MODULE *RollupsPhase*

Specification of the *Rollups* phase changes

$$
\begin{aligned}
\textsc{variables}\ & phase,\\
& inputAccumulationPeriodOver,\\
& challengePeriodOver,\\
& hasClaim,\\
& epochIsSealed
\end{aligned}
$$

Useful global definitions

$$
\begin{aligned}
&Phase \triangleq\\
&\quad \{\text{“InputAccumulation”},\\
&\quad\ \ \text{“AwaitingConsensus”},\\
&\quad\ \ \text{“AwaitingDispute”}\}
\end{aligned}
$$

Invariants

$$
\begin{aligned}
&TypeOK \triangleq\\
&\quad \land\ \ phase \in Phase\\
&\quad \land\ \ inputAccumulationPeriodOver \in \text{BOOLEAN}\\
&\quad \land\ \ challengePeriodOver \in \text{BOOLEAN}\\
&\quad \land\ \ hasClaim \in \text{BOOLEAN}\\
&\quad \land\ \ epochIsSealed \in \text{BOOLEAN}
\end{aligned}
$$

$$
\begin{aligned}
&TimeOK \triangleq\\
&\quad \land\ \ challengePeriodOver \Rightarrow epochIsSealed\\
&\quad \land\ \ epochIsSealed \Rightarrow inputAccumulationPeriodOver
\end{aligned}
$$

$$
\begin{aligned}
&EpochSealOK \triangleq\\
&\quad phase = \text{“InputAccumulation”} \equiv epochIsSealed = \text{FALSE}
\end{aligned}
$$

$$
\begin{aligned}
&HasClaimOK \triangleq\\
&\quad \land\ phase = \text{“InputAccumulation”} \Rightarrow hasClaim = \text{FALSE}\\
&\quad \land\ phase = \text{“AwaitingDispute”} \Rightarrow hasClaim = \text{TRUE}
\end{aligned}
$$

Initial state

$Init \triangleq$

$\quad \land\ phase = \text{"InputAccumulation"}$
$\quad \land\ inputAccumulationPeriodOver = \text{FALSE}$
$\quad \land\ challengePeriodOver = \text{FALSE}$
$\quad \land\ hasClaim = \text{FALSE}$
$\quad \land\ epochIsSealed = \text{FALSE}$

Next state

$EndInputAccumulationPeriod \triangleq$

$\quad \land\ inputAccumulationPeriodOver = \text{FALSE}$
$\quad \land\ inputAccumulationPeriodOver' = \text{TRUE}$
$\quad \land\ \text{UNCHANGED}\ \langle phase,\ challengePeriodOver,\ hasClaim,\ epochIsSealed \rangle$

$EndChallengePeriod \triangleq$

$\quad \land\ inputAccumulationPeriodOver = \text{TRUE}$
$\quad \land\ epochIsSealed = \text{TRUE}$
$\quad \land\ challengePeriodOver = \text{FALSE}$
$\quad \land\ challengePeriodOver' = \text{TRUE}$
$\quad \land\ \text{UNCHANGED}\ \langle phase,\ inputAccumulationPeriodOver,\ hasClaim,\ epochIsSealed \rangle$

We omit the behaviour of adding an input before the input accumulation period is over, because that does not change the state in this spec and behaviours that do not change the state might give a false negative for deadlocks when using the *TLC* Model Checker

$AddLateInput \triangleq$

$\quad \land\ phase = \text{"InputAccumulation"}$
$\quad \land\ inputAccumulationPeriodOver = \text{TRUE}$
$\quad \land\ phase' = \text{"AwaitingConsensus"}$
$\quad \land\ epochIsSealed' = \text{TRUE}$
$\quad \land\ \text{UNCHANGED}\ \langle inputAccumulationPeriodOver,\ challengePeriodOver,\ hasClaim \rangle$

We are abstracting away the validator from the specification, but a richer specification should keep track of claims from each validator so that they can't claim twice

We omit the behavior of a validator sending a non-conflicting claim, because that does not change the state in this specification

$Claim \triangleq$

$\quad \lor$ The input accumulation period is over, no user has sent an input yet, and a validator has submitted a claim, which changes the current phase. Since it is the first claim, there is no conflict.

$\qquad \land\ phase = \text{"InputAccumulation"}$
$\qquad \land\ inputAccumulationPeriodOver = \text{TRUE}$
$\qquad \land\ phase' = \text{"AwaitingConsensus"}$
$\qquad \land\ epochIsSealed' = \text{TRUE}$
$\qquad \land\ hasClaim' = \text{TRUE}$

```
        ∧ UNCHANGED ⟨inputAccumulationPeriodOver, challengePeriodOver⟩
     ∨   A late input has arrived and no validator has claimed yet Since it is the first claim, there
         is no conflict.
        ∧ phase = "AwaitingConsensus"
        ∧ hasClaim = FALSE
        ∧ hasClaim′ = TRUE
        ∧ UNCHANGED ⟨phase, inputAccumulationPeriodOver, challengePeriodOver, epochIsSealed⟩
     ∨   Some validator has claimed already, and now another validator makes a conflicting claim,
         which initiates a dispute
        ∧ phase = "AwaitingConsensus"
        ∧ hasClaim = TRUE
        ∧ phase′ = "AwaitingDispute"
        ∧ UNCHANGED ⟨inputAccumulationPeriodOver, challengePeriodOver, epochIsSealed, hasClaim⟩

ResolveDispute ≜
     ∨   Dispute is resolved before challenge period is over, and so we await for consensus from
         the rest of the validators
        ∧ phase = "AwaitingDispute"
        ∧ challengePeriodOver = FALSE
        ∧ phase′ = "AwaitingConsensus"
        ∧ UNCHANGED ⟨inputAccumulationPeriodOver, challengePeriodOver, epochIsSealed, hasClaim⟩
     ∨   Dispute is resolved after challenge period is over, and so we go directly towards the input
         accumulation period
        ∧ phase = "AwaitingDispute"
        ∧ challengePeriodOver = TRUE
        ∧ phase′ = "InputAccumulation"
        ∧ inputAccumulationPeriodOver′ = FALSE
        ∧ challengePeriodOver′ = FALSE
        ∧ hasClaim′ = FALSE
        ∧ epochIsSealed′ = FALSE

FinalizeEpoch ≜
     ∧ phase = "AwaitingConsensus"
     ∧ challengePeriodOver = TRUE
     ∧ hasClaim = TRUE
     ∧ phase′ = "InputAccumulation"
     ∧ inputAccumulationPeriodOver′ = FALSE
     ∧ challengePeriodOver′ = FALSE
     ∧ hasClaim′ = FALSE
     ∧ epochIsSealed′ = FALSE

Next ≜
     ∨ EndInputAccumulationPeriod
     ∨ EndChallengePeriod
     ∨ AddLateInput
```

$\lor \mathit{Claim}$
$\lor \mathit{ResolveDispute}$
$\lor \mathit{FinalizeEpoch}$

\* Modification History
\* Last modified *Fri Jun* 03 12:40:32 *BRT* 2022 by *guilherme*
\* Created *Wed Jun* 01 09:08:33 *BRT* 2022 by *guilherme*