Obligations de preuve

Qu’est-ce qu’une obligation de preuve ?

Une **obligation de preuve (Proof Obligation - PO)** en Event-B est une assertion qui doit être démontrée vraie afin de garantir la correction logique d’un modèle. Elle joue un rôle crucial pour vérifier que les invariants, les événements, et les raffinements respectent les propriétés spécifiées.
Les obligations de preuve sont générées automatiquement par des outils de preuve et servent de guide pour structurer et compléter les démonstrations nécessaires.

Une obligation de preuve prouvée peut-elle être considérée comme un théorème ?

Oui, en Event-B, une **obligation de preuve prouvée (discharged)** peut être considérée comme un théorème.

- **Théorème** : Une assertion démontrée vraie dans le cadre logique de la méthode.
- **PO** : Une obligation de preuve doit être prouvée (déchargée) pour qu’elle soit validée comme un théorème.

Comment ces obligations de preuve sont-elles générées ?

Les obligations de preuve sont **générées automatiquement** par des outils de preuve à partir des axiomes, définitions et principes fondamentaux.

- Les algorithmes et heuristiques identifient les POs nécessaires pour démontrer la correction d’un modèle.
- Les outils analysent chaque événement, invariant et raffinement pour produire les assertions à prouver.

Pourquoi les obligations de preuve peuvent-elles être générées automatiquement ?

Elles peuvent être générées automatiquement grâce à l’application d’algorithmes standardisés basés sur les règles déterministes du modèle Event-B. Ces outils permettent de garantir que les variables modifiées satisfont toujours les invariants.

Invariant

Quel est le rôle d’un invariant dans la spécification des propriétés d’un système ?

Un invariant est une propriété qui doit toujours être vraie pour un système donné. Il impose des contraintes sur le comportement du système afin de garantir qu'il respecte certaines propriétés essentielles.

Pourquoi l'invariant est-il maintenu par l'événement d'initialisation et chaque événement d'une machine ?

- **Initialisation** : L'événement initialise l'état du système et doit garantir que les invariants sont valides dès le départ.
- **Événements** : Chaque événement représente une transition du système et doit également préserver les invariants pour maintenir la validité tout au long de l'exécution.

Quel mécanisme de preuve est associé à la preuve d'un invariant ?

Les mécanismes courants incluent :

1. **Preuve par induction** : Montrer que l'invariant est vrai à l'état initial et qu'il est préservé par toutes les transitions.
2. **Contraposée** : Prouver que la négation de l'invariant conduit à une contradiction.
3. **Preuve directe** : Utiliser directement les axiomes et définitions pour prouver l'invariant.

Pourquoi différencier invariants et théorèmes dans une machine ?

- **Invariants** : Spécifient les contraintes à respecter en permanence par le système.
- **Théorèmes** : Sont des résultats démontrés basés sur les invariants et les propriétés du modèle.
  La distinction permet de séparer les propriétés à garantir (invariants) des résultats déduits (théorèmes).

Variant

Quelles sont les obligations de preuve associées aux variants ?

- Les événements dits **convergents** doivent prouver que le variant décroît. Cela se traduit par des POs montrant que la valeur du variant diminue strictement après l'exécution de l'événement.

Quelles propriétés peut-on garantir grâce à un variant décroissant ?

1. **Terminaison** : Assure que le système finit par atteindre un état stable sans transitions supplémentaires.
2. **Borne** : Garantit que le système reste dans des limites définies.
3. **Équité** : Permet de s'assurer que tous les événements activables seront exécutés.

Qu’est-ce qu’un variant et quelles propriétés établit-il ?

Un **variant** est une variable qui change au cours de l’exécution d’une machine. Il permet de décrire le comportement dynamique et temporel d’un système.
Caractéristiques :

- **Variable** : Change au fil des événements.
- **Décroissant** : Diminue strictement à chaque événement convergent.
- **Associé aux événements** : Sa modification est liée à des transitions spécifiques.

Raffinement

Quel est le rôle de la relation de simulation dans un raffinement ?

La relation de simulation garantit que la machine concrète respecte et reproduit les propriétés de la machine abstraite. Cela implique que la machine concrète :

1. Ne viole pas les invariants de la machine abstraite.
2. Représente correctement les comportements décrits par la machine abstraite.