# Practical Foundations for Programming Languages

Jason Hu, Zhe Zhou, Ramana Nagasamudram

June 19. 2019

## 1 Essentials

- Practical Foundations for Programming Languages. `https://cs.cmu.edu/~rwh/pfpl`
- Abstract syntax with binding and scope
- Inductive definitions "least relation closed with rules"
- Safety for partial languages

"What does it mean for a PL to exist?"

"A mathematical object subject to rigorous analysis."

1. abstract syntax
2. statics / formation
3. dynamics / execution → the truth about what is going on

All should be coherent.

Statics is approximation of dynamics. Statics is a prediction about the dynamics.

## 2 Abstract Syntax

$$e ::= \ x \mid \texttt{tt} \mid \texttt{ff} \mid \overline{V}_I(e_1, e_2) \mid \overline{V}_{II}(e_1, e_2) \mid \texttt{fix}(x.e)$$

## 3 Statics

We write $e$ `circ` to denote that $e$ is a well-formed circuit.

$$\frac{}{\texttt{tt circ}} \qquad \frac{}{\texttt{ff circ}} \qquad \frac{e_1\ \texttt{circ} \qquad e_2\ \texttt{circ}}{\overline{V}_*(e_1, e_2)\ \texttt{circ}} \qquad \frac{}{x\ \texttt{circ} \vdash x\ \texttt{circ}}$$

$$\frac{x\ \texttt{circ} \vdash e\ \texttt{circ}}{\texttt{fix}(x.e)\ \texttt{circ}}$$

A variable is a placeholder, not a direct participant of computation. It stands for a circuit to be determined.

Consequence relations (entailment, ..) specify the meanings of variables.

# 4 Dynamics

"$e \Downarrow v$" denotes that $e$ evaluates to $v$.

$$\frac{}{\texttt{tt} \Downarrow \texttt{tt}} \qquad \frac{}{\texttt{ff} \Downarrow \texttt{ff}} \qquad \frac{e_1 \Downarrow \texttt{tt}}{\overline{V}_I(e_1, e_2) \Downarrow \texttt{ff}} \qquad \frac{e_1 \Downarrow \texttt{ff} \qquad e_2 \Downarrow \texttt{tt}}{\overline{V}_I(e_1, e_2) \Downarrow \texttt{ff}}$$

$$\frac{e_1 \Downarrow \texttt{ff} \qquad e_2 \Downarrow \texttt{ff}}{\overline{V}_I(e_1, e_2) \Downarrow \texttt{tt}} \qquad \frac{e_2 \Downarrow \texttt{tt}}{\overline{V}_{II}(e_1, e_2) \Downarrow \texttt{ff}} \qquad \frac{e_1 \Downarrow \texttt{tt} \qquad e_2 \Downarrow \texttt{ff}}{\overline{V}_{II}(e_1, e_2) \Downarrow \texttt{ff}}$$

$$\frac{e_1 \Downarrow \texttt{ff} \qquad e_2 \Downarrow \texttt{ff}}{\overline{V}_{II}(e_1, e_2) \Downarrow \texttt{tt}} \qquad \frac{[\texttt{fix}(x.e)/x]e \Downarrow v}{\texttt{fix}(x.e) \Downarrow v}$$

Fact: If $\emptyset \vdash e$ `circ` then there exists $v$, $e \Downarrow v$.

*Proof.* Induction on statics. □

# 5 Structural Properties of Entailment

Structural properties characterizes behavior of variables.

$$\frac{}{x\ \texttt{circ} \vdash x\ \texttt{circ}}\ \textsc{Reflexivity} \qquad \frac{e_1\ \texttt{circ} \qquad x\ \texttt{circ} \vdash x\ \texttt{circ}}{[e_1/x]e_2\ \texttt{circ}}\ \textsc{Substitution}$$

$$\frac{e\ \texttt{circ} \qquad x \notin e}{x\ \texttt{circ} \vdash e\ \texttt{circ}}\ \textsc{Weakening} \qquad \frac{x\ \texttt{circ}, y\ \texttt{circ} \vdash e\ \texttt{circ}}{y\ \texttt{circ}, x\ \texttt{circ} \vdash e\ \texttt{circ}}\ \textsc{Exchange}$$

$$\frac{x\ \texttt{circ}, y\ \texttt{circ} \vdash e\ \texttt{circ}}{z\ \texttt{circ} \vdash [z, z/x, y]e\ \texttt{circ}}\ \textsc{Contraction}$$

Structural properties express limiting conditions on the language. For example, type is a kind of restrictions.

Exercises: BDDS.

$$X \stackrel{?}{=} \overline{V}_I(R, \overline{V}_{II}(X, S))$$

How to deal with recursion?

$$\begin{cases} X = \overline{V}_I(R, Y) \\ Y = \overline{V}_I(X, S) \end{cases}$$

Expanding $X$ gives

$$X = \overline{V}_I(R, \overline{V}_{II}(X, S))$$

Can we take this self reference formula and solve for X? We assume X $= e_X$, and is equal to $f(X)$. Seek a fixpoint for f. which is a f make the things in the left hand and right hand the same.

