

# asm.js

Dave Herman

October 19, 2012

## 1 Abstract syntax

$$\begin{aligned}\rho &::= \textit{runtime lexical environment} \\ \Delta &::= \{f : (\overline{\sigma}) \rightarrow \tau\} \\ \Gamma &::= \{x : \tau\}\end{aligned}$$
$$\begin{aligned}\textit{capsule} &::= \text{function } f(\overline{x}) \{ \overline{fn} \text{ return } f; \} \\ &\quad | \text{function } f(\overline{x}) \{ \overline{fn} \text{ return } \{ \overline{x:f} \}; \} \\ \textit{fn} &::= \text{function } f(\overline{x}) \{ \overline{x} = \kappa_x; \text{ var } \overline{y} = \overline{v}; \text{ ss } \}\end{aligned}$$
$$\begin{aligned}s &::= \{ \text{ ss } \} \\ &\quad | e; \\ &\quad | \text{ if } (e) \text{ } s \\ &\quad | \text{ if } (e) \text{ } s \text{ else } s \\ &\quad | \text{ return } v; \\ &\quad | \text{ while } (e) \text{ } s \\ &\quad | \text{ do } s \text{ while } (e); \\ &\quad | \text{ for } (e; e; e) \text{ } s \\ &\quad | \text{ switch } (e) \{ \overline{c} \} \\ &\quad | \text{ switch } (e) \{ \overline{c} \text{ } d \} \\ &\quad | \text{ break;} \\ &\quad | \text{ break } \textit{lab}; \\ &\quad | \text{ continue;} \\ &\quad | \text{ continue } \textit{lab}; \\ &\quad | \textit{lab} : s\end{aligned}$$
$$\textit{ss} ::= \overline{s}$$
$$\begin{aligned}c &::= \text{case } e : \textit{ss} \\ d &::= \text{default} : \textit{ss} \\ cd &::= c \mid d\end{aligned}$$

$$\kappa_X ::= \begin{array}{l} X \mid 0 \\ X \mid 0\text{xff} \\ X \mid 0\text{xffff} \\ X \ggg 0 \\ +X \\ X + ' ' \\ X + "" \\ !!X \end{array}$$

$$v ::= \begin{array}{l} \kappa_{num} \\ bool \\ str \\ null \end{array}$$

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

$$unop ::= \text{void} \mid \sim \mid - \mid !$$

$$binop ::= \begin{array}{l} < \mid <= \mid > \mid >= \mid == \\ + \mid - \mid * \mid / \mid \% \\ \mid \mid \& \mid \sim \\ << \mid >> \mid >>> \\ \mid \mid \mid \&\& \end{array}$$

$$lval ::= \begin{array}{l} x \\ x[e] \end{array}$$

## 2 Type rules

$$\sigma, \tau ::= \begin{array}{l} \text{boolean} \\ uint8 \mid uint16 \mid uint32 \\ int8 \mid int16 \mid int32 \\ float32 \mid float64 \\ array_\tau \mid \text{Function} \\ any \mid \text{undefined} \mid null \mid \text{string} \mid \text{number} \mid \text{object} \end{array}$$

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

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

$$\begin{aligned}L \cup \text{return} &= L \\ \text{return} \cup L &= L \\ \text{return} \cup \text{return} &= \text{return}\end{aligned}$$

$$\begin{aligned}type(bool) &= \text{boolean} \\ type(str) &= \text{string} \\ type(null) &= \text{null} \\ type(X \& 0xff) &= \text{int8} \\ type(X \& 0xffff) &= \text{int16} \\ type(X \mid 0) &= \text{int32} \\ type(X >>> 0) &= \text{uint32} \\ type(+X) &= \text{float64} \\ type(X + '') &= \text{string} \\ type(X + "") &= \text{string} \\ type(!!X) &= \text{boolean}\end{aligned}$$

$$\begin{aligned}\text{uint8, uint16, uint32} &<: \text{int} \\ \text{int8, int16, int32} &<: \text{int}\end{aligned}$$

Function checking

$$\boxed{\rho; \Delta \vdash fn \text{ ok}}$$

[T-FUNCTION]

$$\frac{\frac{\rho; \Delta; \{\bar{x} : \sigma, y : type(v)\}; \emptyset \vdash ss : \Delta(f)/return}{\forall i. type(\kappa_{x_i}) = \sigma_i}}{\rho; \Delta \vdash \text{function } f(\bar{x}) \{ \bar{x} = \kappa_{\bar{x}}; \text{var } \bar{y} = \bar{v}; ss \} \text{ ok}}$$

Statement list checking

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

[T-STATEMENTS]

$$\frac{\forall i. \rho; \Delta; \Gamma; L \vdash s_i : \tau/\varepsilon_i \quad \varepsilon = \varepsilon_1 ; \dots ; \varepsilon_n}{\rho; \Delta; \Gamma; L \vdash \bar{s} : \tau/\varepsilon}$$

