

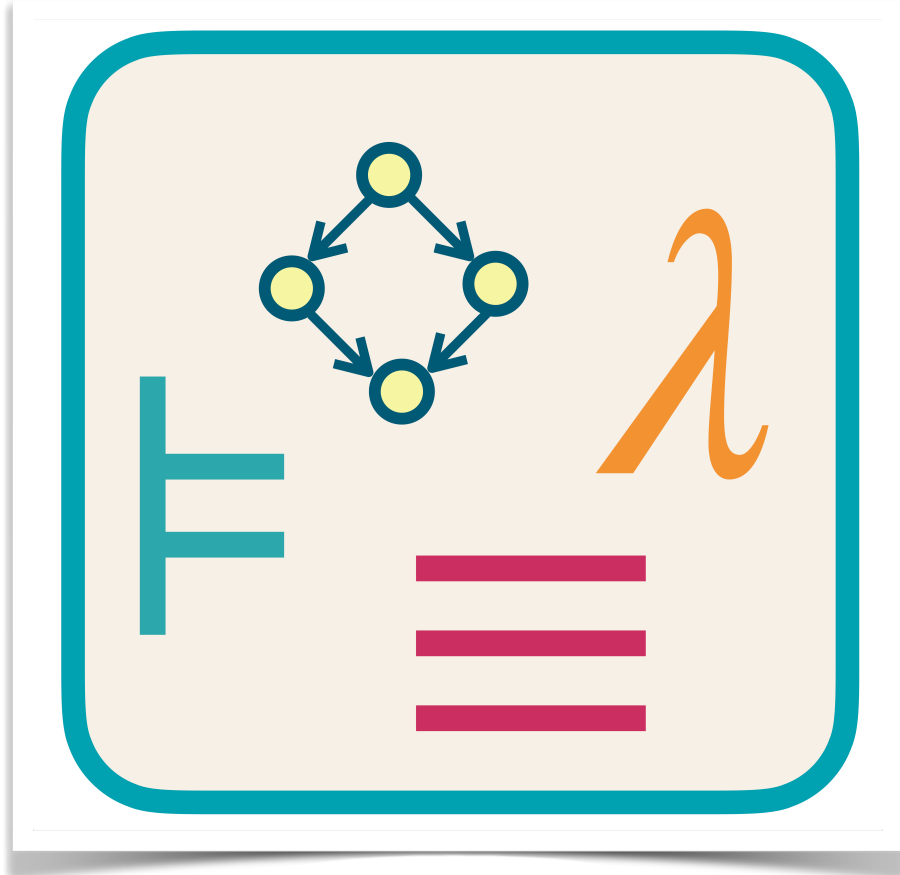
LPP 2026/27 (063AA, 9CFU)

Programming Languages

Fabio Gadducci

and Laboratory

Vincenzo Ciancia



07 - Recursion

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. f\ x : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. f\ x : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

$$\forall x \in X. \forall y \in Y. f\ x\ y \in Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. f\ x : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

$$\forall x \in X. \forall y \in Y. f\ x\ y \in Z$$

$$g : (X \rightarrow Y) \rightarrow Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. f\ x : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

$$\forall x \in X. \forall y \in Y. f\ x\ y \in Z$$

$$g : (X \rightarrow Y) \rightarrow Z$$

$$\forall h \in (X \rightarrow Y). g(h) \in Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. f\ x : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

$$\forall x \in X. \forall y \in Y. f\ x\ y \in Z$$

$$g : (X \rightarrow Y) \rightarrow Z$$

$$g\ x$$

$$\forall h \in (X \rightarrow Y). g(h) \in Z$$

Notation

$$f : X \rightarrow (Y \rightarrow Z)$$

$$f : X \rightarrow Y \rightarrow Z$$

$$\forall x \in X. f(x) : Y \rightarrow Z$$

$$\forall x \in X. f\ x : Y \rightarrow Z$$

$$\forall x \in X. \forall y \in Y. (f(x))(y) \in Z$$

$$\forall x \in X. \forall y \in Y. f\ x\ y \in Z$$

$$g : (X \rightarrow Y) \rightarrow Z$$

$$g\ x$$

$$\forall h \in (X \rightarrow Y). g(h) \in Z$$

$$f\ h$$

Notation

$$f : X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_n$$

$$f : X_1 \rightarrow (X_2 \rightarrow (\cdots \rightarrow X_n))$$

$$f \ x_1 \ x_2 \ \cdots \ x_n$$

$$(((f \ x_1) \ x_2) \ \cdots) \ x_n$$

Notation

$$f : X_1 \times X_2 \rightarrow Y$$

$$f : (X_1 \times X_2) \rightarrow Y$$

$$\forall x_1 \in X_1. \forall x_2 \in X_2. f(x_1, x_2) \in Y$$

Notation

$$f : X_1 \times X_2 \rightarrow Y$$

$$f : (X_1 \times X_2) \rightarrow Y$$

$$\forall x_1 \in X_1. \forall x_2 \in X_2. f(x_1, x_2) \in Y$$

$$f \ x_1$$

Notation

$$f : X \rightarrow Y$$

$$A \subseteq X$$

Notation

$$f : X \rightarrow Y$$

$$A \subseteq X$$

$$f|_A : A \rightarrow Y$$

Notation

$$f : X \rightarrow Y$$

$$A \subseteq X$$

$$f|_A : A \rightarrow Y$$

$$\forall a \in A. f|_A(a) \stackrel{\Delta}{=} f(a)$$

Predecessors

$$A, \prec$$

$$a \in A$$

$$[a] \triangleq \{x \in A \mid x \prec a\}$$

Well-founded recursion

Recursive definitions

$$\mathcal{A}[\![\cdot]\!] : \text{Aexp} \rightarrow \mathbb{M} \rightarrow \mathbb{Z}$$

