Kripke-style semantics for Full Simply Typed Lambda Calculus

Simona Kašterović[1] Silvia Ghilezan[1, 2]

[1]Faculty of Technical Sciences, University of Novi Sad
[2]Mathematical Institute SASA, Belgrade, Serbia

Logic and Applications 2020
September 21-25, 2020

LAP 2019 (reminder)

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

LAP 2019 (reminder)

- Full simply typed lambda calculus

Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

Terms	$M, N ::= x \mid \lambda x.M \mid MN \mid \pi_1(M) \mid \pi_2(M) \mid \langle M, N\rangle \mid \mathsf{in}_1(M) \mid \mathsf{in}_2(M) \mid$ $\mid \mathsf{case}\ M\ \mathsf{of}\ \mathsf{in}_1(x) \Rightarrow N \mid \ \mathsf{in}_2(y) \Rightarrow L \mid \langle\rangle \mid \mathsf{abort}(M)$
Types	$\sigma, \tau ::= a \mid \sigma \to \tau \mid \sigma \times \tau \mid \sigma + \tau \mid 0 \mid 1$

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

$$\frac{x:\sigma \in \Gamma}{\Gamma \vdash x:\sigma} \qquad \frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M:\sigma \to \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash MN:\tau}$$

$$\frac{\Gamma \vdash M:\sigma \quad \Gamma \vdash N:\tau}{\Gamma \vdash \langle M, N\rangle : \sigma \times \tau} \qquad \frac{\Gamma \vdash M:\sigma \times \tau}{\Gamma \vdash \pi_1(M):\sigma} \qquad \frac{\Gamma \vdash M:\sigma \times \tau}{\Gamma \vdash \pi_2(M):\tau}$$

$$\frac{\Gamma \vdash M:\sigma}{\Gamma \vdash \mathrm{in}_1(M):\sigma + \tau} \qquad \frac{\Gamma \vdash M:\tau}{\Gamma \vdash \mathrm{in}_2(M):\sigma + \tau}$$

$$\frac{\Gamma \vdash M:\sigma + \tau \qquad \Gamma, x:\sigma \vdash N:\rho \qquad \Gamma, y:\tau \vdash L:\rho}{\Gamma \vdash \mathrm{case}\ M\ \mathrm{of}\ \mathrm{in}_1(x) \Rightarrow N \mid\ \mathrm{in}_2(y) \Rightarrow L : \rho}$$

$$\frac{}{\Gamma \vdash \langle\rangle : 1} \qquad \frac{\Gamma \vdash M:0}{\Gamma \vdash \mathrm{abort}(M):\sigma}$$

LAP 2019 (reminder)

- Full simply typed lambda calculus
  - Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.
- Kripke-style semantics

LAP 2019 (reminder)

- Full simply typed lambda calculus

Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure

$$\mathcal{K} = \langle W, \leq, \{A_w^\sigma\}, \{i_{w,w'}^\sigma\}\rangle$$

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure

$$\mathcal{K} = \langle W, \leq, \{A_w^\sigma\}, \{i_{w,w'}^\sigma\} \rangle$$

W is a set of "possible worlds" partially ordered by $\leq$

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure

$$\mathcal{K} = \langle W, \leq, \{A^{\sigma}_{w}\}, \{i^{\sigma}_{w,w'}\}\rangle$$

$\{A^{\sigma}_{w}\}$ is a family of sets indexed by types σ and worlds w

LAP 2019 (reminder)

- Full simply typed lambda calculus

Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure

$$\mathcal{K} = \langle W, \leq, \{A^\sigma_w\}, \{i^\sigma_{w,w'}\}\rangle$$

a family $\{i^\sigma_{w,w'}\}$ of "transition functions" $i^\sigma_{w,w'}$: $A^\sigma_w \to A^\sigma_{w'}$ indexed by types of σ and pairs of worlds $w \leq w'$, which satisfy the following conditions:

$$i^\sigma_{w,w} : A^\sigma_w \to A^\sigma_w \text{ is identity} \qquad \text{(id)}$$

$$i^\sigma_{w',w''} \circ i^\sigma_{w,w'} = i^\sigma_{w,w''} \text{ for all } w \leq w' \leq w'' \qquad \text{(comp)}$$

