

asm.js

Dave Herman, Luke Wagner, and Alon Zakai

November 6, 2012

1 Abstract syntax

$b, e, f, g, x, y, z \in \text{Identifier}$
 $\text{arguments}, \text{eval} \notin \text{Identifier}$

$P ::= \text{function}(b, e) \{ \overline{\text{imp}_x} \overline{\text{fn}_f} \text{exp} \}$
 $\text{imp}_x ::= \text{var } x = e.y;$
 $|\text{ var } x = \text{new } e.y(b);$
 $\text{exp} ::= \text{return } f;$
 $|\text{ return } \{ \overline{x:f} \};$
 $\text{fn}_f ::= \text{function } f(\overline{x}) \{ \overline{x = \kappa_x}; \text{var } \overline{y = v}; ss \}$

$s ::= \{ ss \}$
 $|\text{ } e;$
 $|\text{ } ;$
 $|\text{ if } (e) \text{ } s$
 $|\text{ if } (e) \text{ } s \text{ else } s$
 $|\text{ return } e;$
 $|\text{ while } (e) \text{ } s$
 $|\text{ do } s \text{ while } (e);$
 $|\text{ for } (e^?; e^?; e^?) \text{ } s$
 $|\text{ switch } (e) \{ \overline{c} \}$
 $|\text{ switch } (e) \{ \overline{c} \text{ } d \}$
 $|\text{ break};$
 $|\text{ break } lab;$
 $|\text{ continue};$
 $|\text{ continue } lab;$
 $|\text{ lab: } s$

$ss ::= \overline{s}$

$c ::= \text{case } e: ss$
 $d ::= \text{default: } ss$
 $cd ::= c \mid d$

$$\kappa_x ::= \sim\sim x \mid +x \mid x|0 \mid x>>>0$$

$$v ::= r \mid n$$

$$e ::= \begin{array}{l} v \\ lval \\ lval = e \\ f(\bar{e}) \\ unop\ e \\ e\ binop\ e \\ e\ ?\ e : e \\ (\bar{e}) \end{array}$$

$$unop ::= + \mid \sim \mid !$$

$$\begin{array}{l} binop ::= + \mid - \mid * \mid / \mid \% \\ \quad \mid \mid \& \mid \wedge \mid << \mid >> \mid >>> \\ \quad \mid < \mid <= \mid > \mid >= \mid != \mid == \\ lval ::= x \mid x[(e\ \&\ m)\ >>\ n] \end{array}$$

2 Type rules

$$\sigma, \tau ::= \text{bit} \mid \text{double} \mid \text{int} \mid \text{signed} \mid \text{unsigned} \mid \text{boolish} \mid \text{intish} \mid \text{void} \mid \text{unknown}$$

$$\rho ::= \tau \mid \text{array}_\tau^n \mid \text{imul} \mid ((\bar{\sigma}) \rightarrow \tau) \wedge \dots \wedge ((\bar{\sigma}') \rightarrow \tau') \mid \text{function}$$

$$\begin{array}{l} \ell ::= lab \mid \epsilon \\ L ::= \{\bar{\ell}\} \\ \varepsilon ::= L \mid \text{return} \end{array}$$

$$\begin{array}{l} L ; L' = L \cup L' \\ \emptyset ; \text{return} = \text{return} \\ \{\ell, \bar{\ell}'\} ; \text{return} = \{\ell, \bar{\ell}'\} \\ \text{return} ; L = \text{return} \end{array}$$

$$\begin{array}{l} L \cup \text{return} = L \\ \text{return} \cup L = L \\ \text{return} \cup \text{return} = \text{return} \end{array}$$

$type(\sim\sim X) = \text{int}$
 $type(+X) = \text{double}$
 $type(n) = \text{int}$
 $type(r) = \text{double}$
 $type(X|0) = \text{signed}$
 $type(X>>>0) = \text{unsigned}$

$\text{constant} <: \text{signed, unsigned}$
 $\text{signed, unsigned} <: \text{int, extern}$
 $\text{bit, int} <: \text{boolish}$
 $\text{double} <: \text{extern}$
 $\text{unknown, int} <: \text{intish}$
 $((\bar{\sigma}_1) \rightarrow \tau_1) \wedge \dots \wedge ((\bar{\sigma}_n) \rightarrow \tau_n) <: ((\bar{\sigma}_1) \rightarrow \tau_1) \wedge \dots \wedge ((\bar{\sigma}_{n-1}) \rightarrow \tau_{n-1})$
 $\text{imul} <: (\text{intish, intish}) \rightarrow \text{signed}$

