Aggregation (for a monoid B)

$$\mathsf{aggregate}_+ \in \mathbb{P}\,(A \times B) \to (A \mapsto B)$$

$$\mathsf{aggregate}_+\ R = \left\{ a \mapsto \sum_{(a,b)\in R} b \;\mid\; a \in \operatorname{dom} R \right\}$$

Stake Distribution (using functions and maps as relations)

$$\mathsf{stakeDistr} \in \mathsf{UTxO} \to \mathsf{DState} \to \mathsf{PState} \to \mathsf{Snapshot}$$

$$\mathsf{stakeDistr}\ utxo\ dstate\ pstate =$$

$$\left((\operatorname{dom} activeDelegs) \lhd (\mathsf{aggregate}_+\ stakeRelation)\right,\ delegations,\ poolParams\right)$$

where

$$(\ rewards,\ delegations,\ ptrs,\ \_,\ \_,\ \_) = dstate$$

$$(\ poolParams,\ \_,\ \_) = pstate$$

$$stakeRelation = \left(\left({\mathsf{stakeCred_b}}^{-1} \cup (\mathsf{addrPtr} \circ ptr)^{-1}\right) \circ (\operatorname{range} utxo)\right) \cup rewards$$

$$activeDelegs = (\operatorname{dom} rewards) \lhd delegations \rhd (\operatorname{dom} poolParams)$$

Figure 37: Stake Distribution Function

11.5 Snapshot Transition

The state transition types for stake distribution snapshots are given in Figure 38. Each snapshot consists of:

- *stake*, a stake distribution, which is defined in Figure 36 as a mapping of credentials to coin.
- *delegations*, a delegation map, mapping credentials to stake pools.
- *poolParameters*, storing the pool parameters of each stake pool.

The type Snapshots contains the information needing to be saved on the epoch boundary:

- $pstake_{mark}$, $pstake_{set}$ and $pstake_{go}$ are the three snapshots as explained in Section 11.1.
- *feeSS* stores the fees which are added to the reward pot during the next reward update calculation, which is then subtracted from the fee pot on the epoch boundary.

Snapshots

$$\mathsf{Snapshot} = \left(\begin{array}{rcll} \mathit{stake} & \in & \mathsf{Stake} & \text{stake distribution} \\ \mathit{delegations} & \in & \mathsf{Credential} \mapsto \mathsf{KeyHash}_{pool} & \text{stake delegations} \\ \mathit{poolParameters} & \in & \mathsf{KeyHash}_{pool} \mapsto \mathsf{PoolParam} & \text{pool parameters} \end{array}\right)$$

$$\mathsf{Snapshots} = \left(\begin{array}{rcll} pstake_{mark} & \in & \mathsf{Snapshot} & \text{newest stake} \\ pstake_{set} & \in & \mathsf{Snapshot} & \text{middle stake} \\ pstake_{go} & \in & \mathsf{Snapshot} & \text{oldest stake} \\ \mathit{feeSS} & \in & \mathsf{Coin} & \text{fee snapshot} \end{array}\right)$$

Snapshot transitions

$$\_ \vdash \_ \xrightarrow[\text{SNAP}]{} \_ \subseteq \mathbb{P}\left(\mathsf{LState} \times \mathsf{Snapshots} \times \mathsf{Snapshots}\right)$$

Figure 38: Snapshot transition-system types

The snapshot transition rule is given in Figure 39. This transition has no preconditions and results in the following state change:

- The oldest snapshot is replaced with the penultimate one.
- The penultimate snapshot is replaced with the newest one.
- The newest snapshot is replaced with one just calculated.
- The current fees pot is stored in *feeSS*. Note that this value will not change during the epoch, unlike the *fees* value in the UTxO state.

$$\text{Snapshot}\ \frac{\begin{array}{rcl}((utxo,\ \_, fees,\ \_),\ (dstate,\ pstate)) & := & lstate \\ stake & := & \mathsf{stakeDistr}\ utxo\ dstate\ pstate\end{array}}{lstate \vdash \begin{pmatrix} pstake_{mark} \\ pstake_{set} \\ pstake_{go} \\ feeSS \end{pmatrix} \xrightarrow[\text{SNAP}]{} \begin{pmatrix} \boldsymbol{stake} \\ \boldsymbol{pstake_{mark}} \\ \boldsymbol{pstake_{set}} \\ \boldsymbol{fees} \end{pmatrix}} \tag{21}$$

Figure 39: Snapshot Inference Rule

11.6 Pool Reaping Transition

Figure 40 defines the types for the pool reap transition, which is responsible for removing pools slated for retirement in the given epoch.

