

Examen

Modélisation et Vérification de Systèmes Concurrents et Temps-Réel

Durée 2 heures
Tous documents autorisés

Avertissement : Il vous est demandé d'apporter le plus grand soin dans la rédaction. Vous serez jugés plus sur la QUALITÉ que sur la QUANTITÉ de vos réponses.

Le barème est donné à titre purement indicatif.

Partie I Modélisation en LNT (10 points)

Le crible d'Ératosthène est un procédé qui permet de trouver tous les nombres premiers inférieurs à un certain entier naturel donné N . L'algorithme séquentiel procède par élimination : il s'agit de supprimer d'une table des entiers de 2 à N tous les multiples d'un entier. En supprimant tous les multiples, à la fin il ne restera que les entiers qui ne sont multiples d'aucun entier, et qui sont donc les nombres premiers.

On se propose de modéliser en LNT une version distribuée du crible d'Ératosthène.

Dans cet algorithme distribué, on considère un processus **GENERATOR** qui génère les entiers de 2 à N et les envoie vers un pipeline d'unités de calcul **UNIT**, selon le schéma de la figure 1. Chaque unité de calcul fonctionne comme un filtre : elle lit en séquence les entiers qui lui sont envoyés par son voisin de gauche et propage vers son voisin de droite seulement les entiers qui ne sont pas des multiples du premier entier lu. Le premier entier lu par chaque unité est donc un nombre premier, et signalé comme tel sur la porte **OUTPUT**, puisqu'il n'est le multiple d'aucun autre nombre inférieur. Pour calculer les m premiers nombres premiers compris entre 2 et N , il faut donc que le pipeline comporte m unités.

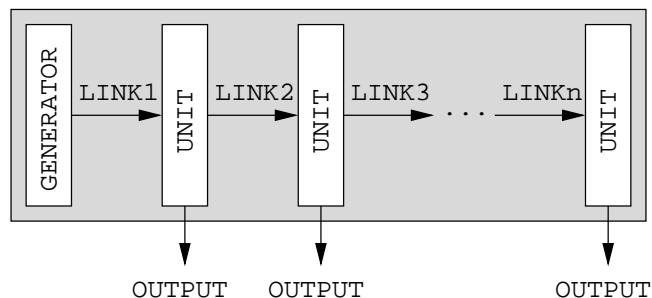


Figure 1: Architecture du crible d'Ératosthène distribué

Question I.1 Modélisation du générateur d'entiers

Le processus **GENERATOR** a le comportement cyclique suivant : il émet en séquence sur sa porte **RIGHT** les entiers naturels de 2 à N (inclus), puis devient inactif.

Compléter le comportement du processus `GENERATOR` ci-dessous.

```
process GENERATOR [RIGHT : Nat_Channel] (N : Nat) is
  -- partie à compléter (10 lignes environ)
  ...
end process
```

Note : On suppose que le channel `Nat_Channel` est défini comme suit :

```
channel Nat_Channel is (Nat) end channel
```

Question I.2 Modélisation d'une unité

Une unité a le comportement suivant :

1. Elle lit sur la porte `LEFT` l'entier naturel qui lui est envoyé par son voisin de gauche et l'envoie sur la porte `OUTPUT`.
2. Puis elle lit en boucle sur la porte `LEFT` chaque nouvel entier qui lui est envoyé par son voisin de gauche. Si cet entier n'est pas un multiple du premier entier lu à l'étape 1, alors elle le propage à destination de son voisin de droite sur la porte `RIGHT` avant de lire le prochain entier sur la porte `LEFT`. Sinon elle passe à l'entier suivant sans rien faire.

Compléter le comportement du processus `UNIT` ci-dessous :

```
process UNIT [LEFT, RIGHT, OUTPUT : Nat_Channel] is
  -- partie à compléter (12 lignes environ)
  ...
end process
```

Note : On suppose que l'on dispose de l'opération infixe `mod` qui calcule le reste de la division de deux entiers ($x \bmod y$).

