Logique et Calculabilité
INF551

$$\exists \Rightarrow \forall$$

Dr. Stéphane Lengrand,

Stephane.Lengrand@Polytechnique.edu

Cours 7
Les machines de Turing

3 notions équivalentes

- Fonctions calculables ("μ-récursives") - Gödel, 1933
- λ-calcul - Church, 1936
- Machines de Turing, 1937

Notion de calcul plus bas niveau, réalisable physiquement (Bletchley Park)

Modèle de coût dans le temps et dans l'espace

I. Les machines de Turing

Souvenez-vous des automates

La lecture d'un mot sur l'alphabet fini Σ par un automate :

a_1	a_2	a_3	a_4	$\ldots$

$$\Uparrow \quad \cdots \longrightarrow \cdots$$

Ensemble d'états S, état initial, état final

Table de transition : fonction de $\Sigma \times S$ dans S

Souvenez-vous des automates

La lecture d'un mot sur l'alphabet fini Σ par un automate :

a_1	a_2	a_3	a_4	$\ldots$

$$\Uparrow \qquad \cdots \longrightarrow \cdots$$

Ensemble d'états S, état initial, état final

Table de transition : fonction de $\Sigma \times S$ dans S

En une phrase :

Machine de Turing = Automate avec droit de lecture ET d'écriture

Souvenez-vous des automates

La lecture d'un mot sur l'alphabet fini Σ par un automate :

a_1	a_2	a_3	a_4	$\ldots$

$$\qquad \Uparrow \quad \cdots \longrightarrow \cdots$$

Ensemble d'états S, état initial, état final

Table de transition : fonction de $\Sigma \times S$ dans S

En une phrase :

Machine de Turing = Automate avec droit de lecture ET d'écriture

Une tête de lecture/écriture peut se déplacer dans les deux sens et revisiter les même cases

Souvenez-vous des automates

La lecture d'un mot sur l'alphabet fini Σ par un automate :

a_1	a_2	a_3	a_4	$\ldots$

$$\Uparrow \qquad \cdots \longrightarrow \cdots$$

Ensemble d'états S, état initial, état final

Table de transition : fonction de $\Sigma \times S$ dans S

En une phrase :

Machine de Turing = Automate avec droit de lecture ET d'écriture

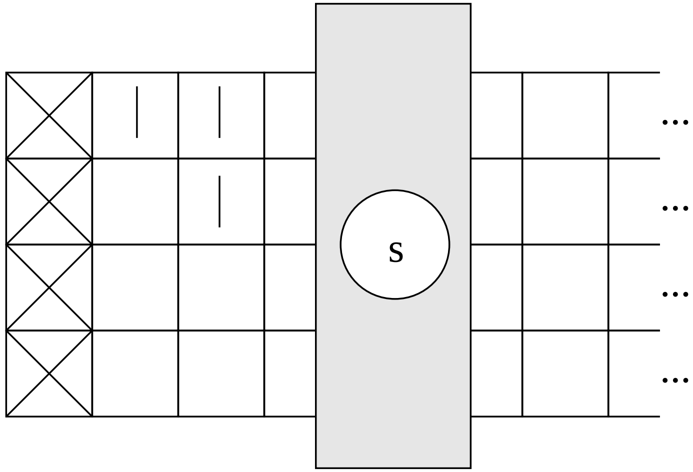
Une tête de lecture/écriture peut se déplacer dans les deux sens et revisiter les même cases

Une video youtube vaut mieux qu'un long discours

http://www.youtube.com/watch?v=cYw2ewoO6c4

La mémoire

k rubans formés d'un nombre infini de cases



Symbole spécial read-only dans la première case : la croix

Chaque autre case contient une lettre d'un alphabet Σ.

