Dequeue Side Semantics

As usual, we assume familiarity with citeOG, adopt its notational conventions, and steal its definitions!

1 Structure and Semantics of Rio Trees

In addition to **Pkt**, **Rk**, and **St**, presuppose the existence of the following opaque sets/functions:

(1) **FIFO**(S), a set of FIFOs with entries in S.
(2) **Class**, a collection of *classes*
(3) **flow** : **Pkt** $\to$ **Class**, a mapping from packets to the class they belong to (flow inference)

Naturally, FIFOs support the partial functions

$$\text{pop} : \mathbf{FIFO}(S) \rightharpoonup S \times \mathbf{FIFO}(S)$$
$$\text{push} : \mathbf{FIFO}(S) \times S \rightharpoonup \mathbf{FIFO}(S)$$

1.1 Structure

Definition 1.1. For topology $t \in \mathbf{Topo}$, the set $\mathbf{RioTree}(t)$ of *Rio trees* over t is defined by

$$\frac{f \in \mathbf{FIFO}(\mathbf{Pkt}) \qquad c \in \mathbf{Class}}{\text{Leaf}(f, c) \in \mathbf{RioTree}(*)} \qquad \frac{ts \in \mathbf{Topo}^n \qquad \forall 1 \le i \le n.\ qs[i] \in \mathbf{RioTree}(ts[i])}{\text{Internal}(qs) \in \mathbf{RioTree}(\text{Node}(ts))}$$

These are trees with leaves decorated by both classes and FIFOs.

Definition 1.2. For topology $t \in \mathbf{Topo}$, the set $\mathbf{OrdTree}(t)$ of *ordered trees* over t is defined by

$$\frac{}{\text{Leaf} \in \mathbf{OrdTree}(*)} \qquad \frac{ts \in \mathbf{Topo}^n \qquad rs \in \mathbf{Rk}^n \qquad \forall 1 \le i \le n.\ os[i] \in \mathbf{OrdTree}(ts[i])}{\text{Internal}(rs, os) \in \mathbf{OrdTree}(\text{Node}(ts))}$$

These are trees with each internal node's child given a rank, thereby inducing a total ordering of children.

1.2 Semantics

Definition 1.3. For $t \in \mathbf{Topo}$, define push : $\mathbf{RioTree}(t) \times \mathbf{Pkt} \rightharpoonup \mathbf{RioTree}(t)$ such that

$$\frac{\mathbf{flow}(\text{pkt}) = c \qquad \text{push}(f, \text{pkt}) = f'}{\text{push}(\text{Leaf}(f, c), \text{pkt}) = \text{Leaf}(f', c)} \qquad \frac{\forall 1 \le i \le |qs|.\ \text{push}(qs[i], \text{pkt}) = qs'[i]}{\text{push}(\text{Internal}(qs), \text{pkt}) = \text{Internal}(qs')} \qquad \frac{\mathbf{flow}(\text{pkt}) \ne c}{\text{push}(\text{Leaf}(f, c), \text{pkt}) = \text{Leaf}(f, c)}$$

Informally, we recursively push to all subtrees but only the FIFOs on leaves with the packet's flow are updated.

Definition 1.4. For $t \in \mathbf{Topo}$, define pop : $\mathbf{RioTree}(t) \times \mathbf{OrdTree}(t) \rightharpoonup \mathbf{Pkt} \times \mathbf{RioTree}(t)$ such that

$$\frac{\text{pop}(f) = (\text{pkt}, f')}{\text{pop}(\text{Leaf}(f, c), \text{Leaf}) = (\text{pkt}, \text{Leaf}(f', c))} \qquad \frac{\begin{array}{c}\exists 1 \le i \le |qs|.\ \text{pop}(qs[i], os[i]) = (\text{pkt}, q) \\ \forall 1 \le j \le |qs|.\ j \ne i \land \text{pop}(qs[j], os[j]) = (\text{pkt}', q') \implies rs[i] < rs[j]\end{array}}{\text{pop}(\text{Internal}(qs), \text{Internal}(rs, os)) = (\text{pkt}, \text{Internal}(qs[q/i]))}$$

Informally, we recursively pop the smallest ranked, poppable subtree.

2 Controls

$$\frac{\mathsf{z}_{\mathsf{in}}(s, \mathsf{pkt}) = s' \qquad \mathsf{push}(q, \mathsf{pkt}) = q'}{\mathsf{push}((s, q, \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}}), \mathsf{pkt}) = (s', q', \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}})}\ \text{RioCtrl-Push}$$

$$\frac{\mathsf{z}_{\mathsf{out}}(s) = (o, s') \qquad \mathsf{pop}(q, o) = (\mathsf{pkt}, q')}{\mathsf{pop}((s, q, \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}})) = (\mathsf{pkt}, (s', q', \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}}))}\ \text{RioCtrl-Pop}$$

$$\frac{\mathsf{z}_{\mathsf{in}}(s, \mathsf{pkt}) = (pt, s') \qquad \mathsf{push}(q, \mathsf{pkt}, pt) = q'}{\mathsf{push}((s, q, \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}}), \mathsf{pkt}) = (s', q', \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}})}\ \text{PIFOCtrl-Push}$$

$$\frac{\mathsf{pop}(q) = (\mathsf{pkt}, q') \qquad \mathsf{z}_{\mathsf{out}}(s, \mathsf{pkt}) = s'}{\mathsf{pop}((s, q, \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}})) = (\mathsf{pkt}, (s', q', \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}}))}\ \text{PIFOCtrl-Pop}$$

Figure 1. Pushing and Popping Controls

For these notes, we'll refer to *controls* from citeOG as *PIFO controls*.

Definition 2.1. For $t \in \mathbf{Topo}$, $\mathbf{PIFOControl}(t)$ is the set of quadruples of $(s, q, \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}})$ where

$$\mathsf{z}_{\mathsf{in}} : \mathbf{St} \times \mathbf{Pkt} \to \mathbf{Path}(t) \times \mathbf{St}$$
$$\mathsf{z}_{\mathsf{out}} : \mathbf{St} \times \mathbf{Pkt} \to \times \mathbf{St}$$

$s \in \mathbf{St}$ and $q \in \mathbf{PIFOTree}(t)$.

Here's the dequeue side version.

Definition 2.2. For $t \in \mathbf{Topo}$, $\mathbf{RioControl}(t)$ is the set of quadruples of $(s, q, \mathsf{z}_{\mathsf{in}}, \mathsf{z}_{\mathsf{out}})$ where

$$\mathsf{z}_{\mathsf{in}} : \mathbf{St} \times \mathbf{Pkt} \to \mathbf{St}$$
$$\mathsf{z}_{\mathsf{out}} : \mathbf{St} \to \mathbf{OrdTree}(t) \times \mathbf{St}$$

$s \in \mathbf{St}$ and $q \in \mathbf{RioTree}(t)$.

Both types of controls support push and pop operations:

$$\mathsf{push} : \mathbf{RioControl}(t) \times \mathbf{Pkt} \to \mathbf{RioControl}(t) \qquad \mathsf{push} : \mathbf{PIFOControl}(t) \times \mathbf{Pkt} \to \mathbf{PIFOControl}(t)$$
$$\mathsf{pop} : \mathbf{RioControl}(t) \to \mathbf{Pkt} \times \mathbf{RioControl}(t) \qquad \mathsf{pop} : \mathbf{PIFOControl}(t) \to \mathbf{Pkt} \times \mathbf{PIFOControl}(t)$$

Their semantics are written out in full in Figure 1.