$M(\text{imul}) : \text{imul}$
 $M(\text{ceil}), M(\text{sin}), M(\text{cos}) : (\text{double}) \rightarrow \text{double}$

$A(\text{Uint8Array}), A(\text{Int8Array}) = \text{array}_{\text{int}}^8$
 $A(\text{Uint16Array}), A(\text{Int16Array}) = \text{array}_{\text{int}}^{16}$
 $A(\text{Uint32Array}), A(\text{Int32Array}) = \text{array}_{\text{int}}^{32}$
 $A(\text{Float32Array}) = \text{array}_{\text{double}}^{32}$
 $A(\text{Float64Array}) = \text{array}_{\text{double}}^{64}$

$+, - : (\text{double, double}) \rightarrow \text{double}$
 $\wedge (\text{int, int}) \rightarrow \text{intish}$
 $* : (\text{double, double}) \rightarrow \text{double}$
 $/, \% : (\text{double, double}) \rightarrow \text{double}$
 $\wedge (\text{signed, signed}) \rightarrow \text{intish}$
 $\wedge (\text{unsigned, unsigned}) \rightarrow \text{intish}$

$|, \&, \wedge, <<, >> : (\text{intish, intish}) \rightarrow \text{signed}$
 $>>> : (\text{intish, intish}) \rightarrow \text{unsigned}$

$<, <=, >, >=, ==, != : (\text{signed, signed}) \rightarrow \text{bit}$
 $\wedge (\text{unsigned, unsigned}) \rightarrow \text{bit}$
 $\wedge (\text{double, double}) \rightarrow \text{bit}$

$+ : (\text{intish}) \rightarrow \text{double}$
 $\sim : (\text{intish}) \rightarrow \text{signed}$
 $! : (\text{boolish}) \rightarrow \text{bit}$

$$\Gamma ::= \{\bar{x} : \bar{\rho}\} \mid \Gamma, \{\bar{x} : \bar{\rho}\}$$

Program checking

$\vdash P \text{ ok}$

$$\frac{[\text{T-PROGRAM}] \quad \begin{array}{c} \{\bar{x}\} \cap \{\bar{f}\} = \emptyset \quad \{\bar{x}\} \cap \{b, e\} = \emptyset \quad \{\bar{f}\} \cap \{b, e\} = \emptyset \\ \forall i. b; e; \Gamma_0 \vdash \text{imp}_x \text{ ok} \quad \forall i. \Gamma_0, \Gamma_1 \vdash \text{fn}_f \text{ ok} \quad \forall i. \Gamma_0, \Gamma_1 \vdash \text{exp} \text{ ok} \end{array}}{\vdash \text{function}(b, e) \{ \overline{\text{imp}_x} \overline{\text{fn}_f} \overline{\text{exp}} \} \text{ ok}}$$

Import checking

$b; e; \Gamma \vdash \text{imp} \text{ ok}$

$$\begin{array}{c} [\text{T-IMPORTSTD}] \quad \frac{\Gamma(x) = M(y)}{b; e; \Gamma \vdash \text{var } x = e.y; \text{ ok}} \quad [\text{T-IMPORTFFI}] \quad \frac{y \notin \text{dom}(M) \quad \Gamma(x) = \text{function}}{b; e; \Gamma \vdash \text{var } x = e.y; \text{ ok}} \\ \\ [\text{T-VIEW}] \quad \frac{\Gamma(x) = \text{array}_{A(y)}^n}{b; e; \Gamma \vdash \text{var } x = e.y(b); \text{ ok}} \quad [\text{T-NEVIEW}] \quad \frac{\Gamma(x) = \text{array}_{A(y)}^n}{b; e; \Gamma \vdash \text{var } x = \text{new } e.y(b); \text{ ok}} \end{array}$$

