machine de Turing non-déterministe (NMT)

quand il y a un choix : depuis une config.,
plusieurs transitions possibles

au lieu de $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$

on a $Q \times \Gamma \to 2^{Q \times \Gamma \times \{L, S, R\}}$

où 2^X est l'ensemble des parties de X

la plupart des définitions sur les MT s'appliquent :

configurations ...

ce qui change : un run dans une MT

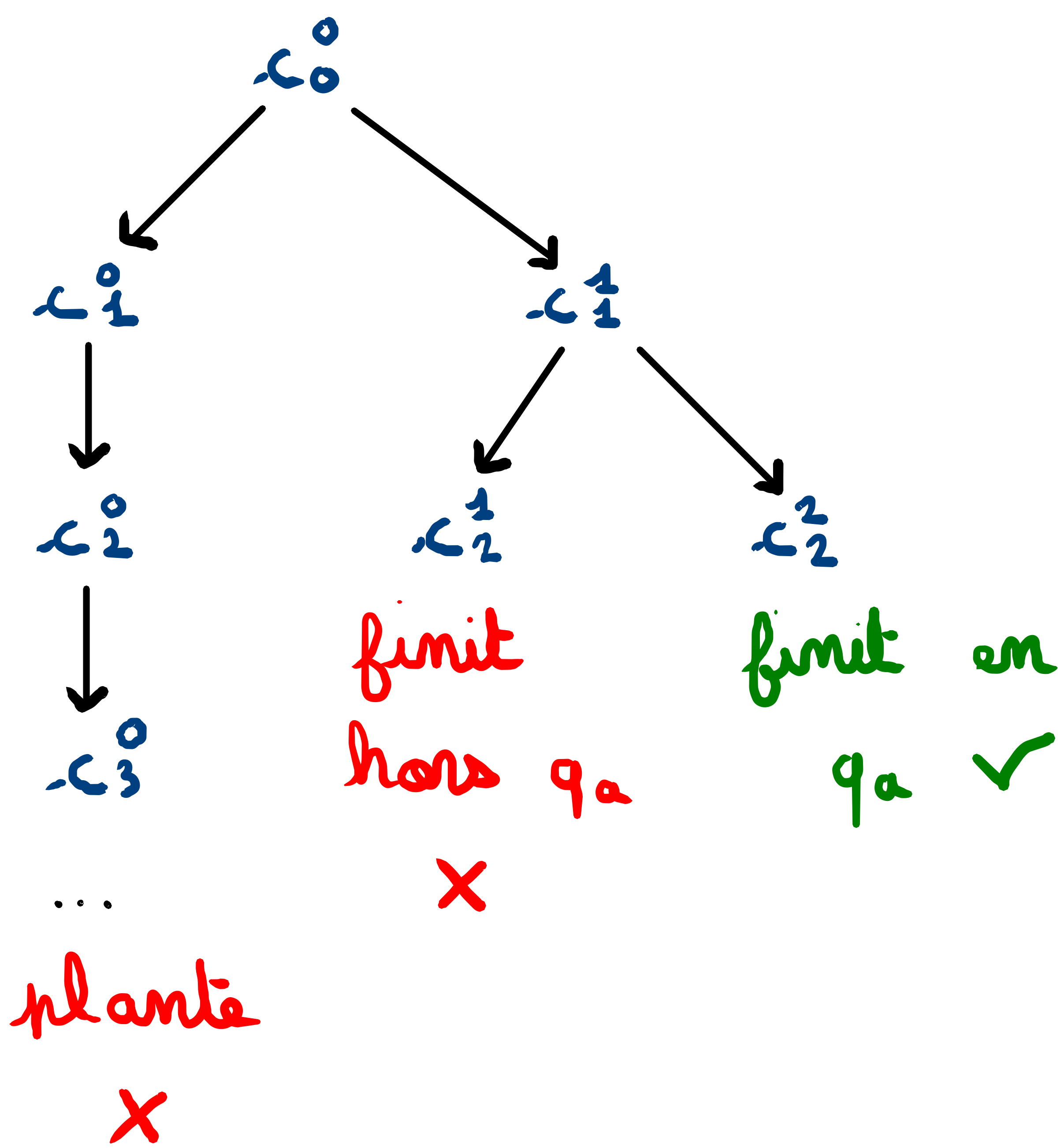
$c_0 \vdash_M c_1 \vdash_M c_2$

devient un arbre d'exécution

$$\begin{tikzcd}
 & & c_0^0 \arrow[dl] \arrow[dr] & \\
 & c_1^0 \arrow[d] & & c_1^1 \arrow[dl] \arrow[dr] \\
 & c_2^0 & c_2^1 & & c_2^2
\end{tikzcd}$$