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure

    $$\mathcal{K} = \langle W, \leq, \{A^\sigma_w\}, \{i^\sigma_{w,w'}\}\rangle$$

  - valuation ρ - partial mapping, $\rho : V \times W \to \bigcup_{\sigma \in \mathsf{Type}, w \in W} A^\sigma_w$,

    If $\rho(x, w) \in A^\sigma_w$ and $w \leq w'$ then $\rho(x, w') = i^\sigma_{w,w'}(\rho(x, w))$.

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure
    $$\mathcal{K} = \langle W, \leq, \{A_w^\sigma\}, \{i_{w,w'}^\sigma\}\rangle$$
  - valuation ρ - partial mapping, $\rho : V \times W \to \bigcup_{\sigma \in \mathrm{Type}, w \in W} A_w^\sigma$,
    $$\text{If } \rho(x, w) \in A_w^\sigma \text{ and } w \leq w' \text{ then } \rho(x, w') = i_{w,w'}^\sigma(\rho(x, w)).$$
  - Kripke lambda model $\mathcal{K}_\rho = \langle \mathcal{K}, \rho \rangle$

$$w \models x : \sigma \text{ if and only if } \rho(x, w) \in A_w^\sigma$$

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure
    $$\mathcal{K} = \langle W, \leq, \{A_w^\sigma\}, \{i_{w,w'}^\sigma\}\rangle$$
  - valuation ρ - partial mapping, $\rho : V \times W \to \bigcup_{\sigma \in \mathsf{Type}, w \in W} A_w^\sigma$,
    $$\text{If } \rho(x, w) \in A_w^\sigma \text{ and } w \leq w' \text{ then } \rho(x, w') = i_{w,w'}^\sigma(\rho(x, w)).$$
  - Kripke lambda model $\mathcal{K}_\rho = \langle \mathcal{K}, \rho \rangle$

$$w \models x : \sigma \text{ if and only if } \rho(x, w) \in A_w^\sigma$$

- Soundness (**proved**)

LAP 2019 (reminder)

- Full simply typed lambda calculus

  Mitchell, J. C., *Foundations for programming languages*, Foundation of computing series. MIT Press, 1996.

- Kripke-style semantics
  - Kripke applicative structure
    $$\mathcal{K} = \langle W, \leq, \{A_w^\sigma\}, \{i_{w,w'}^\sigma\}\rangle$$
  - valuation ρ - partial mapping, $\rho : V \times W \to \bigcup_{\sigma \in \mathsf{Type}, w \in W} A_w^\sigma$,

    If $\rho(x, w) \in A_w^\sigma$ and $w \leq w'$ then $\rho(x, w') = i_{w,w'}^\sigma(\rho(x, w))$.
  - Kripke lambda model $\mathcal{K}_\rho = \langle \mathcal{K}, \rho \rangle$

$w \models x : \sigma$ if and only if $\rho(x, w) \in A_w^\sigma$

- Soundness (**proved**)
- Completeness (**conjectured**)

Failure

Goal:

Inconsistent basis is unsatisfiable.

- Γ is inconsistent if $\Gamma \vdash M : 0$ for some M.

Failure

Goal:

Inconsistent basis is unsatisfiable. ✗

- Γ is inconsistent if $\Gamma \vdash M : 0$ for some M.

NEW Kripke-style semantics

Kašterović, S., Ghilezan, S., *Kripke semantics and completeness for full simply typed lambda calculus*, to appear in Journal of Logic and Computation Volume 30, issue 8 (2020).