Question I.3 Modélisation du système complet

Compléter la définition du processus `MAIN` ci-dessous, modélisant l'architecture du système constitué du générateur d'entiers et de 3 unités. On prendra pour N la valeur 5 et on tiendra compte du fait que la partie grisée de la figure 1 correspond à du comportement interne.

```
process MAIN [ ... (* déclaration des portes à compléter *) ] is
  -- partie à compléter (9 lignes environ)
  ...
end process
```

Question I.4 Système de transitions étiquetées

Le système de transition étiquetées correspondant au processus **MAIN** est donné figure 4 (page 6). Pour chacune des 9 transitions internes (étiquetées par **i**, attention sur les transitions dessinées verticalement l'étiquette **i** est cachée par le trait), indiquer l'action d'origine telle qu'elle était avant le masquage (opérateur **hide**).

Note : On répondra directement sur le sujet, dont on rendra la page 6.

Partie II Automates temporisés (5 points)

On se propose de modéliser en automates temporisés une version distribuée du crible d'Ératosthène. Pour cela on étend avec des contraintes temporelles la description du crible donnée dans la partie I ; l'énoncé de cette deuxième partie reprend l'énoncé de la première partie en mettant les contraintes temporelles en *italique*.

Pour le dessin, il vous est demandé de suivre les conventions syntaxiques suivantes :

- L'état initial est marqué par un trait double (comme en UPPAAL).
- L'invariant d'un état est inscrit à l'intérieur de l'état.
- Toute transition possède une étiquette de la forme " $g/a;r$ ", où g est la garde, a est l'action (un rendez-vous à la LNT) et r l'affectation d'horloges et de variables ; "*true*" représente la garde toujours valide, "*i*" l'action interne et "*null*" l'affectation vide.

Question II.1 Modélisation du générateur d'entiers

Le processus **GENERATOR** a le comportement cyclique suivant : il émet en séquence sur la porte **RIGHT** les entiers naturels de 2 à N (inclus), puis devient inactif. *Deux émissions consécutives sont séparées par un délai compris entre 2 et 4 unités de temps.*

Dessiner l'automate temporisé correspondant.

Question II.2 Propriétés temporelles du système complet

On considère le système composé de l'automate correspondant au processus **GENERATOR** de la question précédente et des deux automates de la figure 3 (page 5), correspondant aux deux premières unités de calcul du crible d'Ératosthène distribué.

Ces automates ont chacun une horloge locale **h** ainsi que deux variables locales **p** et **x** de type **Nat**. Ces automates se synchronisent comme montré sur la figure 1, c'est-à-dire que les seules synchronisations sont :

- sur la porte **LINK1** entre le générateur et la première unité de calcul, et
- sur la porte **LINK2** entre les deux unités de calcul.

Lors de l'exécution du système, l'action "**OUTPUT (3)**" aura lieu à un instant t . Donnez des bornes t_{min} et t_{max} , telles que $t_{min} < t < t_{max}$. Justifiez votre réponse.

Partie III Logique temporelle (5 points)

On considère le système de transitions étiquetées de la figure 2 (page 4) dont 0 est l'état initial. Indiquer pour chacune des dix formules ci-dessous les états qui la satisfont.

1. $\langle \text{true} \rangle \text{ true}$
2. $\langle \text{EAT} \rangle \text{ true}$
3. $[\text{WAKEUP}] \text{ false}$
4. $[\text{WORK}] \text{ true}$
5. $[\text{SLEEP}] \langle \text{WAKEUP} \rangle \text{ true}$
6. $\mu X. (\langle \text{WORK} \rangle \text{ true} \vee \langle \text{true} \rangle X)$
7. $\mu X. (\langle \text{WORK} \rangle \text{ true} \vee (\langle \text{true} \rangle \text{ true} \wedge [\text{true}] X))$
8. $\mu X. (\langle \text{true} \rangle \text{ true} \wedge [\neg \text{WORK}] X)$
9. $\nu X. (\langle \text{true}^* \cdot \text{WORK} \rangle \text{ true} \wedge [\neg \text{DIE}] X)$
10. $\nu X. \langle \neg \text{WORK} \rangle X$

