

MACHINE raffinement2

machine pour prendre en compte les autorisations accès piste

REFINES raffinement1

SEES contexte2

VARIABLES

ensAvionsTarmac ensemble avions sur tarmac inclus dans ensAvions
ensAvionsDecollage ensemble avion sur piste decollage inclus dans ensAvions
ensAvionsAtterrissage ensemble avion sur piste atterrissage inclus dans ensAvions
demandeAcces demande d'accès
pisteOK ensemble de pistes dont l'accès a été autorisée à un avion
pisteKO ensemble de pistes dont l'accès a été refusé à un avion

INVARIANTS

inv1: $ensAvionsTarmac \subseteq ensAvions$
inclus dans ensAvions
inv2: $ensAvionsDecollage \subseteq ensAvions$
inclus dans ensAvions
inv3: $ensAvionsAtterrissage \subseteq ensAvions$
inclus dans ensAvions
inv4: $ensAvionsTarmac \cup ensAvionsDecollage \cup ensAvionsAtterrissage \subseteq dom(clear)$
inv5: $demandeAcces \subseteq clear$
une demande d'accès est une clearance
inv6: $demandeAcces \in ensAvions \leftrightarrow ensPistes \setminus \{tarmac\}$
injection partielle entre les avions et les pistes hors tarmac
inv7: $ensAvionsTarmac \cap ensAvionsDecollage = \emptyset$
inv8: $ensAvionsDecollage \cap ensAvionsAtterrissage = \emptyset$
inv9: $finite(ensAvionsTarmac)$
ensembles finis pour facilité la tâche du prouver
inv10: $finite(ensAvionsDecollage)$
ensembles finis
inv11: $finite(ensAvionsAtterrissage)$
ensembles finis
inv12: $pisteOK \subseteq ensPistes$
inv13: $pisteKO \subseteq ensPistes$
inv14: $finite(pisteKO)$
ensembles finis
inv15: $finite(pisteOK)$
ensembles finissent
inv16: $nbAvionsDecollage = card(ensAvionsDecollage)$
invariant de glue avec raffinement1
inv17: $nbAvionsTarmac = card(ensAvionsTarmac)$
invariant de glue avec raffinement1
inv18: $nbAvionsAtterrissage = card(ensAvionsAtterrissage)$
invariant de glue avec raffinement1

EVENTS

Initialisation

begin

act1: $ensAvionsTarmac := \emptyset$
act2: $ensAvionsDecollage := \emptyset$
act3: $ensAvionsAtterrissage := \emptyset$
act5: $demandeAcces := \emptyset$
act6: $pisteOK := \emptyset$
act7: $pisteKO := \emptyset$

end

Event decoller $\langle ordinary \rangle \hat{=}$

```

refines decoller
  any
    a un avion
  where
    grd1:  $a \in \text{ensAvionsDecollage}$ 
  then
    act1:  $\text{ensAvionsDecollage} := \text{ensAvionsDecollage} \setminus \{a\}$ 
  end
Event atterrir <ordinary>  $\hat{=}$ 
refines atterrir
  any
    a
  where
    grd1:  $a \in \text{ensAvions}$ 
    grd2:  $\text{card}(\text{ensAvionsAtterrissage}) + \text{card}(\text{ensAvionsTarmac}) < \text{ntmax}$ 
    grd3:  $\text{ensAvionsDecollage} = \emptyset$ 
  then
    act1:  $\text{ensAvionsAtterrissage} := \text{ensAvionsAtterrissage} \cup \{a\}$ 
  end
Event entrerPiste <ordinary>  $\hat{=}$ 
refines entrerPiste
  any
    a
    p
  where
    grd1:  $a \in \text{ensAvionsTarmac}$ 
    grd2:  $p \in \text{pisteOK}$ 
    grd3:  $\text{ensAvionsAtterrissage} = \emptyset$ 
    grd4:  $a \mapsto p \in \text{demandeAcces}$ 
  then
    act1:  $\text{ensAvionsTarmac} := \text{ensAvionsTarmac} \setminus \{a\}$ 
    act2:  $\text{ensAvionsDecollage} := \text{ensAvionsDecollage} \cup \{a\}$ 
  end
Event sortirPiste <ordinary>  $\hat{=}$ 
refines sortirPiste
  any
    a
    p
  where
    grd1:  $a \in \text{ensAvionsAtterrissage}$ 
    grd2:  $p \in \text{pisteOK}$ 
  then
    act1:  $\text{ensAvionsTarmac} := \text{ensAvionsTarmac} \cup \{a\}$ 
    act2:  $\text{ensAvionsAtterrissage} := \text{ensAvionsAtterrissage} \setminus \{a\}$ 
  end
Event accepterDemande <ordinary>  $\hat{=}$ 
  any
    a un avion
    p une piste
  where
    grd1:  $a \in \text{ensAvions}$ 
    a est un avion
    grd2:  $p \in \text{ensPistes} \setminus \{\text{tarmac}\}$ 
    la piste est une piste mais pas un tarmac
    grd3:  $a \notin \text{dom}(\text{demandeAcces})$ 
    a ne fait pas partie du domaine
    grd4:  $p \notin \text{ran}(\text{demandeAcces})$ 
    p ne fait pas partie de l'image de la reaction a vers p

```

```

    grd5:  $a \mapsto p \in clear$ 
    grd6:  $p \notin pisteOK \cup pisteKO$ 
           l'avion n'est ni en attente ni déjà autorisé à accéder piste p
  then
    act1:  $demandeAcces(a) := p$ 
  end
Event refuserDemande ⟨ordinary⟩  $\hat{=}$ 
  any
    a un avion
    p une piste
  where
    grd1:  $a \in ensAvions$ 
    grd2:  $p \in ensPistes \setminus \{tarmac\}$ 
    grd3:  $p \notin pisteOK \cup pisteKO$ 
  then
    act1:  $pisteKO := pisteKO \setminus \{p\}$ 
  end
Event libererPisteOK ⟨ordinary⟩  $\hat{=}$ 
  any
    p
  where
    grd1:  $p \in pisteOK$ 
  then
    act1:  $pisteOK := pisteOK \setminus \{p\}$ 
    act2:  $demandeAcces := demandeAcces \triangleright \{p\}$ 
  end
Event libererPisteKO ⟨ordinary⟩  $\hat{=}$ 
  any
    p
  where
    grd1:  $p \in pisteKO$ 
  then
    act1:  $pisteKO := pisteKO \setminus \{p\}$ 
  end
END

```