1 Virtual Machine Abstract Syntax

$$
\begin{array}{llll}
S & ::= & d; S & \textit{statements} \\
  & | & e & \\
\\
d & ::= & \texttt{val}\ f = e & \textit{declarations} \\
  & | & \texttt{def}\ m(\overline{x : \tau}) : \tau = S & \\
  & | & \texttt{type}\ L = \tau & \\
  & | & \texttt{subtype}\ \tau\ \texttt{extends}\ \beta & \\
\\
e & ::= & x & \textit{expressions} \\
  & | & \texttt{new}\ x : \tau\ \{\overline{d}\} & \\
  & | & e.m(\overline{e}) & \\
  & | & e.f & \\
  & | & \mathscr{L} & \\
\\
\mathscr{L} & ::= & n & \textit{literals}
\end{array}
$$

$$
\begin{array}{llll}
\beta & ::= & \texttt{Unit} & \textit{base type} \\
      & | & L & \\
\\
\tau & ::= & \beta\{\overline{\sigma}\} & \textit{type} \\
\\
\sigma & ::= & \texttt{val}\ f : \tau & \textit{decl type} \\
       & | & \texttt{def}\ m(\overline{x : \tau}) : \tau & \\
       & | & \texttt{type}\ L = \tau & \\
       & | & \texttt{subtype}\ \tau\ \texttt{extends}\ \beta &
\end{array}
$$

Notation: overbar means a list of elements, as in Java

2 Standard prelude

```
type Int
    def +(i:Int):Int
    def -(i:Int):Int
    def *(i:Int):Int
    def /(i:Int):Int
```

3 Virtual Machine Typing Rules

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

$$\frac{\Gamma \vdash d : \sigma \quad \Gamma, \sigma \vdash S : \tau}{\Gamma \vdash d; S : \tau}\ (\text{T-S}\textsc{tmt})$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ (\text{T-V}\textsc{ar})$$

$$\frac{\Gamma \mid x : \tau \vdash \overline{d} : \textit{unfold}_\Gamma(\tau)}{\Gamma \vdash \texttt{new}\ x : \tau\ \{\overline{d}\} : \tau}\ (\text{T-N}\textsc{ew})$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \texttt{def}\ m(x : \tau_2') : \tau \in \textit{unfold}_\Gamma(\tau_1) \quad \Gamma \vdash e_2 : \tau_2 \quad \Gamma \vdash \tau_2 <: \tau_2'}{\Gamma \vdash e_1.m(e_2) : \tau}\ (\text{T-I}\textsc{nvk})$$

$$\frac{\Gamma \vdash e : \tau' \quad \texttt{val}\ f : \tau \in \textit{unfold}_\Gamma(\tau')}{\Gamma \vdash e.f : \tau}\ (\text{T-F}\textsc{ield})$$

$$\frac{}{\Gamma \vdash n : \textit{Int}}\ (\text{T-I}\textsc{nt})$$

Technically Γ is a list of σ, but we often write $x : \tau$ for $\texttt{val}\ x : \tau$.

$\boxed{\Gamma \mid x : \tau \vdash d : \sigma}$

$$\frac{\Gamma, x{:}\tau \vdash \overline{\tau}\ \mathit{wf} \quad \Gamma,\ x{:}\tau,\ \overline{y{:}\tau} \vdash e : \tau_2' \quad \Gamma, x{:}\tau \vdash \tau_2' <: \tau_2}{\Gamma \mid x : \tau \vdash \texttt{def}\ m(\overline{y : \tau}) : \tau_2 = e\ \ :\ \ \texttt{def}\ m(\overline{y : \tau}) : \tau_2}\ \text{(T-Def)}$$

$$\frac{\Gamma \vdash e : \tau}{\Gamma \mid x : \tau \vdash \texttt{val}\ f = e\ \ :\ \ \texttt{val}\ f : \tau}\ \text{(T-Val)}$$

$$\frac{\Gamma, x : \tau \vdash \texttt{type}\ L = \tau\ \mathit{wf}}{\Gamma \mid x : \tau \vdash \texttt{type}\ L = \tau : \texttt{type}\ L = \tau}\ \text{(T-Type)}$$

$$\frac{\Gamma, x : \tau \vdash \texttt{subtype}\ \tau\ \texttt{extends}\ \beta\ \mathit{wf}}{\Gamma \mid x : \tau \vdash \texttt{subtype}\ \tau\ \texttt{extends}\ \beta : \texttt{subtype}\ \tau\ \texttt{extends}\ \beta}\ \text{(T-Subtype)}$$

