

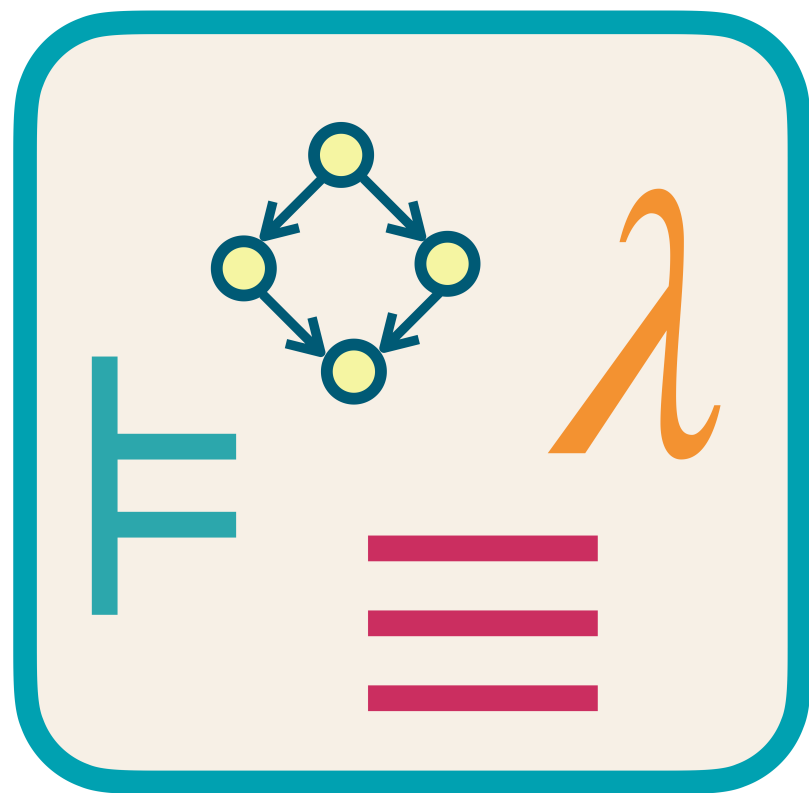
MPP 2025/26 (0077A, 9CFU)

Models for Programming Paradigms

Roberto Bruni

Filippo Bonchi

<http://www.di.unipi.it/~bruni/>



<https://didawiki.di.unipi.it/doku.php/magistraleinformatica/mpp/start>

05b - More Induction

Mathematical induction exercises

Exercise

$$\forall n > 0. P(n)$$

$$P(n) \triangleq n^n \geq n!$$

Exercise

$$a_0 \triangleq 0 \quad a_{n+1} \triangleq 2a_n + n$$

$$\forall n \in \mathbb{N}. P(n) \quad P(n) \triangleq a_n = 2^n - n - 1$$

Exercise: Fibonacci

$$F_1 \triangleq 1 \quad F_2 \triangleq 1 \quad F_{n+2} \triangleq F_n + F_{n+1}$$

$$\forall n > 0. P(n)$$

$$P(n) \triangleq \sum_{i=1}^n F_i = F_{n+2} - 1$$

Determinacy by structural induction

Termination

A result can always be returned

Determinacy

Any two results are the same

Termination vs determinacy

	termination?	determinacy?
$x + y$		
x^y		
$rand()$		
$\pm sqrt(x)$		

Determinacy of arithmetic expressions

$a ::= x \mid n \mid a \text{ op } a$

$x \in \text{Ide} \quad \text{op} \in \{+, \times, -\}$

$n \in \mathbb{Z} \quad \mathbb{M} \triangleq \{\sigma \mid \sigma : \text{Ide} \rightarrow \mathbb{Z}\}$

$$\frac{}{\langle x, \sigma \rangle \longrightarrow \sigma(x)} \quad \frac{}{\langle n, \sigma \rangle \longrightarrow n} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow n_0 \text{ op } n_1}$$

$$P(a) \triangleq \forall \sigma \in \mathbb{M}. \forall m, m' \in \mathbb{Z}. \langle a, \sigma \rangle \longrightarrow m \wedge \langle a, \sigma \rangle \longrightarrow m' \Rightarrow m = m'$$

$$\forall a. P(a) ?$$

Structural induction principle

$$\forall x \in \text{Ide. } P(x)$$

$$\forall n \in \mathbb{Z}. P(n)$$

$$\forall a_0, a_1. P(a_0) \wedge P(a_1) \Rightarrow P(a_0 \text{ op } a_1)$$

$$\forall a. P(a)$$

Base case

$\forall x \in \text{Ide. } P(x)$

Take a generic $x \in \text{Ide}$

We want to prove

$$P(x) \triangleq \forall \sigma, m, m'. \langle x, \sigma \rangle \longrightarrow m \wedge \langle x, \sigma \rangle \longrightarrow m' \Rightarrow m = m'$$

Take generic σ, m, m' s.t. $\langle x, \sigma \rangle \longrightarrow m$ and $\langle x, \sigma \rangle \longrightarrow m'$

We want to prove $m = m'$

Consider the goal $\langle x, \sigma \rangle \longrightarrow m$

Only the rule $\frac{}{\langle x, \sigma \rangle \longrightarrow \sigma(x)}$ is applicable, hence $m = \sigma(x)$

Similarly, since $\langle x, \sigma \rangle \longrightarrow m'$ it must be $m' = \sigma(x)$

and thus we conclude $m = m'$

Base case

$\forall n \in \mathbb{Z}. P(n)$

Take a generic $n \in \mathbb{Z}$

We want to prove

$$P(n) \triangleq \forall \sigma, m, m'. \langle n, \sigma \rangle \longrightarrow m \wedge \langle n, \sigma \rangle \longrightarrow m' \Rightarrow m = m'$$