Statement checking

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

[T-BLOCK]

$$\frac{\rho; \Delta; \Gamma; \emptyset \vdash ss : \tau/\varepsilon}{\rho; \Delta; \Gamma; L \vdash \{ ss \} : \tau/\varepsilon}$$

[T-EXPRSTMT]

$$\frac{\rho; \Delta; \Gamma \vdash e : \sigma}{\rho; \Delta; \Gamma; L \vdash e; : \tau/\emptyset}$$

[T-IF]

$$\frac{\begin{array}{l} \rho; \Delta; \Gamma \vdash e : \text{boolean} \\ \rho; \Delta; \Gamma; \emptyset \vdash s : \tau/\varepsilon \\ \varepsilon' = \varepsilon \cup \emptyset \end{array}}{\rho; \Delta; \Gamma; L \vdash \text{if } (e) \ s : \tau/\varepsilon'}$$

[T-IFELSE]

$$\frac{\begin{array}{l} \rho; \Delta; \Gamma \vdash e : \text{boolean} \\ \rho; \Delta; \Gamma; \emptyset \vdash s_1 : \tau/\varepsilon_1 \quad \rho; \Delta; \Gamma; \emptyset \vdash s_2 : \tau/\varepsilon_2 \\ \varepsilon = \varepsilon_1 \cup \varepsilon_2 \end{array}}{\rho; \Delta; \Gamma; L \vdash \text{if } (e) \ s_1 \text{ else } s_2 : \tau/\varepsilon}$$

[T-RETURN]

$$\frac{}{\rho; \Delta; \Gamma; L \vdash \text{return } v; : type(v)/return}$$

[T-WHILE]

$$\frac{\begin{array}{l} \rho; \Delta; \Gamma \vdash e : \text{boolean} \\ \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash s : \tau/\varepsilon \\ \varepsilon' = \emptyset \cup \varepsilon - (L \cup \{\epsilon\}) \end{array}}{\rho; \Delta; \Gamma; L \vdash \text{while } (e) \ s : \tau/\varepsilon'}$$

[T-DOWHILE]

$$\frac{\begin{array}{l} \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash s : \tau/\varepsilon \\ \rho; \Delta; \Gamma \vdash e : \text{boolean} \\ \varepsilon' = \varepsilon - (L \cup \{\epsilon\}) \end{array}}{\rho; \Delta; \Gamma; L \vdash \text{do } s \text{ while } (e); : \tau/\varepsilon'}$$

[T-FOR]

$$\frac{\begin{array}{l} \forall i \in \{1, 2, 3\}. \rho; \Delta; \Gamma \vdash e_i : \sigma_i \\ \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash s : \tau/\varepsilon \\ \varepsilon' = \emptyset \cup \varepsilon - (L \cup \{\epsilon\}) \end{array}}{\rho; \Delta; \Gamma; L \vdash \text{for } (e_1; e_2; e_3) \ s : \tau/\varepsilon'}$$

Statement checking (cont'd)

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

$$\begin{array}{c}
\begin{array}{c}
\text{[T-BREAK]} \\
\frac{\varepsilon = \{\epsilon\}}{\rho; \Delta; \Gamma; L \vdash \mathbf{break}; : \tau/\varepsilon}
\end{array}
\qquad
\begin{array}{c}
\text{[T-BREAKLABEL]} \\
\frac{\varepsilon = \{lab\}}{\rho; \Delta; \Gamma; L \vdash \mathbf{break} \quad lab; : \tau/\varepsilon}
\end{array}
\\[10pt]
\begin{array}{c}
\text{[T-CONTINUE]} \\
\frac{}{\rho; \Delta; \Gamma; L \vdash \mathbf{continue}; : \tau/\emptyset}
\end{array}
\qquad
\begin{array}{c}
\text{[T-CONTINUELABEL]} \\
\frac{}{\rho; \Delta; \Gamma; L \vdash \mathbf{continue} \quad lab; : \tau/\emptyset}
\end{array}
\\[10pt]
\begin{array}{c}
\text{[T-LABEL]} \\
\frac{\rho; \Delta; \Gamma; L \cup \{lab\} \vdash s : \tau/\varepsilon \quad \varepsilon' = \varepsilon - (L \cup \{lab\})}{\rho; \Delta; \Gamma; L \vdash lab : s : \tau/\varepsilon'}
\end{array}
\\[10pt]
\begin{array}{c}
\text{[T-SWITCH]} \\
\frac{\rho; \Delta; \Gamma \vdash e : \sigma \quad \forall i. \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash c_i : \sigma, \tau/\varepsilon_i \quad \varepsilon = (\bigcup_i \varepsilon_i) - (L \cup \{\epsilon\})}{\rho; \Delta; \Gamma; L \vdash \mathbf{switch} \ (e) \ \{ \ \bar{c} \ \} : \tau/\varepsilon}
\end{array}
\\[10pt]
\begin{array}{c}
\text{[T-SWITCHDEFAULT]} \\
\frac{\rho; \Delta; \Gamma \vdash e : \sigma \quad \forall i. \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash c_i : \sigma, \tau/\varepsilon_i \quad \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash ss : \tau/\varepsilon \quad \varepsilon \neq \mathbf{return} \vee \exists i. \varepsilon_i \cup \emptyset \neq \emptyset \quad \varepsilon' = (\varepsilon \cup \bigcup_i \varepsilon_i) - (L \cup \{\epsilon\})}{\rho; \Delta; \Gamma; L \vdash \mathbf{switch} \ (e) \ \{ \ \bar{c} \ \mathbf{default} : ss \ \} : \tau/\varepsilon'}
\end{array}
\\[10pt]
\begin{array}{c}
\text{[T-SWITCHRETURN]} \\
\frac{\rho; \Delta; \Gamma \vdash e : \sigma \quad \forall i. \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash c_i : \sigma, \tau/\varepsilon_i \quad \forall i. \varepsilon_i \cup \emptyset = \emptyset \quad \rho; \Delta; \Gamma; L \cup \{\epsilon\} \vdash ss : \tau/\mathbf{return}}{\rho; \Delta; \Gamma; L \vdash \mathbf{switch} \ (e) \ \{ \ \bar{c} \ \mathbf{default} : ss \ \} : \tau/\mathbf{return}}
\end{array}
\end{array}$$

