MATHEMATICS FOR INFORMATION SCIENCE

NO.10 CONTINUOUS FUNCTION

Tatsuya Hagino

hagino@sfc.keio.ac.jp

Slides URL

https://vu5.sfc.keio.ac.jp/slide/

Fixed Point Theorem

- Theorem: Any continuous function $f: D \to D$ has the least fixed point.
  - u is a fixed point when $f(u) = u$
- Proof: $f(\bot) \sqsubseteq f(f(\bot))$
  - $\bot \sqsubseteq f(\bot) \sqsubseteq f(f(\bot)) \sqsubseteq f^3(\bot) \sqsubseteq f^4(\bot) \sqsubseteq \cdots \sqsubseteq f^i(\bot) \sqsubseteq \cdots$
  - $\sqcup_{i=0}^{\infty} f^i(\bot)$ is the least fixed point.
    - $f\left(\sqcup_{i=0}^{\infty} f^i(\bot)\right) = \sqcup_{i=1}^{\infty} f^i(\bot) = \sqcup_{i=0}^{\infty} f^i(\bot)$ fixed point
    - For any fixed point $u = f(u)$,
      - $\bot \sqsubseteq u$
      - $f(\bot) \sqsubseteq f(u) = u$
      - $f^2(\bot) \sqsubseteq f(u) = u$
      - $f^i(\bot) \sqsubseteq u$
      - Therefore, $\sqcup_{i=0}^{\infty} f^i(\bot) \sqsubseteq u.$ the least fixed point
- fix: $[D \to D] \to D$
  - where $\mathrm{fix}(f) = \sqcup_{i=0}^{\infty} f^i(\bot)$
  - is also continuous.

Fixed Point Semantics

- Recursive programs are difficult to understand.
  - $\text{fact}(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times \text{fact}(x-1)$

- fact: $N_\perp \to N_\perp$
  - $\text{fact} = \lambda x.\, \text{cond}\big(x = 0, 1, x \times \text{fact}(x-1)\big)$
  - fact is a fixed point of the following F:
    - $F: [N_\perp \to N_\perp] \to [N_\perp \to N_\perp]$
    - $F(f) = \lambda x.\, \text{cond}\big(x = 0, 1, x \times f(x-1)\big)$
  - Define fact as the least fixed point of F.
    - $F(\perp) =$
    - $F^2(\perp) =$
    - $F^3(\perp) =$
    - $\vdots$
    - $\text{fix}(F) = \sqcup_{n=0}^{\infty} F^n(\perp)$

$\mathrm{fact} = \mathrm{fix}(F) = \sqcup_{n=0}^{\infty} F^n(\bot)$

- $F(f) = \lambda x.\,\mathrm{cond}\big(x = 0, 1, x \times f(x-1)\big)$
- $F(\bot) = \lambda x.\,\mathrm{cond}\big(x = 0, 1, x \times \bot\,(x-1)\big)$
  $= \lambda x.\,\mathrm{cond}(x = 0, 1, x \times \bot)$
  $= \lambda x.\,\mathrm{cond}(x = 0, 1, \bot)$
- $F^2(\bot) = \lambda x.\,\mathrm{cond}\big(x = 0, 1, x \times F(\bot)(x-1)\big)$
  $= \lambda x.\,\mathrm{cond}\big(x = 0, 1, x \times \mathrm{cond}(x - 1 = 0, 1, \bot)\big)$
  $= \lambda x.\,\mathrm{cond}\big(x = 0, 1, x \times \mathrm{cond}(x = 1, 1, \bot)\big)$
  $= \lambda x.\,\mathrm{cond}\big(x = 0, 1, \mathrm{cond}(x = 1, x \times 1, x \times \bot)\big)$
  $= \lambda x.\,\mathrm{cond}\big(x = 0, 1, \mathrm{cond}(x = 1, 1, \bot)\big)$
  $= \lambda x.\,\mathrm{cond}(x \leq 1, 1, \bot)$

