Anchored Semi-Unification

Tobias Tebbi

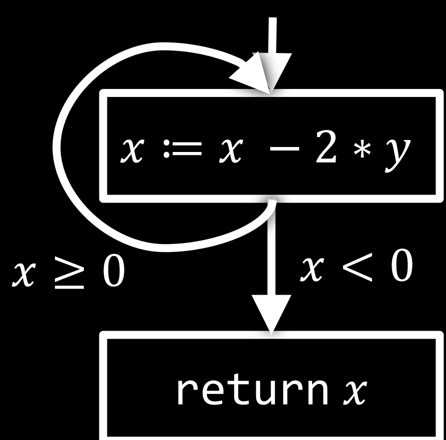
Final Talk – Master Thesis

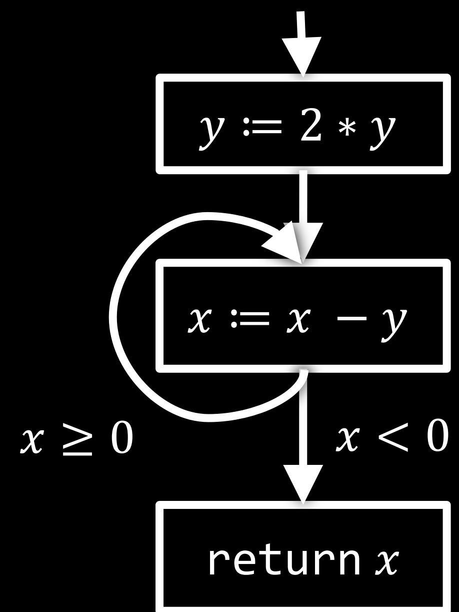
Overview

- Motivation: Translation Validation
- Nonnested Recursion Schemes
- Anchored Semi-Unification

Motivation: Translation Validation

- Check semantic equivalence of input and output of an optimization phase.



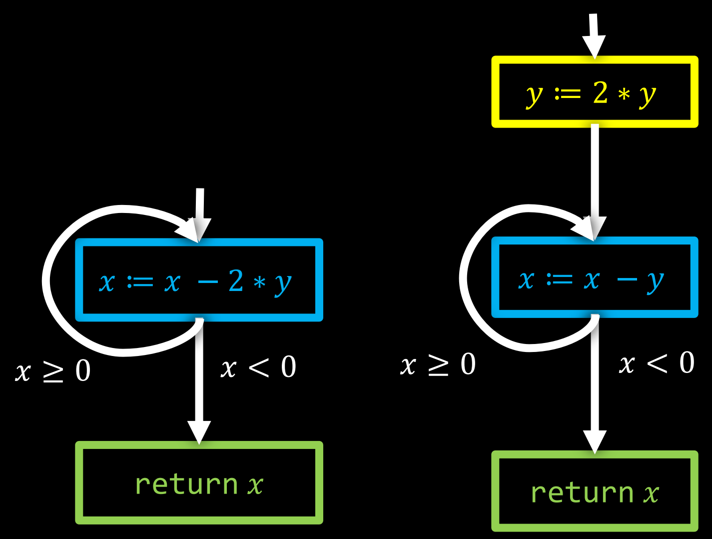


Recursion Schemes

- Encode the CFGs as recursion schemes

$$
\begin{aligned}
&P_1(x, y) \coloneqq \\
&\quad \texttt{if } x \geq 0 \\
&\quad \texttt{then } P_1(x - 2 * y, y) \\
&\quad \texttt{else } P_2(x - 2 * y, y)
\end{aligned}
$$

$$P_2(x, y) \coloneqq \texttt{return } x$$



$$Q_1(x, y) \coloneqq Q_2(x, 2 * y)$$

$$
\begin{aligned}
&Q_2(x, y) \coloneqq \\
&\quad \texttt{if } x \geq 0 \\
&\quad \texttt{then } Q_2(x - y, y) \\
&\quad \texttt{else } Q_3(x - y, y)
\end{aligned}
$$

$$Q_3(x, y) \coloneqq \texttt{return } x$$

Recursion Schemes

- We can unfold the definitions of the procedures.