Une lettre privilegiée (le symbole blanc) :

seul un nombre fini de cases ne contient pas ce symbole (invariant)

Petit pas de calcul

Une table de transition donne les règles d’évolution de la machine

Généralise la notion de table de transition d’un automate :

Petit pas de calcul

Une table de transition donne les règles d'évolution de la machine

Généralise la notion de table de transition d'un automate :

La tête lit le contenu des k rubans à sa position courante

Petit pas de calcul

Une table de transition donne les règles d'évolution de la machine

Généralise la notion de table de transition d'un automate :

La tête lit le contenu des k rubans à sa position courante

En fonction de ce k-uplet de l'état la table prescrit :

– un k-uplet de symboles à écrire sur les rubans

– un nouvel état

– et un déplacement -1 : à gauche, 0 : sur place ou $+1$: à droite

Petit pas de calcul

Une table de transition donne les règles d'évolution de la machine

Généralise la notion de table de transition d'un automate :

La tête lit le contenu des k rubans à sa position courante

En fonction de ce k-uplet de l'état la table prescrit :

– un k-uplet de symboles à écrire sur les rubans

– un nouvel état

– et un déplacement -1 : à gauche, 0 : sur place ou $+1$: à droite

La tête écrit les k symboles, change d'état et de position, puis passe au petit pas de calcul suivant

Petit pas de calcul

Une table de transition donne les règles d'évolution de la machine

Généralise la notion de table de transition d'un automate :

La tête lit le contenu des k rubans à sa position courante

En fonction de ce k-uplet de l'état la table prescrit :

– un k-uplet de symboles à écrire sur les rubans

– un nouvel état

– et un déplacement -1 : à gauche, 0 : sur place ou $+1$: à droite

La tête écrit les k symboles, change d'état et de position, puis passe au petit pas de calcul suivant

Formellement, table de transition =

fonction de $\Sigma^k \times S$ dans $\Sigma^k \times S \times \{-1, 0, +1\}$

(plus fonction de S dans $S \times \{0, +1\}$ quand on lit $(\times, \dots, \times)$)

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Deux états particuliers : un état initial, un état final

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Deux états particuliers : un état initial, un état final

Deux lettres particulières : $|, b$

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Deux états particuliers : un état initial, un état final

Deux lettres particulières : $|, b$

Sur les n premiers rubans on "écrit" les arguments $p_1, \ldots, p_n$:

(sur la ruban i : une croix et p_i bâtons)

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Deux états particuliers : un état initial, un état final

Deux lettres particulières : $|, b$

Sur les n premiers rubans on "écrit" les arguments $p_1, \ldots, p_n$:

(sur la ruban i : une croix et p_i bâtons)

Tête de lecture sur $(\times, \ldots, \times)$ et état initial : la machine démarre

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Deux états particuliers : un état initial, un état final

Deux lettres particulières : $|, b$

Sur les n premiers rubans on "écrit" les arguments $p_1, \ldots, p_n$:

(sur la ruban i : une croix et p_i bâtons)

Tête de lecture sur $(\times, \ldots, \times)$ et état initial : la machine démarre

Si elle atteint un état final, la machine s'arrête (≠ automates)

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Note : la fonction de transition n'est pas forcément totale (comme pour les automates)

la machine peut crasher

Deux états particuliers : un état initial, un état final

Deux lettres particulières : $|, b$

Sur les n premiers rubans on "écrit" les arguments $p_1, \ldots, p_n$:

(sur la ruban i : une croix et p_i bâtons)

Tête de lecture sur $(\times, \ldots, \times)$ et état initial : la machine démarre

Si elle atteint un état final, la machine s'arrête (≠ automates)

On lit le résultat sur la ruban $n+1$ On a pu utiliser plus de rubans,

mais à la fin ils ne contiennent que le symbole blanc

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Definition: Une fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$ est représentable par une machine de Turing à k rubans ($k \geq n+1$) si

pour tous $p_1, \ldots, p_n$ écrits sur les n premiers rubans,

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Definition: Une fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$ est représentable par une machine de Turing à k rubans ($k \geq n + 1$) si

pour tous $p_1, \dots, p_n$ écrits sur les n premiers rubans,

- si f définie en $p_1, \dots, p_n$ la machine atteint l'état final, et à ce moment
  - $p_1, \dots, p_n$ sont écrits sur les n premiers rubans
  - $f(p_1, \dots, p_n)$ est écrit sur le ruban $n + 1$
  - les rubans suivants sont remplis de blancs

