React⚛tRace

Jay Lee

August 20, 2024

1 Syntax and Semantics of Core React

Syntax

Prog ∋	P	$::=$	e	top-level exp
		$\mid$	$c.\lambda x.e\ P$	component def
Expr ∋	e	$::=$	x	identifier
		$\mid$	$\texttt{()} \mid \texttt{true} \mid \texttt{false} \mid n$	constant
		$\mid$	$\texttt{view[}e\texttt{,}\dots\texttt{,}e\texttt{]}$	view
		$\mid$	$\texttt{if}\ e\ e\ e$	conditional
		$\mid$	$\lambda x.e$	function def
		$\mid$	$e\ e$	application
		$\mid$	$\texttt{let}\ x\ \texttt{=}\ e\ \texttt{in}\ e$	let binding
		$\mid$	$\texttt{let (}x\texttt{, }x'_{\texttt{set}}\texttt{)=useState}^{\ell}\ e\ \texttt{in}\ e$	`useState` hook
		$\mid$	$\texttt{let}\ x\ \texttt{= useRef}^{\ell}\ e\ \texttt{in}\ e$	`useRef` hook
		$\mid$	$\texttt{useEffect}\ e$	`useEffect` hook
		$\mid$	$e\texttt{;}\ e$	sequence
		$\mid$	$e \oplus e$	binary op

Semantics

Val ∋	v	$::=$	$\langle\rangle \mid b \mid n$	primitive values
		$\mid$	$[\overline{s}]$	view spec list
		$\mid$	$\langle \lambda x.e, \sigma \rangle$	closure
		$\mid$	$\langle \texttt{set}_{\ell}, p \rangle$	setter closure
		$\mid$	$\langle c.\lambda x.e, \sigma \rangle$	component closure
		$\mid$	$\langle c.\lambda x.e, \sigma, v \rangle$	component spec
Bool ∋	b	$::=$	$\mathbb{tt} \mid \mathbb{ff}$	
ViewSpec ∋	s	$::=$	$\langle\rangle \mid n$	primitive spec
		$\mid$	$\langle c.\lambda x.e, \sigma, v \rangle$	component spec
Env ∋	σ	$::=$	$\{\overline{x \mapsto v}\}$	environment
Phase ∋	ϕ	$::=$	$\mathsf{PInit} \mid \mathsf{PUpdate}$	
		$\mid$	$\mathsf{PRetry} \mid \mathsf{PEffect}$	phase
Decision ∋	d	$::=$	$\mathsf{Idle} \mid \mathsf{Retry} \mid \mathsf{Update}$	decision
PartView ∋	π	$::=$	$\bullet$	root
		$\mid$	$\langle \langle c.\lambda x.e, \sigma, v \rangle, d, \rho_s, \rho_r, q \rangle$	node
SttStore ∋	ρ_s	$::=$	$\{\overline{\ell \mapsto \langle v, q \rangle}\}$	state store
RefStore ∋	ρ_r	$::=$	$\{\overline{\ell \mapsto v}\}$	ref store
JobQ ∋	q	$::=$	$[\overline{\langle \lambda x.e, \sigma \rangle}]$	job queue
Tree ∋	t	$::=$	$\ulcorner\lrcorner \mid \ulcorner n \lrcorner \mid p$	tree
TreeMem ∋	m	$::=$	$\{\overline{p \mapsto \langle \pi, l \rangle}\}$	tree memory

p	$\in$	Path	tree path
l	$::=$	$[\overline{t}]$	children

$$\boxed{m, \sigma \vdash e \Downarrow_p^\phi v, m'}$$

$$\frac{}{m, \sigma \vdash x \Downarrow_p^\phi \sigma(x), m}\ \textsc{Var} \qquad \frac{}{m, \sigma \vdash \texttt{()} \Downarrow_p^\phi \langle\rangle, m}\ \textsc{Unit} \qquad \frac{}{m, \sigma \vdash \texttt{true} \Downarrow_p^\phi \mathtt{tt}, m}\ \textsc{True} \qquad \frac{}{m, \sigma \vdash \texttt{false} \Downarrow_p^\phi \mathtt{ff}, m}\ \textsc{False} \qquad \frac{}{m, \sigma \vdash n \Downarrow_p^\phi n, m}\ \textsc{Int}$$