$\text{fact} = \text{fix}(F) = \sqcup_{n=0}^{\infty} F^n(\bot)$

- $F^3(\bot) = \lambda x.\text{cond}\big(x = 0, 1, x \times F^2(\bot)(x-1)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, x \times \text{cond}(x - 1 \leq 1, 1, \bot)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, x \times \text{cond}(x \leq 2, 1, \bot)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, \text{cond}(x \leq 2, x \times 1, x \times \bot)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, \text{cond}(x \leq 2, x!, \bot)\big)$
  $= \lambda x.\text{cond}(x \leq 2, x!, \bot)$
- $F^n(\bot) = \lambda x.\text{cond}(x < n, x!, \bot)$
- $F^{n+1}(\bot) = \lambda x.\text{cond}\big(x = 0, 1, x \times F^n(\bot)(x-1)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, x \times \text{cond}(x - 1 < n, (x-1)!, \bot)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, x \times \text{cond}(x < n + 1, (x-1)!, \bot)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, \text{cond}(x < n + 1, x \times (x-1)!, x \times \bot)\big)$
  $= \lambda x.\text{cond}\big(x = 0, 1, \text{cond}(x < n + 1, x!, \bot)\big)$
  $= \lambda x.\text{cond}(x < n + 1, x!, \bot)$
- fact $= \sqcup_{n=0}^{\infty} F^n(\bot) = \lambda x.\text{cond}(x \geq 0, x!, \bot)$

Example

- $g(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } g(x-1)$
- $g \equiv \lambda x.\text{cond}\big(x = 0, 1, g(x-1)\big)$
- g is the least fixed point of $G(g) \equiv \lambda x.\text{cond}\big(x = 0, 1, g(x-1)\big)$
- $g = \sqcup_{n=0}^{\infty} G^n(\bot)$
- $G(\bot) =$
- $G^2(\bot) =$
- $G^3(\bot) =$
- $G^n(\bot) =$
- $g = \sqcup_{n=0}^{\infty} G^n(\bot) =$

Example 2

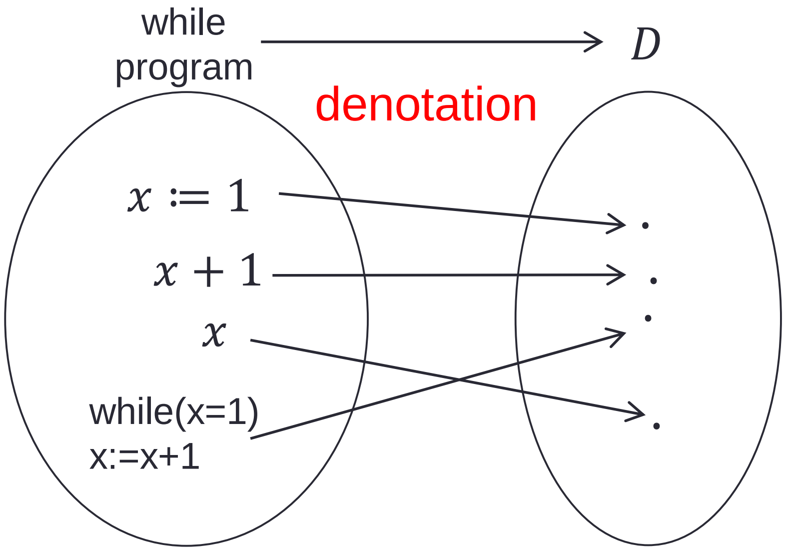
- $h(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } h(x)$
- $h \equiv \lambda x.\, \text{cond}\big(x = 0, 1, h(x)\big)$
- h is the least fixed point of $H(h) \equiv \lambda x.\, \text{cond}\big(x = 0, 1, h(x)\big)$
- $h = \sqcup_{n=0}^{\infty} H^n(\bot)$
- $H(\bot) =$
- $H^2(\bot) =$
- $H^3(\bot) =$
- $H^n(\bot) =$
- $h = \sqcup_{n=0}^{\infty} H^n(\bot) =$

