Machines universelles, Travaux pratiques et dirigés*

Cours d'Alain Colmerauer†

mai 2002

Travaux dirigés numéro 1

1 Schéma d'une machine de Turing

Une machine de Turing se schématise bien en représentant chaque état par un point et chaque instruction (q, a, a', q') par une flèche allant du point q au point q' et étiquetée par $\overleftarrow{aa'}$ ou $\overrightarrow{aa'}$ suivant que $d = \triangleleft$ ou $d = \triangleright$. Dessiner le schéma de la machine additionneuse M, d'alphabet $\{1, \sqcup\}$, qui est telle que $\overline{M}(1^m \sqcup 1^n) = 1^{m+n}$ et dont l'ensemble d'instructions est

$$\begin{array}{ll} \{(q_0, 1, 1, q_0, \triangleright), & (q_0, \sqcup, 1, q_1, \triangleleft), \\ (q_1, 1, 1, q_1, \triangleleft), & (q_1, \sqcup, \sqcup, q_2, \triangleright), \\ (q_2, 1, \sqcup, q_3, \triangleright), & (q_2, \sqcup, \sqcup, q_3, \triangleright)\} \end{array}$$

en omettant la dernière instruction, qui est inutile. Comprendre son fonctionnement à partir de son shéma.

1 Machine duplicatrice sur deux symboles

Dessiner le schéma de la machine duplicatrice M, d'alphabet $\{1, \sqcup\}$, qui est telle que $\overline{M}(x) = xx$, où x est une suite de 1. Prendre son temps.

3 Machine duplicatrice sur trois symboles

S'il reste du temps, dessiner le schéma de la machine duplicatrice M, d'alphabet $\{0, 1, \sqcup\}$, qui est telle que $\overline{M}(x) = xx$, où x est un mot sur {0,1}.

Travaux pratiques numéro 1

1 Outils *Maple* pour machine de Turing

Lancer *Maple* et, par un copier-coller, insérer le contenu du fichier `turingmapleter.txt` qui se trouve sur le site `alain.colmerauer.free.fr`. Jeter un coup d'oeil sur le début de ce fichier. Fixer l'alphabet des machines de Turing à deux symboles en effectuant la commande `fixeralphabet(iu,u);` Tester le fonctionnement de la machine `somme` en lançant par exemple les commandes `sur(somme,iiuiii);` et `sur(somme,iiuiii,1..30);` Attention les symboles $\sqcup, 1$ sont codés u, i.

2 Machine duplicatrice sur deux symboles

Mettre au point la machine duplicatrice à deux symboles conçu au TD précédent. La tester.

3 Machine duplicatrice sur trois symboles

Passer à un alphabet à trois symboles par la commande `fixeralphabet(oiu,u);` Attention les symboles $\sqcup, 0, 1$ sont codés u, o, i. Tester la machine de Turing `copie` qui est donnée comme exemple dans le fichier `Turing.txt` et qui duplique des mots sur l'alphabet $\{o, i\}$.

Travaux dirigés numéro 2

0 Reprise d'une démonstration intéressante

Reprendre et comprendre l'énoncé et la démonstration (reproduits ci-dessous) de l'impossibilté de construire une machine de Turing qui décide de l'arrêt d'une autre machine de Turing. On désigne par $T(\Sigma)$ l'ensemble des machines de Turing d'alphabet Σ et par *vrai* et *faux* deux mots particuliers sur Σ. Si $M \in T(\Sigma)$ alors $\mu(M)$ est un mot sur Σ qui code M. On rappelle que, si M est une machine de Turing, $\overline{M}(x)$ désigne, soit le résultat de l'exécution de M sur le mot x, soit ω si cette exécution ne se termine pas.

Théorème *Il n'existe pas de machine de Turing $A \in T(\Sigma)$ telle que $\overline{A}(x) \in \{vrai, faux\}$ et*

$$\overline{A}(\mu(M)\cdot x) = \begin{cases} \textit{vrai}, & \text{si } \overline{M}(x) \neq \omega, \\ \textit{faux}, & \text{si } \overline{M}(x) = \omega, \end{cases}$$

pour tout $x \in \Sigma^\star$ et $M \in T(\Sigma)$.