$\mathcal{A}[\![a]\!]\sigma$ denotes the value associated to a in σ

$$\mathcal{A}[\![n]\!]\sigma \triangleq n$$

$$\mathcal{A}[\![x]\!]\sigma \triangleq \sigma(x)$$

$$\mathcal{A}[\![a_0 \text{ op } a_1]\!]\sigma \triangleq \mathcal{A}[\![a_0]\!]\sigma \text{ op } \mathcal{A}[\![a_1]\!]\sigma$$

The function is defined recursively:
how do we know one and exactly one value is associated
to each expression? (true)

Recursive definitions

$$N ::= 0 \mid s(N)$$

$$\mathcal{N}[\cdot] : \text{Nexp} \rightarrow \mathbb{N}$$

$$\begin{aligned}\mathcal{N}[0] &\triangleq 0 \\ \mathcal{N}[s(N)] &\triangleq 1 + \mathcal{N}[s(s(N))]\end{aligned}$$

The function is defined recursively:
how do we know one and exactly one value is associated
to each expression? (false)

Well founded recursion

$A, \prec$ w.f.

Well founded recursion

$$A, \prec \text{ w.f.}$$

$$F \triangleq \{F_a : ([a] \rightarrow B) \rightarrow B\}_{a \in A}$$

Well founded recursion

$$A, \prec \text{ w.f.}$$

$$F \triangleq \{F_a : ([a] \rightarrow B) \rightarrow B\}_{a \in A}$$

$$\forall a \in A. \forall h \in [a] \rightarrow B. F_a(h) \in B$$

Well founded recursion

$$A, \prec \text{ w.f.}$$

$$F \triangleq \{F_a : ([a] \rightarrow B) \rightarrow B\}_{a \in A}$$

$$\forall a \in A. \forall h \in [a] \rightarrow B. F_a(h) \in B$$

TH. There exists a unique function $f : A \rightarrow B$ such that

$$\forall a \in A. f(a) = F_a(f|_{[a]})$$

Example

$\mathbb{N}, <$

$$F \triangleq \{F_n : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

Example

$$\mathbb{N}, <$$

$$F \triangleq \{F_n : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0 : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0 \ h \triangleq 1$$

Example

$$\mathbb{N}, <$$

$$F \triangleq \{F_n : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0 : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0 \ h \triangleq 1$$

$$F_{n+1} : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1} \ h \triangleq (n + 1) \cdot h(n)$$

Example

$$\mathbb{N}, <$$

$$F \triangleq \{F_n : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0 : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0 \ h \triangleq 1$$

$$F_{n+1} : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1} \ h \triangleq (n+1) \cdot h(n)$$

$$f(0) = F_0 f|_{\emptyset} = 1$$

$$f(n+1) = F_{n+1} f|_{\{n\}} = (n+1) \cdot f(n)$$

Example

$$\mathbb{N}, <$$

$$F \triangleq \{F_n : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0 : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0 \ h \triangleq 1$$

$$F_{n+1} : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1} \ h \triangleq (n+1) \cdot h(n)$$

$$f(0) = F_0 f|_{\emptyset} = 1$$

$$f(n+1) = F_{n+1} f|_{\{n\}} = (n+1) \cdot f(n)$$

$$f(n) = n!$$

Example

$\mathbb{N}, <$

$m \in \mathbb{N}$

$$F^m \triangleq \{F_n^m : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

Example

$$\mathbb{N}, <$$

$$m \in \mathbb{N}$$

$$F^m \triangleq \{F_n^m : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0^m : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0^m h \triangleq 0$$

Example

$$\mathbb{N}, <$$

$$m \in \mathbb{N}$$

$$F^m \triangleq \{F_n^m : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0^m : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1}^m : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0^m h \triangleq 0$$

Example

$$\mathbb{N}, <$$

$$m \in \mathbb{N}$$

$$F^m \triangleq \{F_n^m : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0^m : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0^m h \triangleq 0$$

$$F_{n+1}^m : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1}^m h \triangleq m + h(n)$$