Semantics of Programming Language

- Syntax of a programming language
  - BNF (or Context Free Grammar) is often used for formal definition.
- Semantics of a programming language
  - Natural language is ambiguous
- Formal Semantics
  - Axiomatic Semantics
    - Embed programs in a logic
  - Operational Semantics
    - Simulate programs in a well-known system
  - Denotational Semantics
    - Embed programs into mathematical object

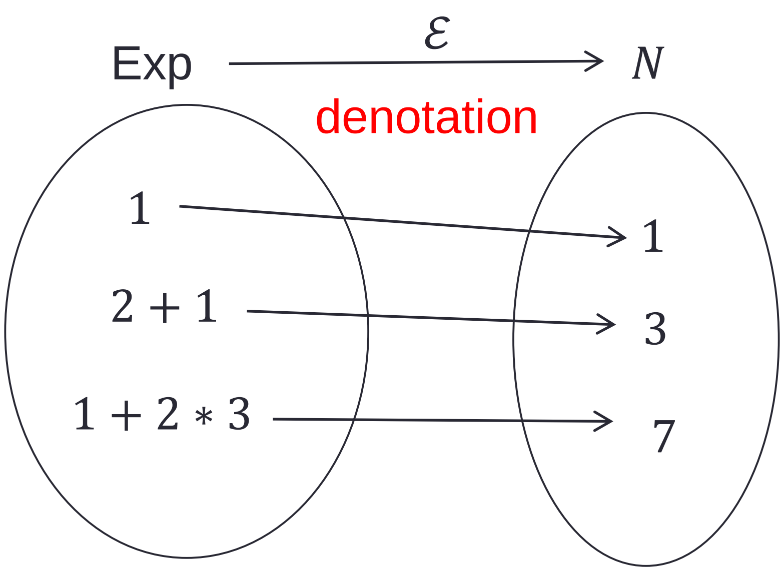
While Program

- While Programs
  - $\text{input}(x_1, x_2, \cdots, x_n)$
  - $\text{output}(y)$
  - $x := e$
  - $\{P_1; P_2;\ \cdots; P_n\}$
  - if $(e_1 = e_2)$ then P else Q
  - while $(e_1 = e_2)$ P



Denotation of Expression

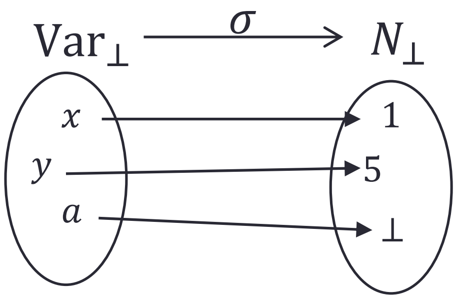
- Abstract syntax for expression $e \in \mathrm{Exp}$
  - $n \in N$
  - $e ::= n \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid e_1 \,/\, e_2$



- $\mathcal{E}[\![n]\!] = n$
- $\mathcal{E}[\![e_1 + e_2]\!] = \mathcal{E}[\![e_1]\!] + \mathcal{E}[\![e_2]\!]$
- $\mathcal{E}[\![e_1 - e_2]\!] = \mathcal{E}[\![e_1]\!] - \mathcal{E}[\![e_2]\!]$
- $\mathcal{E}[\![e_1 * e_2]\!] = \mathcal{E}[\![e_1]\!] \times \mathcal{E}[\![e_2]\!]$
- $\mathcal{E}[\![e_1 \,/\, e_2]\!] = \mathcal{E}[\![e_1]\!] \div \mathcal{E}[\![e_2]\!]$

Denotation of Expression with Variable

- Abstract syntax for expression $e \in \mathrm{Exp}$
  - $n \in N$
  - $v \in \mathrm{Var}$
  - $e ::= n \mid v \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid e_1 \,/\, e_2$
- Denotation of $e \in \mathrm{Exp}$ may depend on the value of variables.
- State S
  - $S = [\mathrm{Var}_\perp \to N_\perp]$
  - $\sigma \in S$ maps variables to their value.



