TYPE SAFETY OF IF+/

- semantic domains of values.
- syntax of IF+/
- rewrite rules.
- reduction
- normal forms VAL, DIV/0, ATM.
- partition theorm for normal forms.
- preservation theorm for VAL*, DIV/0*, ATM*.
- DESTINY thm.
- well typing
- type preservation thm
- type progress thm

$\left.\begin{array}{l}\text{type preservation thm}\\ \text{type progress thm}\end{array}\right\}$ type safety

TYPE SAFETY

semantic domains.

$n \in NUM$

$b \in BOOL = \{true, false\}$

$v \in VAL$

$v ::= \hat{n} \mid \hat{b}$

$+ : [NUM, NUM] \longrightarrow NUM$

$/ : [NUM, NUM] \longrightarrow NUM$

SYNTAX OF IF+/

$e ::= \bar{n}$

$\quad \bar{b}$

$\quad e \oplus e$

$\quad e \oslash e$

$\quad \text{(if)}\ e\ e\ e$

$\boxed{e\ \text{EXP}}$

LITERAL EXPRESSIONS

$$\frac{}{v\ \text{VAL}}\ \text{VAL}$$

$\boxed{e\ \text{VAL}}$

$\boxed{\vdash_{\text{VAL}}\ e\ \text{VAL}}$

SEMANTICS

rewrite rules

$\boxed{e \hookrightarrow e'}$

$\bar{n_1} + \bar{n_2} \hookrightarrow \overline{n_1 + n_2}$ PLUS

$\bar{n_1} / \bar{n_2} \hookrightarrow \overline{n_1 / n_2}$ DIV provided $n_2 \neq 0$

$\text{if true } e_2\ e_3 \hookrightarrow e_2$ IF-TRUE

$\text{if false } e_2\ e_3 \hookrightarrow e_3$ IF-FALSE

SEMANTICS

reduction system $\boxed{e \to e'}$

$$\frac{e \hookrightarrow e'}{e \to e'} \quad \text{REW}$$

$$\frac{e_1 \longrightarrow e_1'}{e_1 \oplus e_2 \longrightarrow e_1' + e_2} \quad \text{PLUS-LEFT}$$

$$\frac{e_1 \text{ VAL} \qquad e_2 \to e_2'}{e_1 \oplus e_2 \longrightarrow e_1 + e_2'} \quad \text{PLUS-RIGHT}$$

notes> reduction is deterministic
left operand reduced first

$$\frac{e_1 \longrightarrow e_1'}{e_1 \oslash e_2 \longrightarrow e_1' \oslash e_2} \quad \text{DIV - LEFT}$$

$$\frac{e_1 \text{ VAL} \qquad e_2 \to e_2'}{e_1 \oslash e_2 \to e_1 \oslash e_2'} \quad \text{DIV-RIGHT}$$

$$\frac{e_1 \to e_1'}{\textcircled{if}\ e_1\ e_2\ e_3 \to \textcircled{if}\ e_1'\ e_2\ e_3} \quad \text{IF-TEST}$$

DIV BY ZERO ERROR

$$\frac{}{\bar{n} \oslash 0 \quad \text{DIV/0}} \ \textcircled{0} \qquad op \in \{\oplus, \oslash\}$$

$$\frac{e_1\ \text{VAL} \qquad e_2\ \text{DIV/0}}{e_1\ \textcircled{op}\ e_2 \quad \text{DIV/0}}\ \text{RIGHT}$$

$$\frac{e_1\ \text{DIV/0}}{e_1\ \textcircled{op}\ e_2 \quad \text{DIV/0}}\ \text{LEFT}$$

$$\frac{e_1\ \text{DIV/0}}{\textcircled{if}\ e_1\ e_2\ e_3 \quad \text{DIV/0}}\ \text{IF}$$

$$\boxed{e\ \ \text{DIV/0}} \qquad \vdash_{\text{DIV/0}} e\ \text{DIV/0}$$

"stuck at"

divide by ZERO

$\vdash e\ \text{DIV/0}$ means ① $e \not\to$

② reason is DIV/0

ex if (4 + 3/0)/2 7+8 false DIV/0

① 3/0 DIV/0 (0)

② (4 + 3/0) DIV/0 (RIGHT)

③ (4 + 3/0)/2 DIV/0 (LEFT)

④ if (4+3/0)/2 7+8 false DIV/0 (IF)

ex a DIV/0 can occor moltple times.

4/0 + 3/0 DIV/0

① 4/0 DIV/0 (0)

② 4/0 + 3/0 DIV/0 (LEFT)

ex | DIV/0 false + 3/0

① 3/0 DIV/0 (0)

② false $\overline{\text{VAL}}$ VAL:VAL ??

③ false + 3/0 DIV/0 (RIGHT)

$$\frac{e' \text{ DIV/0} \quad e \xrightarrow{*} e'}{e \text{ DIV/0*}} \qquad \boxed{e \text{ DIV/0*}}$$

$$\vdash_{\text{DIV/0*}} e \text{ DIV/0*}$$

e simplifies to an expression stuck due to DIV/0 error.

$$\boxed{e \text{ ATM}} \qquad \vdash_{\text{ATM}} e \text{ ATM}$$

arg type mismatch

$$\frac{}{\overline{b}_1 \textcircled{\text{op}} \overline{n}_2 \text{ ATM}} \text{ BOOL-NOM}$$

$$\frac{}{\overline{n}_1 \textcircled{\text{op}} \overline{b}_2 \text{ ATM}} \text{ NUM-BOOL}$$

$$\frac{}{\overline{b}_1 \textcircled{\text{op}} \overline{b}_2 \text{ ATM}} \text{ BOOL-BOOL}$$

$$\frac{e_1 \text{ ATM}}{e_1 \textcircled{\text{op}} e_2 \text{ ATM}} \text{ LEFT}$$

$$\frac{e_2 \text{ ATM}}{\overline{v}_1 \textcircled{\text{op}} e_2 \text{ ATM}} \text{ RIGHT}$$

$$\frac{}{\text{if } \bar{n}_1 \; e_2 \; e_3 \; \text{ATM}} \; \text{IF}$$

$\vdash_{I} e \; \text{ATM}$

intuitively, e is stuck due to argument type mismatch.

$\boxed{e \; \text{ATM*}}$ $\quad$ $\vdash_{\text{ATM*}} e \; \text{ATM*}$

$$\frac{e' \; \text{ATM} \qquad e \xrightarrow{*} e'}{e \; \text{ATM*}} \; *$$

rule

simplification of e results in an expression that is stuck due to an ARG TYPE MISMATCH.

ex true/0 + 3/0

(1) true/0 ATM (BOOL-NUM)

(2) true/0 + 3/0 ATM (LEFT)

$\therefore \vdash_{\text{ATM}}$ true/0 + 3/0 ATM

normal form	description
VAL	values
DIV/0	stuck at DIV/0
ATM	stuck due to arg type mismatch

e VAL $\Rightarrow$ $e \not\rightarrow$

e DIV/0 $\Rightarrow$ $e \not\rightarrow$

e ATM $\Rightarrow$ $e \not\rightarrow$

SPANNING LEMMA

if $e \not\rightarrow$, ① e VAL, or

② e DIV/0, or

③ e ATM