

Outils formels de Modélisation

4^{ème} séance d'exercices

Aurélien Coet, Dimitri Racordon

Dans cette séance d'exercices, nous allons consolider notre étude des propriétés inhérentes aux réseaux de Petri. Notamment, nous étudierons les définitions formelles de ces propriétés.

1 Vous avez dit formelles ... (★★)

Soit un réseau de Petri défini par $N = \langle P, T, *, \Delta, \Delta^*, M_0 \rangle$. Pour chacune des propriétés exprimées formellement ci-dessous, dessinez un exemple de réseau:

1. $|P| = 3 \wedge \exists t \in T \text{ tq. } \forall p \in P : \Delta^*(t, p) = 1$
2. $1 < |P| < |T| \wedge \forall p \in P : \forall t \in T : M \rightarrow^t M' \implies M'(p) > M(p)$
3. $1 < |P| < |T| \wedge \exists s \in T^* \text{ tq. } M \rightarrow^s M' \implies M' \rightarrow^s M$
4. $\forall s \in T^* : M \rightarrow^s M' \implies M \notin R(M') \wedge \nexists M' \in R(M_0) \text{ tq. } \forall p \in P : M'(p) = 0$

2 Electrocardiogramme [🏠] (★★★)

Soit un réseau de Petri défini par $N = \langle P, T, *, \Delta, \Delta^*, M_0 \rangle$. On part de l'hypothèse que le réseau N a nécessairement un nombre fini de marquages atteignables.

1. Écrivez un algorithme qui prend en entrée un réseau N et une transition $t \in T$, et qui retourne vrai ou faux si t est respectivement vivante ou non.
2. Complétez la fonction `isAlive` dans le fichier `Analysis.fs` du projet `Exercise` avec une implémentation de votre algorithme.
3. Le fichier `Program.fs` contient une implémentation du modèle vu en cours à la page 21 des slides *Présentation des propriétés*. Jouez avec le marquage initial de ce modèle et observez son influence sur la vivacité du réseau. Quel est la condition nécessaire pour que le réseau soit vivant ?
4. Comment faudrait-il procéder pour que votre algorithme accepte en entrée un réseau avec un nombre de marquages potentiellement infini?