$$\frac{\left(m_i, \sigma \vdash e_i \Downarrow_p^\phi s_i, m_{i+1}\right)_{i=1}^n}{m_1, \sigma \vdash \texttt{view}\ [\overline{e_i}]_{i=1}^n \Downarrow_p^\phi [\overline{s_i}]_{i=1}^n, m_{n+1}}\ \textsc{ViewSpec} \qquad \frac{m, \sigma \vdash e_1 \Downarrow_p^\phi b, m' \qquad m', \sigma \vdash \llparenthesis b \mathrel{?} e_2 : e_3 \rrparenthesis \Downarrow_p^\phi v, m''}{m, \sigma \vdash \texttt{if}\ e_1\ e_2\ e_3 \Downarrow_p^\phi v, m''}\ \textsc{Cond}$$

$$\frac{}{m, \sigma \vdash \lambda x.e \Downarrow_p^\phi \langle \lambda x.e, \sigma\rangle, m}\ \textsc{Func}$$

$$\frac{m, \sigma \vdash e_1 \Downarrow_p^\phi \langle \lambda x.e, \sigma'\rangle, m_1 \qquad m_1, \sigma \vdash e_2 \Downarrow_p^\phi v_2, m_2 \qquad m_2, \sigma'\{x \mapsto v_2\} \vdash e \Downarrow_p^\phi v', m'}{m, \sigma \vdash e_1\ e_2 \Downarrow_p^\phi v', m'}\ \textsc{AppFunc}$$

$$\frac{m, \sigma \vdash e_1 \Downarrow_p^\phi \langle c.\lambda x.e, \sigma'\rangle, m_1 \qquad m_1, \sigma \vdash e_2 \Downarrow_p^\phi v_2, m_2}{m, \sigma \vdash e_1\ e_2 \Downarrow_p^\phi \langle c.\lambda x.e, \sigma', v_2\rangle, m_2}\ \textsc{AppComp}$$

$$\frac{m, \sigma \vdash e_1 \Downarrow_p^\phi \langle \texttt{set}_\ell, p'\rangle, m_1 \qquad m_1, \sigma \vdash e_2 \Downarrow_p^\phi \langle \lambda x.e, \sigma'\rangle, m_2}{m, \sigma \vdash e_1\ e_2 \Downarrow_p^\phi \langle\rangle, m_2\left\{(p').\pi\left\langle \begin{array}{l} d \mapsto \llparenthesis (p = p' \wedge \phi \neq \mathsf{PEffect}) \mathrel{?} \mathsf{Retry} : \mathsf{Update} \rrparenthesis \\ \rho_s\{(\ell).q \mapsto q \triangleright \langle \lambda x.e, \sigma'\rangle\} \end{array}\right\rangle\right\}}\ \textsc{AppSetStt}$$

$$\frac{m, \sigma \vdash e_1 \Downarrow_p^\phi v_1, m_1 \qquad m_1, \sigma\{x \mapsto v_1\} \vdash e_2 \Downarrow_p^\phi v_2, m_2}{m, \sigma \vdash \texttt{let}\ x\ \texttt{=} e_1\ \texttt{in}\ e_2 \Downarrow_p^\phi v_2, m_2}\ \textsc{LetBind}$$

$$\frac{m, \sigma \vdash e_1 \Downarrow_p^{\mathsf{PInit}} v_1, m_1 \qquad m_1\{(p).\pi.\rho_s.(\ell) \mapsto \langle v_1, []\rangle\}, \sigma\{x \mapsto v_1, x'_{\texttt{set}} \mapsto \langle \texttt{set}_\ell, p\rangle\} \vdash e_2 \Downarrow_p^{\mathsf{PInit}} v_2, m_2}{m, \sigma \vdash \texttt{let (}x\texttt{, }x'_{\texttt{set}}\texttt{)=useState}^\ell\ e_1\ \texttt{in}\ e_2 \Downarrow_p^{\mathsf{PInit}} v_2, m_2}\ \textsc{SttBind}$$