Pool Reap State

$$\mathsf{PIReapState} = \left(\begin{array}{rcll} \mathit{utxoSt} & \in & \mathsf{UTxOState} & \text{utxo state} \\ \mathit{acnt} & \in & \mathsf{Acnt} & \text{accounting} \\ \mathit{dstate} & \in & \mathsf{DState} & \text{delegation state} \\ \mathit{pstate} & \in & \mathsf{PState} & \text{pool state} \end{array} \right)$$

Pool Reap transitions

$$\_ \vdash \_ \xrightarrow[\text{POOLREAP}]{} \_ \in \mathbb{P}\ (\mathsf{PParams} \times \mathsf{PIReapState} \times \mathsf{Epoch} \times \mathsf{PIReapState})$$

Figure 40: Pool Reap Transition

The pool-reap transition rule is given in Figure 41. This transition has no preconditions and results in the following state change:

- For each retiring pool, the refund for the pool registration deposit is added to the pool's registered reward account, provided the reward account is still registered.
- The sum of all the refunds attached to unregistered reward accounts are added to the treasury.
- The deposit pool is reduced by the amount of claimed and unclaimed refunds.
- Any delegation to a retiring pool is removed.
- Each retiring pool is removed from all four maps in the pool state.

$$
\begin{aligned}
\mathit{retired} &:= \mathrm{dom}\,(\mathit{retiring}^{-1}\,e)\\
\mathit{pr} &:= \{\mathit{hk} \mapsto (\mathsf{poolDeposit}\;\mathit{pp}) \mid \mathit{hk} \in \mathit{retired}\}\\
\mathit{rewardAcnts} &:= \{\mathit{hk} \mapsto \mathsf{poolRAcnt}\;\mathit{pool} \mid \mathit{hk} \mapsto \mathit{pool} \in \mathit{retired} \lhd \mathit{poolParams}\}\\
\mathit{rewardAcnts}' &:= \left\{a \mapsto \sum \mathit{pr}(\mathit{rewardAcnts}^{-1}(a)) \;\middle|\; a \in \mathrm{range}\,\mathit{rewardAcnts}\right\}\\
\mathit{refunds} &:= \mathrm{dom}\,\mathit{rewards} \lhd \mathit{rewardAcnts}'\\
\mathit{mRefunds} &:= \mathrm{dom}\,\mathit{rewards} \not\lhd \mathit{rewardAcnts}'\\
\mathit{refunded} &:= \sum_{\_ \mapsto c \in \mathit{refunds}} c\\
\mathit{unclaimed} &:= \sum_{\_ \mapsto c \in \mathit{mRefunds}} c
\end{aligned}
$$

Pool-Reap

$$
\mathit{pp} \vdash \begin{pmatrix} \mathit{utxo}\\ \mathit{deposited}\\ \mathit{fees}\\ \mathit{ppup}\\ \\ \mathit{treasury}\\ \mathit{reserves}\\ \\ \mathit{rewards}\\ \mathit{delegations}\\ \mathit{ptrs}\\ \mathit{genDelegs}\\ \mathit{fGenDelegs}\\ i_{rwd}\\ \\ \mathit{poolParams}\\ \mathit{fPoolParams}\\ \mathit{retiring} \end{pmatrix} \xrightarrow[\text{POOLREAP}]{e} \begin{pmatrix} \mathit{utxo} & & \\ \mathit{deposited} & - & (\mathit{unclaimed} + \mathit{refunded})\\ \mathit{fees} & & \\ \mathit{ppup} & & \\ & & \\ \mathit{treasury} & + & \mathit{unclaimed}\\ \mathit{reserves} & & \\ & & \\ \mathit{rewards} & \cup_+ & \mathit{refunds}\\ \mathit{delegations} & \rhd & \mathit{retired}\\ \mathit{ptrs} & & \\ \mathit{genDelegs} & & \\ \mathit{fGenDelegs} & & \\ i_{rwd} & & \\ & & \\ \mathit{retired} & \not\lhd & \mathit{poolParams}\\ \mathit{retired} & \not\lhd & \mathit{fPoolParams}\\ \mathit{retired} & \not\lhd & \mathit{retiring} \end{pmatrix} \tag{22}
$$

Figure 41: Pool Reap Inference Rule