11.7 Protocol Parameters Update Transition

Finally, reaching the epoch boundary may trigger a change in the protocol parameters. The protocol parameters environment consists of the delegation and pool states, and the signal is an optional new collection of protocol parameters The state change is a change of the UTxOState, the Acnt states and the current PParams. The type of this state transition is given in Figure 42.

New Proto Param environment

$$\mathsf{NewPParamEnv} = \left(\begin{array}{rcll} \mathit{dstate} & \in & \mathsf{DState} & \text{delegation state} \\ \mathit{pstate} & \in & \mathsf{PState} & \text{pool state} \end{array}\right)$$

New Proto Param States

$$\mathsf{NewPParamState} = \left(\begin{array}{rcll} \mathit{utxoSt} & \in & \mathsf{UTxOState} & \text{utxo state} \\ \mathit{acnt} & \in & \mathsf{Acnt} & \text{accounting} \\ \mathit{pp} & \in & \mathsf{PParams} & \text{current protocol parameters} \end{array}\right)$$

New Proto Param transitions

$$\_ \vdash \_ \xrightarrow[\mathsf{NEWPP}]{\_} \_ \subseteq \mathbb{P}\,(\mathsf{NewPParamEnv} \times \mathsf{NewPParamState} \times \mathsf{PParams}^{?} \times \mathsf{NewPParamState})$$

Helper Functions

$$\mathsf{updatePpup} \in \mathsf{UTxOState} \to \mathsf{PParams} \to \mathsf{UTxOState}$$

$$\mathsf{updatePpup}\ \mathit{utxoSt}\ \mathit{pp} = \begin{cases} (\mathit{utxo}, \mathit{deposited}, \mathit{fees}, (\mathit{fpup},\ \emptyset)) & \mathit{canFollow} \\ (\mathit{utxo}, \mathit{deposited}, \mathit{fees}, (\emptyset,\ \emptyset)) & \text{otherwise} \end{cases}$$

where

$$\mathit{canFollow} = \forall \mathit{ps} \in \operatorname{range} \mathit{pup},\ \mathit{pv} \mapsto v \in \mathit{ps} \implies \mathsf{pvCanFollow}\ (\mathsf{pv}\ \mathit{pp})\ v$$

$$(\mathit{utxo}, \mathit{deposited}, \mathit{fees}, (\mathit{pup},\ \mathit{fpup})) = \mathit{utxoSt}$$

Figure 42: New Proto Param transition-system types

Figure 43 defines the new protocol parameter transition. The transition has two rules, depending on whether or not the new protocol parameters meet some requirements. In particular, we require that the new parameters would not incur a debt of the system that can not be covered by the reserves, and that the max block size is greater than the sum of the max transaction size and the max header size. If the requirements are met, the new protocol parameters are accepted, the proposal is reset, and the reserves are adjusted to account for changes in the deposits. Otherwise, the only change is that the proposal is reset.

The NEWPP rule also cleans up the protocol parameter update proposals, by calling updatePpup on the UTxO state. The updatePpup sets the protocol parameter updates to the future protocol parameter updates provided the protocol versions all can follow from the version given in the protocol parameters, or the emptyset otherwise. In any case, the future protocol parameters update proposals are set to the empty set. If new protocol parameters are being adopted, then these is the value given to updatePpup, otherwise the old parameters are given.

Regarding adjusting the reserves for changes in the deposits, one of three things happens:

- If the new protocol parameters mean that **fewer** funds are required in the deposit pot to cover all possible refunds, then the excess is moved to the reserves.
- If the new protocol parameters mean that **more** funds are required in the deposit pot to

cover all possible refunds and the difference is **less** than the reserve pot, then funds are moved from the reserve pot to cover the difference.

- If the new protocol parameters mean that **more** funds are required in the deposit pot to cover all possible refunds and the difference is **more** than the reserve pot, then Rule 24 meets the precondition and the only change of state is that the update proposals are reset.