Example

$$\mathbb{N}, <$$

$$m \in \mathbb{N}$$

$$F^m \triangleq \{F_n^m : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0^m : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0^m h \triangleq 0$$

$$F_{n+1}^m : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1}^m h \triangleq m + h(n)$$

$$f^m(0) = 0$$

$$f^m(n+1) = m + f^m(n)$$

Example

$$\mathbb{N}, <$$

$$m \in \mathbb{N}$$

$$F^m \triangleq \{F_n^m : ([n] \rightarrow \mathbb{N}) \rightarrow \mathbb{N}\}_{n \in \mathbb{N}}$$

$$F_0^m : (\emptyset \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_{n+1}^m : (\{n\} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$

$$F_0^m h \triangleq 0$$

$$F_{n+1}^m h \triangleq m + h(n)$$

$$f^m(0) = 0$$

$$f^m(n+1) = m + f^m(n)$$

$$f^m(n) = m \cdot n$$

Ackermann function

a computable function that is total but not primitive recursive

Ackermann function

a computable function that is total but not primitive recursive

$$m \in \mathbb{N}$$

Ackermann function

a computable function that is total but not primitive recursive

$$m \in \mathbb{N} \quad \text{ack}_m : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

Ackermann function

a computable function that is total but not primitive recursive

$$m \in \mathbb{N} \quad \text{ack}_m : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$\text{ack}_m(0, 0) \triangleq m$$

$$\text{ack}_m(0, n+1) \triangleq \text{ack}_m(0, n) + 1$$

$$\text{ack}_m(1, 0) \triangleq 0$$

$$\text{ack}_m(k+1, n+1) \triangleq \text{ack}_m(k, \text{ack}_m(k+1, n))$$

$$\text{ack}_m(k+2, 0) \triangleq 1$$

Ackermann function

a computable function that is total but not primitive recursive

$$m \in \mathbb{N} \quad \text{ack}_m : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$\text{ack}_m(0, 0) \stackrel{\triangle}{=} m$$

$$\text{ack}_m(0, n+1) \stackrel{\triangle}{=} \text{ack}_m(0, n) + 1$$

$$\text{ack}_m(1, 0) \stackrel{\triangle}{=} 0$$

$$\text{ack}_m(k+1, n+1) \stackrel{\triangle}{=} \text{ack}_m(k, \text{ack}_m(k+1, n))$$

$$\text{ack}_m(k+2, 0) \stackrel{\triangle}{=} 1$$

$\mathbb{N} \times \mathbb{N}$, $\prec$ lexicographic precedence relation

Ackermann function

a computable function that is total but not primitive recursive

$$m \in \mathbb{N} \quad \text{ack}_m : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$\text{ack}_m(0, 0) \stackrel{\triangle}{=} m$$

$$\text{ack}_m(0, n + 1) \stackrel{\triangle}{=} \text{ack}_m(0, n) + 1$$

$$\text{ack}_m(1, 0) \stackrel{\triangle}{=} 0$$

$$\text{ack}_m(k + 1, n + 1) \stackrel{\triangle}{=} \text{ack}_m(k, \text{ack}_m(k + 1, n))$$

$$\text{ack}_m(k + 2, 0) \stackrel{\triangle}{=} 1$$

$\mathbb{N} \times \mathbb{N}$, $\prec$ lexicographic precedence relation

$$(k, n) \prec (k + 1, n')$$

Ackermann function

a computable function that is total but not primitive recursive

$$m \in \mathbb{N} \quad \text{ack}_m : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$\text{ack}_m(0, 0) \stackrel{\Delta}{=} m$$

$$\text{ack}_m(0, n+1) \stackrel{\Delta}{=} \text{ack}_m(0, n) + 1$$

$$\text{ack}_m(1, 0) \stackrel{\Delta}{=} 0$$

$$\text{ack}_m(k+1, n+1) \stackrel{\Delta}{=} \text{ack}_m(k, \text{ack}_m(k+1, n))$$