Take generic σ, m, m' s.t. $\langle n, \sigma \rangle \longrightarrow m$ and $\langle n, \sigma \rangle \longrightarrow m'$

We want to prove $m = m'$

Consider the goal $\langle n, \sigma \rangle \longrightarrow m$

Only the rule $\frac{}{\langle n, \sigma \rangle \longrightarrow n}$ is applicable, hence $m = n$

Similarly, since $\langle n, \sigma \rangle \longrightarrow m'$ it must be $m' = n$

and thus we conclude $m = m'$

Inductive case

$\forall a_0, a_1. P(a_0) \wedge P(a_1) \Rightarrow P(a_0 \text{ op } a_1)$ Take generic a_0, a_1

We assume (inductive hypotheses)

$$P(a_i) \triangleq \forall \sigma, m_i, m'_i. \langle a_i, \sigma \rangle \longrightarrow m_i \wedge \langle a_i, \sigma \rangle \longrightarrow m'_i \Rightarrow m_i = m'_i$$

We want to prove

$$P(a_0 \text{ op } a_1) \triangleq \forall \sigma, m, m'. \langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow m \wedge \langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow m' \Rightarrow m = m'$$

Take generic σ, m, m' such that $\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow m$ and $\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow m'$

We want to prove $m = m'$

Inductive case (ctd)

Consider the goal $\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow m$

Only the rule $\frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow n_0 \text{ op } n_1}$ is applicable

hence $m = n_0 \text{ op } n_1$ with $\langle a_0, \sigma \rangle \longrightarrow n_0$ and $\langle a_1, \sigma \rangle \longrightarrow n_1$

Similarly, since $\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow m'$

it must be $m' = n'_0 \text{ op } n'_1$ with $\langle a_0, \sigma \rangle \longrightarrow n'_0$ and $\langle a_1, \sigma \rangle \longrightarrow n'_1$

By inductive hypotheses, $n_0 = n'_0$ and $n_1 = n'_1$

and thus we conclude $m = n_0 \text{ op } n_1 = n'_0 \text{ op } n'_1 = m'$

Many sorted signatures

Different types of expressions

$$e ::= x \mid n \mid e + e \mid e \leq e \mid e \& e$$

$$((x + 1) \leq (y - 1)) \& (x \leq 42)$$

$$(((x + 1) \leq y) - 1) \& x \leq 42$$

$$x + (1 \leq (y - (1 \& (x \leq 42))))$$

Many sorted signatures

$a ::= x \mid n \mid a + a$ arithmetic expressions

$b ::= a \leq a \mid b \& b$ Boolean expressions

Sorts: identifiers x

numerals n

arithmetic expressions a

Boolean expressions b

Operators:

- $+$ • takes two arithmetic expressions and returns an arithmetic expression
- $\leq$ • takes two arithmetic expressions and returns a Boolean expression
- $\&$ • takes two Boolean expressions and returns a Boolean expression

Terms over a signature

$S = \{s, \dots\}$ a set of **sorts**

$\Sigma = \{\Sigma_{s_1 \dots s_n, s}\}_{s_1, \dots, s_n, s \in S}$ a **many sorted signature**

$$f \in \Sigma_{s_1 \dots s_n, s} \qquad f : (s_1 \times \dots \times s_n) \rightarrow s$$

$T_{\Sigma, s}$ denotes the set of all **terms of sort s**

it is the least set such that:

- if $c \in \Sigma_{\epsilon, s}$, then $c \in T_{\Sigma, s}$
- if $f \in \Sigma_{s_1 \dots s_n, s}$ and $\forall i. t_i \in T_{\Sigma, s_i}$, then $f(t_1, \dots, t_n) \in T_{\Sigma, s}$

$T_{\Sigma} = \{T_{\Sigma, s}\}_{s \in S}$ denotes the set of all **well-sorted terms**

Boolean expressions

$$x \in \text{Ide} \quad n \in \mathbb{Z} \quad \text{op} \in \{+, \times, -\}$$

$$v \in \mathbb{B} \quad \text{bop} \in \{\wedge, \vee\} \quad \text{cmp} \in \{<, \leq, >, \geq, =, \neq\}$$

$$a ::= x \mid n \mid a \text{ op } a$$

$$b ::= v \mid a \text{ cmp } a \mid \neg b \mid b \text{ bop } b$$

$$S \triangleq \{\text{Aexp}, \text{Bexp}\}$$

$$\Sigma_{\epsilon, \text{Aexp}} \triangleq \text{Ide} \cup \mathbb{Z}$$

$$\Sigma_{\text{Aexp} \text{Aexp}, \text{Aexp}} \triangleq \{+, \times, -\}$$

$$\Sigma_{\epsilon, \text{Bexp}} \triangleq \mathbb{B}$$

$$\Sigma_{\text{Aexp} \text{Aexp}, \text{Bexp}} \triangleq \{<, \leq, >, \geq, =, \neq\}$$

$$\Sigma_{\text{Bexp}, \text{Bexp}} \triangleq \{\neg\}$$

$$\Sigma_{\text{Bexp} \text{Bexp}, \text{Bexp}} \triangleq \{\wedge, \vee\}$$

Semantics of expressions

$a ::= x \mid n \mid a \text{ op } a$

$b ::= v \mid a \text{ cmp } a \mid \neg b \mid b \text{ bop } b$

$$\frac{}{\langle x, \sigma \rangle \longrightarrow \sigma(x)} \quad \frac{}{\langle n, \sigma \rangle \longrightarrow n} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow n_0 \text{ op } n_1}$$

$$\frac{}{\langle v, \sigma \rangle \longrightarrow v} \quad \frac{\langle b, \sigma \rangle \longrightarrow v}{\langle \neg b, \sigma \rangle \longrightarrow \neg v} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow n_0 \text{ cmp } n_1}$$

$$\frac{\langle b_0, \sigma \rangle \longrightarrow v_0 \quad \langle b_1, \sigma \rangle \longrightarrow v_1}{\langle b_0 \text{ bop } b_1, \sigma \rangle \longrightarrow v_0 \text{ bop } v_1}$$