- Denotation of Exp
  - $\mathcal{E} \in \mathrm{Exp} \to [S \to N_\perp]$

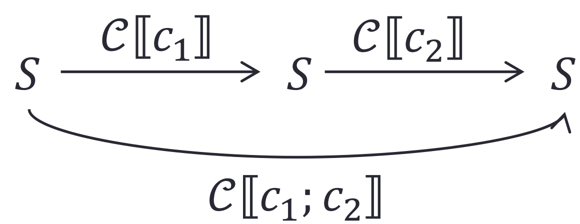
- $\mathcal{E}[\![n]\!]\sigma = n$
- $\mathcal{E}[\![v]\!]\sigma = \sigma[\![v]\!]$
- $\mathcal{E}[\![e_1 + e_2]\!]\sigma = \mathcal{E}[\![e_1]\!]\sigma + \mathcal{E}[\![e_2]\!]\sigma$
- $\mathcal{E}[\![e_1 - e_2]\!]\sigma = \mathcal{E}[\![e_1]\!]\sigma - \mathcal{E}[\![e_2]\!]\sigma$
- $\mathcal{E}[\![e_1 * e_2]\!]\sigma = \mathcal{E}[\![e_1]\!]\sigma \times \mathcal{E}[\![e_2]\!]\sigma$
- $\mathcal{E}[\![e_1 \,/\, e_2]\!]\sigma = \mathcal{E}[\![e_1]\!]\sigma \div \mathcal{E}[\![e_2]\!]\sigma$

Denotation of Command

- Abstract syntax for command $c \in \mathrm{Cmd}$
  - $c ::= \mathrm{null} \mid x \coloneqq e \mid \mathrm{if}\ (e_1 = e_2)\ c_1\ \mathrm{else}\ c_2 \mid \mathrm{while}\ (e_1 = e_2)\ c \mid c_1; c_2$
- Denotation of Cmd
  - $\mathcal{C} \in \mathrm{Cmd} \to [S \to S]$

$$S \xrightarrow{\mathcal{C}[\![c]\!]} S$$

- $\mathcal{C}[\![\mathrm{null}]\!]\sigma = \sigma$
- $\mathcal{C}[\![c_1; c_2]\!]\sigma = \mathcal{C}[\![c_2]\!](\mathcal{C}[\![c_1]\!]\sigma)$
  - $\mathcal{C}[\![c_1; c_2]\!] = \mathcal{C}[\![c_2]\!] \circ \mathcal{C}[\![c_1]\!]$



- $\mathcal{C}[\![x \coloneqq e\,]\!]\sigma = \sigma[\mathcal{E}[\![e]\!]\sigma / x]$

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

- $\mathcal{C}[\![\mathrm{if}\ (e_1 = e_2)\ c_1\ \mathrm{else}\ c_2]\!]\sigma = \mathrm{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, \mathcal{C}[\![c_1]\!]\sigma, \mathcal{C}[\![c_2]\!]\sigma)$

Denotation of Command (cont)

- $\mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!]\sigma$
- $\text{while } (e_1 = e_2)\ c \equiv \text{if } (e_1 = e_2)\ \{c; \text{while } (e_1 = e_2)\ c\} \text{ else null}$
- $\mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!]\sigma = \mathcal{C}[\![\text{if } (e_1 = e_2)\ \{c; \text{while } (e_1 = e_2)\ c\} \text{ else null }]\!]\sigma$

$$= \text{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, \mathcal{C}[\![c; \text{while } (e_1 = e_2)\ c]\!]\sigma, \mathcal{C}[\![\text{null}]\!]\sigma)$$

$$= \text{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, \mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!](\mathcal{C}[\![c]\!]\sigma), \sigma)$$

