Techinical Setting for **X** Language

Apprentice

Jan. 2000

Contents

1 syntax **2**

2 type system **3**

3 state **4**

4 semantics **5**

References **6**

1 syntax

We can see the grammar of X language in table 1. It's written in BNF[1].

$(Integer)\ n$	$\in$	Int
$(Bool)\ b$	$\in$	$\{\texttt{true}, \texttt{false}\}$
$passive\_operation$	$::=$	$\texttt{int}/n(-;-)$
		$\texttt{bool}/b(-;-)$
		$\texttt{func}/n(v_1 \ldots v_n.e)$
$active\_operation$	$::=$	int_arith
		$bool\_arith$

Table 1: Syntax

Describe the scope of variable binding informally...

And we can write a sample program like 1.

```
&&(
    >(2, 1);
    _.||(!=(0, 1), false)
)
```

Listing 1: Sample Code

If you like, you can have 中文.

2 type system

We have a sample type system, in table 2.

$$\frac{}{\vdash \mathit{Int} : \mathit{int}} \qquad \frac{}{\vdash \mathit{Real} : \mathit{real}} \qquad \frac{\vdash e : \tau}{\vdash \texttt{(}e\texttt{)} : \tau}$$

$$\frac{\vdash e_1 : \tau \qquad \vdash e_2 : \tau}{\vdash e_1 \ \mathit{op} \ e_2 : \tau}$$

Table 2: Sample Type System

3 state

(*Value*) w, r	$\in$	$\mathbb{O}_{\checkmark}$
(*LVMap*) s	$\in$	$Nat \rightharpoonup Term$
(*RefMap*) h	$\in$	$Nat \rightharpoonup Term$
(*BindState*) σ	$::=$	(s, h)
(*Tag-Block*) T_b	$\in$	$\{\texttt{ok}, \texttt{break}, \texttt{continue}\}$
(*BlockState*) γ	$::=$	$(T_b, \texttt{i}_L, r)$
(*State*) $\mathcal{P}$	$::=$	(σ, γ, δ)

Table 3: State Design

(*VarAccess*) $\texttt{i}_v \mapsto_s t$	$::=$	$s(\texttt{i}_v) = t$
(*ReferenceAccess*) $\texttt{i}_a \mapsto_h v$	$::=$	$h(\texttt{i}_a) = v$
(*MapUpdate*) $s\{k \rightsquigarrow v\}$	$::=$	$\lambda z. \begin{cases} s(z), & \text{if } z \neq k; \\ v, & \text{if } z = k. \end{cases}$

Table 4: Notations

4 semantics

Eval-Loc-Var

$$\frac{\mathtt{i}_v \mapsto_s t}{(\mathtt{i}_v, (s,h), -, -, -) \longrightarrow (t, (s,h), -, -, -)}$$

Bind-Var

$$\frac{}{(\mathtt{bind}\ x = t\ \mathtt{in}\ e, \sigma, (\vec{x}, \vec{t^x}, \vec{a}, \vec{t^a}), -, -) \longrightarrow (e, \sigma, (\vec{x} :: x, \vec{t^x} :: t, \vec{a}, \vec{t^a}), -, -)}$$

Bind-Name

$$\frac{}{(\mathtt{new}\ x = t\ \mathtt{in}\ e, \sigma, (\vec{x}, \vec{t^x}, \vec{a}, \vec{t^a}), -, -) \longrightarrow (e, \sigma, (\vec{x}, \vec{t^x}, \vec{a} :: x, \vec{t^a} :: t), -, -)}$$

Bind-OK

$$\frac{\begin{array}{c} k_x = |\vec{x}| \quad k_a = |\vec{a}| \quad k_x > 0 \vee k_a > 0 \\ \forall i \in \{1, \ldots, k_x\}\ \mathtt{i}_{vi} = new\_loc_v() \qquad \forall j \in \{1, \ldots, k_a\}\ \mathtt{i}_{aj} = new\_loc_a() \\ \forall i \in \{1, \ldots, k_x\}\ t_i^{x\prime} = t_i^x[\mathtt{i}_{v1}/x_1] \ldots [\mathtt{i}_{vk_x}/x_i][\mathtt{i}_{a1}/a_1] \ldots [\mathtt{i}_{ak_a}/a_{k_a}] \\ \forall j \in \{1, \ldots, k_a\}\ t_j^{a\prime} = t_j^a[\mathtt{i}_{v1}/x_1] \ldots [\mathtt{i}_{vk_x}/x_{k_x}][\mathtt{i}_{a1}/a_1] \ldots [\mathtt{i}_{ak_a}/a_{k_a}] \\ t' = t[\mathtt{i}_{v1}/x_i] \ldots [\mathtt{i}_{vk_x}/x_i][\mathtt{i}_{a1}/a_1] \ldots [\mathtt{i}_{ak_a}/a_{k_a}] \\ s' = s\{\mathtt{i}_{v1} \rightsquigarrow t_1^{x\prime}, \ldots, \mathtt{i}_{vk_x} \rightsquigarrow t_{k_x}^{x\prime}\} \qquad h' = h\{\mathtt{i}_{a1} \rightsquigarrow t_1^{a\prime}, \ldots, \mathtt{i}_{ak_a} \rightsquigarrow t_{k_a}^{a\prime}\} \end{array}}{(t, (s,h), (\vec{x}, \vec{t^x}, \vec{a}, \vec{t^a}), -, -) \longrightarrow (t', (s', h'), -, -, -)}$$

Eval-Thunk-Apply

$$\frac{n = |\vec{x}| \quad |\vec{x}| = |\vec{r}| \quad \forall\ i \in \{1, \ldots, n\} \quad \mathtt{i}_{vi} = new\_loc_v()}{\begin{array}{c} (\vec{x}(\vec{r}).e, (s,h), -, -, -) \longrightarrow \\ (e[\mathtt{i}_{v1}/x_1] \ldots [\mathtt{i}_{vn}/x_n], (s\{\mathtt{i}_{v1} \rightsquigarrow r_1, \ldots, \mathtt{i}_{vn} \rightsquigarrow r_n\}, h), -, -, -) \end{array}}$$

Table 5: Operational Semantics for Spartan

References

[1] Daniel D. McCracken and Edwin D. Reilly. “Backus-Naur Form (BNF)”. In: *Encyclopedia of Computer Science*. GBR: John Wiley and Sons Ltd., 2003, pp. 129∼131. ISBN: 0470864125.