$$\text{ack}_m(k+2, 0) \stackrel{\Delta}{=} 1$$

$\mathbb{N} \times \mathbb{N}$, $\prec$ lexicographic precedence relation

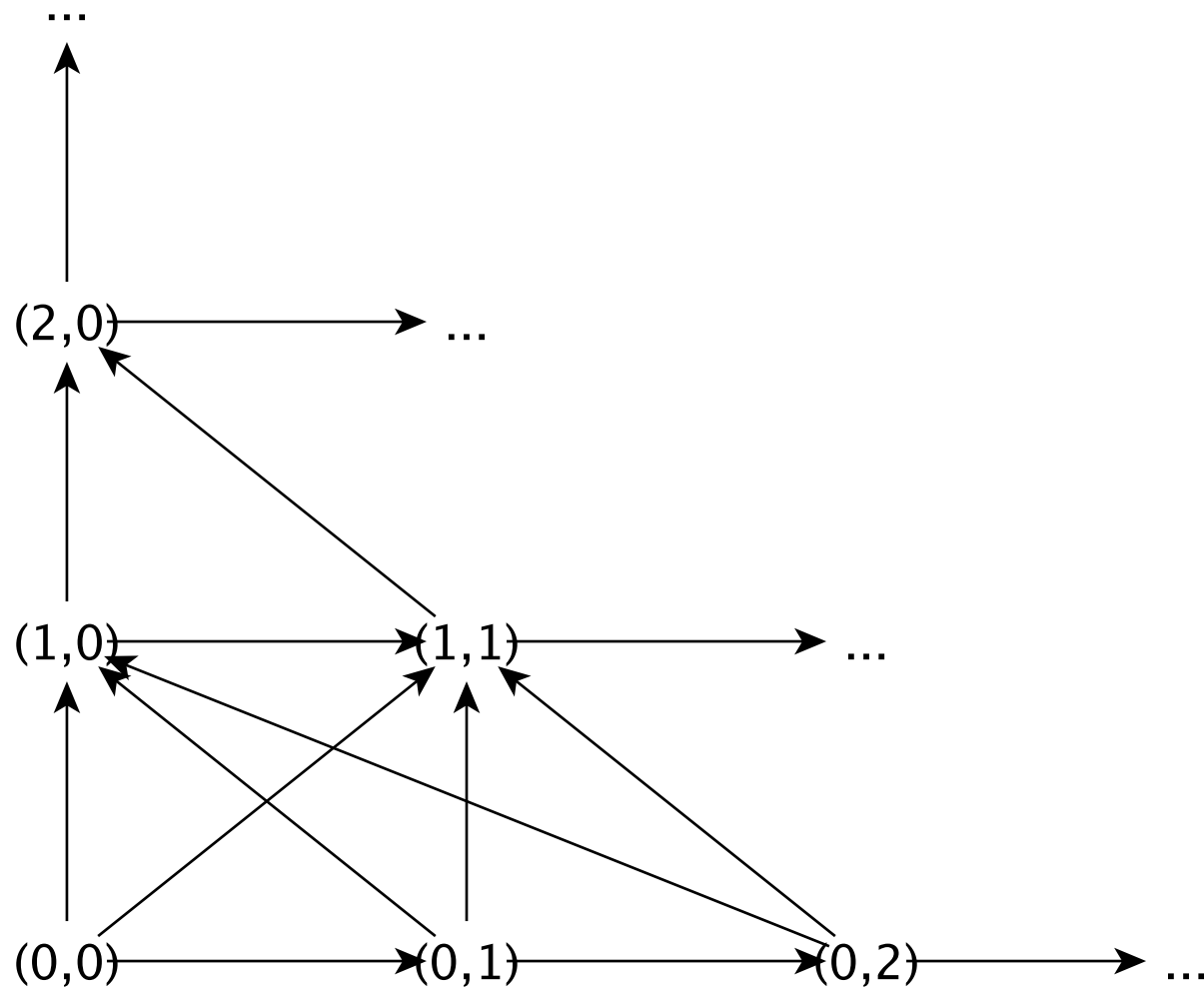
$$(k, n) \prec (k+1, n')$$

$$(k, n) \prec (k, n+1)$$

Ackermann function

$$(k, n) \prec (k + 1, n')$$

