

# Theoretical Properties

- Structural Operational Semantics
- Correctness of Live Variables Analysis

# The Semantics

A *state* is a mapping from variables to integers:

$$\sigma \in \text{State} = \text{Var} \rightarrow \mathbf{Z}$$

The semantics of arithmetic and boolean expressions

$$\mathcal{A} : \text{AExp} \rightarrow (\text{State} \rightarrow \mathbf{Z}) \quad (\text{no errors allowed})$$

$$\mathcal{B} : \text{BExp} \rightarrow (\text{State} \rightarrow \mathbf{T}) \quad (\text{no errors allowed})$$

The *transitions* of the semantics are of the form

$$\langle S, \sigma \rangle \rightarrow \sigma' \quad \text{and} \quad \langle S, \sigma \rangle \rightarrow \langle S', \sigma' \rangle$$

## Transitions

$$\langle [x := a]^\ell, \sigma \rangle \rightarrow \sigma[x \mapsto \mathcal{A}[\![a]\!]\sigma]$$

$$\langle [\text{skip}]^\ell, \sigma \rangle \rightarrow \sigma$$

$$\frac{\langle S_1, \sigma \rangle \rightarrow \langle S'_1, \sigma' \rangle}{\langle S_1; S_2, \sigma \rangle \rightarrow \langle S'_1; S_2, \sigma' \rangle}$$

$$\frac{\langle S_1, \sigma \rangle \rightarrow \sigma'}{\langle S_1; S_2, \sigma \rangle \rightarrow \langle S_2, \sigma' \rangle}$$

$$\langle \text{if } [b]^\ell \text{ then } S_1 \text{ else } S_2, \sigma \rangle \rightarrow \langle S_1, \sigma \rangle \quad \text{if } \mathcal{B}[\![b]\!]\sigma = \text{true}$$

$$\langle \text{if } [b]^\ell \text{ then } S_1 \text{ else } S_2, \sigma \rangle \rightarrow \langle S_2, \sigma \rangle \quad \text{if } \mathcal{B}[\![b]\!]\sigma = \text{false}$$

$$\langle \text{while } [b]^\ell \text{ do } S, \sigma \rangle \rightarrow \langle (S; \text{while } [b]^\ell \text{ do } S), \sigma \rangle \quad \text{if } \mathcal{B}[\![b]\!]\sigma = \text{true}$$

$$\langle \text{while } [b]^\ell \text{ do } S, \sigma \rangle \rightarrow \sigma \quad \text{if } \mathcal{B}[\![b]\!]\sigma = \text{false}$$

## Example:

```
<[y:=x]1; [z:=1]2; while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ300>
→ <[z:=1]2; while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ330>
→ <while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ331>
→ <[z:=z*y]4; [y:=y-1]5;
    while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ331>
→ <[y:=y-1]5; while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ333>
→ <while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ323>
→ <[z:=z*y]4; [y:=y-1]5;
    while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ323>
→ <[y:=y-1]5; while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ326>
→ <while [y>1]3 do ([z:=z*y]4; [y:=y-1]5); [y:=0]6, σ316>
→ <[y:=0]6, σ316>
→ σ306
```

# Equations and Constraints

Equation system  $\text{LV}^=(S_*)$ :

$$\text{LV}_{\text{exit}}(\ell) = \begin{cases} \emptyset & \text{if } \ell \in \text{final}(S_*) \\ \cup\{\text{LV}_{\text{entry}}(\ell') \mid (\ell', \ell) \in \text{flow}^R(S_*)\} & \text{otherwise} \end{cases}$$

$$\text{LV}_{\text{entry}}(\ell) = (\text{LV}_{\text{exit}}(\ell) \setminus \text{kill}_{\text{LV}}(B^\ell)) \cup \text{gen}_{\text{LV}}(B^\ell)$$

where  $B^\ell \in \text{blocks}(S_*)$

Constraint system  $\text{LV}^{\subseteq}(S_*)$ :

$$\text{LV}_{\text{exit}}(\ell) \supseteq \begin{cases} \emptyset & \text{if } \ell \in \text{final}(S_*) \\ \cup\{\text{LV}_{\text{entry}}(\ell') \mid (\ell', \ell) \in \text{flow}^R(S_*)\} & \text{otherwise} \end{cases}$$

$$\text{LV}_{\text{entry}}(\ell) \supseteq (\text{LV}_{\text{exit}}(\ell) \setminus \text{kill}_{\text{LV}}(B^\ell)) \cup \text{gen}_{\text{LV}}(B^\ell)$$

where  $B^\ell \in \text{blocks}(S_*)$

## Lemma

Each solution to the equation system  $\text{LV}^=(S_*)$  is also a solution to the constraint system  $\text{LV}^\subseteq(S_*)$ .