Subterms

$$t_i \prec f(t_1, \dots, t_n)$$

sometimes, an additional requirement:
sort-preserving relation

$$f \in \Sigma_{s_1 \dots s_n, s} \wedge s_i = s$$

One sorted = Unsorted

a special case:

$S = \{*\}$ a singleton set of sorts

$$\Sigma_{\underbrace{*\dots*}_n,*} \simeq \Sigma_n$$

Termination of Boolean expressions

Termination of expressions

$$a ::= x \mid n \mid a \text{ op } a$$

$$b ::= v \mid a \text{ cmp } a \mid \neg b \mid b \text{ bop } b$$

$$\frac{}{\langle x, \sigma \rangle \longrightarrow \sigma(x)} \quad \frac{}{\langle n, \sigma \rangle \longrightarrow n} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow n_0 \text{ op } n_1}$$

$$\frac{}{\langle v, \sigma \rangle \longrightarrow v} \quad \frac{\langle b, \sigma \rangle \longrightarrow v}{\langle \neg b, \sigma \rangle \longrightarrow \neg v} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow n_0 \text{ cmp } n_1}$$

$$\frac{\langle b_0, \sigma \rangle \longrightarrow v_0 \quad \langle b_1, \sigma \rangle \longrightarrow v_1}{\langle b_0 \text{ bop } b_1, \sigma \rangle \longrightarrow v_0 \text{ bop } v_1}$$

$$P(b) \triangleq \forall \sigma \in \mathbb{M}. \exists v \in \mathbb{B}. \langle b, \sigma \rangle \longrightarrow v$$

$$\forall b. P(b) ?$$

Base case

$\forall v \in \mathbb{B}. P(v)$

Take a generic $v \in \mathbb{B}$

We want to prove $P(v) \triangleq \forall \sigma. \exists u. \langle v, \sigma \rangle \longrightarrow u$

Take a generic $\sigma \in \mathbb{M}$ and consider the goal $\langle v, \sigma \rangle \longrightarrow u$

the only
variable

By rule $\frac{}{\langle v, \sigma \rangle \longrightarrow v}$

we have $\langle v, \sigma \rangle \longrightarrow u \nearrow_{[u=v]} \square$

And we are done (taking $u = v$)

A surprising base case

$\forall a_0, a_1. P(a_0 \text{ cmp } a_1)$ Take generic a_0, a_1

We want to prove $P(a_0 \text{ cmp } a_1) \triangleq \forall \sigma. \exists v. \langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow v$

Consider the goal $\langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow v$

By rule $\frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow n_0 \text{ cmp } n_1}$ we have

$\langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow v \nwarrow_{[v=n_0 \text{ cmp } n_1]} \langle a_0, \sigma \rangle \longrightarrow n_0, \langle a_1, \sigma \rangle \longrightarrow n_1$

By **termination of arithmetic expressions**, such n_0, n_1 exist

And we are done (taking $v = n_0 \text{ cmp } n_1$)

Proof obligations



try to complete on your own
the missing inductive cases of the proof

$$\forall b. P(b) \Rightarrow P(\neg b)$$

$$\forall b_0, b_1. (P(b_0) \wedge P(b_1)) \Rightarrow P(b_0 \text{ bop } b_1)$$

Commands

Commands

$a ::= x \mid n \mid a + a \mid \dots$

$b ::= v \mid a \leq a \mid \dots$

$c ::= \mathbf{skip} \mid x := a \mid c; c \mid \mathbf{if } b \mathbf{ then } c \mathbf{ else } c \mid \mathbf{while } b \mathbf{ do } c$

$S \triangleq \{\text{Aexp}, \text{Bexp}, \text{Com}\}$

$\Sigma_{\epsilon, \text{Aexp}} \triangleq \text{Ide} \cup \mathbb{Z}$

$\Sigma_{\text{AexpAexp}, \text{Aexp}} \triangleq \{+, \times, -\}$

$\Sigma_{\epsilon, \text{Bexp}} \triangleq \mathbb{B}$

$\Sigma_{\text{AexpAexp}, \text{Bexp}} \triangleq \{<, \leq, >, \geq, =, \neq\}$

$\Sigma_{\text{Bexp}, \text{Bexp}} \triangleq \{\neg\}$

$\Sigma_{\text{BexpBexp}, \text{Bexp}} \triangleq \{\wedge, \vee\}$

$\Sigma_{\epsilon, \text{Com}} \triangleq \{\mathbf{skip}\}$

$\Sigma_{\text{Aexp}, \text{Com}} \triangleq \{x := \mid x \in \text{Ide}\}$

$\Sigma_{\text{ComCom}, \text{Com}} \triangleq \{;\}$

$\Sigma_{\text{BexpComCom}, \text{Com}} \triangleq \{\mathbf{if}\}$

$\Sigma_{\text{BexpCom}, \text{Com}} \triangleq \{\mathbf{while}\}$

Big-step operational semantics of commands

$\langle a, \sigma \rangle \longrightarrow n$ (the evaluation of a in σ returns the integer n)

$\langle b, \sigma \rangle \longrightarrow v$ (the evaluation of b in σ returns the Boolean v)

$\langle c, \sigma \rangle \longrightarrow \sigma'$ (the evaluation of c in σ returns the memory σ')