$$(k, n) \prec (k, n + 1)$$

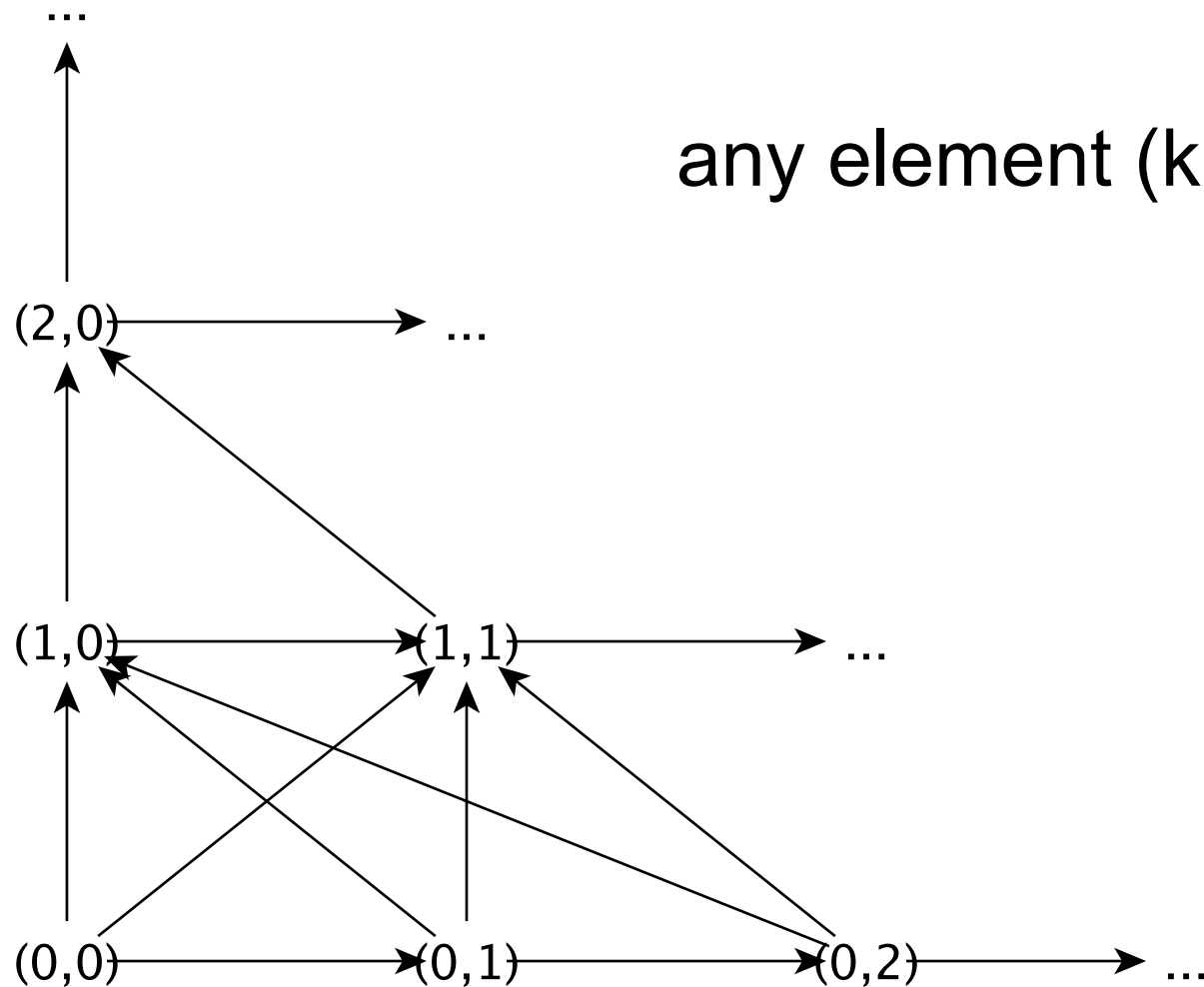


Ackermann function

$$(k, n) \prec (k + 1, n')$$

$$(k, n) \prec (k, n + 1)$$

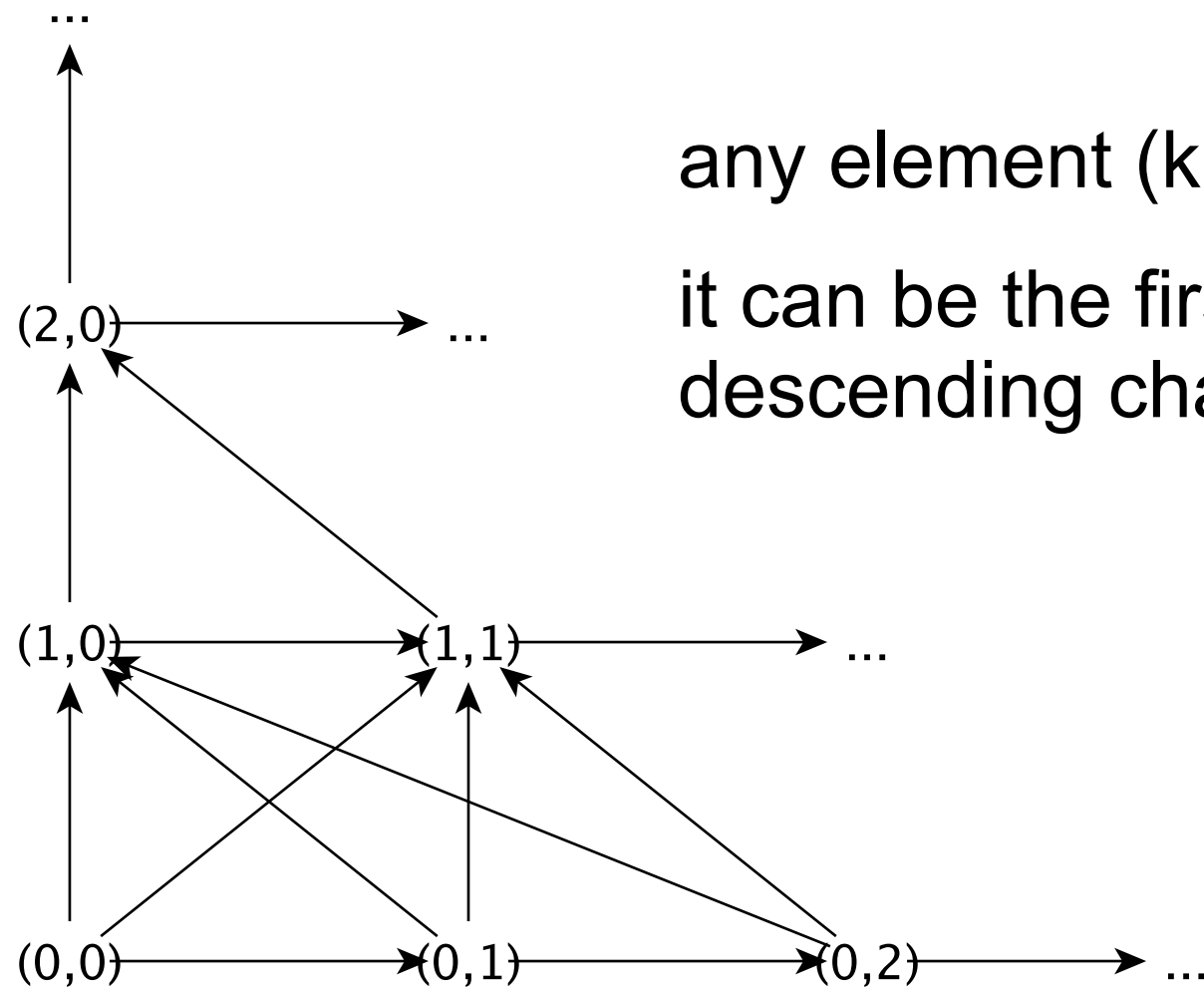
any element $(k+1, n)$ has infinitely many predecessors



