Examen du cours «Analyse d'algorithmes»

Rappel : seule une feuille de notes personnelles est autorisée pendant l'examen. Tout autre document est interdit.

Soyez concis et précis. On rappelle que le soin apporté aux justifications et aux preuves est déterminant.

1 Automates finis

1.1 Hors d'œuvre

Pour chaque langage régulier ci-dessous, donnez l'automate fini **déterministe et minimal** qui le reconnait.

1. le langage des mots sur l'alphabet $\{a, b\}$ dont le nombre de a est pair.
2. le langage des mots sur l'alphabet $\{a, b\}$ tels que tous les a apparaîssent par paire.
3. le langage des mots sur l'alphabet $\{a, b\}$ dont la longueur est un multiple de 3.

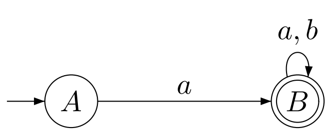
1.2 Produit d'automates

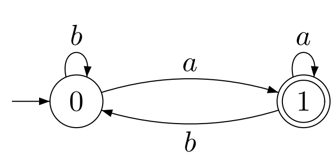
On rappelle qu'un automate fini $A = (Q, q_0, \Sigma, \delta, F)$ est défini par un ensemble fini d'états Q, un état initial $q_0 \in Q$, un alphabet Σ, une relation de transition $\delta \subseteq Q \times \Sigma \times Q$ et un ensemble d'états acceptants $F \subseteq Q$.

Soient $A = (Q, q_0, \Sigma, \delta, F)$ et $A' = (Q', q'_0, \Sigma', \delta', F')$ deux automates finis. L'automate produit $A \times A' = (Q'', q''_0, \Sigma'', \delta'', F'')$ de A et A' est défini par :

- $Q'' = Q \times Q'$,
- $q''_0 = (q_0, q'_0)$,
- $\Sigma'' = \Sigma \cap \Sigma'$,
- δ'' est définie pour tous états $(q, q'), (r, r') \in Q''$ et pour tout symbole $e \in \Sigma''$ par : $((q, q'), e, (r, r')) \in \delta''$ si $(q, e, r) \in \delta$ et $(q', e, r') \in \delta'$.
- $F'' = F \times F'$.

1. Calculez l'automate produit des deux automates suivants.





2. Quelle relation lie $\mathcal{L}(A)$, $\mathcal{L}(A')$ et $\mathcal{L}(A \times A')$, les langages reconnus par A, A' et $A \times A'$ respectivement ? Vous argumenterez votre réponse.

2 Preuves d'algorithmes

2.1 Première partie

On considère le while-program suivant qui prend en entrée un tableau d'entiers `T` triés en ordre croissant et un entier `x`.

```
i := 1;
tantque (i <= taille(T)) && (T[i] < x) faire
  i := i + 1
fintantque
```

Pour un tableau `T`, on note `taille(T)` le nombre d'éléments de `T`. On donne les préconditions et postconditions suivantes pour ce while-program :

- **Précond :** $\forall k \in [1; \texttt{taille}(\texttt{T}) - 1].\ \texttt{T}[k] \leq \texttt{T}[k+1]$
- **Postcond :** $\forall k \in [1; i-1].\ \texttt{T}[k] < x \wedge \forall k \in [i; \texttt{taille}(\texttt{T})].\ x \leq \texttt{T}[k] \wedge \forall k \in [1; \texttt{taille}(\texttt{T}) - 1].\ \texttt{T}[k] \leq \texttt{T}[k+1]$

1. Que fait ce while-program ?
2. Prouvez la correction partielle de ce while-program en prenant soin d'expliquer le choix de votre invariant de boucle.
3. Prouvez la terminaison de ce while-program. Vous expliquerez le choix de la mesure de convergence.

2.2 Deuxième partie

On complète le while-program précédent de la façon suivante :

```
i := 1;
tantque (i <= taille(T)) && (T[i] < x) faire
  i := i + 1
fintantque;
si (T[i] = x) alors
  skip
sinon
  i := 0
finsi
```

1. Que fait ce while-program ? Que signifie la valeur de `i` à la fin de son exécution ?
2. Sachant que la précondition du while-program est :

   $$\forall k \in [1; \texttt{taille}(\texttt{T}) - 1].\ \texttt{T}[k] \leq \texttt{T}[k+1]$$

   donnez une postcondition pour cet algorithme qui exprime le résultat de son exécution. Vous justifierez votre réponse.
3. Prouvez la correction partielle de ce while-program.
4. Que pouvez-vous dire au sujet de la terminaison de ce while-program ? Justifiez votre réponse.

Règles de la logique de Hoare

Règle de conséquence :

$$\frac{P' \Rightarrow P, \{P\}\,S\,\{Q\}\ ,\ Q \Rightarrow Q'}{\{P'\}\,S\,\{Q'\}}$$

Règle d'affectation :

$$\frac{}{\{P[e/x]\}\,x{:=}e\,\{P\}}$$

Règle de séquence :

$$\frac{\{P\}\,S_1\,\{Q\}\ ,\ \{Q\}\,S_2\,\{R\}}{\{P\}\,S_1;S_2\,\{R\}}$$

Règle du «`si`» :

$$\frac{\{P \wedge B\}\,S_1\,\{R\}\ ,\ \{P \wedge \neg B\}\,S_2\,\{R\}}{\{P\}\,\texttt{si}\ B\ \texttt{alors}\ S_1\ \texttt{sinon}\ S_2\ \texttt{finsi}\,\{R\}}$$

Règle du «`tantque`» :

$$\frac{\{P \wedge B\}\,S\,\{P\}}{\{P\}\,\texttt{tantque (B) faire S fintantque}\,\{P \wedge \neg B\}}$$

Terminaison des while-programs

On rappelle que pour prouver la terminaison d'une boucle :

```
tantque B faire
  S
fintantque
```

Il s'agit de trouver un terme t appelé «mesure» qui satisfait :

1. t est à valeurs dans un ensemble bien fondé $(\mathbb{D}, \leq)$: $\{B \wedge t \in \mathbb{D} \wedge P\}\,S\,\{t \in \mathbb{D}\}$
2. t est strictement décroissant : $\{B \wedge t = X \wedge P\}\,S\,\{t < X\}$

où P est n'importe quelle formule invariante : $\{P \wedge B\}\,S\,\{P\}$, X est une variable qui n'apparaît pas dans le while-program, et $t < X$ est défini par $t \leq X \wedge X \not\leq t$.