Memories

$$\mathbb{M} \triangleq \{ \sigma : \text{Ide} \rightarrow \mathbb{Z} \mid \sigma \text{ has finite support} \}$$

$$\{x \in \text{Ide} \mid \sigma(x) \neq 0\} \text{ is finite}$$

$$(n_1/x_1, \dots, n_k/x_k) : \text{Ide} \rightarrow \mathbb{Z}$$

all different

$$(n_1/x_1, \dots, n_k/x_k)(x) \triangleq \begin{cases} n_i & \text{if } x = x_i \\ 0 & \text{otherwise} \end{cases}$$

$\sigma_0 \triangleq ()$ is the typical initial memory

Memory updates

$$\sigma[n/y](x) \triangleq \begin{cases} n & \text{if } x = y \\ \sigma(x) & \text{otherwise} \end{cases}$$

$$\forall \sigma, m, n, y. \sigma[m/y][n/y] = \sigma[n/y]$$

$$\sigma[m/y][n/y](x) \triangleq \begin{cases} n & \text{if } x = y \\ \sigma[m/y](x) = \sigma(x) & \text{otherwise} \end{cases}$$

$$\forall \sigma, m, n, y, z. \ y \neq z \Rightarrow \sigma[n/y][m/z] = \sigma[m/z][n/y]$$

we write $\sigma[n/y, m/z]$ in such cases

$$(n_1/x_1, \dots, n_k/x_k) = \sigma_0[n_1/x_1, \dots, n_k/x_k]$$

Semantics of commands

$c ::= \text{skip} \mid x := a \mid c; c \mid \text{if } b \text{ then } c \text{ else } c \mid \text{while } b \text{ do } c$

$$\frac{}{\langle \text{skip}, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle a, \sigma \rangle \longrightarrow n}{\langle x := a, \sigma \rangle \longrightarrow \sigma[n/x]} \quad \frac{\langle c_0, \sigma \rangle \longrightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \longrightarrow \sigma'}{\langle c_0; c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff} \quad \langle c_1, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c_0, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff}}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \text{while } b \text{ do } c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma'}$$

Multiple rules for the same construct!

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff} \quad \langle c_1, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c_0, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff}}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \text{while } b \text{ do } c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{}{\langle x, \sigma \rangle \longrightarrow \sigma(x)} \quad \frac{}{\langle n, \sigma \rangle \longrightarrow n} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ op } a_1, \sigma \rangle \longrightarrow n_0 \text{ op } n_1} \quad \frac{}{\langle \text{skip}, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle a, \sigma \rangle \longrightarrow n}{\langle x := a, \sigma \rangle \longrightarrow \sigma[n/x]} \quad \frac{\langle c_0, \sigma \rangle \longrightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \longrightarrow \sigma'}{\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{}{\langle v, \sigma \rangle \longrightarrow v} \quad \frac{\langle b, \sigma \rangle \longrightarrow v}{\langle \neg b, \sigma \rangle \longrightarrow \neg v} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \text{ cmp } a_1, \sigma \rangle \longrightarrow n_0 \text{ cmp } n_1}$$

$$\frac{\langle b_0, \sigma \rangle \longrightarrow v_0 \quad \langle b_1, \sigma \rangle \longrightarrow v_1}{\langle b_0 \text{ bop } b_1, \sigma \rangle \longrightarrow v_0 \text{ bop } v_1}$$

Recursive definition!



one premise is as complex
as the conclusion

$$\frac{\langle b, \sigma \rangle \longrightarrow \mathbf{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{}{\langle x, \sigma \rangle \longrightarrow \sigma(x)} \quad \frac{}{\langle n, \sigma \rangle \longrightarrow n} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \ \mathbf{op} \ a_1, \sigma \rangle \longrightarrow n_0 \ \mathbf{op} \ n_1}$$

$$\frac{}{\langle \mathbf{skip}, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle a, \sigma \rangle \longrightarrow n}{\langle x := a, \sigma \rangle \longrightarrow \sigma[n/x]} \quad \frac{\langle c_0, \sigma \rangle \longrightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \longrightarrow \sigma'}{\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{}{\langle v, \sigma \rangle \longrightarrow v} \quad \frac{\langle b, \sigma \rangle \longrightarrow v}{\langle \neg b, \sigma \rangle \longrightarrow \neg v} \quad \frac{\langle a_0, \sigma \rangle \longrightarrow n_0 \quad \langle a_1, \sigma \rangle \longrightarrow n_1}{\langle a_0 \ \mathbf{cmp} \ a_1, \sigma \rangle \longrightarrow n_0 \ \mathbf{cmp} \ n_1}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \mathbf{ff} \quad \langle c_1, \sigma \rangle \longrightarrow \sigma'}{\langle \mathbf{if} \ b \ \mathbf{then} \ c_0 \ \mathbf{else} \ c_1, \sigma \rangle \longrightarrow \sigma'} \quad \frac{\langle b, \sigma \rangle \longrightarrow \mathbf{tt} \quad \langle c_0, \sigma \rangle \longrightarrow \sigma'}{\langle \mathbf{if} \ b \ \mathbf{then} \ c_0 \ \mathbf{else} \ c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b_0, \sigma \rangle \longrightarrow v_0 \quad \langle b_1, \sigma \rangle \longrightarrow v_1}{\langle b_0 \ \mathbf{bop} \ b_1, \sigma \rangle \longrightarrow v_0 \ \mathbf{bop} \ v_1}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \mathbf{ff}}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \longrightarrow \sigma}$$