Ackermann function

$$(k, n) \prec (k + 1, n')$$

$$(k, n) \prec (k, n + 1)$$



any element $(k+1, n)$ has infinitely many predecessors

it can be the first element of infinitely many descending chains (of unbounded length, but finite)

Ackermann function

$$(k, n) \prec (k + 1, n')$$

$$(k, n) \prec (k, n + 1)$$

$\prec^+$ is w.f.

Ackermann function

$$\begin{aligned}(k, n) &\prec (k + 1, n') \\ (k, n) &\prec (k, n + 1)\end{aligned}\quad \prec^+ \text{ is w.f.}$$

Take a non-empty set $Q \subseteq \mathbb{N} \times \mathbb{N}$

Ackermann function

$$\begin{aligned}(k, n) &\prec (k + 1, n') \\ (k, n) &\prec (k, n + 1)\end{aligned}\quad \prec^+ \text{ is w.f.}$$

Take a non-empty set $Q \subseteq \mathbb{N} \times \mathbb{N}$

can we find m minimal in Q ?

Ackermann function

$$\begin{aligned}(k, n) &\prec (k + 1, n') \\ (k, n) &\prec (k, n + 1)\end{aligned}\quad \prec^+ \text{ is w.f.}$$

Take a non-empty set $Q \subseteq \mathbb{N} \times \mathbb{N}$

can we find m minimal in Q ?

$$\hat{k} \triangleq \min \{k \mid (k, n) \in Q\} \text{ (non-empty because } Q \neq \emptyset)$$

Ackermann function

$$\begin{aligned}(k, n) &\prec (k + 1, n') \\ (k, n) &\prec (k, n + 1)\end{aligned}\quad \prec^+ \text{ is w.f.}$$

Take a non-empty set $Q \subseteq \mathbb{N} \times \mathbb{N}$

can we find m minimal in Q ?

$$\hat{k} \triangleq \min \{k \mid (k, n) \in Q\} \text{ (non-empty because } Q \neq \emptyset)$$

$$\hat{n} \triangleq \min \{n \mid (\hat{k}, n) \in Q\} \text{ (non-empty by def of } \hat{k})$$

Ackermann function

$$\begin{aligned}(k, n) &\prec (k + 1, n') \\ (k, n) &\prec (k, n + 1)\end{aligned}\quad \prec^+ \text{ is w.f.}$$

Take a non-empty set $Q \subseteq \mathbb{N} \times \mathbb{N}$

can we find m minimal in Q ?

$$\hat{k} \triangleq \min \{k \mid (k, n) \in Q\} \text{ (non-empty because } Q \neq \emptyset)$$

$$\hat{n} \triangleq \min \{n \mid (\hat{k}, n) \in Q\} \text{ (non-empty by def of } \hat{k})$$

clearly $(\hat{k}, \hat{n}) \in Q$ is minimal

Ackermann function

$$\begin{aligned}ack_m(0, 0) &\triangleq m \\ack_m(0, n + 1) &\triangleq ack_m(0, n) + 1\end{aligned}$$

Ackermann function

$$\begin{aligned}ack_m(0, 0) &\triangleq m \\ack_m(0, n + 1) &\triangleq ack_m(0, n) + 1\end{aligned}$$

n increments of base case m

Ackermann function

$$ack_m(0, 0) \triangleq m$$

$$ack_m(0, n + 1) \triangleq ack_m(0, n) + 1$$

n increments of base case m

$$ack_m(0, n) \triangleq m + n$$

Ackermann function

$$\begin{aligned}ack_m(1, 0) &\triangleq 0 \\ack_m(k + 1, n + 1) &\triangleq ack_m(k, ack_m(k + 1, n))\end{aligned}$$

Ackermann function

$$\begin{aligned}ack_m(1, 0) &\triangleq 0 \\ack_m(k + 1, n + 1) &\triangleq ack_m(k, ack_m(k + 1, n)) \\ack_m(1, n + 1) &\triangleq ack_m(0, ack_m(1, n))\end{aligned}$$

Ackermann function

$$\begin{aligned}ack_m(1, 0) &\triangleq 0 \\ack_m(k + 1, n + 1) &\triangleq ack_m(k, ack_m(k + 1, n)) \\ack_m(1, n + 1) &\triangleq ack_m(0, ack_m(1, n)) \\ack_m(0, n) &\triangleq m + n\end{aligned}$$

Ackermann function

$$\begin{aligned}ack_m(1, 0) &\triangleq 0 \\ack_m(k + 1, n + 1) &\triangleq ack_m(k, ack_m(k + 1, n)) \\ack_m(1, n + 1) &\triangleq ack_m(0, ack_m(1, n)) \\ack_m(0, n) &\triangleq m + n \\ack_m(1, n + 1) &\triangleq m + ack_m(1, n)\end{aligned}$$