Preuve Supposons qu'il existe une telle machine A et montrons que l'on aboutit à une contradiction. En ajoutant quelques instructions à la machine A, on peut la modifier de façon à ce qu'elle boucle au lieu de sortir *vrai*, c'est-à-dire de façon à ce que

$$\overline{A}(\mu(M)\cdot x) = \begin{cases} \omega, & \text{si } \overline{M}(x) \neq \omega, \\ \textit{faux}, & \text{si } \overline{M}(x) = \omega, \end{cases}$$

*Cours de 2e année de DEUG (Diplôme d'études universitaires générales), option MIAS (Mathématiques, Informatique et Applications aux Sciences) et option MASS (Mathématiques Appliquées et Sciences Sociales)

†Laboratoire d'Informatique Fondamentale de Marseille, CNRS et Universités de Provence et de la Méditerranée

Il s'ensuit que

$$\overline{A}(\mu(M)\cdot x) = \omega \text{ ssi } \overline{M}(x) \neq \omega, \tag{1}$$

pour tout $x \in \Sigma^\star$ et $M \in T(\Sigma)$. Soit B la machine duplicatrice, définie dans le cours et qui est telle que $\overline{B}(x) = x\cdot x$. En enchaînant les exécutions de B et de cette nouvelle machine A, on peut alors construire une machine C telle que $\overline{C}(x) = \overline{A}(\overline{B}(x))$, donc telle que, pour tout $x \in \Sigma^\star$,

$$\overline{C}(x) = \overline{A}(x\cdot x). \tag{2}$$

En donnant à x la valeur $\mu(C)$ et à M la valeur C dans (1) et (2), on aboutit à :

$$\overline{C}(\mu(C)) = \overline{A}(\mu(C)\cdot\mu(C)),$$

$$\overline{A}(\mu(C)\cdot\mu(C)) = \omega \text{ ssi } \overline{C}(\mu(C)) \neq \omega,$$

d'où on déduit la contradiction

$$\overline{C}(\mu(C)) = \omega \text{ ssi } \overline{C}(\mu(C)) \neq \omega.$$

1 Machine : Décalage à droite

En s'aidant d'un dessin, écrire les instructions de la machine qui passe d'une configuration de la forme

$$Q0u: \; xu\underline{y_1}\dots y_n uu,$$

où uu n'est pas sous-mot de $uy_1\dots y_n u$, à la configuration

$$R0: \; xuuy_1\dots \underline{y_n}u.$$

Comme noms d'états, à part $R0$, utiliser uniquement des noms commençant par Q.

2 Machine : Propagation de o, i, u à gauche

En s'aidant d'un dessin, écrire les instructions de la machine qui passe d'une configuration de la forme

$$P0: \; uuy_1\dots y_n u\underline{x},$$

où uu n'est pas sous-mot de $uy_1\dots y_n u$, à la configuration

$$Q0u: \; xu\underline{y_1}\dots y_n uu.$$

Comme noms d'états, à part $Q0$, utiliser uniquement des noms commençant par P. Comprendre ce que fait cette machine suivie de la précédente.

3 Machine : Décalage à gauche

En s'aidant d'un dessin, écrire les instructions de la machine qui passe d'une configuration de la forme

$$Mu0: \; xuuy_1\dots \underline{y_n}u,$$

où uu n'est pas sous-mot de $uy_1\dots y_n u$, à la configuration

$$N0: \; \underline{x}uy_1\dots y_n uu.$$

Comme noms d'états, à part $R0$, utiliser uniquement des noms commençant par M.

4 Machine : Propagation de o, i, u à droite

En s'aidant d'un dessin, écrire les instructions de la machine qui passe d'une configuration de la forme

$$N0: \; \underline{x}uy_1\dots y_n uu,$$

où uu n'est pas sous-mot de $uy_1\dots y_n u$, à la configuration

$$R0: \; uuy_1\dots \underline{y_n}ux,$$

Comme noms d'états, à part $R0$, utiliser uniquement des noms commençant par N. Comprendre ce que fait cette machine précédée de la machine précédente.

Travaux pratiques numéro 2

1 Machine : Décalage à droite

Mettre au point cette machine en effectuant plusieurs tests.

2 Machine : Propagation de o, i, u à gauche

Mettre au point cette machine en effectuant plusieurs tests. Créer la machine 2 suivie de la machine 1 et la tester.

3 Machine : Décalage à gauche

Mettre au point cette machine en effectuant plusieurs tests.

4 Machine : Propagation de o, i, u à droite

Mettre au point cette machine en effectuant plusieurs tests. Créer la machine 3 suivie de la machine 4 et la tester.

Travaux dirigés numéro 3

1 Architecture d'une machine universelle sur $\{o, i, u\}$

Prendre connaissance

Prendre connaissance de l'architecture de la machine universelle U décrite ci-dessous.