profondeur potentiellement infinie mais degré fini

acceptation



accepte !

il suffit d'une seule branche finissant en q_a pour accepter.

arrêt : toutes les branches s'arrêtent

refuser : toutes les branches s'arrêtent hors de q_a

attention : $\neg$ accepter $\neq$ refuser

semi-équivalence

la notion de sortie n'a pas de sens sur une NMT

car différentes branches peuvent avoir des sorties $\neq$

M et N sont semi-équivalentes sur x ssi

M	s'arrête	$\Leftrightarrow$	N	s'arrête
	refuse			refuse
	accepte			accepte

noté $M(x) \sim N(x)$

$M \sim N$ ssi $\forall x \quad M(x) \sim N(x)$

expressivité des NMT

$\forall$ N NMT $\exists$ M MT M $\sim$ N

idée : créer M qui parcourt l'arbre d'exécution de N en largeur grâce à une queue sur son ruban utiliser le non-déterminisme

mq RE est stable par .

càd pour L_1 et L_2 RE, $\exists N$ NMT
 M_1 M_2

$N(x)$ accepte ssi $x \in L_1 . L_2$
 refuse $\notin$

idée : découper $x = x_1 . x_2$

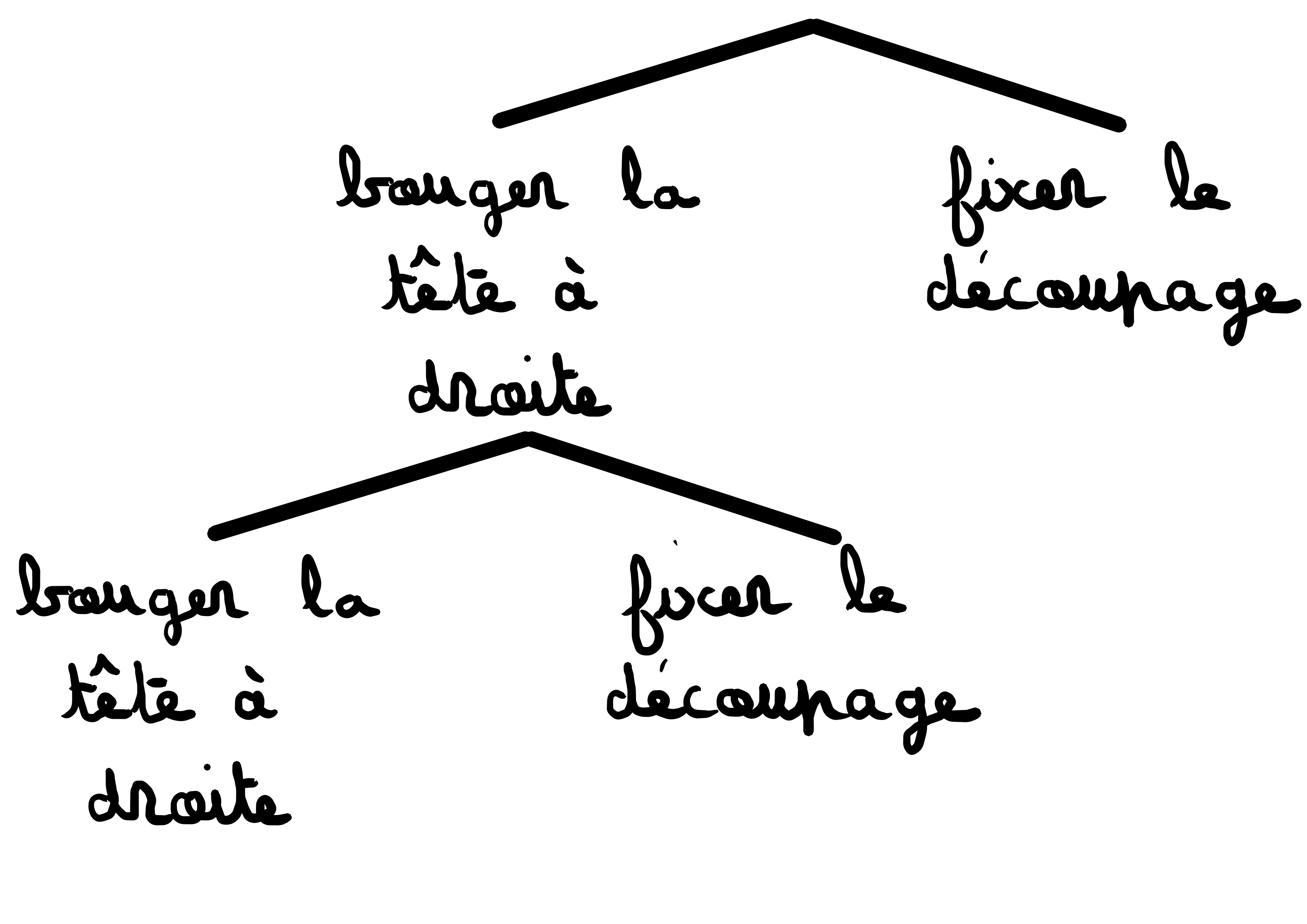
puis tester $M_1(x)$ et $M_2(x)$ entrelacés

