls } \llbracket \text{work} \rrbracket^\# &= \{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto a_1\} \\ \text{dann } H_{\text{work}} \llbracket \text{work} \rrbracket^\# &= \text{Id} \oplus \{\text{ret} \mapsto ((\llbracket \text{work} \rrbracket^\# \circ E_f) \text{ret})\} \\ &= \text{Id} \oplus \{\text{ret} \mapsto (\{a_1 \mapsto b_1, b_1 \mapsto 0, \text{ret} \mapsto b_1\} \text{ret})\} \\ &= \{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto b_1\} \end{aligned}$$

Damit können wir die Fixpunkt-Iteration durchführen :-)



	1
7	$\{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto \text{ret}\}$
10	$\{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto \text{ret}\}$
11	$\{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto a_1\}$
8	$\{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto \text{ret}\}$
9	$\{a_1 \mapsto a_1, b_1 \mapsto a_1, \text{ret} \mapsto \text{ret}\}$

$$\begin{aligned}
 \llbracket (9, \dots, 10) \rrbracket^\# \circ \llbracket 9 \rrbracket^\# &= \{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto b_1\} \circ \\
 &\quad \{a_1 \mapsto a_1, b_1 \mapsto a_1, \text{ret} \mapsto \text{ret}\} \\
 &= \{a_1 \mapsto a_1, b_1 \mapsto a_1, \text{ret} \mapsto a_1\}
 \end{aligned}$$



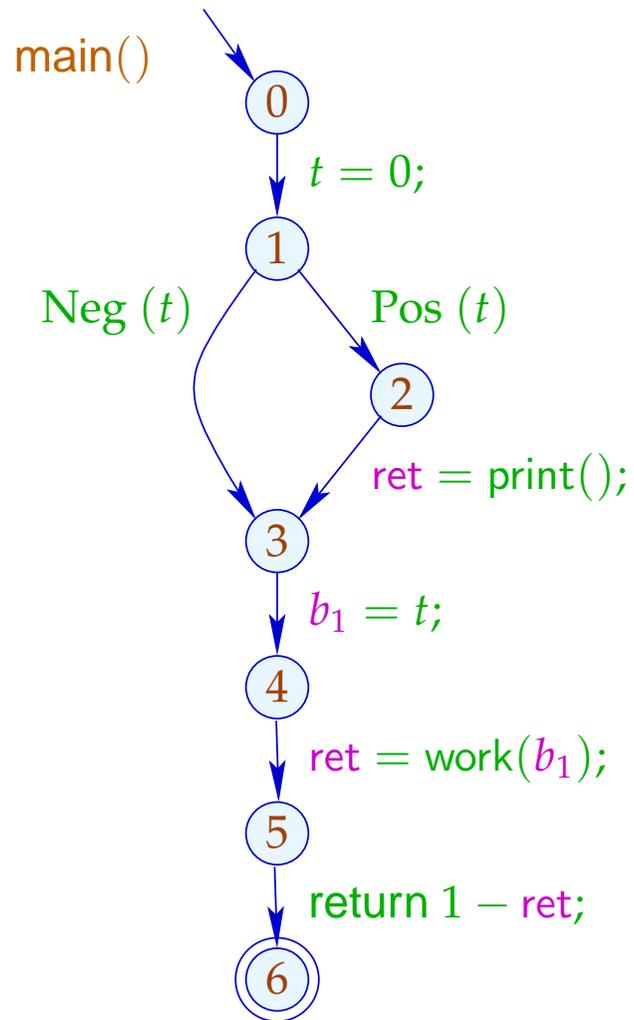
	2
7	$\{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto \text{ret}\}$
10	$\{a_1 \mapsto a_1, b_1 \mapsto a_1 \sqcup b_1, \text{ret} \mapsto a_1 \sqcup \text{ret}\}$
11	$\{a_1 \mapsto a_1, b_1 \mapsto a_1 \sqcup b_1, \text{ret} \mapsto a_1\}$
8	$\{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto \text{ret}\}$
9	$\{a_1 \mapsto a_1, b_1 \mapsto a_1, \text{ret} \mapsto \text{ret}\}$

$$\begin{aligned}
 \llbracket (9, \dots, 10) \rrbracket^\# \circ \llbracket 9 \rrbracket^\# &= \{a_1 \mapsto a_1, b_1 \mapsto b_1, \text{ret} \mapsto b_1\} \circ \\
 &\quad \{a_1 \mapsto a_1, b_1 \mapsto a_1, \text{ret} \mapsto \text{ret}\} \\
 &= \{a_1 \mapsto a_1, b_1 \mapsto a_1, \text{ret} \mapsto a_1\}
 \end{aligned}$$