Soit M une machine de Turing quelconque sur l'alphabet $\Sigma = \{o, i, u\}$ et soit un de ses couples : *configuration initiale, configuration finale,*

$$\dots \underset{\displaystyle x_1}{\overset{\boxed{q}}{\downarrow}} \; x_2 \dots x_m \dots$$

$$\dots \underset{\displaystyle x'_1}{\overset{\boxed{q'}}{\downarrow}} \; x'_2 \dots x'_{m'} \dots$$

où q, q' sont des états de M et les x_i, x'_i des éléments de Σ. Sur le même alphabet Σ, on se propose de construire une machine universelle U qui admettra un couple, *configuration initiale, configuration finale,* de la forme :

$$\dots \underset{\displaystyle y_1}{\overset{\boxed{r}}{\downarrow}} \; y_2 \dots y_d \, x_1 x_2 \dots x_m \dots$$

$$\dots y'_1 y'_2 \dots y'_d \; \underset{\displaystyle x'_1}{\overset{\boxed{r'}}{\downarrow}} \; x'_2 \dots x'_{m'} \dots$$

où r, r' sont des états de U, les y_i, y'_i des éléments de Σ et $y_1 y_2 \dots y_d$ un codage de M par un mot sur Σ.

A toute configuration courante de M de la forme

$$\dots v_1 \dots v_{i-1} \; \underset{\displaystyle v_i}{\overset{\boxed{s}}{\downarrow}} \; \dots v_k \dots \tag{3}$$

correspond une configurations de U de la forme

$$\dots v_1 \dots v_{i-1} \, w_1 \dots \underset{\displaystyle w_j}{\overset{\boxed{t}}{\downarrow}} \dots w_d \, v_i \dots v_k \dots \tag{4}$$

où s est un état de M, où t est un état de U et où les v_i, w_i sont des éléments de Σ. Le mot $w_1 \dots w_d$ code le couple (M, s). Il est de la forme

$$uuI_m uI_{m-1}u \dots uI_1 uI_0 uuI_k o \dots oiu$$

où chaque I_i est un mot sur $\{o, i\}$ qui représente une instruction de M et où $I_k o \dots o$ est une copie de la prochaine instruction à exécuter dans la configuration 3, complétée par des o à droite, de façon à obtenir un mot de taille fixe. D'une façon plus précise, si M a n états $q_0, q_1, \dots, q_n$ avec q_n, unique état final,

- $m = 3(n - 2)$,
- pour chaque état q_i non final, $I_{3i}, I_{3i+1}, I_{3i+2}$ sont les codages des instructions qu'il faut exécuter dans l'état q_i, suivant qu'on lit le symbole o, i ou u,
- I_k est une copie de la prochaine instruction à exécuter.

Chaque instruction codée I_j est de la forme

$$\underbrace{z_1}_{\text{mouvement}} \; \underbrace{z_2 z_3}_{\text{écriture}} \; \underbrace{z_4 \dots z_p}_{\text{où aller}}, \tag{5}$$

où

- z_1 vaut o ou i, suivant qu'il faut se déplacer à gauche ou à droite,
- $z_2 z_3$ vaut oo, io, oi, suivant qu'il faut écrire le symbole o, i, u,
- $z_4 \dots z_p$, un mot de taille variable pouvant être vide, représente un entier de la forme $3k$, où k est le numéro de l'état dans lequel il faut passer. Ce nombre $3k$ est écrit de gauche à droite, sous forme d'une nombre binaire, avec o pour 0 et i pour 1. S'il faut passer dans l'état final, $z_4 \dots z_p$ est une suite $i \dots i$ de i, toujours la même, qui code un entier strictement plus grand que le nombre m.

2 Calcul d'un codage de machine

Calculer le mot qui code la machine *complément*, d'état initial q_0, d'état final $q2$, et dont les instructions sont

$$(q_0, o, i, q_0, \triangleright), (q_0, i, o, q_0, \triangleright), (q_0, u, u, q_1, \triangleleft),$$
$$(q_1, o, o, q_1, \triangleleft), (q_1, i, i, q_1, \triangleleft), (q_1, u, u, q_2, \triangleright).$$

Il faut trouver

$$uu \, \overbrace{ioiiiiuoioiiuoooii}^{\text{instructions pour } q_1} u \; \overbrace{ooiiiuioouiio}^{\text{instructions pour } q_0} \; uu \, \overbrace{oooooo}^{\text{aller à } q_0} iu$$

où *oooooo* est une instruction fictive, de la forme (5), avec $z_1 z_2 z_3 = ooo$ et $z_4 \dots z_p$ mot vide. Elle est complétée par *ooo* et demande de passer à l'état initial q_0.

