15-317 Lecture 24: Ordered and Linear Session Types

- Recap of $\oplus$, $\&$, 1
- $\backslash$ $/$ $\multimap$]
- $\bullet$ $\circ$ $\otimes$]
- Example: Lists
- Type safety
  - Session fidelity / preservation
  - Deadlock freedom / progress

$$\frac{\Omega \Vdash P :: (x : A)}{\Omega \Vdash z.\mathrm{inl}(x); P :: (z : A \oplus B)} \oplus R_1$$

$$\frac{\Omega_L, (x : A)\, \Omega_R \Vdash P :: (w : C) \qquad \Omega_L\, (y : B)\, \Omega_R \vdash Q :: (w : C)}{\Omega_L, (z : A \oplus B)\, \Omega_R \Vdash \mathrm{case}\ z\ (\mathrm{inl}(x) \Rightarrow P \mid \mathrm{inr}(y) \Rightarrow Q) :: (w : C)} \oplus L$$

$$\frac{}{\Omega \Vdash \mathrm{close}\ x :: (x : 1)} 1R$$

$$\frac{\Omega_L\, \Omega_R \Vdash P :: (z : C)}{\Omega_L\, (x : 1)\, \Omega_R \Vdash \mathrm{wait}\ x; P :: (z : C)} 1L$$

$$\dfrac{(x{:}A)\ \Omega \Vdash P :: (y{:}B)}{\Omega \Vdash \mathrm{case}\ z\ (\langle x, y\rangle \Rightarrow P) :: (z{:}A \backslash B)}\ \backslash R$$

$$\dfrac{\Omega \Vdash P :: (x{:}A) \qquad \Omega_L\ (y{:}B)\ \Omega_R \Vdash Q :: (w{:}C)}{\Omega_L\ \Omega\ (z{:}A \backslash B)\ \Omega_R \Vdash x \leftarrow P\ ;\ z.\langle x, y\rangle\ ;\ Q :: (w{:}C)}\ \backslash L$$

$$\dfrac{\Omega \Vdash P :: (x{:}A) \qquad \left(\dfrac{\Omega_L\ (y{:}B)\ \Omega_R \Vdash Q :: (w{:}C)}{\Omega_L\ (x{:}A)\ (z{:}A \backslash B)\ \Omega_R \Vdash z.\langle x, y\rangle\ ;\ Q :: (w{:}C)}\ \backslash L^{*}\right)}{\Omega_L\ \Omega\ (z{:}A \backslash B)\ \Omega_R \Vdash (x \leftarrow P\ ;\ (z.\langle x, y\rangle\ ;\ Q)) :: (w{:}C)}\ \mathrm{cut}$$

$$\frac{\Delta, (x{:}A) \Vdash P :: (y{:}B)}{\Delta \Vdash \mathsf{case}\ z\ (\langle x, y\rangle \Rightarrow P) :: (z : A \multimap B)}\ {\multimap}R
\qquad
\frac{\Delta, y{:}B \Vdash Q :: (w{:}C)}{\Delta, x{:}A, z{:}A \multimap B \Vdash z.\langle x, y\rangle ; Q :: (w{:}C)}\ {\multimap}L^{*}$$

$$\frac{\Delta_1 \Vdash A \qquad \Delta_2 \Vdash B}{\Delta_1, \Delta_2 \Vdash A \otimes B}\ {\otimes}R
\qquad
\frac{\Delta \Vdash B}{\Delta, A \Vdash A \otimes B}\ {\otimes}R^{*}
\qquad
\frac{\Delta \Vdash A}{\Delta, B \Vdash A \otimes B}\ {\otimes}R^{+}$$

$$\frac{\Delta \Vdash P :: (y{:}B)}{\Delta, (x{:}A) \Vdash z.\langle x, y\rangle ; P :: (z{:}A \otimes B)}\ {\otimes}R^{*}
\qquad
\frac{\Delta, x{:}A, y{:}B \Vdash P :: (w{:}C)}{\Delta, (z{:}A \otimes B) \Vdash \mathsf{case}\ z\ (\langle x, y\rangle \Rightarrow P) :: (w{:}C)}\ {\otimes}L$$

$\otimes R^*$

$$\Delta_2 \Vdash Q :: (y : B)$$

$$\dfrac{}{x : A,\ y : B \Vdash z.\langle x, y\rangle :: (z : A \otimes B)}\ \otimes R^0$$

cut

$$\Delta_2,\ x : A \Vdash y \leftarrow Q\ ;\ z.\langle x, y\rangle$$

$$\Delta_1 \Vdash P :: (x : A)$$

cut

$$\Delta_1, \Delta_2 \Vdash x \leftarrow P\ ;\ y \leftarrow Q\ ;\ z.\langle x, y\rangle :: (z : A \otimes B)$$

Lists

$$\text{list}_A = \oplus\{\text{cons} : A \overset{\otimes}{\underset{\circ ?}{\bullet}} \text{list}_A,\ \text{nil} : 1\}$$

$(l : \text{list}_A)\,(k : \text{list}_A) \Vdash \text{append} :: (r : \text{list}_A)$

append = case l (nil(x) ⇒ wait x ; $r \leftrightarrow k$
| cons(y) ⇒ case y (⟨a, l'⟩ ⇒
r' ← append ← l' k ;
z ← (z.⟨a, r'⟩) ;
r.cons(z)))

```
case l of
  [] => k
| cons(a, l') =>
  let r' = append l' k
  let z = (a, r')
  in
    cons (z)
  end
```

$(\ell : \mathrm{list}_A) \Vdash \mathrm{rev} : (r : \mathrm{list}_A)$

```
rev = case l (nil(x) => r.nil(x)
             | cons(y) => case y (<a, l'> =>
                 r' <- rev <- l';
                 k <- (z <- (w <- (v <-
                        close v;
                        w.nil(v);
                        z.<a, w>;
                        k.cons(z)))))          [a]
                 r <- append l' k
```

Type safety

- Process configurations

$$\frac{}{\Delta \vdash \cdot :: \Delta}\ \text{emp}$$

$$\frac{}{\cdot \vdash \cdot :: \cdot}\ \text{emp?}$$

$P_1 \mid P_2 \mid \ldots \mid P_n$ ← subs ingleton

⤳ $\mathcal{C} ::= \cdot \mid \text{proc}(a, P) \mid \mathcal{C}_1\ \mathcal{C}_2$

- a: channel provided by P
- P: Process term
- $\mathcal{C}_1\ \mathcal{C}_2$: associative, commutative

$$\frac{}{\Delta \vdash \cdot :: \Delta}\ \text{emp}$$

$$\frac{\Delta_1 \vdash \ell_1 :: \Delta_2 \quad \Delta_2 \vdash \ell_2 :: \Delta_3}{\Delta_1 \vdash \ell_1\, \ell_2 :: \Delta_3}\ \text{comp}$$

$$\frac{\Delta_2 \Vdash P :: (a : A)}{\Delta_1, \Delta_2 \vdash \text{proc}(a, P) :: \Delta_1, (a : A)}\ \text{proc}$$

- Session fidelity (preservation)

  If $\Delta_1 \vdash e :: \Delta_2$ and $e \to e'$,

  then $\Delta_1 \vdash e' :: \Delta_2$.

- Deadlock freedom (progress)

  If $\Delta_1 \vdash e :: \Delta_2$, then either

  There is e' s.t. $e \to e'$

  or every proc(a, P) is trying to communicate on a channel

  $(c : C)$ in Δ_1 or Δ_2