



Examen en Outils Formels

Date: 11/01/2026

Exercice 01: 3 pts

Gestion d'un guichet de banque

On souhaite modéliser le fonctionnement d'un guichet bancaire qui traite des clients arrivant dans une file d'attente. Le guichet ne peut servir qu'un client à la fois.

Description du système

Les clients arrivent aléatoirement et se placent dans la file d'attente.

Le guichet fonctionne selon les règles suivantes :

1. File d'attente

Chaque client qui arrive se place dans la file.

Tant qu'un client n'est pas en cours de service, il reste en attente.

2. Guichet / Service

Le guichet ne peut servir qu'un seul client à la fois.

Lorsqu'il est libre, il prend le premier client de la file.

Une fois le service terminé, le guichet redevient libre.

Question : spécifiez le système en termes de réseaux de Petri.

Exercice 02: 6 pts

On souhaite spécifier en **Z** un système de gestion d'un **parking** permettant de contrôler les entrées et sorties de voitures. Le parking possède une **capacité maximale** et ne peut jamais contenir plus de voitures que cette capacité. Chaque voiture est identifiée par une **immatriculation** (un élément d'un ensemble donné **VOITURE**). Le système doit gérer et contrôler les opérations suivantes :

1. État du système

Le système doit mémoriser :

- L'ensemble des **voitures actuellement dans le parking**,
- La **capacité maximale** du parking (un entier naturel strictement positif).

2. Opérations du système

Vous devez spécifier les opérations suivantes en **Z** :

✓ENTREE(parking) :

Permet l'entrée d'une voiture dans le parking. Conditions :

- La voiture ne doit pas déjà être présente,
- Le parking ne doit pas être plein (nombre de voitures < capacité).

Effets :

La voiture est ajoutée à l'ensemble des voitures présentes.

✓SORTIE(parking) :

Permet la sortie de la voiture du parking. Conditions :

- La voiture doit être présente dans le parking.

Effets :

La voiture est retirée de l'ensemble.

✓**EST_PRESENT** :

Opération de **consultation** indiquant si la voiture se trouve dans le parking ou non.

✓**NB_PLACES_LIBRES**

Renvoie le nombre de places encore disponibles (capacité – cardinal du parking).

3. Propriétés attendues

Votre spécification doit garantir :

Invariant :

$\#voitures \leq capacite$

Sécurité :

Aucune voiture ne peut entrer si le parking est plein ou si elle y est déjà.

Question : Donnez la spécification **Z** de ce système.

Exercice 03: 6 pts

Feux tricolores à un carrefour

Contexte : Deux routes croisées, une seule peut avoir le feu vert à la fois.

Objectif :

Modéliser deux feux (FeuRouteA, FeuRouteB) en alternance.

Ajouter des voitures qui respectent les feux.

Question 01 : Donnez la spécification **Promela** de ce système.

Question 02 : À vérifier :

Sûreté : Pas deux feux verts en même temps.

Exercice 04: 5 pts

Soit le théorème suivant : $\forall n, m \in \mathbb{N}: n + m = m + n$.

Question : Ecrire une preuve **Coq** de ce théorème (en d'autre terme montrer que l'opération $+$ est commutative).

Bon courage