3 Fonctionnement de la machine universelle

Prendre connaissance du fonctionnement de la machine universelle décrit ci-dessous.

1. Initialisation

$$\overbrace{\underline{u}uI_mu\ldots uI_1uI_0uuoooKiux}^{\text{grosse tête}}$$

Ici K est un entier naturel codé sous forme binaire, avec o, i pour $0, 1$ et les poids faibles à gauches. On se positionne sur le premier o qui suit le deuxième uu et on passe à $\mathbf{2}$.

2. Test d'arrêt

$$uuI_mu\ldots uI_1uI_0uu\underline{o}ooKiux$$

Si K est une suite de i, on se positionne sur x et on arrête tout le processus. Sinon, étant positionné dans Ki, on passe à $\mathbf{3}$.

3. Lecture

$$uuI_mu\ldots uI_1uI_0uuoooKiux$$

On est positionné dans Ki. Suivant que x est o, i, u on ajoute $0, 1, 2$ à K. On remplace iux par ouu. On passe à $\mathbf{4}$, étant positionné sur le dernier symbole de K.

4. Recherche de l'instruction

$$uuI_mu\ldots uI_1uI_0u\,\overbrace{uoooKo}^{\text{petite tête}}\,uu$$

On est sur le dernier symbole de K. On suppose que K vaut k. On recule la petite tête, c'est-à-dire le bloc $uoooKo$. Chaque fois qu'on franchit un u on décrémente K d'une unité. Quand K devient négatif, c'est-à-dire une suite de i, on le remplace par une suite de o suivie de u. Etant positionné sur un o de le petite tête ainsi raccourcie, on passe à $\mathbf{5}$.

5. Enrégistrement de l'instruction

$$uuI_mu\ldots uI_k\,\overbrace{uo\ldots o}^{\text{petite tête}}\,uuI_{k-1}u\ldots uI_0uuu$$

On est positionné sur un o de la petite tête. On enrégistre dans la petite tête une copie (complétée de zéros à droite) de l'instruction I_k. Ceci se fait par reculs de la petites tête à droite et enregistrement au fur et à mesure des éléments de I_k jusqu'à arriver à une configuration de la forme

$$uuI_mu\ldots uI_{k+1}u\,\overbrace{\underline{u}I_ko\ldots ou}^{\text{petite tête}}\,I_kuI_{k-1}u\ldots uI_0uuu.$$

On remplace alors le dernier u de la petite tête, on se positionne sur le premier symbole de $I_ko\ldots o$ et en passe à $\mathbf{6}$.

6. Propagation de l'instruction à droite

$$uuI_mu\ldots uI_{k+1}uu\,\overbrace{I_ko\ldots oo}^{\text{petite tête}}\,I_kuI_{k-1}u\ldots uI_0uuu.$$

On consisdère maintenant que la petite tête est $I_ko\ldots o$ et on est positionné sur son premier symbole. On ramène la petite tête jusqu'à l'extrémité droite de la grosse tête. On remplace uuu par iiu, on se positionne deux symboles après le deuxième uu et on passe à $\mathbf{7}$.

7. Ecriture

$$uuI_mu\ldots uI_kuI_{k-1}u\ldots uI_0uuz_1\underline{z_2}z_3Kiiu$$

Ici $z_1z_2z_3K$ n'est rien d'autre que $I_ko\ldots o$. Le compteur K a donc changé de valeur. On remplace iiu par iuy, où y vaut o, i, u, suivant que z_2z_3 vaut oo, io, oi. Dans la foulée on remplace z_2z_3 par oo. Etant positionné sur le premier i de iuy, on passe à $\mathbf{8}$.

8. Test de mouvement

$$uuI_mu\ldots uI_kuI_{k-1}u\ldots uI_0uuz_1ooK\underline{i}uy$$

On remplace le deuxième uu par uo. Puis, si z_1 vaut o, on se repostionne sur le i qui suit K et on passe à $\mathbf{9}$. Si z_1 vaut i, on remplace z_1 par o, on se positionne sur y et on passe à $\mathbf{10}$.

9. Mouvement gauche

$$xuuI_mu\ldots uI_kuI_{k-1}u\ldots uI_0uooooK\underline{i}uy$$

On enlève x et on l'insère entre u et y. On se positionne sur le i qui suit K et on passe à $\mathbf{11}$.

10. Mouvement droit

$$uuI_mu\ldots uI_kuI_{k-1}u\ldots uI_0uooooKiu\underline{y}x$$

On enlève y pour le mettre devant le tout premier uu. On se positionne sur le i qui suit K et on passe à $\mathbf{11}$.