Note that here, unlike most of the inference rules in this document, the $utxoSt'$ and the $acnt'$ do not come from valid UTxO or accounts transitions in the antecedent. We simply define the consequent transition using these directly (instead of listing all the fields in both states in the consequent transition). It is done this way here for ease of reading.

$$\text{New-Proto-Param-Accept}\ \frac{\begin{array}{c} pp_{new} \neq \Diamond \\ \\ \begin{array}{rcl} (utxo,~deposited,~fees,~ppup) & := & utxoSt \\ (rewards,~\_,~\_,~\_,~\_,~i_{rwd}) & := & dstate \\ (poolParams,~\_,~\_) & := & pstate \\ oblg_{cur} & := & \mathsf{obligation}~pp~rewards~poolParams \\ oblg_{new} & := & \mathsf{obligation}~pp_{new}~rewards~poolParams \\ diff & := & oblg_{cur} - oblg_{new} \end{array} \\ \\ oblg_{cur} = deposited \\ reserves + diff \geq \sum\limits_{\_\mapsto val \in i_{rwd}} val \\ \mathsf{maxTxSize}~pp_{new} + \mathsf{maxHeaderSize}~pp_{new} < \mathsf{maxBlockSize}~pp_{new} \\ \\ utxoSt' := (utxo,~\boldsymbol{oblg_{new}},~fees,~ppup) \\ utxoSt'' := \mathsf{updatePpup}~utxoSt'~pp_{new} \\ \\ (treasury,~reserves) := acnt \\ acnt' := (treasury,~\boldsymbol{reserves + diff}) \end{array}}{\begin{array}{l} dstate \\ pstate \end{array} \vdash \left(\begin{array}{r} utxoSt \\ acnt \\ pp \end{array}\right) \xrightarrow[\text{NEWPP}]{pp_{new}} \left(\begin{array}{r} \boldsymbol{utxoSt''} \\ \boldsymbol{acnt'} \\ \boldsymbol{pp_{new}} \end{array}\right)} \tag{23}$$

$$\text{New-Proto-Param-Denied}\ \frac{\begin{array}{c} \left(\begin{array}{c} pp_{new} = \Diamond \\ \vee \\ reserves + diff < \sum\limits_{\_\mapsto val \in i_{rwd}} val \\ \vee \\ \mathsf{maxTxSize}~pp_{new} + \mathsf{maxHeaderSize}~pp_{new} \geq \mathsf{maxBlockSize}~pp_{new} \end{array}\right) \\ \\ \begin{array}{rcl} (rewards,~\_,~\_,~\_,~\_,~i_{rwd}) & := & dstate \\ (poolParams,~\_,~\_) & := & pstate \\ oblg_{cur} & := & \mathsf{obligation}~pp~rewards~poolParams \\ oblg_{new} & := & \mathsf{obligation}~pp_{new}~rewards~poolParams \\ diff & := & oblg_{cur} - oblg_{new} \end{array} \\ \\ utxoSt' := \mathsf{updatePpup}~utxoSt~pp \end{array}}{\begin{array}{l} dstate \\ pstate \end{array} \vdash \left(\begin{array}{r} utxoSt \\ acnt \\ pp \end{array}\right) \xrightarrow[\text{NEWPP}]{pp_{new}} \left(\begin{array}{r} \boldsymbol{utxoSt'} \\ acnt \\ pp \end{array}\right)} \tag{24}$$

Figure 43: New Proto Param Inference Rule

11.8 Complete Epoch Boundary Transition

Finally, it is possible to define the complete epoch boundary transition type, which is defined in Figure 44. The transition has no evironment. The state is made up of the the accounting state, the snapshots, the ledger state and the protocol parameters. The transition uses a helper function votedValue which returns the consensus value of update proposals in the event that consensus is met. **Note that** votedValue **is only well-defined if** *quorum* **is greater than half the number of core nodes, i.e.** Quorum $> |genDelegs|/2$ **.**

Epoch States

$$\mathsf{EpochState} = \left(\begin{array}{rcll} acnt & \in & \mathsf{Acnt} & \text{accounting} \\ ss & \in & \mathsf{Snapshots} & \text{snapshots} \\ ls & \in & \mathsf{LState} & \text{ledger state} \\ prevPp & \in & \mathsf{PParams} & \text{previous protocol parameters} \\ pp & \in & \mathsf{PParams} & \text{protocol parameters} \end{array}\right)$$

Epoch transitions

$$\vdash \_ \xrightarrow[\mathsf{EPOCH}]{\_} \_ \subseteq \mathbb{P}\,(\mathsf{EpochState} \times \mathsf{Epoch} \times \mathsf{EpochState})$$