$$\frac{\begin{array}{c} \phi \in \{\mathsf{PUpdate}, \mathsf{PRetry}\} \qquad m_1(p).\pi.\rho_s(\ell) = \left\langle v_1, \left[\overline{\langle \lambda x_i.e'_i, \sigma_i\rangle}\right]_{i=1}^n\right\rangle \\ \left(m_i, \sigma_i\{x_i \mapsto v_i\} \vdash e'_i \Downarrow_p^\phi v_{i+1}, m_{i+1}\right)_{i=1}^n \\ m_{n+1}\left\{(p).\pi.\left\langle \begin{array}{l} d \mapsto \llparenthesis v_{n+1} \not\equiv v_1 \mathrel{?} \mathsf{Update} : d \rrparenthesis \\ \rho_s\{(\ell) \mapsto \langle v_{n+1}, []\rangle\} \end{array}\right\rangle\right\}, \sigma\left\{ \begin{array}{l} x \mapsto v_{n+1} \\ x'_{\texttt{set}} \mapsto \langle \texttt{set}_\ell, p\rangle \end{array}\right\} \vdash e_2 \Downarrow_p^\phi v, m' \end{array}}{m_1, \sigma \vdash \texttt{let (}x\texttt{, }x'_{\texttt{set}}\texttt{)=useState}^\ell\ e_1\ \texttt{in}\ e_2 \Downarrow_p^\phi v, m'}\ \textsc{SttReBind}$$

$$\frac{m, \sigma \vdash e_1 \Downarrow_p^{\mathsf{PInit}} v_1, m_1 \qquad m_1\{(p).\pi.\rho_r.(\ell) \mapsto v_1\}, \sigma\{x \mapsto v_1\} \vdash e_2 \Downarrow_p^{\mathsf{PInit}} v_2, m_2}{m, \sigma \vdash \texttt{let}\ x\ \texttt{=}\ \texttt{useRef}^\ell\ e_1\ \texttt{in}\ e_2 \Downarrow_p^{\mathsf{PInit}} v_2, m_2}\ \textsc{RefBind}$$

$$\frac{\phi \in \{\mathsf{PUpdate}, \mathsf{PRetry}\} \qquad m(p).\pi.\rho_r(\ell) = v_1 \qquad m, \sigma\{x \mapsto v_1\} \vdash e_2 \Downarrow_p^\phi v_2, m_2}{m, \sigma \vdash \texttt{let}\ x\ \texttt{=}\ \texttt{useRef}^\ell\ e_1\ \texttt{in}\ e_2 \Downarrow_p^\phi v_2, m_2}\ \textsc{RefReBind}$$

$$\frac{\phi \in \{\mathsf{PInit}, \mathsf{PUpdate}, \mathsf{PRetry}\}}{m, \sigma \vdash \texttt{useEffect}\ e \Downarrow_p^\phi \langle\rangle, m\{(p).\pi.\langle q \mapsto q \triangleright \langle \lambda\_.e, \sigma\rangle\rangle\}}\ \textsc{Eff}$$

$$\boxed{m, \sigma \vdash e \Downarrow\!\!\Downarrow_p^\phi v, m'}$$

$$\frac{m, \sigma \vdash e \Downarrow_p^\phi v, m' \qquad m'(p).\pi.d \in \{\mathsf{Idle}, \mathsf{Update}\}}{m, \sigma \vdash e \Downarrow\!\!\Downarrow_p^\phi v, m'}\ \textsc{EvalOnce}$$

$$\frac{m, \sigma \vdash e \Downarrow_p^\phi v, m' \qquad m'(p).\pi.d = \mathsf{Retry} \qquad m', \sigma \vdash e \Downarrow\!\!\Downarrow_p^\phi v', m''}{m, \sigma \vdash e \Downarrow\!\!\Downarrow_p^\phi v', m''}\ \textsc{EvalMult}$$

$$\boxed{m \vdash \mathit{render1}(s) = \langle t, m'\rangle}$$

$$\frac{}{m \vdash \mathit{render1}(\langle\rangle) = \langle \ulcorner\lrcorner, m\rangle}\ \textsc{Render1Null} \qquad \frac{}{m \vdash \mathit{render1}(n) = \langle \ulcorner n \lrcorner, m\rangle}\ \textsc{Render1Int}$$