11. Retour au test d'arrêt

$$uuI_mu\ldots uI_kuI_{k-1}u\ldots uI_0uooooK\underline{i}uyx$$

Juste après I_0 on remplace uo par uu, on se positionne sur le symbole qui suit se uu et on retourne à $\mathbf{2}$.

4 Module : Ecriture

Concevoir le module d'écriture. Utiliser la lettre J pour les noms d'états en commençant par J0. L'état final est K0.

5 Module : Test de mouvement

Concevoir le module d'écriture. Utiliser la lettre K pour les noms d'états en commençant par K0. L'état final est P0, si on passe au mouvement droit, et M0u, si on passe au mouvement gauche.

6 Module : Retour au test d'arrêt

Concevoir le module de retour au test d'arrêt. Utiliser la lettre R pour les noms d'états en commençant par R0. L'état final est B0.

Travaux pratiques numéro 3

1 Mise en piste, nouvelle version

Récupérer le fichier `turingmapleter.txt` sur le site `alain.colmerauer.free.fr` qui, à signaler, ne commence pas par `www`. Insérer son contenu dans une fenêtre *Maple*. Faire `beau(complement);` pour voir les instructions de la machine *complément* et faire `code(complement);` pour voir son codage.

2 Module : Ecriture

Mettre au point le module 7. Ecriture.

3 Module : Test de mouvement

Mettre au point le module 8. Test de mouvement.

4 Module : Retour au test d'arrêt

Mettre au point le module 11. Test de mouvement.

5 Décodage et exécution d'une instruction.

Récupérer le canevas de la machine universelle, `canevasuniverselle.txt`, sur le site `alain.colmerauer.free.fr`. Insérer aux endroits prévus les programmes du TP 2 et les modules écrits au cours de ce TP et vérifier qu'ils s'enchaînent bien. Obtenir ainsi une partie de machine universelle, composée des modules 7,8,9,10,11. Cette partie permet de décoder une instruction et de l'exécuter.

Effectuer tous les tests concernant le TP 3, donnés en commentaire à la fin de `canevasuniverselle.txt`. Tant qu'ils ne produisent pas les résultats annoncés, ne pas passer au TP 4.

Travaux dirigés numéro 4

1 Module : Initialisation

Concevoir le module d'initialisation. Utiliser la lettre A pour les noms d'états en commençant par A0. L'état final est B0.

2 Module : Test d'arrêt

Concevoir le module de test d'arrêt. Utiliser la lettre B pour les noms d'états en commençant impérativement par B0. L'état final est Fin ou C0, suivant qu'on s'arrête ou qu'on ne s'arrête pas.

3 Module : Lecture

Concevoir le module de lecture. Utiliser la lettre C pour les noms d'états en commençant par C0. L'état final est impérativement D2u. Le plus simple est de programmer la boucle suivante, où x est le caractère qu'on doit lire :

1. avancer jusqu'à u puis se positionner sur x,
2. si $x = o$, remplacer x par u, remplacer le i qui précède x par o et c'est fini,
3. si $x = i$, remplacer x par o et passer à 5,
4. si $x = u$, remplace x par i et passer à 5,
5. reculer jusqu'à u, avancer jusqu'au début de K, ajouter 1 à K (dont les poids faibles sont à gauche) et retourner en 1.

Travaux pratiqes numéro 4

1 Module : Initialisation

Mettre au point le module 1. Initialisation.

2 Module : Test d'arrêt

Mettre au point le module 2. Test d'arrêt.

3 Module : Lecture

Mettre au point le module 3. Lecture.

4 Assemblage du début de la machine universelle

Récupérer le canevas de la machine universelle, `canevasuniverselle.txt`, sur le site `alain.colmerauer.free.fr`. Insérer aux endroits prévus les modules écrits au cours de ce TP et vérifier qu'ils s'enchaînent bien. Obtenir ainsi le début de la machine universelle, composée des modules 1,2,3.

Effectuer tous les tests concernant le TP 4, donnés en commentaire à la fin de `canevasuniverselle.txt`. Tant qu'ils ne produisent pas les résultats annoncés, ne pas passer au TP 5.

Travaux dirigés numéro 5

Reprendre l'examen de l'année précédente.

Travaux pratiques numéro 5

Mettre ensembles, dans un même canevas, le début de la machine de universelle, mis au point lors du TP 4, le milieu de la machine universelle, qui se trouve dans le fichier `milieuuniverselle.txt` sur les site `alain.colmerauer.free.fr` et la fin de la machine universelle, mise au point lors du TP 3. Effectuer les tests du TP 5, donnés en commentaire à la fin de `milieuuniverselle.txt`.