MODULE *ProofStatus*

EXTENDS *Naturals*, *Sequences*

PROOF OBLIGATION STATUSES

We define a possible proof-obligation status to be a mapping from provers to outcomes.

CONSTANT *Prover*

The set of all back-end provers, containing the elements "zenon" and "isabelle", among others.

$$ProofOutcome \triangleq \{\text{"notTried"}, \text{"failed"}, \text{"succeeded"}, \text{"stopped"}\}$$

$ObligationStatus \triangleq$

The set of all possible statuses for an obligation, where a status consists of a mapping from provers to the result of running that prover on the obligation, plus a triviality bit indicating whether or not *TLAPM* has determined the obligation to be trivial.

$$\begin{array}{ll}[proverStatus & : [Prover \to ProofOutcome],\\ \ triviality & : \text{BOOLEAN}\\ ]\end{array}$$

$$NullStatus \triangleq \begin{array}{ll}[proverStatus & \mapsto [p \in Prover \mapsto \text{"notTried"}],\\ \ triviality & \mapsto \text{FALSE}]\end{array}$$

THEOREM $NullStatus \in ObligationStatus$
BY DEF *NullStatus*, *ObligationStatus*, *ProofOutcome*

PROOF -OUTCOME TREES

We define an abstract proof tree that represents a step or theorem and proof statuses of all its obligations.

CONSTANTS *Obligation*, *Step*

We assume uninterpreted sets of obligations (or obligation ids) and steps.

$ProofOutcomeTree \triangleq$

An abstract representation of a theorem or proof step and its proof. This uses a standard TLA+ recursive definition of this sort of tree, where $P[0]$ is the set of steps with a (possibly missing or omitted) leaf proof, and $Pf[n]$ is the set of all proof trees of depth at most n.

$$\begin{array}{l}\text{LET } Pf[n \in Nat] \triangleq\\ \quad \text{IF } n = 0 \text{ THEN } \begin{array}{ll}[step & : Step,\\ \ leafProof & : \text{TRUE},\\ \ proofPresence & : \{\text{"missing"}, \text{"OMITTED"}\}]\\ \ \cup & \\ [step & : Step,\\ \ leafProof & : \text{TRUE},\\ \ proofPresence & : \{\text{"present"}\},\\ \ obligations & : \text{SUBSET } (Obligation \times ObligationStatus)]\end{array}\\ \qquad \text{ELSE } \begin{array}{ll}[step & : Step,\\ \ leafProof & : \text{FALSE},\end{array}\end{array}$$

$\quad\quad\quad\quad children \;\; : Seq(Pf[n-1])]$

$\quad\quad\quad\quad \cup\, Pf[n-1]$

IN UNION $\{Pf[n] : n \in Nat\}$

STATUS SPECIFICATIONS

CONSTANT $NumberOfStepStatuses$

For simplicity, a step status is a number from 1 to $NumberOfStepStatuses$.

$StepStatus \triangleq 1\,..\,NumberOfStepStatuses$

$StepStatusPredicate \triangleq$

A set of elements, each representing a predicate on a proof tree, as defined by $ApplyStatusPredicate$ below.

$[presence : \{\text{"missing"}, \text{"OMITTED"}\},$
$\; leaf : \text{BOOLEAN}$
$]$
$\cup$
$[presence : \{\text{"present"}\},$
$\; oStatus : Prover \times ProofOutcome,$
$\; leaf : \text{BOOLEAN}$
$]$

$ApplyStatusPredicate(pred, pfTree) \triangleq$

Defines the result of applying the $StepStatusPredicate$ $pred$ to the $ProofTree$ $pfTree$.

LET $ASP[pf \in ProofOutcomeTree] \triangleq$
$\quad \land\, pred.leaf \equiv pf.leaf$
$\quad \land\, pred.presence = pf.presence$
$\quad \land\, (pred.presence = \text{"present"}) \Rightarrow$
$\quad\quad$ IF $pred.leaf$
$\quad\quad\quad$ THEN $\exists\, ob \in pf.obligations :$
$\quad\quad\quad\quad ob[pred.oStatus[1]] = pred.oStatus[2]$
$\quad\quad\quad$ ELSE $\exists\, i \in pf.children : ASP[pf.children[i]]$
IN $ASP[pfTree]$

---

* Modification History
* Last modified *Wed Jun* 30 03:06:52 *PDT* 2010 by *lamport*
* Created *Wed Jun* 30 01:51:58 *PDT* 2010 by *lamport*