$$\frac{m \vdash p \text{ fresh} \qquad m\{(p) \mapsto \langle\langle\langle c.\lambda x.e, \sigma, v\rangle, \mathsf{Idle}, \{\}, []\rangle, []\rangle\}, \sigma\{x \mapsto v\} \vdash e \Downarrow_p^{\mathsf{PInit}} [\bar{s}], m' \qquad m' \vdash \mathit{render}(p, [\bar{s}]) = m''}{m \vdash \mathit{render1}(\langle c.\lambda x.e, \sigma, v\rangle) = \langle p, m''\rangle}\ \textsc{Render1Comp}$$

$$\boxed{m \vdash \mathit{render}(p, [\bar{s}]) = m'}$$

$$\frac{(m_i \vdash \mathit{render1}(s_i) = \langle t_i, m_i'\rangle)_{i=1}^{n} \qquad (m_{i+1} = m_i'\{(p).l \mapsto l \triangleright t_i\})_{i=1}^{n}}{m_1 \vdash \mathit{render}\left(p, [\bar{s_i}]_{i=1}^{n}\right) = m'}\ \textsc{Render}$$

$$\boxed{m \vdash \mathit{update1}(t, v_\varepsilon) = \langle b, m'\rangle}$$

$$\frac{}{m \vdash \mathit{update1}(\ulcorner\lrcorner, \varepsilon) = \langle \mathtt{ff}, m\rangle}\ \textsc{Update1Null} \qquad \frac{}{m \vdash \mathit{update1}(\ulcorner n \lrcorner, \varepsilon) = \langle \mathtt{ff}, m\rangle}\ \textsc{Update1Int} \qquad \frac{m \vdash \mathit{update}(p, v_\varepsilon) = \langle b, m'\rangle}{m \vdash \mathit{update1}(p, v_\varepsilon) = \langle b, m'\rangle}\ \textsc{Update1Path}$$

$$\boxed{m \vdash \mathit{update}(p, v_\varepsilon) = \langle b, m'\rangle}$$

$$\frac{m_1(p) = \left\langle \bullet, [\bar{t_i}]_{i=1}^{n}\right\rangle \qquad (m_i \vdash \mathit{update1}(t_i, \varepsilon) = \langle b_i, m_{i+1}\rangle)_{i=1}^{n}}{m_1 \vdash \mathit{update}(p, \varepsilon) = \left\langle \bigvee_{i=1}^{n} b_i, m_{n+1}\right\rangle}\ \textsc{UpdateRoot}$$

$$\frac{m_1(p) = \left\langle \pi, [\bar{t_i}]_{i=1}^{n}\right\rangle \qquad \pi.d = \mathsf{Idle} \qquad (m_i \vdash \mathit{update1}(t_i, \varepsilon) = \langle b_i, m_{i+1}\rangle)_{i=1}^{n}}{m_1 \vdash \mathit{update}(p, \varepsilon) = \left\langle \bigvee_{i=1}^{n} b_i, m_{n+1}\right\rangle}\ \textsc{UpdateNodeIdle}$$

$$\frac{\begin{array}{c} m(p) = \left\langle \langle\langle c.\lambda x.e, \sigma, v'\rangle, d, \rho_s, q\rangle, [\bar{t_i}]_{i=1}^{n}\right\rangle \qquad (d = \mathsf{Update}) \vee (d = \mathsf{Idle} \wedge v_\varepsilon \neq \varepsilon) \\ v = (\!|\, v_\varepsilon \neq \varepsilon \mathrel{?} v_\varepsilon : v' \,|\!) \qquad m\{(p).\pi\langle d \mapsto \mathsf{Idle}\rangle\}, \sigma\{x \mapsto v\} \vdash e \Downarrow_p^{\mathsf{PUpdate}} [\bar{s_i}]_{i=1}^{n'}, m' \\ m'\{(p).\pi\langle l \mapsto []\rangle\} \vdash \mathit{reconcile}\left([\bar{t_i}]_{i=1}^{\min\{n,n'\}}, [\bar{s_i}]_{i=1}^{n'}\right) = \langle b, m''\rangle \end{array}}{m \vdash \mathit{update}(p, v_\varepsilon) = \langle b \vee (m'(p).\pi.d \neq \mathsf{Idle}), m''\rangle}\ \textsc{UpdateNodeReconcile}$$