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

NEW Kripke-style semantics

$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A^\sigma_w\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$

W is a set of *possible worlds* partially ordered by $\leq$

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{D_w\}_{w \in W}$ of sets, indexed by worlds w, where the set D_w represents the domain of a world w,

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{A_w^\sigma\}$ of sets indexed by types σ and worlds w

Sets A_w^σ satisfy conditions:

- for all $w \in W$, for all $\sigma \in$ Type, $A_w^\sigma \subseteq D_w$, A_w^0 is empty, i.e. $A_w^0 = \emptyset$, and A_w^1 has one element, i.e. $A_w^1 = \{1_w\}$, $1_w \in D_w$,
- there exists an injection function $H : D_w \uplus D_w \to D_w$, such that for all $\sigma, \tau \in$ Type, $H : A_w^\sigma \uplus A_w^\tau \to A_w^{\sigma+\tau}$,
- there exists an injection function $G : D_w \to D_w \times D_w$ such that for all $\sigma, \tau \in$ Type, $G : A_w^{\sigma\times\tau} \to A_w^\sigma \times A_w^\tau$,

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{App_w\}$ of *application functions* $App_w : D_w \times D_w \to D_w$ indexed by worlds w

$$App_w \restriction (A_w^{\sigma\to\tau} \times A_w^\sigma) : A_w^{\sigma\to\tau} \times A_w^\sigma \to A_w^\tau$$

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{Proj_{1,w}\}$ of *first projection functions* $Proj_{1,w} : D_w \to D_w$ indexed by worlds w

$$Proj_{1,w} \restriction A_w^{\sigma\times\tau} : A_w^{\sigma\times\tau} \to A_w^\sigma$$

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\} \rangle$$

a family $\{Proj_{2,w}\}$ of *second projection functions* $Proj_{2,w} : D_w \to D_w$ indexed by worlds w

$$Proj_{2,w} \restriction A_w^{\sigma\times\tau} : A_w^{\sigma\times\tau} \to A_w^\tau$$

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{Inl_w\}$ of *left injection functions* $Inl_w : D_w \to D_w$ indexed by worlds w

$$Inl_w \restriction A_w^\sigma : A_w^\sigma \to A_w^{\sigma+\tau}$$

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A_w^\sigma\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{Inr_w\}$ of *right injection functions* $Inr_w : D_w \to D_w$ indexed by worlds w

$$Inr_w \restriction A_w^\tau : A_w^\tau \to A_w^{\sigma+\tau}$$

NEW Kripke-style semantics

$$\mathcal{K} = \langle W, \leq, \{D_w\}, \{A^\sigma_w\}, \{App_w\}, \{Proj_{1,w}\}, \{Proj_{2,w}\}, \{Inl_w\}, \{Inr_w\}, \{i_{w,w'}\}\rangle$$

a family $\{i_{w,w'}\}$ of *transition functions* $i_{w,w'} : D_w \to D_{w'}$ indexed by pairs of worlds $w \leq w'$

$i_{w,w'} \restriction A^\sigma_w : A^\sigma_w \to A^\sigma_{w'}$ and all transition functions satisfy the following conditions:

$$i_{w,w} : D_w \to D_w \text{ is the identity} \qquad \text{(id)}$$

$$i_{w',w''} \circ i_{w,w'} = i_{w,w''} \text{ for all } w \leq w' \leq w'' \qquad \text{(comp)}$$

We also require that the application functions, the projection functions and the injection functions commute with the transition in a natural way:

$$(\forall f \in D_w)\ (\forall a \in D_w)\ (\forall w' \in W, w \leq w')$$

$$i_{w,w'}(App_w(f,a)) = App_{w'}(i_{w,w'}(f), i_{w,w'}(a)) \quad \text{(comm1)}$$

$$i_{w,w'}(Proj_{1,w}(a)) = Proj_{1,w'}(i_{w,w'}(a)) \quad \text{(comm2)}$$

$$i_{w,w'}(Proj_{2,w}(a)) = Proj_{2,w'}(i_{w,w'}(a)) \quad \text{(comm3)}$$

$$i_{w,w'}(Inl_w(a)) = Inl_{w'}(i_{w,w'}(a)) \quad \text{(comm4)}$$

$$i_{w,w'}(Inr_w(a)) = Inr_{w'}(i_{w,w'}(a)) \quad \text{(comm5)}$$

;

Kašterović, S., Ghilezan, S., *Kripke semantics and completeness for full simply typed lambda calculus,* to appear in Journal of Logic and Computation Volume 30, issue 8 (2020).