Représentation des fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$

Definition: Une fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$ est représentable par une machine de Turing à k rubans ($k \geq n+1$) si

pour tous $p_1, \ldots, p_n$ écrits sur les n premiers rubans,

- si f définie en $p_1, \ldots, p_n$ la machine atteint l'état final, et à ce moment
  - $p_1, \ldots, p_n$ sont écrits sur les n premiers rubans
  - $f(p_1, \ldots, p_n)$ est écrit sur le ruban $n+1$
  - les rubans suivants sont remplis de blancs

- si f pas définie en $p_1, \ldots, p_n$ la machine n'atteint jamais l'état final

Un modèle de coût

Le coût d'un calcul

Un modèle de coût

Le coût d'un calcul

– En temps : nombre de transitions pour aller de l'état initial à l'état final

Un modèle de coût

Le coût d'un calcul

– En temps : nombre de transitions pour aller de l'état initial à l'état final

– En espace : position maximale de la tête de lecture/écriture lors du calcul

Un exemple

Deux rubans

Trois états s_0 (initial), s_1 et s_2 (final)

$$M((\times, \times), s_0) = ((\times, \times), s_0, 1)$$

$$M((|, b), s_0) = ((|, |), s_0, 1)$$

$$M((b, b), s_0) = ((b, |), s_1, -1)$$

$$M((|, |), s_1) = ((|, |), s_1, -1)$$

$$M((\times, \times), s_1) = ((\times, \times), s_2, 0)$$

Quelle fonction ?

Un exemple

Deux rubans

Trois états s_0 (initial), s_1 et s_2 (final)

$$M((\times,\times),s_0) = ((\times,\times),s_0,1)$$

$$M((|,b),s_0) = ((|,|),s_0,1)$$

$$M((b,b),s_0) = ((b,|),s_1,-1)$$

$$M((|,|),s_1) = ((|,|),s_1,-1)$$

$$M((\times,\times),s_1) = ((\times,\times),s_2,0)$$

Quelle fonction ?

Successeur !

II. La représentation des fonctions calculables par des machines de Turing

Modifier la numérotation des rubans :

Si M est une machine qui lit n arguments sur les rubans $\leq n$
et écrit son résultat sur le ruban $n+1$ en utilisant a rubans auxiliaires
$n+2, \ldots, n+2+a\ldots$

...on peut facilement la transformer en une machine M' qui lit ses arguments sur des rubans arbitraires, qui écrit son résultat sur un ruban arbitraire, et qui utilise a rubans vierges qu'on lui fournit.

Modifier la numérotation des rubans :

Si M est une machine qui lit n arguments sur les rubans $\leq n$

et écrit son résultat sur le ruban $n+1$ en utilisant a rubans auxiliaires

$n+2, \ldots, n+2+a \ldots$

…on peut facilement la transformer en une machine M' qui lit ses arguments sur des rubans arbitraires, qui écrit son résultat sur un ruban arbitraire, et qui utilise a rubans vierges qu'on lui fournit.

La machine HOME : s_0 initial, s_1 final

$$\begin{aligned} \textsc{Home}((\times, \ldots, \times), s_0) &= ((\times, \ldots, \times), s_1, 0) \\ \textsc{Home}((a_1, \ldots, a_n), s_0) &= ((a_1, \ldots, a_n), s_0, -1) \end{aligned}$$

La composition de 2 machines

Soient M_1 et M_2 deux machines à k rubans.

