MODULE *DAGConsensus*

Specification of a very simple DAG-based *BFT* consensus protocol.

Model-checking with *TLC* seems intractable beyond 4 rounds.

EXTENDS *FiniteSets*, *Integers*

CONSTANTS
N The set of nodes
, R set of rounds
, $Quorum$ The set of quorums
, $Leader(\_)$ operator mapping each round to its leader

ASSUME $\exists\, n \in R : R = 0 \,..\, n$

DAG vertices are just pairs consisting of a node and a round:

$V \triangleq N \times R$
$Node(v) \triangleq v[1]$
$Round(v) \triangleq v[2]$

A *digraph* is just a set of edges:

$IsDigraph(digraph) \triangleq \forall\, e \in digraph :$
$\quad \land\; e = \langle e[1],\, e[2] \rangle$
$\quad \land\; \{e[1],\, e[2]\} \subseteq V$

$Vertices(digraph) \triangleq \text{UNION } \{\{e[1],\, e[2]\} : e \in digraph\}$

$Children(v,\, digraph) \triangleq$
$\quad \{c \in V : \langle v,\, c \rangle \in digraph\}$

RECURSIVE $Reachable(\_,\, \_,\, \_)$
$Reachable(v1,\, v2,\, dag) \triangleq$
$\quad \lor\; v1 = v2$
$\quad \lor\; \exists\, c \in Children(v1,\, dag) : Reachable(c,\, v2,\, dag)$

$Parents(v,\, digraph) \triangleq$
$\quad \{e[1] : e \in \{e \in digraph : e[2] = v\}\}$

--algorithm *DAGConsensus* **{**
variables
$vs = \{\}$, the vertices of the *DAG*
$es = \{\}$ the edges of the *DAG*
define {
$Committed(v) \triangleq$
$\quad \land\; v \in vs$
$\quad \land\; Node(v) = Leader(Round(v))$
$\quad \land\; Round(v)\%2 = 0$
$\quad \land \{Node(p) : p \in Parents(v,\, es)\} \in Quorum$

$Correctness \triangleq$
$\forall v1, v2 \in vs :$
$\land\ Committed(v1)$
$\land\ Committed(v2)$
$\land\ Round(v1) \leq Round(v2)$
$\Rightarrow Reachable(v2, v1, es)$

}

process $(node \in N)$

variables

$round = 0$ **;** current round

$delivered = \{\}$ **;** delivered DAG vertices

{

$l0$: **while (** TRUE **)**

either {

deliver a vertice

with ($v \in vs \setminus delivered$ **)**

$delivered := delivered \cup \{v\}$

}

or {

create a new vertice

if ($round = 0$ **)**

$vs := vs \cup \{\langle self, round\rangle\}$

else with ($prev = \{v \in delivered : Round(v) = round - 1\}$ **) {**

when $(\{Node(p) : p \in prev\} \in Quorum)$ **;**

with ($v = \langle self, round\rangle$ **) {**

$vs := vs \cup \{v\}$ **;**

$es := es \cup \{\langle v, p\rangle : p \in prev\}$

}

} ;

$round := round + 1$

}

}

}

$TypeOK \triangleq$
$\land\ vs \subseteq V$
$\land\ \forall e \in es :$
$\land\ e = \langle e[1], e[2]\rangle$
$\land\ \{e[1], e[2]\} \subseteq V$
$\land\ Round(e[1]) > Round(e[2])$
$\land\ \forall n \in N :$
$\land\ round[n] \in Nat$
$\land\ delivered[n] \subseteq vs$

Model-checking stuff:

To define leaders, let's first order the nodes arbitrarily:

$NodeSeq \triangleq \text{CHOOSE } s \in [1 \,..\, Cardinality(N) \rightarrow N] :$
$\quad \forall\, i, j \in 1 \,..\, Cardinality(N) : i \neq j \Rightarrow s[i] \neq s[j]$

Example assignment of leaders to rounds (changes every 2 rounds):

$ModLeader(r) \triangleq NodeSeq[((r \div 2) \% Cardinality(N)) + 1]$

$StateConstraint \triangleq$
$\quad \text{LET } Max(S) \triangleq \text{CHOOSE } x \in S : \forall\, y \in S : y \leq x \text{IN}$
$\qquad \forall\, n \in N : round[n] \in 0 \,..\, (Max(R) + 1)$

$Falsy1 \triangleq \neg($
$\quad \exists\, v1, v2 \in vs :$
$\qquad \land\ v1 \neq v2$
$\qquad \land\ Committed(v1)$
$\qquad \land\ Committed(v2)$
$)$

$Falsy2 \triangleq \neg($
$\quad \exists\, v \in vs : Round(v) \neq 0 \land Committed(v)$
$)$