Wenn wir die Effekte von Funktionsaufrufen kennen, können wir ein Constraintsystem aufstellen, um den abstrakten Zustand bei Erreichen eines Punkts ermitteln:

$$\begin{array}{ll}
 \mathcal{R}[\text{main}] & \sqsupseteq \text{enter}_0 d_0 \\
 \mathcal{R}[f] & \sqsupseteq \text{enter}_f^\# (\mathcal{R}[u]) \quad k = (u, \text{ret} = f(a_1, \dots, a_k);, \_) \quad \text{Aufruf} \\
 \mathcal{R}[v] & \sqsupseteq \mathcal{R}[f] \quad v \text{ Anfangspunkt von } f \\
 \mathcal{R}[v] & \sqsupseteq \llbracket k \rrbracket^\# (\mathcal{R}[u]) \quad k = (u, \_, v) \quad \text{Kante}
 \end{array}$$

... im Beispiel:



0	$\{b_1 \mapsto 0, \text{ret} \mapsto 0, t \mapsto 0\}$
1	$\{b_1 \mapsto 0, \text{ret} \mapsto 0, t \mapsto 0\}$
2	$\{b_1 \mapsto 0, \text{ret} \mapsto 0, t \mapsto 0\}$
3	$\{b_1 \mapsto 0, \text{ret} \mapsto \top, t \mapsto 0\}$
4	$\{b_1 \mapsto 0, \text{ret} \mapsto \top, t \mapsto 0\}$
5	$\{b_1 \mapsto 0, \text{ret} \mapsto 0, t \mapsto 0\}$
6	$\{b_1 \mapsto 0, \text{ret} \mapsto \top, t \mapsto 0\}$

## Diskussion:

- Zumindest **Kopier-Konstanten** lassen sich interprozedural ermitteln.
- Dazu mussten wir Bedingungen und kompliziertere Zuweisungen ignorieren :-)
- In der zweiten Phase hätten wir allerdings exakter rechnen können :-)
- Die weitere Abstrahierung war aus zwei Gründen notwendig:
  - (1) Die Menge der auftretenden Transformer  $\mathbb{M} \subseteq \mathbb{D} \rightarrow \mathbb{D}$  muss **endlich** sein;
  - (2) Die Funktionen  $M \in \mathbb{M}$  müssen **effizient** implementierbar sein :-)
- Auf die zweite Bedingung kann evt. verzichtet werden ...

## Beobachtung:

Sharir/Pnueli, Cousot

- Oft werden Funktionen nur mit **wenigen** verschiedenen abstrakten Argumenten aufgerufen.
- Man könnte dann doch jede Funktion für nur genau diese Aufrufe analysieren :-)
- Stelle das folgende Constraint-System auf:

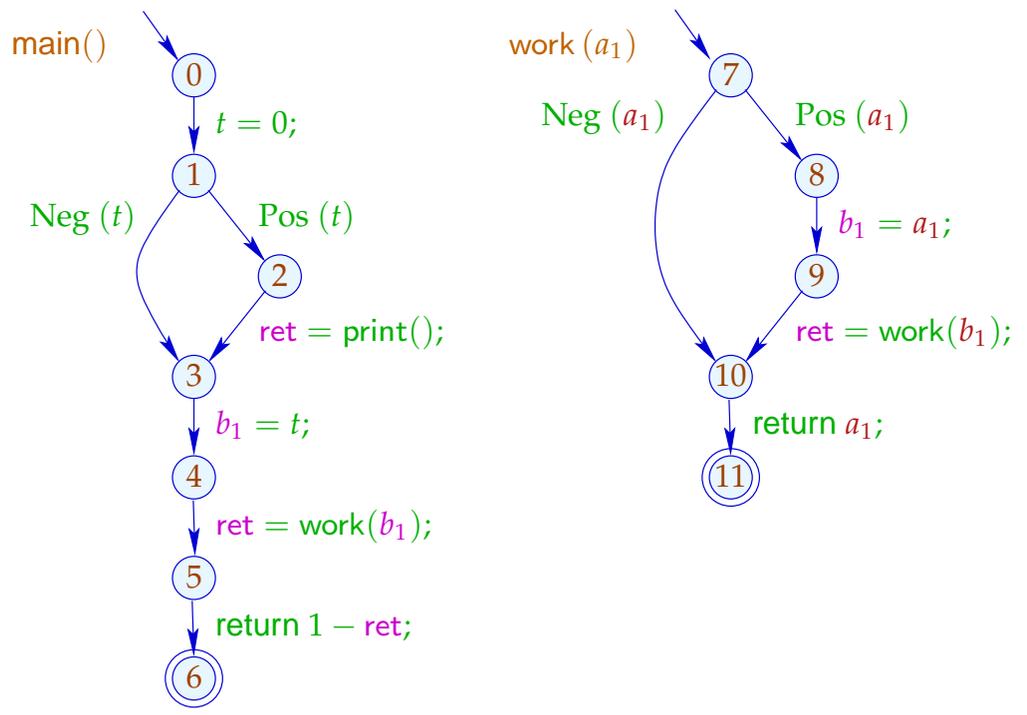
$$\begin{aligned} \llbracket v, a \rrbracket^\# &\sqsupseteq a && v \text{ Eintrittspunkt} \\ \llbracket v, a \rrbracket^\# &\sqsupseteq \text{combine}^\# (\llbracket u, a \rrbracket, \llbracket f, \text{enter}_f^\# \llbracket u, a \rrbracket^\# \rrbracket^\#) \\ &&& (u, \text{ret} = f(a_1, \dots, a_k);, v) \text{ Aufruf} \\ \llbracket v, a \rrbracket^\# &\sqsupseteq \llbracket \text{lab} \rrbracket^\# \llbracket u \rrbracket^\# && k = (u, \text{lab}, v) \text{ Kante} \\ \llbracket f, a \rrbracket^\# &\sqsupseteq \llbracket \text{stop}_f, a \rrbracket^\# && \text{stop}_f \text{ Endpunkt von } f \\ // \llbracket v, a \rrbracket^\# &= && \text{Wert des Effekts für das Argument } a. \end{aligned}$$

## Diskussion:

- Dieses Constraint-System ist i.a. riesengroß :-)
- Wir wollen es aber gar nicht komplett lösen !!!
- Uns reicht es, die korrekten Werte für alle Aufrufe zu ermitteln, die vorkommen, d.h. für die Berechnung des Werts  $\llbracket \text{main}(), a_0 \rrbracket^\#$  benötigt werden  $\implies$  wir verwenden unseren lokalen Fixpunkt-Algorithmus :-))
- Der Fixpunkt-Algo liefert uns sogar noch die Menge der aktuellen Parameter  $a \in \mathbb{D}$ , für die eine Funktion (möglicherweise) aufgerufen wird sowie die Werte an allen ihren Programm-Punkten für jeden dieser Aufrufe :-)

... im Beispiel:

Versuchen wir einfach einmal eine **volle** Konstanten-Propagation ...



0	$\lambda x.0$	$\lambda x.0$
1	$\lambda x.0$	$\lambda x.0$
2	$\lambda x.0$	$\perp$
3	$\lambda x.0$	$\lambda x.0$
4	$\lambda x.0$	$\lambda x.0$
7	$\lambda x.0$	$\lambda x.0$
8	$\lambda x.0$	$\perp$
9	$\lambda x.0$	$\perp$
10	$\lambda x.0$	$\lambda x.0$
11	$\lambda x.0$	$\lambda x.0$
5	$\lambda x.0$	$\lambda x.0$
6	$\lambda x.0$	$(\lambda x.0) \oplus \{\text{ret} \mapsto 1\}$
main()	$\lambda x.0$	$(\lambda x.0) \oplus \{\text{ret} \mapsto 1\}$