$M_1; M_2$ est la machine où :

Ensemble d'états : union disjointe des ensembles d'états de $M_1; M_2$

Table : réunion des tables de toutes ces machines (fonctions de domaines disjoints) + transition entre l'état final de M_1 et l'état initial de M_2

Etat initial = celui de M_1

Etat final = celui de M_2

Les projections : facile

Les projections : facile

Les fonctions nulles : facile

$M((\times, \times, ..., \times), s_0) = ((\times, \times, ..., \times), s_1, 0)$

Les projections : facile

Les fonctions nulles : facile

$$M((\times, \times, ..., \times), s_0) = ((\times, \times, ..., \times), s_1, 0)$$

L'addition, la multiplication, la comparaison : facile

La composition

Fonctions h, g_1, ..., g_m représentées par N et $M_1, \ldots, M_m$

La composition

Fonctions h, g_1, ..., g_m représentées par N et $M_1, \ldots, M_m$

Rubans arguments/résultat + un certain nombre de rubans auxiliaires

La composition

Fonctions h, g_1, ..., g_m représentées par N et $M_1, ..., M_m$

Rubans arguments/résultat + un certain nombre de rubans auxiliaires

On les modifie pour que

- elles aient $n + m + 1 + a$ rubans où a plus grand nombre de rubans auxiliaires de $M_1, ..., M_m, N$
- M_i lise arguments sur $1, ..., n$ et écrive résultat sur $n + 1 + i$,
- N lise arguments sur $n + 2, ..., n + m + 1$ et écrive résultat sur $n + 1$
- les rubans auxiliaires soient au-delà

La composition

Fonctions h, g_1, ..., g_m représentées par N et $M_1, ..., M_m$

Rubans arguments/résultat + un certain nombre de rubans auxiliaires

On les modifie pour que

- elles aient $n + m + 1 + a$ rubans où a plus grand nombre de rubans auxiliaires de $M_1, ..., M_m, N$
- M_i lise arguments sur $1, ..., n$ et écrive résultat sur $n + 1 + i$,
- N lise arguments sur $n + 2, ..., n + m + 1$ et écrive résultat sur $n + 1$
- les rubans auxiliaires soient au-delà

$\circ^n_m(h, g_1, \ldots, g_m)$ est représentée par $M_1; \ldots; M_m; N$

Fonction g de $\mathbb{N}^n$ dans $\mathbb{N}$ représentée par N avec a rubans auxiliaires

Le théorème

Toute fonction calculable est représentable par une machine de Turing

Variantes des machines de Turing

– avoir des rubans isomorphes à $\mathbb{Z}$ plutôt que $\mathbb{N}$

Variantes des machines de Turing

– avoir des rubans isomorphes à $\mathbb{Z}$ plutôt que $\mathbb{N}$

– faire des déplacements $-p$, $-(p-1)$, ..., 0, ..., $(p-1)$, p
(p fixé)

Variantes des machines de Turing

– avoir des rubans isomorphes à $\mathbb{Z}$ plutôt que $\mathbb{N}$

– faire des déplacements $-p$, $-(p-1)$, …, 0, …, $(p-1)$, p
(p fixé)

– avoir une ruban qui compte l'espace mémoire

Variantes des machines de Turing

– avoir des rubans isomorphes à $\mathbb{Z}$ plutôt que $\mathbb{N}$

– faire des déplacements $-p$, $-(p-1)$, ..., 0, ..., $(p-1)$, p (p fixé)

– avoir une ruban qui compte l'espace mémoire

– passer de $k+1$ rubans à k rubans (plus difficile)

Variantes des machines de Turing

– avoir des rubans isomorphes à $\mathbb{Z}$ plutôt que $\mathbb{N}$

– faire des déplacements $-p$, $-(p-1)$, ..., 0, ..., $(p-1)$, p (p fixé)

– avoir une ruban qui compte l'espace mémoire

– passer de $k+1$ rubans à k rubans (plus difficile)

– passer d'un alphabet de cardinal n à un alphabet de cardinal $\sqrt{n}+1$

Toutes ces variantes sont polynomialement équivalents en termes de coût

La suite

En PC : comment gagner $\$10^6$

La prochaine fois : la démonstration automatique

Questions?