- $\mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!] = \lambda\sigma.\,\text{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, \mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!](\mathcal{C}[\![c]\!]\sigma), \sigma)$
- $\mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!]$ is the fixed point of
  - $\lambda w.\,\lambda\sigma.\,\text{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, w(\mathcal{C}[\![c]\!]\sigma), \sigma)$
- $\mathcal{C}[\![\text{while } (e_1 = e_2)\ c]\!] = \text{fix}\big(\lambda w.\,\lambda\sigma.\,\text{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, w(\mathcal{C}[\![c]\!]\sigma), \sigma)\big)$

$$\text{fix}(f) \equiv \sqcup_{i=0}^{\infty} f^{i}(\perp)$$

Denotation of Command (summary)

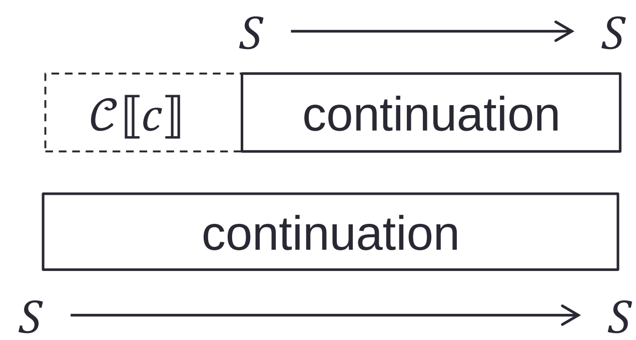
- Abstract syntax for command $c \in \mathrm{Cmd}$
  - $c ::= \mathrm{null} \mid x \coloneqq e \mid \mathrm{if}\ (e_1 = e_2)\ c_1\ \mathrm{else}\ c_2 \mid \mathrm{while}\ (e_1 = e_2)\ c \mid c_1; c_2$
- Denotation of Cmd
  - $\mathcal{C} \in \mathrm{Cmd} \to [S \to S]$

$$S \xrightarrow{\mathcal{C}[\![c]\!]} S$$

- $\mathcal{C}[\![\mathrm{null}]\!]\sigma = \sigma$
- $\mathcal{C}[\![c_1; c_2]\!] = \mathcal{C}[\![c_2]\!] \circ \mathcal{C}[\![c_1]\!]$
- $\mathcal{C}[\![x \coloneqq e\ ]\!]\sigma = \sigma[\mathcal{E}[\![e]\!]\sigma / x]$
- $\mathcal{C}[\![\mathrm{if}\ (e_1 = e_2)\ c_1\ \mathrm{else}\ c_2]\!]\sigma = \mathrm{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, \mathcal{C}[\![c_1]\!]\sigma, \mathcal{C}[\![c_2]\!]\sigma)$
- $\mathcal{C}[\![\mathrm{while}\ (e_1 = e_2)\ c]\!] = \mathrm{fix}\big(\lambda w. \lambda\sigma. \mathrm{cond}(\mathcal{E}[\![e_1]\!]\sigma = \mathcal{E}[\![e_2]\!]\sigma, w(\mathcal{C}[\![c]\!]\sigma), \sigma)\big)$

Continuation

- Difficult to handle side effect
  - $\mathcal{C}[\![x := e\,]\!]\sigma = \sigma[\mathcal{E}[\![e]\!]\sigma/x]$
- Continuation is
  - the rest of the computation
  - $C = [S \to S]$



- $\mathcal{C}[\![c]\!]$
  - receive the rest of the computation
  - returns the computation including c
  - $\mathcal{C}[\![c]\!] \in [C \to C]$

$$[S \to S] \xrightarrow{\mathcal{C}[\![c]\!]} [S \to S]$$

- $\mathcal{E}[\![e]\!] \in [K \to C]$
  - $K = [N_\bot \to C]$
  - K is continuation of expression

$$[N_\bot \to [S \to S]] \xrightarrow{\mathcal{E}[\![e]\!]} [S \to S]$$

Denotation of Command (Continuation)