$\boxed{\Gamma \vdash \tau_1 <: \tau_2}$

$$\frac{\Gamma \vdash \overline{\sigma_1} <: \overline{\sigma_2}}{\Gamma \vdash \beta\{\overline{\sigma_1}\} <: \beta\{\overline{\sigma_2}\}}$$

$$\frac{\Gamma \vdash \tau_1 <: \tau_2 \quad \Gamma \vdash \tau_2 <: \tau_3}{\Gamma \vdash \tau_1 <: \tau_3}$$

$$\frac{}{\Gamma \vdash \tau <: \texttt{Unit}}$$

$$\frac{\Gamma \vdash \beta_1\{\overline{\sigma_1}\} <: \beta_2 \quad \Gamma \vdash \overline{\sigma_1} <: \overline{\sigma_2}}{\Gamma \vdash \beta_1\{\overline{\sigma_1}\} <: \beta_2\{\overline{\sigma_2}\}}$$

$\boxed{\Gamma \vdash \beta_1\{\overline{\sigma}\} <: \beta_2}$ and $\boxed{\Gamma \vdash \overline{\sigma_1} <: \overline{\sigma_2}}$ and $\boxed{\Gamma \vdash \sigma_1 <: \sigma_2}$

$$\frac{}{\Gamma \vdash \beta\{\overline{\sigma}\} <: \beta}$$

$$\frac{L_1\{\overline{\sigma_1}\} <: \text{Ł}_2 \in \Gamma \quad \Gamma \vdash \overline{\sigma_1'} <: \overline{\sigma_1} \quad \Gamma \vdash L_2\{\overline{\sigma_1'}\} <: \text{Ł}_3}{\Gamma \vdash L_1\{\overline{\sigma_1'}\} <: \text{Ł}_3}$$

$$\frac{\sigma_1' \subset \sigma_1 \quad \overline{\Gamma \vdash \sigma_1' <: \sigma_2}}{\Gamma \vdash \overline{\sigma_1} <: \overline{\sigma_2}}$$

$$\frac{\Gamma \vdash \tau <: \tau'}{\Gamma \vdash \texttt{val}\ f : \tau <: \texttt{val}\ f : \tau'}$$

$$\frac{\Gamma \vdash \tau_2 <: \tau_2' \quad \Gamma \vdash \overline{\tau_1'} <: \overline{\tau_1}}{\Gamma \vdash \texttt{def}\ m(\overline{x : \tau_1}) : \tau_2 <: \texttt{def}\ m(\overline{x : \tau_1'}) : \tau_2'}$$

$$\frac{}{\Gamma \vdash \texttt{type}\ L = \tau <: \texttt{type}\ L = \tau}$$

$\boxed{\mathit{unfold}_\Gamma(\tau) = \overline{\sigma}}$

$$\frac{}{\mathit{unfold}_\Gamma(\texttt{Unit}) = \bullet}$$

$$\frac{\texttt{type}\ L = \tau \in \Gamma \quad \mathit{unfold}_\Gamma(\tau) = \overline{\sigma}}{\mathit{unfold}_\Gamma(L) = \overline{\sigma}}$$

$$\frac{\mathit{unfold}_\Gamma(\beta) = \overline{\sigma}}{\mathit{unfold}_\Gamma(\beta\{\overline{\sigma'}\}) = \overline{\sigma} \leftarrow \overline{\sigma'}}$$

Note: $\overline{\sigma} \leftarrow \overline{\sigma'}$ means that we append the two lists, except that when the same symbol is defined in both $\overline{\sigma}$ and $\overline{\sigma'}$, we include only the (overriding) definition in $\overline{\sigma'}$.

Now, finally, type and declaration type well-formedness rules:

$\boxed{\Gamma \vdash \tau\ \mathit{wf}}$

$$\frac{}{\Gamma \vdash \texttt{Unit}\ \ \mathit{wf}}$$

$$\frac{\mathit{unfold}_\Gamma(L) = \overline{\sigma}}{\Gamma \vdash L\ \mathit{wf}}$$

$$\frac{\Gamma \vdash \beta\ \mathit{wf} \quad \Gamma \vdash \overline{\sigma'}\ \mathit{wf}}{\Gamma \vdash \beta\{\overline{\sigma'}\}\ \mathit{wf}}$$

$\boxed{\Gamma \vdash \sigma \ wf}$

$$\frac{\Gamma \vdash \overline{y : \tau} \ wf \quad \Gamma \vdash \tau_2 \ wf}{\Gamma \vdash \texttt{def} \ m(\overline{y : \tau}) : \tau_2 \ wf}$$

$$\frac{\Gamma \vdash \tau \ wf}{\Gamma \vdash \texttt{val} \ f : \tau \ wf}$$

$$\frac{\Gamma, \texttt{type} \ L = \tau \vdash \tau \ wf}{\Gamma \vdash \texttt{type} \ L = \tau \ wf}$$

$$\frac{\mathit{unfold}_\Gamma(\tau) = \overline{\sigma} \quad \mathit{unfold}_\Gamma(\beta) = \overline{\sigma'} \quad \Gamma, \texttt{subtype} \ \tau \ \texttt{extends} \ \beta \vdash \overline{\sigma} <: \overline{\sigma'}}{\Gamma \vdash \texttt{subtype} \ \tau \ \texttt{extends} \ \beta \ wf}$$