Triangular numbers

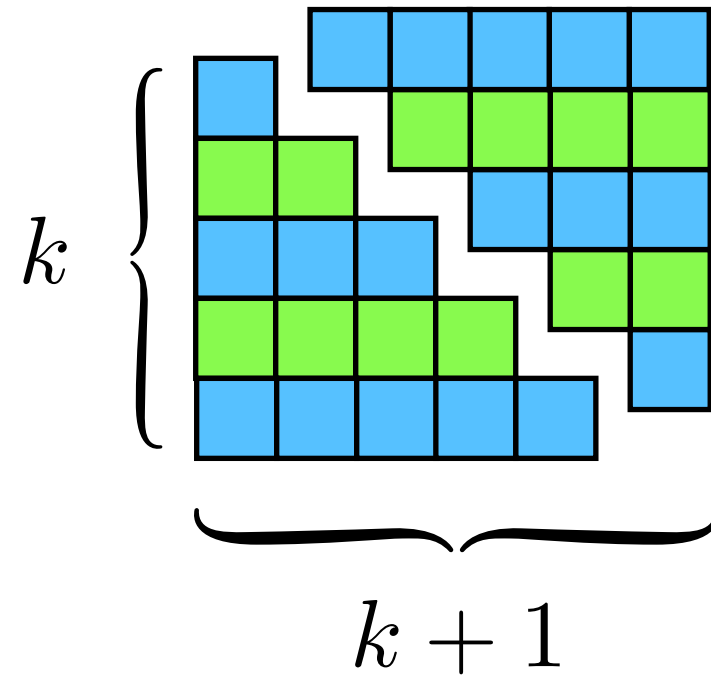
$$T_1 \quad \begin{array}{|c|} \hline \square \\ \hline \end{array} \quad 1$$

$$T_2 \quad \begin{array}{|c|c|} \hline \square & \\ \hline \square & \square \\ \hline \end{array} \quad 3$$

$$T_3 \quad \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array} \quad 6$$

$$T_4 \quad \begin{array}{|c|c|c|c|} \hline \square & & & \\ \hline \square & \square & & \\ \hline \square & \square & \square & \\ \hline \square & \square & \square & \square \\ \hline \end{array} \quad 10$$

$$T_5 \quad \begin{array}{|c|c|c|c|c|} \hline \square & & & & \\ \hline \square & \square & & & \\ \hline \square & \square & \square & & \\ \hline \square & \square & \square & \square & \\ \hline \square & \square & \square & \square & \square \\ \hline \end{array} \quad 15$$



$$T_k \triangleq \sum_{i=1}^k i = \frac{k(k+1)}{2}$$

Compute T_k without div

$$\begin{array}{l}
 c_0 \{ \\
 c_1 \{ \\
 w \{
 \end{array}
 \left\{
 \begin{array}{l}
 t := 0; \\
 i := 1; \overbrace{\quad b \quad} \\
 \mathbf{while} \ i \leq k \ \mathbf{do} \ (\\
 \qquad t := t + i; \\
 \qquad i := i + 1) \\
 \end{array}
 \right\}
 \begin{array}{l}
 c_2 \\
 c_3
 \end{array}
 \left. \vphantom{\begin{array}{l} c_0 \\ c_1 \\ w \end{array}} \right\} c$$

initial memory

let's find σ such that $\langle (c_0; c_1); w, (2/k) \rangle \longrightarrow \sigma$

let's find σ', σ such that $\left\{ \begin{array}{l} \langle c_0; c_1, (2/k) \rangle \longrightarrow \sigma' \\ \langle w, \sigma' \rangle \longrightarrow \sigma \end{array} \right.$

Let's derive

let's find σ' such that $\langle c_0; c_1, (2/k) \rangle \longrightarrow \sigma'$

$\langle c_0; c_1, (2/k) \rangle \longrightarrow \sigma'$

$\nwarrow \langle t := 0, (2/k) \rangle \longrightarrow \sigma_1, \langle i := 1, \sigma_1 \rangle \longrightarrow \sigma'$

$\nwarrow_{\sigma_1 = (2/k)[n_1/t]} \langle 0, (2/k) \rangle \longrightarrow n_1, \langle i := 1, (2/k, n_1/t) \rangle \longrightarrow \sigma'$

$\nwarrow_{n_1=0} \langle i := 1, (2/k, 0/t) \rangle \longrightarrow \sigma'$

$\nwarrow_{\sigma' = (2/k, 0/t)[n_2/i]} \langle 1, (2/k, 0/t) \rangle \longrightarrow n_2$

$\nwarrow_{n_2=1} \square$

$\sigma' = (2/k, 0/t, 1/i)$

more concisely: $\langle c_0; c_1, (2/k) \rangle \longrightarrow \sigma'$

$\nwarrow^*_{\sigma' = (2/k, 0/t, 1/i)} \square$

$c_0 \{$
 $c_1 \{$
 $w \{$
 $t := 0;$
 $i := 1;$
 $\text{while } i \leq k \text{ do } ($
 $t := t + i;$
 $i := i + 1)$
 $\} c_2 \}$
 $\} c_3 \}$
 c

$\sigma \triangleq (2/n)$

Let's derive

let's find σ such that $\langle w, (2/k, 0/t, 1/i) \rangle \longrightarrow \sigma$

$\langle w, (2/k, 0/t, 1/i) \rangle \longrightarrow \sigma$

$\nwarrow \langle b, (2/k, 0/t, 1/i) \rangle \longrightarrow \mathbf{tt}, \langle c, (2/k, 0/t, 1/i) \rangle \longrightarrow \sigma_1, \langle w, \sigma_1 \rangle \longrightarrow \sigma$

$\nwarrow^* \langle c, (2/k, 0/t, 1/i) \rangle \longrightarrow \sigma_1, \langle w, \sigma_1 \rangle \longrightarrow \sigma$

$\nwarrow_{\sigma_1 = (2/k, 0/t, 1/i)[1/t, 2/i]}^* \langle w, (2/k, 1/t, 2/i) \rangle \longrightarrow \sigma$