Function checking

$\Gamma \vdash \text{fn} \text{ ok}$

$$\begin{array}{c} [\text{T-FUNCTION}] \quad \frac{\begin{array}{c} \{\bar{x}\} \cap \{\bar{y}\} = \emptyset \quad \Gamma(f) = (\bar{\sigma}) \rightarrow \tau \quad \bar{\sigma} = \overline{\text{type}(\kappa_x)} \quad \tau \neq \text{void} \\ \Gamma, \{\bar{x} : \bar{\sigma}, y : \text{type}(v)\}; \emptyset \vdash ss : \tau / \text{return} \end{array}}{\Gamma \vdash \text{function } f(\bar{x}) \{ \overline{x = \kappa_x}; \overline{\text{var } y = v}; \overline{ss} \} \text{ ok}} \\ \\ [\text{T-VOIDFUNCTION}] \quad \frac{\begin{array}{c} \{\bar{x}\} \cap \{\bar{y}\} = \emptyset \quad \Gamma(f) = (\bar{\sigma}) \rightarrow \text{void} \quad \bar{\sigma} = \overline{\text{type}(\kappa_x)} \\ \Gamma, \{\bar{x} : \bar{\sigma}, y : \text{type}(v)\}; \emptyset \vdash ss : \text{void} / \varepsilon \end{array}}{\Gamma \vdash \text{function } f(\bar{x}) \{ \overline{x = \kappa_x}; \overline{\text{var } y = v}; \overline{ss} \} \text{ ok}} \end{array}$$

Export checking

$\Gamma \vdash \text{exp} \text{ ok}$

$$\begin{array}{c} [\text{T-SINGLETON}] \quad \frac{\Gamma(f) = (\bar{\sigma}) \rightarrow \tau \quad \tau <: \text{extern}}{\Gamma \vdash \text{return } f; \text{ ok}} \quad [\text{T-MODULE}] \quad \frac{\forall f. (\Gamma(f) = (\bar{\sigma}) \rightarrow \tau \wedge \tau <: \text{extern})}{\Gamma \vdash \text{return } \{ \overline{x : f} \}; \text{ ok}} \end{array}$$

Statement list checking

$$\boxed{\Gamma; L \vdash ss : \tau/\varepsilon}$$

$$\frac{[T\text{-NoStatements}] \quad \Gamma; L \vdash \epsilon : \tau/\emptyset}{\Gamma; L \vdash \epsilon : \tau/\emptyset} \quad \frac{[T\text{-Statements}] \quad \begin{array}{l} \forall i. \Gamma; L \vdash s_i : \tau/\varepsilon_i \\ n > 0 \quad \varepsilon = \varepsilon_1 ; \dots ; \varepsilon_n \end{array}}{\Gamma; L \vdash \bar{s} : \tau/\varepsilon}$$

Statement checking

$$\boxed{\Gamma; L \vdash s : \tau/\varepsilon}$$

$$\begin{array}{c} [T\text{-Block}] \quad \frac{\Gamma; \emptyset \vdash ss : \tau/\varepsilon}{\Gamma; L \vdash \{ ss \} : \tau/\varepsilon} \quad [T\text{-ExprStmt}] \quad \frac{\Gamma \vdash e : \sigma}{\Gamma; L \vdash e ; : \tau/\emptyset} \\ \\ [T\text{-If}] \quad \frac{\begin{array}{l} \Gamma \vdash e : \text{boolish} \\ \Gamma; \emptyset \vdash s : \tau/\varepsilon \\ \varepsilon' = \varepsilon \cup \emptyset \end{array}}{\Gamma; L \vdash \text{if } (e) \text{ } s : \tau/\varepsilon'} \quad [T\text{-IfElse}] \quad \frac{\begin{array}{l} \Gamma \vdash e : \text{boolish} \\ \Gamma; \emptyset \vdash s_1 : \tau/\varepsilon_1 \quad \Gamma; \emptyset \vdash s_2 : \tau/\varepsilon_2 \\ \varepsilon = \varepsilon_1 \cup \varepsilon_2 \end{array}}{\Gamma; L \vdash \text{if } (e) \text{ } s_1 \text{ else } s_2 : \tau/\varepsilon} \\ \\ [T\text{-ReturnExpr}] \quad \frac{\text{type}(e) <: \tau \quad \Gamma \vdash e : \tau}{\Gamma; L \vdash \text{return } e ; : \tau/\text{return}} \quad [T\text{-ReturnVoid}] \quad \frac{}{\Gamma; L \vdash \text{return}; : \text{void}/\text{return}} \\ \\ [T\text{-While}] \quad \frac{\begin{array}{l} \Gamma \vdash e : \text{boolish} \\ \Gamma; L \cup \{\epsilon\} \vdash s : \tau/\varepsilon \\ \varepsilon' = \emptyset \cup \varepsilon - (L \cup \{\epsilon\}) \end{array}}{\Gamma; L \vdash \text{while } (e) \text{ } s : \tau/\varepsilon'} \quad [T\text{-DoWhile}] \quad \frac{\begin{array}{l} \Gamma; L \cup \{\epsilon\} \vdash s : \tau/\varepsilon \\ \Gamma \vdash e : \text{boolish} \\ \varepsilon' = \varepsilon - (L \cup \{\epsilon\}) \end{array}}{\Gamma; L \vdash \text{do } s \text{ while } (e); : \tau/\varepsilon'} \\ \\ [T\text{-For}] \quad \frac{\begin{array}{l} e_1^? = \varepsilon \vee \Gamma \vdash e_1^? : \sigma_1 \quad e_2^? = \varepsilon \vee \Gamma \vdash e_2^? : \text{boolish} \quad e_3^? = \varepsilon \vee \Gamma \vdash e_3^? : \sigma_3 \\ \Gamma; L \cup \{\epsilon\} \vdash s : \tau/\varepsilon \quad \varepsilon' = \emptyset \cup \varepsilon - (L \cup \{\epsilon\}) \end{array}}{\Gamma; L \vdash \text{for } (e_1; e_2; e_3) \text{ } s : \tau/\varepsilon'}$$

