MODULE *UpdateSystem*

EXTENDS *Integers*, *TLC*

VARIABLES *node_state*
, *blockchain_state*
, *syncedBlocksTick*
, *installerVersionBlock*
, *latestInstallerVersion*

This algorithm CAN deadlock, since we don't update the blockchain when running it. Why do we use values that are so small? To avoid combinatorial explosion of the state space.

We start with the fresh node, no block synced

$$
\begin{aligned}
Init \triangleq\ & \land\ node\_state && = 0 \\
& \land\ blockchain\_state && \in (1 \,..\, 216) \\
& \quad \text{The number of updates on the blockchain} \\
& \quad \land\ numOfUpdates && \in (0 \,..\, blockchain\_state) \\
& \land\ installerVersionBlock && \in (1 \,..\, 216) \\
& \land\ syncedBlocksTick && \in (1 \,..\, 20) \\
& \land\ latestInstallerVersion && = 0
\end{aligned}
$$

The formulas that need to be true for all states

$$
\begin{aligned}
Invariants \triangleq\ & \land\ node\_state \leq blockchain\_state \\
& \land\ \ \lor\ latestInstallerVersion = 0 \quad \text{We can make this a bit more precise} \\
& \qquad \lor\ latestInstallerVersion = installerVersionBlock
\end{aligned}
$$

We sync 1-50 blocks in one tick

$$
\begin{aligned}
SyncBlocks \triangleq\ & node\_state' = \text{IF}\ (node\_state + syncedBlocksTick) > blockchain\_state \\
& \qquad \text{THEN}\ blockchain\_state \\
& \qquad \text{ELSE}\ node\_state + syncedBlocksTick
\end{aligned}
$$

The situation when we change the installer version

$$
\begin{aligned}
CheckUpdates \triangleq\ & \text{IF}\ (node\_state < installerVersionBlock \land node\_state' > installerVersionBlock) \\
& \quad \text{THEN}\ latestInstallerVersion' = installerVersionBlock \\
& \quad \text{ELSE}\ \text{UNCHANGED}\ latestInstallerVersion
\end{aligned}
$$

Ideally, the blockchain would advance one block per tick, but the state explosion is VERY large.

$$
\begin{aligned}
RunNode \triangleq\ & \land\ SyncBlocks \\
& \land\ CheckUpdates \\
& \land\ \text{UNCHANGED}\ blockchain\_state \\
& \land\ \text{UNCHANGED}\ installerVersionBlock \\
& \quad \text{We want to change this every tick, since it's not a constant} \\
& \land\ syncedBlocksTick' = RandomElement(1 \,..\, 20)
\end{aligned}
$$

We stop if the blockchain is synced up

$$Next \triangleq RunNode$$

\ * Modification History
\ * Last modified *Tue Mar* 05 09:11:23 *CET* 2019 by *ksaric*
\ * Created *Tue Feb* 26 15:16:50 *CET* 2019 by *ksaric*