$\nwarrow \langle b, (2/k, 1/t, 2/i) \rangle \longrightarrow \mathbf{tt}, \langle c, (2/k, 1/t, 2/i) \rangle \longrightarrow \sigma_2, \langle w, \sigma_2 \rangle \longrightarrow \sigma$

$\nwarrow^* \langle c, (2/k, 1/t, 2/i) \rangle \longrightarrow \sigma_2, \langle w, \sigma_2 \rangle \longrightarrow \sigma$

$\nwarrow_{\sigma_2 = (2/k, 1/t, 2/i)[3/t, 3/i]}^* \langle w, (2/k, 3/t, 3/i) \rangle \longrightarrow \sigma$

$\nwarrow_{\sigma = (2/k, 3/t, 3/i)} \langle b, (2/k, 3/t, 3/i) \rangle \longrightarrow \mathbf{ff}$

$\nwarrow^* \square$

$c_0 \{$
 $c_1 \{$
 $w \{$
 $t := 0;$
 $i := 1;$
 $\text{while } i \leq k \text{ do } ($
 $t := t + i;$
 $i := i + 1)$
 $\} c_2 \}$
 $\} c_3 \}$
 c

$\sigma \triangleq (2/n)$

$\sigma = (2/k, 3/t, 3/i)$

Divergence

Termination of commands?

$c ::= \text{skip} \mid x := a \mid c; c \mid \text{if } b \text{ then } c \text{ else } c \mid \text{while } b \text{ do } c$

$$\frac{}{\langle \text{skip}, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle a, \sigma \rangle \longrightarrow n}{\langle x := a, \sigma \rangle \longrightarrow \sigma[n/x]} \quad \frac{\langle c_0, \sigma \rangle \longrightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \longrightarrow \sigma'}{\langle c_0; c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff} \quad \langle c_1, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c_0, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff}}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \text{while } b \text{ do } c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma'}$$

$$P(c) \triangleq \forall \sigma \in \mathbb{M}. \exists \sigma' \in \mathbb{M}. \langle c, \sigma \rangle \longrightarrow \sigma'$$

$$\forall c. P(c) ?$$

Termination?

$$P(c) \triangleq \forall \sigma. \exists \sigma'. \langle c, \sigma \rangle \longrightarrow \sigma' \qquad \forall c. P(c) ?$$

take $w \triangleq$ while tt do skip

$$\langle w, \sigma \rangle \longrightarrow \sigma'$$

$$\swarrow \langle \mathbf{tt}, \sigma \rangle \longrightarrow \mathbf{tt}, \langle \mathbf{skip}, \sigma \rangle \longrightarrow \sigma'', \langle w, \sigma'' \rangle \longrightarrow \sigma'$$

$$\swarrow \langle \mathbf{skip}, \sigma \rangle \longrightarrow \sigma'', \langle w, \sigma'' \rangle \longrightarrow \sigma'$$

$$\swarrow_{\sigma''=\sigma} \langle w, \sigma \rangle \longrightarrow \sigma'$$

same goal from which we started!
no other options: $\langle \mathbf{tt}, \sigma \rangle \not\rightarrow \mathbf{ff}$
no way to complete the derivation!

$$\neg P(w)$$

we have found a counterexample to termination

Divergence

$\langle c, \sigma \rangle \not\rightarrow$ means $\neg \exists \sigma'. \langle c, \sigma \rangle \longrightarrow \sigma'$

and we say that c diverges on σ

sometimes divergence is very difficult to detect
(well... it is undecidable)

Divergence

take $w \triangleq \mathbf{while} \ x > 0 \ \mathbf{do} \ x := x + 1$

take a generic σ

if $\sigma(x) \leq 0$: $\langle w, \sigma \rangle \longrightarrow \sigma' \quad \nwarrow_{\sigma'=\sigma} \langle x > 0, \sigma \rangle \longrightarrow \mathbf{ff} \quad \nwarrow^* \square$

hence $\langle w, \sigma \rangle \longrightarrow \sigma$

if $\sigma(x) > 0$: $\langle w, \sigma \rangle \longrightarrow \sigma'$

$\nwarrow \langle x > 0, \sigma \rangle \longrightarrow \mathbf{tt}, \langle x := x + 1, \sigma \rangle \longrightarrow \sigma'', \langle w, \sigma'' \rangle \longrightarrow \sigma'$

$\nwarrow^* \langle x := x + 1, \sigma \rangle \longrightarrow \sigma'', \langle w, \sigma'' \rangle \longrightarrow \sigma'$

$\nwarrow_{\sigma''=\sigma[n/x]} \langle x + 1, \sigma \rangle \longrightarrow n, \langle w, \sigma[n/x] \rangle \longrightarrow \sigma'$

$\nwarrow_{n=\sigma(x)+1}^* \langle w, \sigma[\sigma(x) + 1/x] \rangle \longrightarrow \sigma'$

not the same goal from which we started!
how can we prove divergence? $\sigma(x) > 0 \Rightarrow \sigma[\sigma(x) + 1/x](x) = \sigma(x) + 1 > 0$

Proving divergence (if possible)

consider $w \triangleq \text{while } b \text{ do } c$

suppose we find a set of memories $S \subseteq \mathbb{M}$ such that

- $\forall \sigma \in S. \langle b, \sigma \rangle \longrightarrow \text{tt}$
- $\forall \sigma \in S. \forall \sigma' \in \mathbb{M}. \langle c, \sigma \rangle \longrightarrow \sigma' \Rightarrow \sigma' \in S$

then we can conclude $\forall \sigma \in S. \langle w, \sigma \rangle \not\rightarrow$

note that if $\langle c, \sigma \rangle \not\rightarrow$, then $\langle c, \sigma \rangle \longrightarrow \sigma' \Rightarrow \sigma' \in S$ trivially holds