Statement checking (cont'd)

$$\boxed{\Gamma; L \vdash s : \tau/\varepsilon}$$

$$\begin{array}{c} \text{[T-BREAK]} \\ \hline \Gamma; L \vdash \mathbf{break}; : \tau/\{\epsilon\} \end{array} \quad \begin{array}{c} \text{[T-BREAKLABEL]} \\ \hline \Gamma; L \vdash \mathbf{break} \text{ } lab; : \tau/\{lab\} \end{array}$$

$$\begin{array}{c} \text{[T-CONTINUE]} \\ \hline \Gamma; L \vdash \mathbf{continue}; : \tau/\emptyset \end{array} \quad \begin{array}{c} \text{[T-CONTINUELABEL]} \\ \hline \Gamma; L \vdash \mathbf{continue} \text{ } lab; : \tau/\emptyset \end{array}$$

$$\begin{array}{c} \text{[T-LABEL]} \\ \Gamma; L \cup \{lab\} \vdash s : \tau/\varepsilon \\ \varepsilon' = \varepsilon - (L \cup \{lab\}) \\ \hline \Gamma; L \vdash lab : s : \tau/\varepsilon' \end{array}$$

[T-SWITCH]

$$\begin{array}{c} \Gamma \vdash e : \sigma \\ \forall i. \Gamma; L \cup \{\epsilon\} \vdash c_i : \sigma, \tau/\varepsilon_i \\ \Gamma; L \cup \{\epsilon\} \vdash cd : \sigma, \tau/\varepsilon \\ \varepsilon \neq \mathbf{return} \vee \exists i. \varepsilon_i \cup \emptyset \neq \emptyset \\ \varepsilon' = (\varepsilon \cup \bigcup_i \varepsilon_i) - (L \cup \{\epsilon\}) \\ \hline \Gamma; L \vdash \mathbf{switch} \text{ } (e) \text{ } \{ \bar{c} \text{ } cd \} : \tau/\varepsilon' \end{array} \quad \begin{array}{c} \text{[T-SWITCHRETURN]} \\ \Gamma \vdash e : \sigma \\ \forall i. \Gamma; L \cup \{\epsilon\} \vdash c_i : \sigma, \tau/\varepsilon_i \\ \forall i. \varepsilon_i \cup \emptyset = \emptyset \\ \Gamma; L \cup \{\epsilon\} \vdash cd : \sigma, \tau/\mathbf{return} \\ \hline \Gamma; L \vdash \mathbf{switch} \text{ } (e) \text{ } \{ \bar{c} \text{ } cd \} : \tau/\mathbf{return} \end{array}$$

[T-EMPTYSWITCH]

$$\frac{\Gamma \vdash e : \sigma}{\Gamma; L \vdash \mathbf{switch} \text{ } (e) \text{ } \{ \} : \tau/\emptyset}$$