- Abstract syntax for command $c \in \mathrm{Cmd}$
  - $c ::= \mathrm{null} \mid x := e \mid \mathrm{if}\ (e_1 = e_2)\ c_1\ \mathrm{else}\ c_2 \mid \mathrm{while}\ (e_1 = e_2)\ c \mid c_1; c_2$
- Denotation of Cmd
  - $C = [S \to S]$
  - $\mathcal{C} \in \mathrm{Cmd} \to [C \to C]$

$$[S \to S] \xrightarrow{\mathcal{C}[\![c]\!]} [S \to S]$$

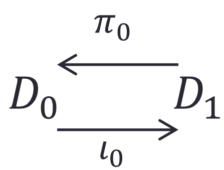
- $\mathcal{C}[\![c]\!]\theta\sigma = \theta(\sigma')$
  - Execute a command c in state σ and the modified state σ' is passed to θ.
- $\mathcal{C}[\![\mathrm{null}]\!]\theta = \theta$
- $\mathcal{C}[\![c_1; c_2]\!]\theta = \mathcal{C}[\![c_1]\!](\mathcal{C}[\![c_2]\!]\theta)$
- $\mathcal{C}[\![x := e\ ]\!]\theta\sigma = \mathcal{E}[\![e]\!](\lambda n.\theta(\sigma[n/x]))$
- $\mathcal{C}[\![\mathrm{if}\ (e_1 = e_2)\ c_1\ \mathrm{else}\ c_2]\!]\theta = \mathcal{E}[\![e_1]\!]\left(\lambda n_1.\mathcal{E}[\![e_2]\!]\left(\lambda n_2.\mathrm{cond}(n_1 = n_2, \mathcal{C}[\![c_1]\!]\theta, \mathcal{C}[\![c_2]\!]\theta)\right)\right)$
- $\mathcal{C}[\![\mathrm{while}\ (e_1 = e_2)\ c]\!] = \mathrm{fix}\left(\lambda w.\lambda\theta.\mathcal{E}[\![e_1]\!]\left(\lambda n_1.\mathcal{E}[\![e_2]\!]\left(\lambda n_2.\mathrm{cond}\left(n_1 = n_2, \mathcal{C}[\![c]\!](w(\theta)), \theta\right)\right)\right)\right)$

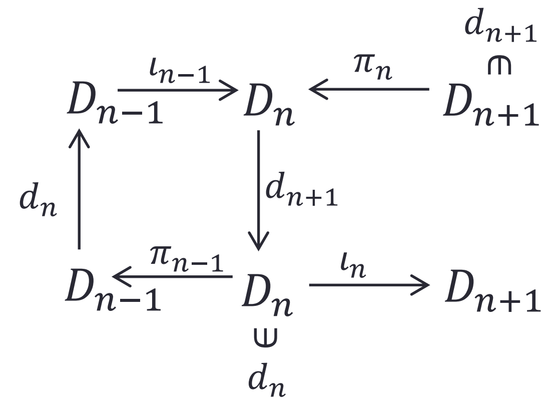
Denotation of Expression (Continuation)

- Abstract syntax for expression $e \in \mathrm{Exp}$
  - $n \in N$
  - $v \in \mathrm{Var}$
  - $e ::= n \mid v \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid e_1 / e_2 \mid v{+}{+}$
- Denotation of Exp
  - $\mathcal{E} \in \mathrm{Exp} \to [K \to C]$
  - $K = [N_\perp \to C]$
- $\mathcal{E}[\![e]\!]\kappa\sigma = \kappa(n)(\sigma')$
  - Calculate the value of e in state σ and the result n is passed to κ with the modified state σ'.
- $\mathcal{E}[\![n]\!]\kappa\sigma = \kappa(n)(\sigma)$
- $\mathcal{E}[\![v]\!]\kappa\sigma = \kappa(\sigma[\![v]\!])\sigma$
- $\mathcal{E}[\![e_1 + e_2]\!]\kappa\sigma = \mathcal{E}[\![e_1]\!]\left(\lambda n_1. \mathcal{E}[\![e_2]\!](\lambda n_2. \kappa(n_1 + n_2))\right)\sigma$
- $\mathcal{E}[\![v{+}{+}]\!]\kappa\sigma = \kappa(\sigma[\![v]\!])\sigma[\sigma[\![v]\!] + 1/v]$