$$P_1(x,y)$$

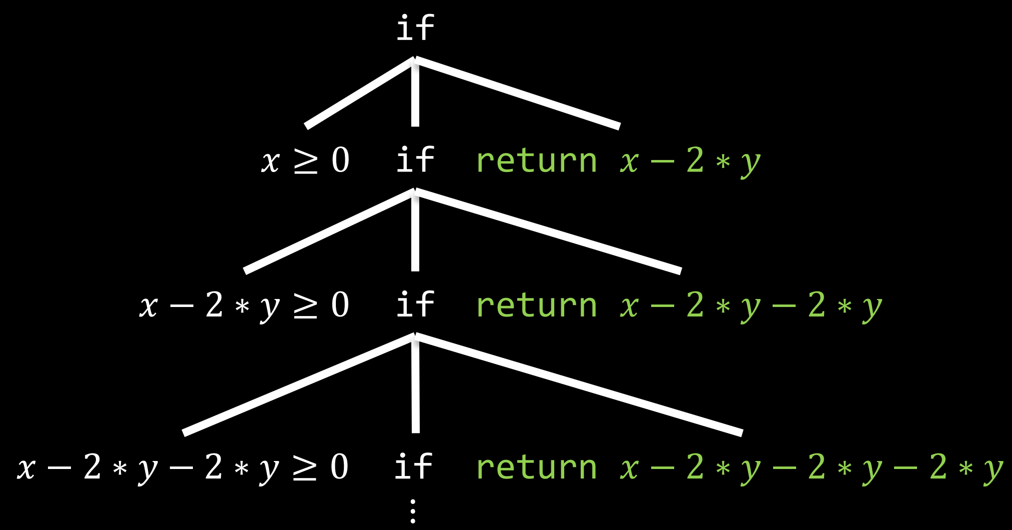
$$\texttt{if}\ x \geq 0\ \texttt{then}\ P_1(x-2*y,y)\ \texttt{else}\ P_2(x-2*y,y)$$

$$\texttt{if}\ x \geq 0\ \texttt{then}\ P_1(x-2*y,y)\ \texttt{else}\ \texttt{return}\ x-2*y$$

$$\texttt{if}\ x \geq 0\ \texttt{then}\ \texttt{if}\ x-2*y \geq 0\ \texttt{then}\ P_1(x-2*y-2*y,y)\ \texttt{else}\ \texttt{return}\ x-2*y-2*y\ \texttt{else}\ \texttt{return}\ x-2*y$$

$P_1(x,y) \coloneqq$
  $\texttt{if}\ x \geq 0$
  $\texttt{then}\ P_1(x-2*y,y)$
  $\texttt{else}\ P_2(x-2*y,y)$

$P_2(x,y) \coloneqq \texttt{return}\ x$



$Q_1(x,y) \coloneqq Q_2(x, 2*y)$

$Q_2(x,y) \coloneqq$
  $\texttt{if}\ x \geq 0$
  $\texttt{then}\ Q_2(x-y,y)$
  $\texttt{else}\ Q_3(x-y,y)$

$Q_3(x,y) \coloneqq \texttt{return}\ x$

Recursion Schemes

- Q_1 results in the same infinite tree.