That is the "Y combinator" a.k.a. "fixed point combinator".

Example:

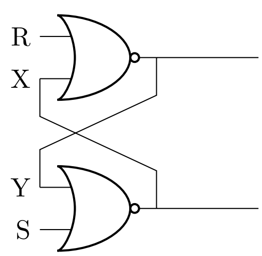


Figure 1: RS Latch

$$RSL(R, S) = \texttt{fix}(x.\overline{V}_I(R, \overline{V}_{II}(x, S)))$$

Attempting to evaluate $RSL(\texttt{ff}, \texttt{tt})$:

$$\begin{aligned} RSL(\texttt{ff}, \texttt{tt}) &= \texttt{fix}(x.\overline{V}_I(\texttt{ff}, \overline{V}_{II}(x, \texttt{tt}))) \\ &= \overline{V}_I(\texttt{ff}, \overline{V}_{II}(RSL(\texttt{ff}, \texttt{tt}), \texttt{tt})) \\ &= \overline{V}_I(\texttt{ff}, \texttt{ff}) \\ &= \texttt{tt} \end{aligned}$$

$RSL(\texttt{ff}, \texttt{ff})$ has no value as it diverges.

# 6 Check Coherence

We may have some undefined terms: $\texttt{fix}(x.x)$, $\frac{1}{0}$...

What is the meaning of well defined?

Structural properties state that we may substitute computations (v.s. values) for variables. It is obvious that $e$'s need not evaluate to $v$'s. With `fix`, coherence can no longer be expressed as evaluating values. So what do we say instead? The issue is, in general there are many sorts of undefinedness. To handle this, the idea is to distinguish divergence from nonsense.

Idea: To introduce "time" (though it introduces complexity):

1. Using time to express coherence;
2. Using time to express "propagation delay".

$e$ val "final state"

$$\frac{}{\texttt{tt}\ \text{val}} \qquad \frac{}{\texttt{ff}\ \text{val}}$$

**Structural operational semantics(SOS)**

$e \mapsto e'$ denotes a transition system.

$$\frac{e_1 \mapsto e_1'}{\overline{V}_I(e_1, e_2) \mapsto \overline{V}_I(e_1', e_2)} \qquad \frac{}{\overline{V}_I(\texttt{tt}, e_2) \mapsto \texttt{ff}} \qquad \frac{e_2 \mapsto e_2'}{\overline{V}_I(\texttt{ff}, e_2) \mapsto \overline{V}_I(\texttt{ff}, e_2')}$$

$$\frac{}{\overline{V}_I(\texttt{ff}, \texttt{tt}) \mapsto \texttt{ff}} \qquad \frac{}{\overline{V}_I(\texttt{ff}, \texttt{ff}) \mapsto \texttt{tt}} \qquad \frac{e_2 \mapsto e_2'}{\overline{V}_{II}(e_1, e_2) \mapsto \overline{V}_I(e_1, e_2')}$$

$$\frac{}{\overline{V}_{II}(e_1, \texttt{tt}) \mapsto \texttt{ff}} \qquad \frac{e_1 \mapsto e_1'}{\overline{V}_{II}(e_1, \texttt{ff}) \mapsto \overline{V}_I(e_1', \texttt{ff})} \qquad \frac{}{\overline{V}_{II}(\texttt{tt}, \texttt{ff}) \mapsto \texttt{ff}}$$

$$\frac{}{\overline{V}_{II}(\texttt{ff}, \texttt{ff}) \mapsto \texttt{tt}}$$

**Preservation** If $e$ `circ` and $e \mapsto e'$, then $e'$ `circ`.

**Progression** If $e$ `circ` then $e$ val or exists $e'$, $e \mapsto e'$.

Progress-and-preservation denotes safety or coherence. It allows no ill-defined situations yet divergence is OK.

## 7 Exercises

Exercises:.

- Give an alternate formulation of the circuit language using BDD's (binary decision diagrams). The abstract syntax for this language is given by,

$$e ::= \ x \mid \texttt{tt} \mid \texttt{ff} \mid \texttt{if}(e; e; e)$$

- Prove that $e \Downarrow v$ if and only if $e \mapsto^* v$, where $\mapsto^*$ is a multi-step transition. We have the additional rules,

$$\frac{}{v \mapsto^* v} \qquad\qquad \frac{e \mapsto e' \qquad e' \mapsto^* v}{e \mapsto^* v}$$

- Using structural operational semantics, calculate the number of steps it takes for $RSL(\texttt{ff}, \texttt{tt})$ to transition to $\texttt{tt}$