$$\boxed{m \vdash \mathit{reconcile1}(t, s) = \langle b, t, m'\rangle}$$

$$m \vdash \mathit{reconcile1}(t, s) = \begin{cases} \langle \mathtt{ff}, \ulcorner\lrcorner, m\rangle & \text{if } \langle t, s\rangle = \langle \ulcorner\lrcorner, \langle\rangle\rangle \\ \langle \mathtt{ff}, \ulcorner n \lrcorner, m\rangle & \text{if } \langle t, s\rangle = \langle \ulcorner n \lrcorner, \langle n\rangle\rangle \\ \langle b, p, m'\rangle & \text{if } \langle t, s\rangle = \langle p, \langle c.\lambda x.e, \sigma, v\rangle\rangle \wedge m(p).\pi.s = \langle c.\lambda x.e, \sigma, v'\rangle, \\ & \quad \text{where } (m \vdash \mathit{update1}(p, (\!|\, v \not\equiv v' \mathrel{?} v : \varepsilon \,|\!)) = \langle b, m'\rangle) \\ \langle \mathtt{tt}, t', m'\rangle & \text{otherwise, where } (m \vdash \mathit{render1}(s) = \langle t', m'\rangle) \end{cases}$$

$$\boxed{m \vdash \mathit{reconcile}(p, [\bar{t}], [\bar{s}]) = \langle b, m'\rangle}$$

$$\frac{\left\{\begin{array}{ll} m_i \vdash \mathit{reconcile1}(t_i, s_i) = \langle b_i, t_i, m_i'\rangle & \text{if } i \leq n' \\ m_i \vdash \mathit{render1}(s_i) = \langle t_i, m_i'\rangle, \quad b_i = \mathtt{tt} & \text{otherwise} \\ \multicolumn{2}{c}{\text{where } m_{i+1} = m_i'\{(p).l \mapsto l \triangleright t_i\}} \end{array}\right\}_{i=1}^{n}}{m_1 \vdash \mathit{reconcile}\left(p, [\bar{t_i}]_{i=1}^{n'}, [\bar{s_i}]_{i=1}^{n}\right) = \left\langle \bigvee_{i=1}^{n} b_i, m_{n+1}\right\rangle}\ \textsc{Reconcile}$$

$$\boxed{m \vdash \mathit{commitEffs1}(t) = m'}$$

$$\frac{\textsc{CommitEffs1Unit}}{m \vdash \mathit{commitEffs1}(\langle\rangle) = m} \quad \frac{\textsc{CommitEffs1Int}}{m \vdash \mathit{commitEffs1}(n) = m} \quad \frac{\overset{\textsc{CommitEffs1Path}}{m \vdash \mathit{commitEffs}(p) = m'}}{m \vdash \mathit{commitEffs1}(p) = m'}$$

$$\boxed{m \vdash \mathit{commitEffs}(p) = m'}$$

$$\textsc{CommitEffsRoot}\ \frac{m_1(p) = \left\langle \bullet, \left[\bar{t}_i\right]_{i=1}^{n} \right\rangle \qquad (m_i \vdash \mathit{commitEffs1}(t_i) = m_{i+1})_{i=1}^{n}}{m_1 \vdash \mathit{commitEffs}(p) = m_{n+1}}$$

$$\textsc{CommitEffsNode}\ \frac{\begin{array}{c} m_1(p) = \left\langle \pi, \left[\bar{t}_i\right]_{i=1}^{n} \right\rangle \qquad \pi \neq \bullet \qquad \pi.q = \left[\overline{\langle \lambda\_.e_i, \sigma_i \rangle}\right]_{i=1}^{k} \qquad \left(m_i, \sigma_i \vdash e_i \Downarrow_p^{\text{PEffect}} v_i, m_{i+1}\right)_{i=1}^{k} \\ m_1' = m_{k+1}\{(p).\pi\langle q \mapsto [\,]\rangle\} \qquad (m_i' \vdash \mathit{commitEffs1}(t_i) = m_{i+1}')_{i=1}^{n} \end{array}}{m_1 \vdash \mathit{commitEffs}(p) = m_{n+1}'}$$

$$\boxed{\sigma \vdash P \downarrow \left[\bar{s}\right]}$$

$$\textsc{GatherComp}\ \frac{\sigma\{c \mapsto \langle c.\lambda x.e, \sigma \rangle\} \vdash P \downarrow \left[\bar{s}\right]}{\sigma \vdash c.\lambda x.e\ P \downarrow \left[\bar{s}\right]} \quad \textsc{TopExp}\ \frac{\{\}, \sigma \vdash e \Downarrow \left[\bar{s}\right], \{\}}{\sigma \vdash e \downarrow \left[\bar{s}\right]}$$