Case checking

$$\boxed{\rho; \Delta; \Gamma; L \vdash c : \sigma, \tau/\varepsilon}$$

$$\begin{array}{c}
\text{[T-CASE]} \\
\frac{\rho; \Delta; \Gamma \vdash e : \sigma \quad \rho; \Delta; \Gamma; L \vdash ss : \tau/\varepsilon}{\rho; \Delta; \Gamma; L \vdash \mathbf{case} \ e : ss : \sigma, \tau/\varepsilon}
\end{array}$$

Expression checking

$$\boxed{\rho; \Delta; \Gamma \vdash e : \tau}$$

[T-VALUE]

$$\frac{}{\rho; \Delta; \Gamma \vdash v : \text{type}(v)}$$

[T-VARREF]

$$\frac{\Gamma(x) = \tau}{\rho; \Delta; \Gamma \vdash x : \tau}$$

[T-ASSIGN]

$$\frac{\Gamma(x) = \tau \quad \rho; \Delta; \Gamma \vdash e : \tau}{\rho; \Delta; \Gamma \vdash x = e : \tau}$$

[T-LOAD]

$$\frac{\text{type}(\rho(x)) = \text{array}_\tau \quad \rho; \Delta; \Gamma \vdash e : \sigma \quad \sigma <: \text{int}}{\rho; \Delta; \Gamma \vdash x[e] : \tau}$$

[T-STORE]

$$\frac{\text{type}(\rho(x)) = \text{array}_\tau \quad \rho; \Delta; \Gamma \vdash e_1 : \sigma \quad \sigma <: \text{int} \quad \rho; \Delta; \Gamma \vdash e_2 : \tau}{\rho; \Delta; \Gamma \vdash x[e_1] = e_2 : \tau}$$

[T-FUNCALL]

$$\frac{\Delta(f) = (\bar{\sigma}) \rightarrow \tau \quad \forall i. \rho; \Delta; \Gamma \vdash e_i : \sigma_i}{\rho; \Delta; \Gamma \vdash f(\bar{e}) : \tau}$$

[T-FFI]

$$\frac{f \notin \text{dom}(\Delta) \quad f \notin \text{dom}(\Gamma) \quad \text{type}(\rho(f)) = \text{Function} \quad \forall i. \rho; \Delta; \Gamma \vdash e_i : \sigma_i}{\rho; \Delta; \Gamma \vdash f(\bar{e}) : \text{any}}$$

[T-UNARYOP]

$$\frac{\text{type}(\text{unop}) = \sigma \rightarrow \tau \quad \rho; \Delta; \Gamma \vdash e : \sigma}{\rho; \Delta; \Gamma \vdash \text{unop } e : \tau}$$

[T-BINARYOP]

$$\frac{\forall i \in \{1, 2\}. \rho; \Delta; \Gamma \vdash e_i : \sigma_i \quad \text{type}(\text{binop}) = \sigma_1 \times \sigma_2 \rightarrow \tau}{\rho; \Delta; \Gamma \vdash e_1 \text{ binop } e_2 : \tau}$$

[T-CONDITIONAL]

$$\frac{\rho; \Delta; \Gamma \vdash e_1 : \text{boolean} \quad \forall i \in \{2, 3\}. \rho; \Delta; \Gamma \vdash e_i : \tau}{\rho; \Delta; \Gamma \vdash e_1 ? e_2 : e_3 : \tau}$$

[T-PAREN]

$$\frac{\forall i \leq n. \rho; \Delta; \Gamma \vdash e_i : \tau_i}{\rho; \Delta; \Gamma \vdash (\bar{e}) : \tau_n}$$