Ackermann function

$$ack_m(1, 0) \triangleq 0$$

$$ack_m(k + 1, n + 1) \triangleq ack_m(k, ack_m(k + 1, n))$$

$$ack_m(1, n + 1) \triangleq ack_m(0, ack_m(1, n))$$

$$ack_m(0, n) \triangleq m + n$$

$$ack_m(1, n + 1) \triangleq m + ack_m(1, n)$$

add m for n times to the base case 0

Ackermann function

$$ack_m(1, 0) \triangleq 0$$

$$ack_m(k + 1, n + 1) \triangleq ack_m(k, ack_m(k + 1, n))$$

$$ack_m(1, n + 1) \triangleq ack_m(0, ack_m(1, n))$$

$$ack_m(0, n) \triangleq m + n$$

$$ack_m(1, n + 1) \triangleq m + ack_m(1, n)$$

add m for n times to the base case 0

$$ack_m(1, n) \triangleq m \cdot n$$

Ackermann function

$$\begin{aligned}ack_m(k+1, n+1) &\triangleq ack_m(k, ack_m(k+1, n)) \\ack_m(k+2, 0) &\triangleq 1\end{aligned}$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(2, 0) \triangleq 1$$

$$ack_m(2, n+1) \triangleq ack_m(1, ack_m(2, n))$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(2, 0) \triangleq 1$$

$$ack_m(2, n+1) \triangleq ack_m(1, ack_m(2, n))$$

$$ack_m(1, n) \triangleq m \cdot n$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(2, 0) \triangleq 1$$

$$ack_m(2, n+1) \triangleq ack_m(1, ack_m(2, n))$$

$$ack_m(1, n) \triangleq m \cdot n$$

$$ack_m(2, n+1) \triangleq m \cdot ack_m(2, n)$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(2, 0) \triangleq 1$$

$$ack_m(2, n+1) \triangleq ack_m(1, ack_m(2, n))$$

$$ack_m(1, n) \triangleq m \cdot n$$

$$ack_m(2, n+1) \triangleq m \cdot ack_m(2, n)$$

multiplies by m for n times the base case 1

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(2, 0) \triangleq 1$$

$$ack_m(2, n+1) \triangleq ack_m(1, ack_m(2, n))$$

$$ack_m(1, n) \triangleq m \cdot n$$

$$ack_m(2, n+1) \triangleq m \cdot ack_m(2, n)$$

multiplies by m for n times the base case 1

$$ack_m(2, n) \triangleq m^n$$

Ackermann function

$$\begin{aligned}ack_m(k+1, n+1) &\triangleq ack_m(k, ack_m(k+1, n)) \\ack_m(k+2, 0) &\triangleq 1\end{aligned}$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(3, 0) \triangleq 1$$

$$ack_m(3, n+1) \triangleq ack_m(2, ack_m(3, n))$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(3, 0) \triangleq 1$$

$$ack_m(3, n+1) \triangleq ack_m(2, ack_m(3, n))$$

$$ack_m(2, n) \triangleq m^n$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(3, 0) \triangleq 1$$

$$ack_m(3, n+1) \triangleq ack_m(2, ack_m(3, n))$$

$$ack_m(2, n) \triangleq m^n$$

$$ack_m(3, n+1) \triangleq m^{ack_m(3, n)}$$

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(3, 0) \triangleq 1$$

$$ack_m(3, n+1) \triangleq ack_m(2, ack_m(3, n))$$

$$ack_m(2, n) \triangleq m^n$$

$$ack_m(3, n+1) \triangleq m^{ack_m(3, n)}$$

n times exponentiation

Ackermann function

$$ack_m(k+1, n+1) \triangleq ack_m(k, ack_m(k+1, n))$$

$$ack_m(k+2, 0) \triangleq 1$$

$$ack_m(3, 0) \triangleq 1$$

$$ack_m(3, n+1) \triangleq ack_m(2, ack_m(3, n))$$

$$ack_m(2, n) \triangleq m^n$$

$$ack_m(3, n+1) \triangleq m^{ack_m(3, n)}$$

n times exponentiation

$$ack_m(3, n) \triangleq m^{m^{m \dots m}}$$

Ackermann function

it grows faster than any primitive recursive function

$$ack_3(0, 3) \triangleq 3 + 3 = 6$$

$$ack_3(1, 3) \triangleq 3 \cdot 3 = 9$$

$$ack_3(2, 3) \triangleq 3^3 = 27$$

$$ack_3(3, 3) \triangleq 3^{3^3} = 3^{27} \simeq 7.6 \cdot 10^{12}$$

Arithmetic expressions

$\text{Aexp}, <$

$a_i < a_0 \text{ op } a_1$

$\mathcal{A}[\cdot] : \text{Aexp} \rightarrow \mathbb{M} \rightarrow \mathbb{Z}$

$\mathcal{A}[n]\sigma \stackrel{\Delta}{=} n$

$\mathcal{A}[x]\sigma \stackrel{\Delta}{=} \sigma(x)$

$\mathcal{A}[a_0 \text{ op } a_1]\sigma \stackrel{\Delta}{=} \mathcal{A}[a_0]\sigma \text{ op } \mathcal{A}[a_1]\sigma$