Kripke lambda model

$$\mathcal{K}_\rho = \langle \mathcal{K}, \rho \rangle$$

- $\mathcal{K}$ is a Kripke applicative structure which is **extensional** and **has combinators**
- ρ - partial mapping, for $x \in V$, $w \in W$, if $\rho(x, w)$ is defined, then $\rho(x, w) \in D_w$,

$$\text{if } \rho(x, w) \in D_w \text{ and } w \leq w', \text{ then } \rho(x, w') = i_{w,w'}(\rho(x, w)).$$

$[\![M]\!]^w_\rho$ - meaning of the term M

- $[\![x]\!]^w_\rho = \rho(x, w)$;
- $[\![\lambda x.M]\!]^w_\rho =$ a unique element $d \in D_w$, such that

$$(\forall w' \geq w)(\forall a \in D_{w'})\ (App_{w'}(i_{w,w'}(f), a) = [\![M]\!]^{w'}_{\rho(x:=a)})$$

- $[\![MN]\!]^w_\rho = App_w([\![M]\!]^w_\rho, [\![N]\!]^w_\rho)$;
- $[\![\pi_1(M)]\!]^w_\rho = Proj_{1,w}([\![M]\!]^w_\rho)$;
- $[\![\pi_2(M)]\!]^w_\rho = Proj_{2,w}([\![M]\!]^w_\rho)$;
- $[\![\langle M, N\rangle]\!]^w_\rho =$ a unique $p \in D_w$ such that

$$Proj_{1,w}(p) = [\![M]\!]^w_\rho$$
$$Proj_{2,w}(p) = [\![N]\!]^w_\rho$$

$[\![M]\!]^w_\rho$ - meaning of the term M ...

- $[\![\mathrm{in}_1(M)]\!]^w_\rho = Inl_w([\![M]\!]^w_\rho)$;
- $[\![\mathrm{in}_2(M)]\!]^w_\rho = Inr_w([\![M]\!]^w_\rho)$;
- $[\![\mathrm{case}\ M\ \mathrm{of}\ \mathrm{in}_1(x) \Rightarrow N \mid\ \mathrm{in}_2(y) \Rightarrow L]\!]^w_\rho = App_w(f, [\![M]\!]^w_\rho)$, where f is a unique element of D_w such that

$$(\forall w' \geq w)(\forall a \in D_{w'})(\forall b \in D_{w'})$$
$$App_{w'}(i_{w,w'}(f), Inl_{w'}(a)) = [\![N]\!]^{w'}_{\rho(x:=a)}$$
$$App_{w'}(i_{w,w'}(f), Inr_{w'}(b)) = [\![L]\!]^{w'}_{\rho(y:=b)}$$

- $[\![\langle\ \rangle]\!]^w_\rho = 1_w$ a unique element of A^1_w;
- $[\![\mathrm{abort}(M)]\!]^w_\rho = App_w(Z_w, [\![M]\!]^w_\rho)$.

Satisfiability

$$w \models M : \sigma \text{ if and only if } [\![M]\!]_\rho^w \in A_w^\sigma.$$

$$\mathcal{K}_\rho \models M : \sigma \text{ if and only if } w \models M : \sigma \text{ for all } w.$$

Satisfiability

$$w \models M : \sigma \text{ if and only if } [\![M]\!]_\rho^w \in A_w^\sigma.$$

$$\mathcal{K}_\rho \models M : \sigma \text{ if and only if } w \models M : \sigma \text{ for all } w.$$

Theorem (Soundness)

If $\Gamma \vdash M : \sigma$, then $\Gamma \models M : \sigma$.

- We defined a canonical model $\mathcal{K}^\Gamma$ such that $\mathcal{K}^\Gamma \models M : \sigma$ if and only if $\Gamma \vdash M : \sigma$.

Completeness

If $\Gamma \models M : \sigma$, then $\Gamma \vdash M : \sigma$.


THANK
YOU