MODULE *Token*

EXTENDS *Naturals*, *TLC*, *FiniteSets*

CONSTANTS *CONTRACTS*, set of contracts in *Tezos*
INIT_TOKEN initial token amount

VARIABLES *tokenMap*, token amount state of contracts
pick current pick for model checking

---

some common helper operators

$Range(T) \triangleq \{\langle T[x], x\rangle : x \in \text{DOMAIN } T\}$
$Pick(S) \triangleq \text{CHOOSE } s \in S : \text{TRUE}$

RECURSIVE $SetReduce(\_, \_, \_)$
$SetReduce(Op(\_, \_), S, value) \triangleq$
IF $S = \{\}$ THEN $value$
ELSE LET $s \triangleq Pick(S)$
IN IF $Op(s[1], value) = Op(value, s[1])$
THEN $SetReduce(Op, S \setminus \{s\}, Op(s[1], value))$
ELSE $Assert(\text{FALSE}, \text{"error"})$

$Sum(S) \triangleq \text{LET } \_op(a, b) \triangleq a + b$
IN $SetReduce(\_op, S, 0)$

$TOKENTransfer(owner, receiver, amount) \triangleq$
IF $owner = receiver$
THEN UNCHANGED $tokenMap$
ELSE
$tokenMap' = [x \in CONTRACTS \mapsto$
CASE $x = owner \to tokenMap[x] - amount$
$\square$ $x = receiver \to tokenMap[x] + amount$
$\square$ OTHER $\to tokenMap[x]]$

$tokenMapChecker \triangleq$
$Sum(Range(tokenMap)) = Cardinality(CONTRACTS) * INIT\_TOKEN$

$Init \triangleq$
$\land\ tokenMap = [x \in CONTRACTS \mapsto INIT\_TOKEN]$
$\land\ pick = [$
$owner \mapsto RandomElement(CONTRACTS),$
$receiver \mapsto RandomElement(CONTRACTS),$
$amount \mapsto RandomElement(0 \,..\, INIT\_TOKEN * 2)$
$]$

$$
\begin{array}{l}
Next \triangleq \\
\quad \land\ pick' = [\\
\qquad owner \mapsto RandomElement(CONTRACTS), \\
\qquad receiver \mapsto RandomElement(CONTRACTS), \\
\qquad amount \mapsto RandomElement(0 \,..\, INIT\_TOKEN * 2) \\
\quad\ ] \\
\quad \land\ \text{IF } tokenMap[pick.owner] \geq pick.amount \\
\qquad \text{THEN } TOKENTransfer(pick.owner,\ pick.receiver,\ pick.amount) \\
\qquad \text{ELSE UNCHANGED } tokenMap
\end{array}
$$