**Proof:** Trivial.

## Lemma

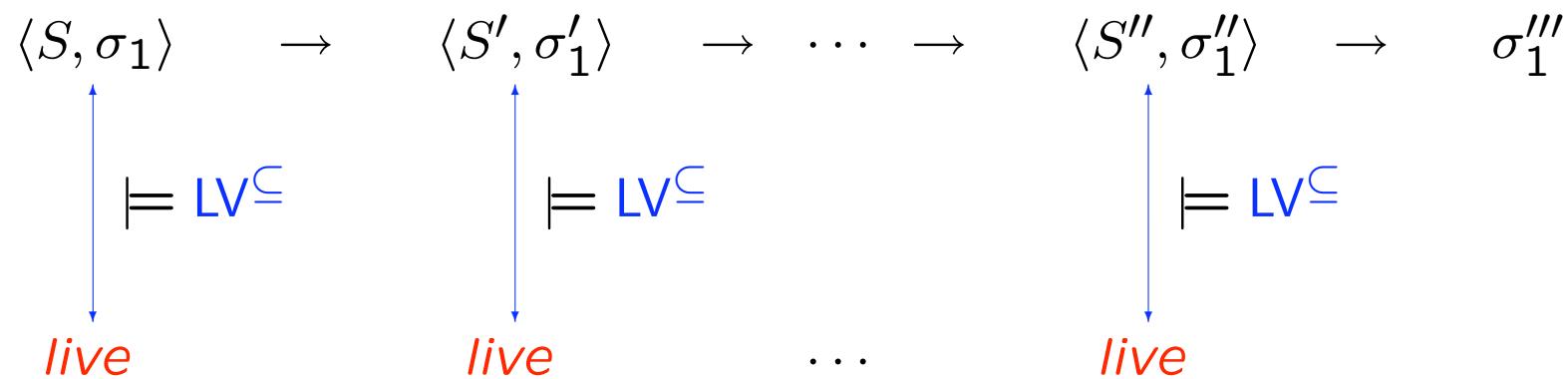
The **least** solution to the equation system  $\text{LV}^=(S_*)$  is also the **least** solution to the constraint system  $\text{LV}^\subseteq(S_*)$ .

**Proof:** Use Tarski's Theorem.

**Naive Proof:** Proceed by contradiction. Suppose some LHS is strictly greater than the RHS. Replace the LHS by the RHS in the solution. Argue that you still have a solution. This establishes the desired contradiction.

## Lemma

A solution *live* to the constraint system is preserved during computation



Proof: requires a lot of machinery — see the book.

## Correctness Relation

$$\sigma_1 \sim_V \sigma_2$$

means that for all practical purposes the two states  $\sigma_1$  and  $\sigma_2$  are equal: only the values of the live variables of  $V$  matters and here the two states are equal.

### Example:

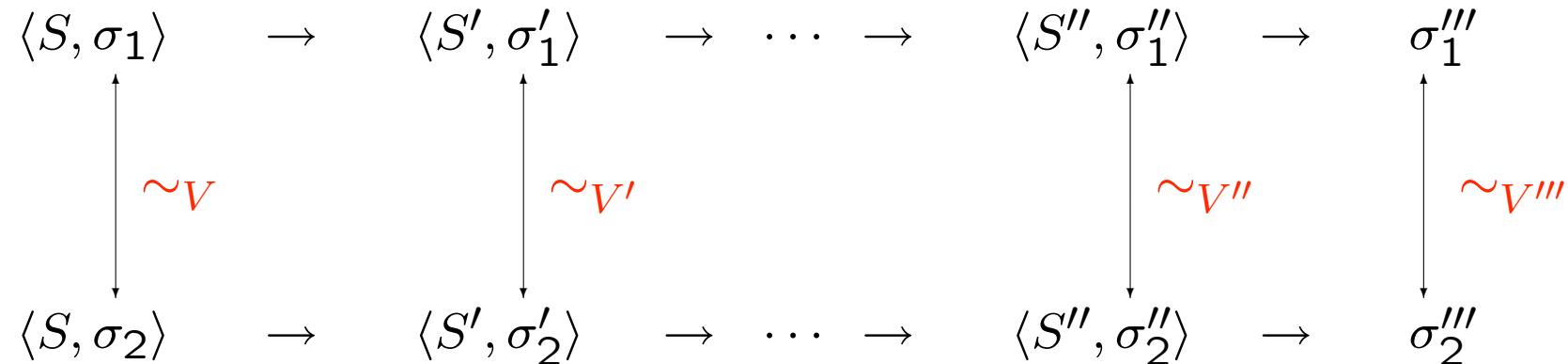
Consider the statement  $[x := y + z]^\ell$

Let  $V_1 = \{y, z\}$ . Then  $\sigma_1 \sim_{V_1} \sigma_2$  means  $\sigma_1(y) = \sigma_2(y) \wedge \sigma_1(z) = \sigma_2(z)$

Let  $V_2 = \{x\}$ . Then  $\sigma_1 \sim_{V_2} \sigma_2$  means  $\sigma_1(x) = \sigma_2(x)$

# Correctness Theorem

The relation “ $\sim$ ” is *invariant* under computation: the live variables for the initial configuration remain live throughout the computation.



$$V = \text{live}_{\text{entry}}(\text{init}(S))$$

$$V'' = \text{live}_{\text{entry}}(\text{init}(S''))$$

$$V' = \text{live}_{\text{entry}}(\text{init}(S'))$$

$$V''' = \text{live}_{\text{exit}}(\text{init}(S''))$$

$$= \text{live}_{\text{exit}}(\ell)$$

for some  $\ell \in \text{final}(S)$