$$\boxed{P \text{ or } \langle p, m \rangle \hookrightarrow \langle p, m' \rangle}$$

$$\textsc{StepProg}\ \frac{\{\} \vdash P \downarrow \left[\bar{s}\right] \qquad \{\} \vdash p \text{ fresh} \qquad \{p \mapsto \langle \bullet, [\,] \rangle\} \vdash \mathit{render}\left(p, \left[\bar{s}\right]\right) = m \qquad m \vdash \mathit{commitEffs}(p) = m'}{P \hookrightarrow \langle p, m' \rangle}$$

$$\textsc{StepPathUpdate}\ \frac{m \vdash \mathit{update}(p, \varepsilon) = \langle \mathtt{tt}, m' \rangle \qquad m' \vdash \mathit{commitEffs}(q) = m''}{\langle p, m \rangle \hookrightarrow \langle p, m'' \rangle} \quad \textsc{StepPathDone}\ \frac{m \vdash \mathit{update}(p, \varepsilon) = \langle \mathtt{ff}, m' \rangle}{\langle p, m \rangle \hookrightarrow \langle p, m' \rangle}$$

2 Language extension

We extend the core React with objects and the heap.

Syntax

$$\begin{array}{rrcll} \text{Expr} \ni & e & ::= & \{\overline{x : e}\} & \text{object} \\ & & \mid & e.x & \text{field access} \\ & & \mid & e.x := e & \text{assignment} \end{array}$$

Semantics

$$\begin{array}{rrcll} \text{Prim} \ni & \epsilon & ::= & \langle\rangle \mid b \mid n & \text{primitive values} \\ \text{Object} \ni & o & ::= & \left[\bar{s}\right] & \text{view spec list} \\ & & \mid & \langle \lambda x.e, \sigma \rangle & \text{closure} \\ & & \mid & \langle \mathtt{set}_\ell, p \rangle & \text{setter closure} \\ & & \mid & \langle c.\lambda x.e, \sigma \rangle & \text{component closure} \\ & & \mid & \langle c.\lambda x.e, \sigma, v \rangle & \text{component spec} \\ \text{Bool} \ni & b & ::= & \mathtt{tt} \mid \mathtt{ff} & \\ \text{ViewSpec} \ni & s & ::= & \langle\rangle \mid n & \text{primitive spec} \end{array}$$

$$
\begin{array}{rcll}
 & & | \ \langle c.\lambda x.e, \sigma, v \rangle & \text{component spec} \\
\text{Env} \ni \sigma & ::= & \left\{ \overline{x \mapsto v} \right\} & \text{environment} \\
\text{Phase} \ni \phi & ::= & \mathsf{PInit} \mid \mathsf{PUpdate} & \\
 & | & \mathsf{PRetry} \mid \mathsf{PEffect} & \text{phase} \\
\text{Decision} \ni d & ::= & \mathsf{Idle} \mid \mathsf{Retry} \mid \mathsf{Update} & \text{decision} \\
\text{PartView} \ni \pi & ::= & \bullet & \text{root} \\
 & | & \langle \langle c.\lambda x.e, \sigma, v \rangle, d, \rho_s, \rho_r, q \rangle & \text{node} \\
\text{SttStore} \ni \rho_s & ::= & \left\{ \overline{\ell \mapsto \langle v, q \rangle} \right\} & \text{state store} \\
\text{RefStore} \ni \rho_r & ::= & \left\{ \overline{\ell \mapsto v} \right\} & \text{ref store} \\
\text{JobQ} \ni q & ::= & \left[\overline{\langle \lambda x.e, \sigma \rangle} \right] & \text{job queue} \\
\text{Tree} \ni t & ::= & \ulcorner \lrcorner \mid \ulcorner n \lrcorner \mid p & \text{tree} \\
\text{TreeMem} \ni m & ::= & \left\{ \overline{p \mapsto \langle \pi, l \rangle} \right\} & \text{tree memory} \\
p & \in & \mathsf{Path} & \text{tree path} \\
l & ::= & \left[\bar{t} \right] & \text{children}
\end{array}
$$