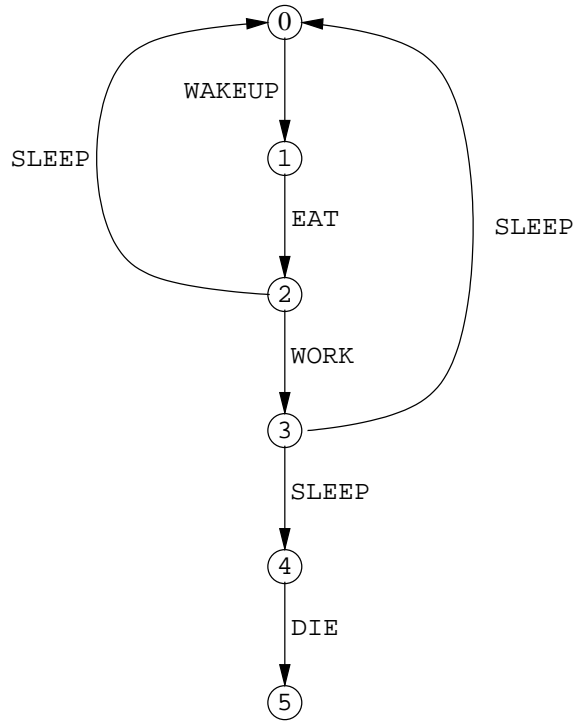


Figure 2: Système de transitions étiquetées de la Partie III

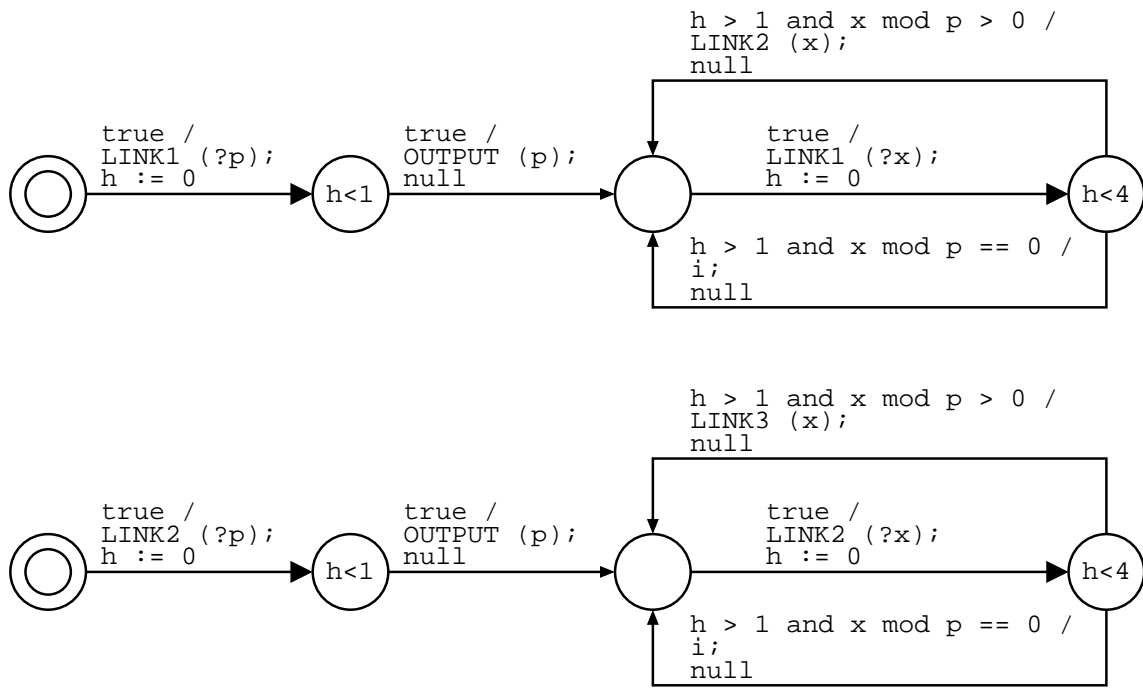


Figure 3: Automates temporisés des deux premières unités du crible d'Ératosthène (Question II.2)