Revisiting the example

take $w \triangleq \mathbf{while} \ x > 0 \ \mathbf{do} \ x := x + 1$

take a generic σ

if $\sigma(x) \leq 0$: $\langle w, \sigma \rangle \longrightarrow \sigma$

let $S \triangleq \{\sigma \mid \sigma(x) > 0\}$

• $\forall \sigma \in S. \langle x > 0, \sigma \rangle \longrightarrow \mathbf{tt}$ ✓

• $\forall \sigma \in S. \forall \sigma'. \langle x := x + 1, \sigma \rangle \longrightarrow \sigma' \Rightarrow \sigma' \in S$ ✓

in fact $\langle x := x + 1, \sigma \rangle \longrightarrow \sigma' \Rightarrow \sigma' = \sigma[\sigma(x) + 1/x]$

$$\sigma(x) > 0 \Rightarrow \sigma[\sigma(x) + 1/x](x) = \sigma(x) + 1 > 0$$

therefore, if $\sigma(x) > 0$, then $\langle w, \sigma \rangle \not\rightarrow$



Exercise

take $w \triangleq$ **while** $x \neq 0$ **do** $x := x - 2$

find the set of all and only memories σ such that $\langle w, \sigma \rangle \not\rightarrow$

$$S_1 \triangleq \{\sigma \mid \sigma(x) < 0\}$$

$$S_2 \triangleq \{\sigma \mid \exists k \in \mathbb{Z}. \sigma(x) = 2k + 1\}$$

- $\forall \sigma \in S. \langle x \neq 0, \sigma \rangle \longrightarrow \mathbf{tt}$
- $\forall \sigma \in S. \forall \sigma'. \langle x := x - 2, \sigma \rangle \longrightarrow \sigma' \Rightarrow \sigma' \in S$
 $\langle x := x - 2, \sigma \rangle \longrightarrow \sigma' \Rightarrow \sigma' = \sigma[\sigma(x) - 2/x]$

Collatz's conjecture: half or triple plus one

$w \triangleq \text{while } x > 1 \text{ do } (\text{ if } x \% 2 = 0 \text{ then } x := x/2$
 $\text{ else } x := (3 \times x) + 1)$

$\forall \sigma. \sigma(x) \leq 1 \Rightarrow \langle w, \sigma \rangle \longrightarrow \sigma$ 

open conjecture: $\forall \sigma. \exists \sigma'. \langle w, \sigma \rangle \longrightarrow \sigma' \equiv \neg(\exists \sigma. \langle w, \sigma \rangle \not\rightarrow)$

more precisely: $\forall \sigma. \sigma(x) > 1 \Rightarrow \langle w, \sigma \rangle \longrightarrow \sigma[1/x]$

Experimental evidence:

as of 2020, the conjecture has been verified for all values up to 2^{68} .

Barina, David. "Convergence verification of the Collatz problem".

The Journal of Supercomputing (2020). doi:10.1007/s11227-020-03368-x

Determinacy?

Determinacy of commands?

$c ::= \text{skip} \mid x := a \mid c; c \mid \text{if } b \text{ then } c \text{ else } c \mid \text{while } b \text{ do } c$

$$\frac{}{\langle \text{skip}, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle a, \sigma \rangle \longrightarrow n}{\langle x := a, \sigma \rangle \longrightarrow \sigma[n/x]} \quad \frac{\langle c_0, \sigma \rangle \longrightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \longrightarrow \sigma'}{\langle c_0; c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff} \quad \langle c_1, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c_0, \sigma \rangle \longrightarrow \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow \text{ff}}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma} \quad \frac{\langle b, \sigma \rangle \longrightarrow \text{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \text{while } b \text{ do } c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma'}$$

$$P(c) \triangleq \forall \sigma, \sigma_1, \sigma_2. \langle c, \sigma \rangle \longrightarrow \sigma_1 \wedge \langle c, \sigma \rangle \longrightarrow \sigma_2 \Rightarrow \sigma_1 = \sigma_2 \quad \forall c. P(c) ?$$

Structural induction principle

$$\forall x, a. P(x := a) \qquad P(\text{skip})$$

$$\forall c_0, c_1. P(c_0) \wedge P(c_1) \Rightarrow P(c_0 ; c_1)$$

$$\forall b, c_0, c_1. P(c_0) \wedge P(c_1) \Rightarrow P(\text{if } b \text{ then } c_0 \text{ else } c_1)$$

$$\forall b, c. P(c) \Rightarrow P(\text{while } b \text{ do } c)$$

$$\forall c \in \text{Com}. P(c)$$

Base case

$\forall x, a. P(x := a)$

Take generic $x \in \text{Ide}, a \in \text{Aexp}$

We want to prove

$$P(x := a) \triangleq \forall \sigma, \sigma_1, \sigma_2. \langle x := a, \sigma \rangle \longrightarrow \sigma_1 \wedge \langle x := a, \sigma \rangle \longrightarrow \sigma_2 \Rightarrow \sigma_1 = \sigma_2$$

Take generic $\sigma, \sigma_1, \sigma_2$ s.t. $\langle x := a, \sigma \rangle \longrightarrow \sigma_1$ and $\langle x := a, \sigma \rangle \longrightarrow \sigma_2$

We want to prove $\sigma_1 = \sigma_2$

Consider the goal $\langle x := a, \sigma \rangle \longrightarrow \sigma_1$

Only the rule $\frac{\langle a, \sigma \rangle \longrightarrow n}{\langle x := a, \sigma \rangle \longrightarrow \sigma[n/x]}$ is applicable, hence $\sigma_1 = \sigma[n/x]$
with $\langle a, \sigma \rangle \longrightarrow n$