$Q_1(x, y)$

$Q_2(x, 2*y)$

if $x \geq 0$ then $Q_2(x - 2*y, 2*y)$ else $Q_3(x - 2*y, 2*y)$

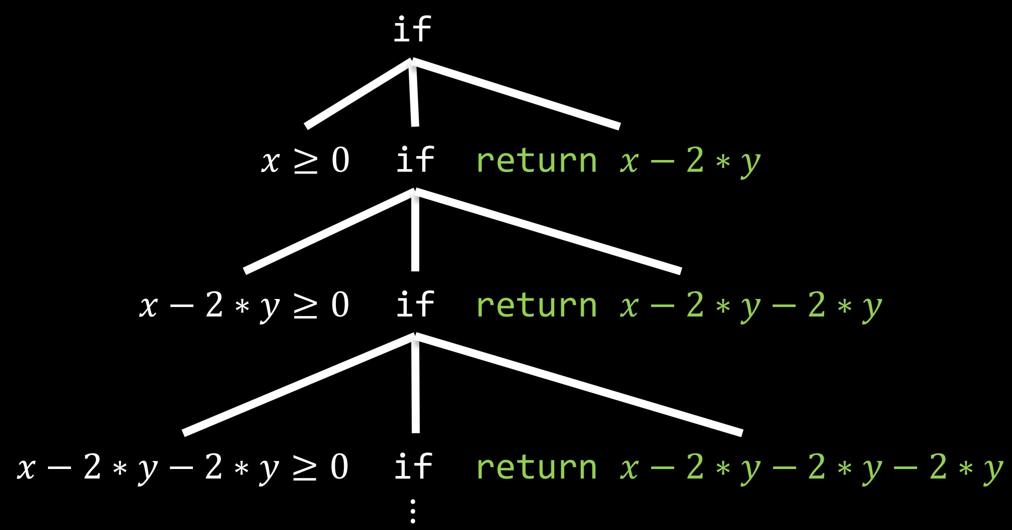
if $x \geq 0$ then $Q_2(x - 2*y, 2*y)$ else return $x - 2*y$

if $x \geq 0$ then if $x - 2*y \geq 0$ then $Q_2(x - 2*y - 2*y, 2*y)$ else return $x - 2*y - 2*y$ else return $x - 2*y$

if $x \geq 0$ then if $x - 2*y \geq 0$ then $P_1(x - 2*y - 2*y, y)$ else return $x - 2*y - 2*y$ else return $x - 2*y$

$P_1(x, y) :=$
  if $x \geq 0$
  then $P_1(x - 2*y, y)$
  else $P_2(x - 2*y, y)$

$P_2(x, y) :=$ return x



$Q_1(x, y) := Q_2(x, 2*y)$

$Q_2(x, y) :=$
  if $x \geq 0$
  then $Q_2(x - y, y)$
  else $Q_3(x - y, y)$

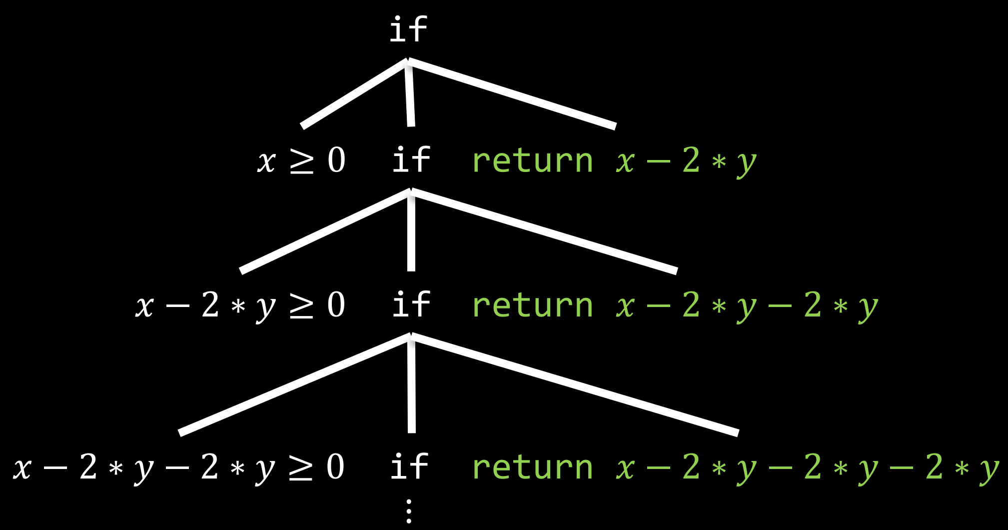
$Q_3(x, y) :=$ return x

Recursion Schemes

- Q_1 results in the same infinite tree.
- If the schemes produce the same tree, then the initial CFGs are equivalent.