Domain for Lambda Expression

- Construct a domain D where $D \cong [D \to D]$
  - $D_0 = \{\cdot\}_\perp$
  - $D_1 = [D_0 \to D_0]$
  - $D_2 = [D_1 \to D_1]$
  - $\cdots$
  - $D_{n+1} = [D_n \to D_n]$





- $D_0 \lhd D_1 \lhd D_2 \lhd D_3 \cdots \lhd D_n \lhd \cdots$
  - $\pi_0 \in D_1 \to D_0 = [D_0 \to D_0] \to D_0$ $\quad$ $\pi_0(d_1) = d_1\left(\perp_{D_0}\right)$
  - $\iota_0 \in D_0 \to D_1 = D_0 \to [D_0 \to D_0]$ $\quad$ $\iota_0(d_0) = \lambda x_1 \in D_0.\, d_0$
  - $\pi_n \in D_{n+1} \to D_n = [D_n \to D_n] \to D_n$ $\quad$ $\pi_n(d_{n+1}) = \pi_{n-1} \circ d_{n+1} \circ \iota_{n-1}$
  - $\iota_n \in D_n \to D_{n+1} = D_n \to [D_n \to D_n]$ $\quad$ $\iota_n(d_n) = \iota_{n-1} \circ d_n \circ \pi_{n-1}$

- $D_\infty = \{\, (d_0, d_1, \cdots, d_n, \cdots) \mid d_n \in D_n ,\, \pi_n(d_{n+1}) = d_n \}$
  - $D_n \lhd D_\infty$
  - $D_\infty \cong [\mathrm{D}_\infty \to D_\infty]$

Denotation of Lambda Expression

- Abstract syntax for lambda expression $M \in \Lambda$
  - $x \in \mathrm{Var}$
  - $M ::= x \mid \lambda x. M \mid M_1\ M_2$

- $D_\infty \cong [\mathrm{D}_\infty \to D_\infty]$
  - $\pi \in [D_\infty \to D_\infty] \to D_\infty$
  - $\iota \in D_\infty \to [D_\infty \to D_\infty]$

- Denotation of Lambda Expression
  - $\sigma \in S = \mathrm{Var} \to D_\infty$
  - $\mathcal{L} \in \Lambda \to [S \to D_\infty]$

- $\mathcal{L}[\![x]\!]\sigma = \sigma[\![x]\!]$
- $\mathcal{L}[\![\lambda x. M]\!]\sigma = \pi(\lambda v \in D_\infty. \mathcal{L}[\![M]\!]\sigma[v/x])$
- $\mathcal{L}[\![M_1\ M_2]\!]\sigma = \iota(\mathcal{L}[\![M_1]\!]\sigma)(\mathcal{L}[\![M_2]\!]\sigma)$

Summary

- Fixed point
  - Fixed point theorem
  - Fixed point semantics for recursive functions

- Semantics of Programming Language
  - Axiomatic Semantics
  - Operational Semantics
  - Denotational Semantics

- Denotational Semantics
  - Semantic Function
  - Continuation
  - D_∞

Homework 10

- Find out what the following functions actually are using fixed point semantics.
  - $f(x) \equiv \text{if } x = 0 \text{ then } 0 \text{ else } f\big(f(x-1)\big) + 1$
  - $g(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } g\big(g(x-1)\big) + 1$

- Hint:
  - f is the least fixed point of
    $F(f) = \lambda x.\mathrm{cond}\big(x = 0, 0, f\big(f(x-1)\big) + 1\big)$
    Calculate $\sqcup_{n=0}^{\infty} F^n(\bot)$

  - g is the least fixed point of
    $G(g) = \lambda x.\mathrm{cond}\big(x = 0, 1, g\big(g(x-1)\big) + 1\big)$
    Calculate $\sqcup_{n=0}^{\infty} G^n(\bot)$