Bordeaux INP
ENSEIRB
MATMECA

Calculabilité & Complexité
Devoir Maison n° 2

21 février 2021

Rémi Dehenne
Sébastien Delpeuch
Tom Moënne-Loccoz
Aurélien Moinel

Exercice 1 : Réductions

On rappelle que le langage $L_{ACCEPT} = \{(M, w) | M \text{ accepte } w\}$ est indécidable.

Q1 Montrer que le langage $L_{REACH} = \{(M, q, w) | M \text{ visite l'état } q \text{ sur } w\}$ est indécidable en réduisant L_{ACCEPT}.

Entrées de $ACCEPT$: M Machine de Turing, w mot.

Entrées de $REACH$: M Machine de Turing, q état, w mot.

Pour toute entrée (M, x) de $ACCEPT$, nous associons l'entrée (M, q_{yes}, x).

Si $x \in L(M)$, M accepte x, donc M sur l'entrée x termine sur l'état q_{yes} en temps fini et passe donc par cet état.

Au contraire, si $x \notin L(M)$, M n'accepte pas x, M ne passe par l'état q_{yes}. En effet, cet état est terminal donc si M passe par l'état q_{yes}, M termine sur ce même état immédiatement. Or, M n'accepte pas x donc M ne termine pas sur q_{yes}. Cela implique que M ne passe pas par l'état q_{yes} sur l'entrée x.

Nous avons donc :

"$x \in L(M)$" $\iff$ "$M(x)$ passe par l'état q_{yes}"

Supposons qu'il existe une machine de Turing M_{REACH} qui calcule $REACH$.

Prenons la fonction $r : \{0,1\}^\star \to \{0,1\}^\star$ telle que $r(M, x) = (M, q_{yes}, x)$. Nous supposons que les doublets (machine et mot) et triplets (machine, état et mot) sont représentés par une chaîne binaire.

Nous avons que r est calculable dans la mesure où les paramètres M et x fournissent la sortie M et x sans transformation et q_{yes} est une constante.

Nous avons alors que $M_{ACCEPT}(M, x) = M_{REACH}(r(M, x))$. Cela implique que $ACCEPT$ se réduit à $REACH$, donc $ACCEPT$ est calculable puisque $REACH$ l'est. Cela est absurde puisque $ACCEPT$ n'est pas calculable, donc M_{REACH} n'existe pas et le langage L_{REACH} est indécidable.

Q2 Montrer que le langage $L^{\infty}_{REACH} = \{(M, q, w) | M \text{ visite l'état } q \text{ infiniment souvent sur } w\}$ est indécidable en réduisant L_{ACCEPT}.

Entrées de $ACCEPT$: M Machine de Turing, w mot.

Entrées de $REACH^{\infty}$: M Machine de Turing, q état, w mot.

Pour toute entrée (M, x) de $ACCEPT$, nous associons l'entrée (N, q_{trap}, x). La machine N consiste en la machine M avec :

- l'ajout d'un état supplémentaire q_{trap} ;
- toutes les transitions menant à q_{yes} sont redirigées sur l'état q_{trap} ;
- une transition de l'état q_{trap} à q_{trap} quelle que soit la lecture des rubans.

Si $x \in L(M)$, M accepte x donc M sur l'entrée x termine sur l'état q_{yes}. Or, par construction de N, il y a équivalence entre "M visite l'état q_{yes}" et "N visite l'état q_{trap}". Cela implique que N sur l'entrée x arrive à l'état q_{trap} et puisque celui-ci n'a qu'une transition vers lui-même quelle que soit la lecture, N visite cet état un nombre infini de fois.

Au contraire, si $x \notin L(M)$, M n'accepte pas x donc M sur l'entrée x ne termine pas sur l'état q_{yes} et par le même raisonnement que dans l'exercice précédent, M ne visite jamais l'état q_{yes}. Puisqu'il y a équivalence entre "M visite l'état q_{yes}" et "N visite l'état q_{trap}", N ne visite pas l'état q_{trap} donc en particulier, N ne visite pas cet état un nombre infini de fois.

Nous avons donc :

"$x \in L(M)$" $\iff$ "$N(x)$ passe par l'état q_{trap} un nombre infini de fois".

Supposons qu'il existe une machine de Turing M^{∞}_{REACH} qui calcule $REACH^{\infty}$.

Prenons la fonction $r : \{0,1\}^{\star} \to \{0,1\}^{\star}$ telle que $r(M,x) = (N, q_{trap}, x)$. Nous supposons que les doublets (machine et mot) et triplets (machine, état et mot) sont représentés par une chaîne binaire.

Nous avons que r est calculable dans la mesure où la machine N est issue du paramètre M selon une transformation définie sans ambiguïté et nécessitant un nombre fini de modifications de M. En outre, q_{trap} est une constante et x est issue du paramètre x sans transformation.