$P_1(x, y) \coloneqq$
  $\texttt{if}\ x \geq 0$
  $\texttt{then}\ P_1(x - 2 * y, y)$
  $\texttt{else}\ P_2(x - 2 * y, y)$

$P_2(x, y) \coloneqq \texttt{return}\ x$



$Q_1(x, y) \coloneqq Q_2(x, 2 * y)$

$Q_2(x, y) \coloneqq$
  $\texttt{if}\ x \geq 0$
  $\texttt{then}\ Q_2(x - y, y)$
  $\texttt{else}\ Q_3(x - y, y)$

$Q_3(x, y) \coloneqq \texttt{return}\ x$

Unification Modulo Nonnested Recursion Schemes

- Restriction to nonnested schemes: $\cancel{P(Q(\ldots))}$
- Unification: Find substitution of variables such that $P(s_1, \ldots, s_n) \equiv Q(t_1, \ldots, t_m)$
- Tree equivalence problem was known to be decidable [Sabelfeld00,Courcelle78]
- We reduce the unification problem to anchored semi-unification

Semi-Unification

- Semi-unification is undecidable
- We reduce to a new decidable fragment: anchored semi-unification
- We skip the reduction for this talk

Ordinary Unification

- Terms: $s, t ::= x \mid a \mid s \cdot t$
- Substitutions σ, τ substitute variables by terms
- Given $s \doteq t$,
  find a substitution σ
  such that $\sigma s = \sigma t$.
- We call σ a unifier of $s \doteq t$
- Example $x \cdot a \doteq b \cdot y$
  Unifier $\sigma = \{x \mapsto b, y \mapsto a\}$

Semi-Unification

- Terms: $s, t ::= x \mid a \mid s \cdot t \mid \alpha x$
- An instance variable αx is a variant of x
- This must be respected by substitutions:
$$\sigma(\alpha x) = \sigma\big(\hat{\alpha}(\sigma x)\big)$$
where $\hat{\alpha} s$ is s with every variable y being replaced by αy
- Example:
$\sigma \coloneqq \{x \mapsto a \cdot y, \alpha x \mapsto a \cdot z, \alpha y = z\}$ is ok
$\sigma \coloneqq \{x \mapsto a \cdot y, \alpha x \mapsto y\}$ is forbidden

Semi-Unification Example

We search for a solution σ

- $\alpha x \doteq a \cdot y, \qquad \alpha z \doteq a \cdot (a \cdot x),\ \alpha y \doteq a \cdot x,\ \alpha x \doteq z$

Semi-Unification Example

$$\alpha x \Longrightarrow z \text{ since } \sigma(\alpha x) = \sigma z$$

- $\alpha x \doteq a \cdot y, \qquad \alpha z \doteq a \cdot (a \cdot x),\ \alpha y \doteq a \cdot x,\ \alpha x \doteq z$
- $z \doteq a \cdot y, \qquad \alpha z \doteq a \cdot (a \cdot x),\ \alpha y \doteq a \cdot x,\ \alpha x \doteq z$

Semi-Unification Example

$$z \Longrightarrow a \cdot y \text{ since } \sigma z = \sigma(a \cdot y)$$
$$\alpha z \Longrightarrow \hat{\alpha}(a \cdot y) \text{ since } \sigma(\alpha z) = \sigma(\hat{\alpha}(\sigma z))$$

- $\alpha x \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \; \alpha y \doteq a \cdot x, \; \alpha x \doteq z$
- $z \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \; \alpha y \doteq a \cdot x, \; \alpha x \doteq z$
- $z \doteq a \cdot y, \; \hat{\alpha}(a \cdot y) \doteq a \cdot (a \cdot x), \; \alpha y \doteq a \cdot x, \; \alpha x \doteq a \cdot y$

Semi-Unification Example

By definition of $\hat{\alpha}$

- $\alpha x \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \ \hat{\alpha}(a \cdot y) \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \cdot \alpha y \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$