[T-EMPTYSTATEMENT]

$$\frac{}{\Gamma; L \vdash ; : \tau/\emptyset}$$

Case checking

$$\boxed{\Gamma; L \vdash cd : \sigma, \tau/\varepsilon}$$

[T-CASE]

$$\frac{\Gamma \vdash e : \sigma \quad \Gamma; L \vdash ss : \tau/\varepsilon}{\Gamma; L \vdash \mathbf{case} \text{ } e : ss : \sigma, \tau/\varepsilon}$$

[T-DEFAULT]

$$\frac{\Gamma; L \vdash ss : \tau/\varepsilon}{\Gamma; L \vdash \mathbf{default} : ss : \sigma, \tau/\varepsilon}$$

Expression checking

$\boxed{\Gamma \vdash e : \tau}$

$\frac{[T\text{-CONSTANT}] \quad -2^{31} \leq n < 2^{32}}{\Gamma \vdash n : \mathbf{constant}}$	$\frac{[T\text{-DOUBLE}]}{\Gamma \vdash r : \mathbf{double}}$
$\frac{[T\text{-VARREF}] \quad \Gamma(x) = \tau}{\Gamma \vdash x : \tau}$	$\frac{[T\text{-ASSIGN}] \quad \Gamma \vdash e : \Gamma(x)}{\Gamma \vdash x = e : \tau}$
$\frac{[T\text{-LOAD}] \quad \begin{array}{c} m = 2^k - 1 \\ \Gamma(x) = \mathbf{array}_{\tau}^n \quad \Gamma \vdash e : \mathbf{int} \end{array}}{\Gamma \vdash x[(e \ \& \ m) \gg n/8] : \tau}$	$\frac{[T\text{-STORE}] \quad \begin{array}{c} m = 2^k - 1 \quad \Gamma(x) = \mathbf{array}_{\tau}^n \\ \Gamma \vdash e_1 : \mathbf{int} \quad \Gamma \vdash e_2 : \tau \end{array}}{\Gamma \vdash x[(e_1 \ \& \ m) \gg n/8] = e_2 : \tau}$
$\frac{[T\text{-FUNCALL}] \quad \begin{array}{c} \Gamma(f) = (\bar{\sigma}) \rightarrow \tau \\ \forall i. \Gamma \vdash e_i : \sigma_i \end{array}}{\Gamma \vdash f(\bar{e}) : \tau}$	$\frac{[T\text{-FFI}] \quad \begin{array}{c} \Gamma(f) = \mathbf{function} \\ \forall i. \Gamma \vdash e_i : \mathbf{extern} \end{array}}{\Gamma \vdash f(\bar{e}) : \mathbf{unknown}}$
$\frac{[T\text{-CONDITIONAL}] \quad \begin{array}{c} \Gamma \vdash e_1 : \mathbf{boolish} \\ \Gamma \vdash e_2 : \tau \quad \Gamma \vdash e_3 : \tau \end{array}}{\Gamma \vdash e_1 ? e_2 : e_3 : \tau}$	$\frac{[T\text{-PAREN}] \quad \forall i \leq n. \Gamma \vdash e_i : \tau_i}{\Gamma \vdash (\bar{e}) : \tau_n}$
$\frac{[T\text{-UNOP}] \quad \begin{array}{c} \mathit{unop} : _ \wedge (\sigma) \rightarrow \tau \wedge _ \quad \Gamma \vdash e : \sigma \end{array}}{\Gamma \vdash \mathit{unop} \ e : \tau}$	$\frac{[T\text{-BINOP}] \quad \begin{array}{c} \mathit{binop} : _ \wedge (\sigma_1, \sigma_2) \rightarrow \tau \wedge _ \\ \Gamma \vdash e_1 : \sigma_1 \quad \Gamma \vdash e_2 : \sigma_2 \end{array}}{\Gamma \vdash e_1 \ \mathit{binop} \ e_2 : \tau}$
$\frac{[T\text{-SUB}] \quad \Gamma \vdash e : \sigma \quad \sigma <: \tau}{\Gamma \vdash e : \tau}$	$\frac{[T\text{-CAST}] \quad \Gamma \vdash e : \mathbf{double}}{\Gamma \vdash \sim\sim e : \mathbf{signed}}$