Nous avons alors que $M_{ACCEPT}(M,x) = M^{\infty}_{REACH}(r(M,x))$. Cela implique que $ACCEPT$ se réduit à $REACH^{\infty}$, donc $ACCEPT$ est calculable puisque $REACH$ l'est. Cela est absurde puisque $ACCEPT$ n'est pas calculable, donc M^{∞}_{REACH} n'existe pas et le langage L^{∞}_{REACH} est indécidable.

Exercice 2 : Décidabilité

Q3 Montrer que le problème qui suit est décidable :

— entrée : une machine de Turing à une bande M, un mot w et un entier positif n

— sortie : OUI si au cours de l'exécution de M sur l'entrée w, la tête de lecture/écriture sort du segment constitué des n premières cases de la bande, NON sinon

Pour prouver que ce problème est décidable, nous proposons une machine de Turing N qui décide de ce problème.

La machine N simule la machine M en effectuant trois vérifications.

D'abord, elle vérifie après chaque transition de la machine N si la tête de lecture de M sort du segment constitué des n premières cases de la bande. Si elle en sort, la machine N passe sur l'état q_{yes} et termine. Sinon, la vérification suivante est opérée.

Cette deuxième vérification consiste à vérifier que la machine M se trouve sur un état terminal, auquel cas la machine N transitionne sur l'état q_{no} et termine. Dans le cas contraire, une troisième vérification intervient.

Cette dernière vérification consiste à vérifier que la machine M a effectué plus de $|\Sigma|^n \times n \times |Q|$ transitions, auquel cas la machine N transitionne sur l'état q_{no} et termine. Si cette vérification n'est pas concluante, une nouvelle transition de la machine M est initiée et le cycle se répète.

Nous pouvons maintenant vérifier qu'une telle machine N décide bien du problème énoncé précédemment.

Si la machine M sur le mot d'entrée w sort du segment constitué des n premières cases en temps fini inférieur à $|\Sigma|^n \times n \times |Q|$, la première vérification de la machine N l'obligera à passer dans l'état q_{yes} et termine, et les deux autres vérifications n'arrêteront pas la machine M avant cela. Ce résultat est cohérent avec nos attentes et a lieu en temps fini (inférieur à $\texttt{constante} \times |\Sigma|^n \times n \times |Q|$).

Si la machine M termine sans être sortie du segment constitué par les n premières cases en temps fini inférieur à $|\Sigma|^n \times n \times |Q|$, la deuxième vérification de la machine N l'obligera à passer dans l'état q_{no} et termine, et les deux autres vérifications n'arrêteront pas la machine M avant cela. Ce résultat est lui aussi cohérent avec nos attentes et a lieu en temps fini (inférieur à $\texttt{constante} \times |\Sigma|^n \times n \times |Q|$).

Enfin, si la machine M ne termine pas en temps fini inférieur à $|\Sigma|^n \times n \times |Q|$ et ne sort pas des n premières cases pendant ce temps, la troisième vérification de la machine N l'obligera à passer dans l'état q_{no} et termine, et les deux autres vérifications n'arrêteront pas la machine M avant cela. Nous remarquons que ce résultat est lui aussi rendu en temps fini (inférieur à $\texttt{constante} \times |\Sigma|^n \times n \times |Q| + 1$).

Il faut maintenant montrer que ce dernier résultat est celui attendu par l'énoncé du problème. Cela revient à démontrer que si la machine M ne termine pas en temps fini inférieur à $|\Sigma|^n \times n \times |Q|$ et ne sort pas des n premières cases pendant ce temps, cette dernière ne sortira jamais du segment constitué des n premières cases de la bande.

Pour cela, il suffit de remarquer que sous l'hypothèse que la tête de lecture de M ne sort pas des n premières cases, seul le contenu de ces n premières cases est sujet à modification. L'état de la mémoire de la machine M est donc entièrement représenté par les symboles écrits sur ses n premières cases, la valeur de son registre d'état et la position de sa tête de lecture. Si la machine M devait se retrouver deux fois dans le même état de mémoire, le déterminisme des machines de Turing implique que la machine se trouverait dans une boucle de transitions où chaque boucle ne connaîtrait aucune variation de l'état de la mémoire à aucune transition.

Or, il n'y a que $|\Sigma|^n \times n \times |Q|$ états de la mémoire possibles selon l'hypothèse énoncée précédemment. Il est donc obligatoire qu'après plus de $|\Sigma|^n \times n \times |Q| + 1$ transitions, la machine M se soit retrouvée au moins deux fois dans l'exact même état de la mémoire. En particulier, cela implique qu'après un tel nombre de transitions,

il est impossible que la machine entre dans un état de mémoire différent d'un état de mémoire déjà visité et donc impossible que la machine M sorte des n premières cases quel que soit l'échéance temporelle.

Nous avons donc que dans ce dernier cas aussi, le résultat de notre machine N correspond bien aux attentes du problème.

Ces trois cas sur x forment l'univers complet des évènements possibles pour la machine N et tous ces cas sont traités correctement en temps fini. Nous avons donc que la machine N décide du problème énoncé initialement.