Semi-Unification Example

Splitting the equation

- $\alpha x \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \ \hat{\alpha}(a \cdot y) \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \cdot \alpha y \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \doteq a, \ \alpha y \doteq a \cdot x, \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$

Semi-Unification Example

Removing trivial equations

- $\alpha x \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \ \hat{\alpha}(a \cdot y) \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \cdot \alpha y \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \doteq a, \ \alpha y \doteq a \cdot x, \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$

Semi-Unification Example

Now we can construct a solution

- $\alpha x \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \quad \alpha z \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq z$
- $z \doteq a \cdot y, \ \hat{\alpha}(a \cdot y) \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \cdot \alpha y \doteq a \cdot (a \cdot x), \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad a \doteq a, \ \alpha y \doteq a \cdot x, \ \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$
- $z \doteq a \cdot y, \quad \alpha y \doteq a \cdot x, \ \alpha x \doteq a \cdot y$

$$\sigma \coloneqq \{z \mapsto a \cdot y, \alpha y \mapsto a \cdot x, \alpha x \mapsto a \cdot y, \alpha z \mapsto a \cdot (a \cdot x)\}$$

Semi-Unification Rules

- $E, \ \alpha x \doteq s \ \Longrightarrow \ E[s \,/\, \alpha x], \ \alpha x \doteq s$
  if s instance-free
- $E, \ x \doteq s \ \Longrightarrow \ E[s/x], \ x \doteq s$
  if s instance-free
- $E, \ s_1 \cdot s_2 \doteq t_1 \cdot t_2 \ \Longrightarrow \ E, \ s_1 \doteq t_1, \ s_2 \doteq t_2$
- $E, \ s \doteq s \ \Longrightarrow \ E$

Semi-Unification Rules

- But semi-unification is undecidable!
- The rules always terminate but …
- … they can get stuck.
- Example:

$$\alpha x \doteq a, x \doteq \alpha y$$

Substituting x would produce $\alpha(\alpha y)$.
We forbid this to ensure termination.

- $E,\ \alpha x \doteq \boldsymbol{s} \implies E[s/\alpha x],\ \alpha x \doteq \boldsymbol{s}$
  if $\boldsymbol{s}$ instance-free
- $E,\ x \doteq \boldsymbol{s} \implies E[s/x],\ x \doteq \boldsymbol{s}$
  if $\boldsymbol{s}$ instance-free
- $E,\ s_1 \cdot s_2 \doteq t_1 \cdot t_2 \implies E,\ s_1 \doteq t_1,\ s_2 \doteq t_2$
- $E,\ s \doteq s \implies E$

Anchored Semi-Unification

- Anchoredness: Invariant to ensure progress
- Intuition: Whenever we see an instance variable, we can replace it by an instance-free term.
- Definition: There is a partial equivalence relation $\sim$ on variables such that
  - If $\alpha x \in (\sim)$, then $\alpha x \doteq s$ with s instance-free
  - If $s[x] \doteq t[y]$, then $x \sim y$
  - If $s[\alpha x] \doteq t[\beta y]$, then $\alpha x \sim \beta y$
  - If $x \sim y$ and $\alpha x \in (\sim)$, then $\alpha x \sim \alpha y$

Complexity

- The rules I presented need exponential time
- There is an algorithm with time $O(n^3\alpha(n))$
- Idea from Unification-Closure [Huet78]: Don' t perform substitutions but just record equivalences in a union-find structure.

References

- **B. Courcelle**. A representation of trees by languages II. Theoretical Computer Science, 7(1):25-55, 1978.
- **V. Sabelfeld**. The tree equivalence of linear recursion schemes. Theoretical Computer Science, 238(1-2):1-29, 2000.
- **G. P. Huet**, Résolution d'Equations dans des Langages d'ordre 1,2,...,ω, Thèse d`État, Université de Paris VII, 1978
- **F. Baader and W. Snyder**. Unification theory. In Handbook of Automated Reasoning, volume 1, pages 445-534. Elsevier, 2001.