Similarly, since $\langle x := a, \sigma \rangle \longrightarrow \sigma_2$ it must be $\sigma_2 = \sigma[m/x]$
with $\langle a, \sigma \rangle \longrightarrow m$

by determinacy of Aexp we have $n = m$ and thus $\sigma_1 = \sigma_2$

Inductive case

$$\forall c_0, c_1. P(c_0) \wedge P(c_1) \Rightarrow P(c_0 ; c_1)$$

Take generic c_0, c_1

We assume

(inductive hypotheses)

$$P(c_i) \triangleq \forall \sigma, \sigma_1, \sigma_2. \langle c_i, \sigma \rangle \longrightarrow \sigma_1 \wedge \langle c_i, \sigma \rangle \longrightarrow \sigma_2 \Rightarrow \sigma_1 = \sigma_2$$

We want to prove

$$P(c_0 ; c_1) \triangleq \forall \sigma, \sigma_1, \sigma_2. \langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma_1 \wedge \langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma_2 \Rightarrow \sigma_1 = \sigma_2$$

Take generic $\sigma, \sigma_1, \sigma_2$ such that $\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma_1$ and $\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma_2$

We want to prove $\sigma_1 = \sigma_2$

Inductive case (ctd)

Consider the goal $\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma_1$

Only the rule
$$\frac{\langle c_0, \sigma \rangle \longrightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \longrightarrow \sigma'}{\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma'}$$
 is applicable

hence $\sigma_1 = \sigma'_1$ with $\langle c_0, \sigma \rangle \longrightarrow \sigma''_1$ and $\langle c_1, \sigma''_1 \rangle \longrightarrow \sigma'_1$

Similarly, since $\langle c_0 ; c_1, \sigma \rangle \longrightarrow \sigma_2$

it must be $\sigma_2 = \sigma'_2$ with $\langle c_0, \sigma \rangle \longrightarrow \sigma''_2$ and $\langle c_1, \sigma''_2 \rangle \longrightarrow \sigma'_2$

By inductive hypothesis $P(c_0)$, we have $\sigma''_1 = \sigma''_2$

and thus $\langle c_1, \sigma''_2 \rangle \longrightarrow \sigma'_1$ and $\langle c_1, \sigma''_2 \rangle \longrightarrow \sigma'_2$

By inductive hypothesis $P(c_1)$, we then have $\sigma'_1 = \sigma'_2$

and thus we conclude $\sigma_1 = \sigma'_1 = \sigma'_2 = \sigma_2$

Inductive case

$\forall b, c. P(c) \Rightarrow P(\text{while } b \text{ do } c)$ Take generic b, c

We assume (inductive hypothesis)

$$P(c) \triangleq \forall \sigma, \sigma_1, \sigma_2. \langle c, \sigma \rangle \longrightarrow \sigma_1 \wedge \langle c, \sigma \rangle \longrightarrow \sigma_2 \Rightarrow \sigma_1 = \sigma_2$$

We want to prove

$$P(\text{while } b \text{ do } c) \triangleq \forall \sigma, \sigma_1, \sigma_2. \langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma_1 \wedge \langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma_2 \Rightarrow \sigma_1 = \sigma_2$$

Take $\sigma, \sigma_1, \sigma_2$ such that $\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma_1$ and $\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma_2$

We want to prove $\sigma_1 = \sigma_2$

By determinacy of boolean expressions, there are two cases

$$\langle b, \sigma \rangle \longrightarrow \text{ff}$$

$$\langle b, \sigma \rangle \longrightarrow \text{tt}$$

Inductive case (ctd)

if $\langle b, \sigma \rangle \longrightarrow \text{ff}$

Consider the goal $\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma_1$

Only the rule $\frac{\langle b, \sigma \rangle \longrightarrow \text{ff}}{\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma}$ is applicable hence $\sigma_1 = \sigma$

Similarly, since $\langle \text{while } b \text{ do } c, \sigma \rangle \longrightarrow \sigma_2$ it must be $\sigma_2 = \sigma$
and thus we conclude $\sigma_1 = \sigma = \sigma_2$

Inductive case (ctd)

if $\langle b, \sigma \rangle \longrightarrow \mathbf{tt}$

Consider the goal $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \longrightarrow \sigma_1$

Only the rule
$$\frac{\langle b, \sigma \rangle \longrightarrow \mathbf{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \longrightarrow \sigma'}$$
 is applicable

hence $\sigma_1 = \sigma'_1$ with $\langle c, \sigma \rangle \longrightarrow \sigma''_1$ and $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma''_1 \rangle \longrightarrow \sigma'_1$

Similarly, since $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \longrightarrow \sigma_2$

it must be $\sigma_2 = \sigma'_2$ with $\langle c, \sigma \rangle \longrightarrow \sigma''_2$ and $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma''_2 \rangle \longrightarrow \sigma'_2$

By inductive hypothesis $P(c)$, we have $\sigma''_1 = \sigma''_2$

thus $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma''_2 \rangle \longrightarrow \sigma'_1$ and $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma''_2 \rangle \longrightarrow \sigma'_2$

but there is no inductive hypothesis $P(\mathbf{while} \ b \ \mathbf{do} \ c)$!

Recursive definition!



one premise is as much complex as the conclusion

$$\frac{\langle b, \sigma \rangle \longrightarrow \mathbf{tt} \quad \langle c, \sigma \rangle \longrightarrow \sigma'' \quad \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma'' \rangle \longrightarrow \sigma'}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \longrightarrow \sigma'}$$

to close the proof of determinacy
we need a more convenient induction principle:

Rule induction