Accessor Functions

$$\mathsf{getIR} \in \mathsf{EpochState} \to (\mathsf{Credential}_{stake} \mapsto \mathsf{Coin}) \quad \text{get instantaneous rewards}$$

Helper Functions

$$\mathsf{votedValue} \in (\mathsf{KeyHash}_{\mathsf{G}} \mapsto \mathsf{PParamsUpdate}) \to \mathsf{PParams} \to \mathbb{N} \to \mathsf{PParamsUpdate}^{?}$$

$$\mathsf{votedValue}\ pup\ pp\ quorum = \begin{cases} pp \underrightarrow{\cup} p & \exists! p \in \operatorname{range} pup\ (|pup \rhd p| \geq quorum) \\ \Diamond & \text{otherwise} \end{cases}$$

Figure 44: Epoch transition-system types

The epoch transition rule calls SNAP, POOLREAP and NEWPP in sequence. It also stores the previous protocol parameters in *prevPp*. The previous protocol parameters will be used for the reward calculation in the upcoming epoch, note that they correspond to the epoch for which the rewards are being calculated. Additionally, this transition also adopts the pool parameters *fPoolParams* corresponding to the pool re-registration certificates which we submitted late in the ending epoch. The ordering of these rules is important. The stake pools which will be updated by *fPoolParams* or reaped during the POOLREAP transition must still be a part of the new snapshot, and so SNAP must occur before these two actions. Moreover, SNAP sets the deposit pot equal to current obligation, which is a property that is preserved by POOLREAP and which is necessary for the preservation of Ada property in the NEWPP transition.

$$
\text{Epoch}\;
\frac{
\begin{array}{c}
\mathit{lstate}\ \vdash \mathit{ss} \xrightarrow[\text{SNAP}]{} \mathit{ss}' \\[1ex]
(\mathit{utxoSt},\ (\mathit{dstate},\ \mathit{pstate})) := \mathit{ls} \\
(\mathit{poolParams},\ \mathit{fPoolParams},\ \mathit{retiring}) := \mathit{pstate} \\
\mathit{pstate}' := (\mathit{poolParams} \underset{\rightarrow}{\cup} \mathit{fPoolParams},\ \emptyset,\ \mathit{retiring}) \\[2ex]
\mathit{pp} \vdash \begin{pmatrix} \mathit{utxoSt} \\ \mathit{acnt} \\ \mathit{dstate} \\ \mathit{pstate}' \end{pmatrix} \xrightarrow[\text{POOLREAP}]{e} \begin{pmatrix} \mathit{utxoSt}' \\ \mathit{acnt}' \\ \mathit{dstate}' \\ \mathit{pstate}'' \end{pmatrix} \\[2ex]
(\_,\ \_,\ \_,\ (\mathit{pup}, \_)) := \mathit{utxoSt}' \\
\mathit{pp}_{\mathit{new}} := \mathsf{votedValue}\ \mathit{pup}\ \mathit{pp}\ \mathsf{Quorum} \\
\begin{array}{c} \mathit{dstate}' \\ \mathit{pstate}'' \end{array} \vdash \begin{pmatrix} \mathit{utxoSt}' \\ \mathit{acnt}' \\ \mathit{pp} \end{pmatrix} \xrightarrow[\text{NEWPP}]{\mathit{pp}_{\mathit{new}}} \begin{pmatrix} \mathit{utxoSt}'' \\ \mathit{acnt}'' \\ \mathit{pp}' \end{pmatrix} \\[2ex]
\mathit{ls}' := (\mathit{utxoSt}'',\ (\mathit{dstate}',\ \mathit{pstate}''))
\end{array}
}{
\vdash \begin{pmatrix} \mathit{acnt} \\ \mathit{ss} \\ \mathit{ls} \\ \mathit{prevPp} \\ \mathit{pp} \end{pmatrix} \xrightarrow[\text{EPOCH}]{e} \begin{pmatrix} \boldsymbol{\mathit{acnt}''} \\ \boldsymbol{\mathit{ss}'} \\ \boldsymbol{\mathit{ls}'} \\ \boldsymbol{\mathit{pp}} \\ \boldsymbol{\mathit{pp}'} \end{pmatrix}
}
\tag{25}
$$

Figure 45: Epoch Inference Rule