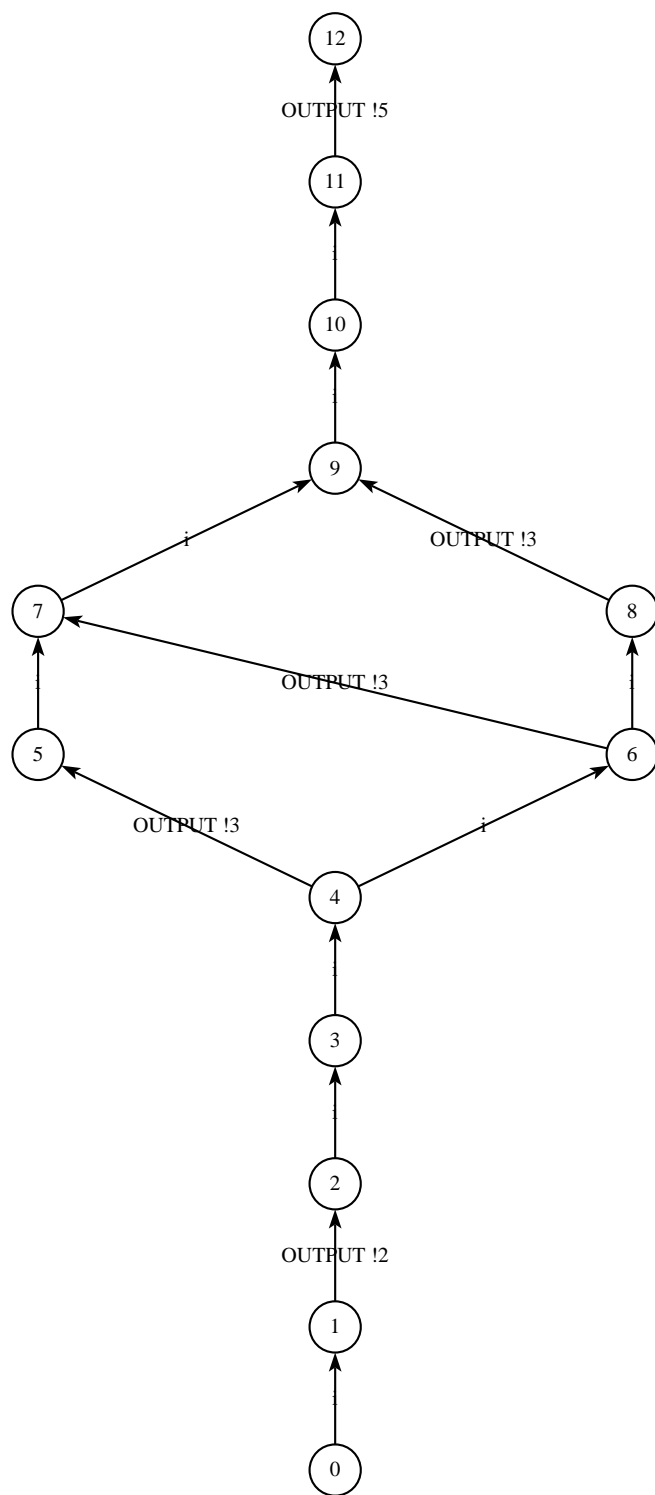


Figure 4: Système de transitions étiquetées de la Question I.4