Boolean expressions

$\text{Bexp}, \prec$

$b_i \prec b_0 \text{ bop } b_1$

$b \prec \neg b$

$\mathcal{B}[\cdot] : \text{Bexp} \rightarrow \mathbb{M} \rightarrow \mathbb{Z}$

$\mathcal{B}[v]\sigma \triangleq v$

$\mathcal{B}[a_0 \text{ cmp } a_1]\sigma \triangleq \mathcal{A}[a_0]\sigma \text{ cmp } \mathcal{A}[a_1]\sigma$

$\mathcal{B}[\neg b]\sigma \triangleq \neg \mathcal{B}[b]\sigma$

$\mathcal{B}[b_0 \text{ bop } b_1]\sigma \triangleq \mathcal{B}[b_0]\sigma \text{ bop } \mathcal{B}[b_1]\sigma$

Recursive definitions

$$\mathcal{C}[\![\cdot]\!] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M}$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

Recursive definitions

for divergence

$$\mathcal{C}[\![\cdot]\!] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\![\text{skip}]\!]\sigma \quad \triangleq$$

$$\mathcal{C}[\![x := a]\!]\sigma \quad \triangleq$$

$$\mathcal{C}[\![c_0; c_1]\!]\sigma \quad \triangleq$$

$$\mathcal{C}[\![\text{if } b \text{ then } c_0 \text{ else } c_1]\!]\sigma \quad \triangleq$$

$$\mathcal{C}[\![\text{while } b \text{ do } c]\!]\sigma \quad \triangleq$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma)$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \stackrel{\Delta}{=} \sigma$$

$$\mathcal{C}[x := a]\sigma \stackrel{\Delta}{=} \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \stackrel{\Delta}{=} \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma) \text{ almost...}$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \stackrel{\Delta}{=}$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \stackrel{\Delta}{=}$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma) \text{ almost...}$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq \begin{cases} \mathcal{C}[c_0]\sigma & \text{if } \mathcal{B}[b]\sigma \\ \mathcal{C}[c_1]\sigma & \text{otherwise} \end{cases}$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma) \text{ almost...}$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq \begin{cases} \mathcal{C}[c_0]\sigma & \text{if } \mathcal{B}[b]\sigma \\ \mathcal{C}[c_1]\sigma & \text{otherwise} \end{cases}$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq \begin{cases} \sigma & \text{if } \neg \mathcal{B}[b]\sigma \\ \mathcal{C}[\text{while } b \text{ do } c](\mathcal{C}[c]\sigma) & \text{otherwise} \end{cases}$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma) \text{ almost...}$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq \begin{cases} \mathcal{C}[c_0]\sigma & \text{if } \mathcal{B}[b]\sigma \\ \mathcal{C}[c_1]\sigma & \text{otherwise} \end{cases}$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq \begin{cases} \sigma & \text{if } \neg \mathcal{B}[b]\sigma \\ \mathcal{C}[\text{while } b \text{ do } c](\mathcal{C}[c]\sigma) & \text{otherwise} \end{cases} \text{ almost...}$$

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma) \text{ almost...}$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq \begin{cases} \mathcal{C}[c_0]\sigma & \text{if } \mathcal{B}[b]\sigma \\ \mathcal{C}[c_1]\sigma & \text{otherwise} \end{cases}$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq \begin{cases} \sigma & \text{if } \neg \mathcal{B}[b]\sigma \\ \mathcal{C}[\text{while } b \text{ do } c](\mathcal{C}[c]\sigma) & \text{otherwise} \end{cases}$$

almost...

not well-founded recursion!

Recursive definitions

for divergence

$$\mathcal{C}[\cdot] : \text{Com} \rightarrow \mathbb{M} \rightarrow \mathbb{M} \cup \{\perp\}$$

$$\mathcal{C}[\text{skip}]\sigma \triangleq \sigma$$

$$\mathcal{C}[x := a]\sigma \triangleq \sigma[\mathcal{A}[a]\sigma / x]$$

$$\mathcal{C}[c_0; c_1]\sigma \triangleq \mathcal{C}[c_1](\mathcal{C}[c_0]\sigma) \text{ almost...}$$

$$\mathcal{C}[\text{if } b \text{ then } c_0 \text{ else } c_1]\sigma \triangleq \begin{cases} \mathcal{C}[c_0]\sigma & \text{if } \mathcal{B}[b]\sigma \\ \mathcal{C}[c_1]\sigma & \text{otherwise} \end{cases}$$

$$\mathcal{C}[\text{while } b \text{ do } c]\sigma \triangleq \begin{cases} \sigma & \text{if } \neg \mathcal{B}[b]\sigma \\ \mathcal{C}[\text{while } b \text{ do } c](\mathcal{C}[c]\sigma) & \text{otherwise} \end{cases} \text{ almost...}$$

not well-founded recursion!

how do we know one solution exists? how do we know it is unique?