pb : quel est le bon découpage ?

oracle

idée : deviner le découpage en le choisissant

de manière non-déterministe



$x \in L_1 . L_2$ ssi il existe un bon découpage

ce genre de méthodes s'appelle un oracle :

postuler un choix parmi un ensemble dénombrable

puis le vérifier

une autre application

mq RE est stable par $*$

soit M acceptant L ; construire N qui accepte L^* en postulant un découpage $x = x_1 \cdot \ldots \cdot x_n$ puis vérifier $\forall i \; x_i \in L$

calculer avec des MT

soit $f : E \subseteq \Sigma^* \to \Sigma'^*$

f est calculable (ou récursive) ssi $\exists$ M MT

sur Σ tq :

. $\Sigma' \subset \Gamma$

. $\forall x \in E$, $M(x)$ accepte avec la sortie $f(x)$

un exemple

la multiplication unaire est calculable

$f : 0^n 1 0^m \rightarrow 0^{nm}$

équivaut à $f(0^n, 0^m)$

on utilise le séparateur 1 pour passer 2 arguments

notons que $0^{nm} = \underbrace{0^n \ldots 0^n}_{m \text{ fois}}$

utilisons M à 2 rubans où l'on copie 0^n m fois sur le second ruban.

Théorème d'itération de Kleene   optionnel

il existe une fct s récursive totale sur

$B_\# = \{0, 1, \#\}$ telle que $\forall$ M MT,

$\forall x, y \in B^*$,

$$\mathcal{M}_{s(\langle M \rangle \# x)}(y) \equiv M(x \# y)$$

c'est au fond une currification

$$s(\langle M \rangle, x) = M_x : y \longrightarrow M(x, y)$$

optionnel

M_x est la MT qui sur l'entrée y

1) écrit $x\#$ devant y

2) puis applique M à $x\#y$

Théorème de récursion de Kleene optionnel

soit f récursive totale binaire ; $\exists$ M MT tq

$M_{f(<M>)} \equiv M$

notons qui si $f(x)$ est la fonction qui accepte tjrs avec la sortie x alors le pt fixe de f par Kleene est un programme qui affiche son propre code source (Quine)

comparer des problèmes

on regarde des problèmes d'appartenance $x \in L$?

A est Turing - réductible à B ssi

$x \in A \iff f(x) \in B$ $\quad f$ calculable

on note $A \leq_T B$

f est une réduction de A à B

B est "plus complexe" que A

B est R ssi A est R

$\quad\;\;$ RE $\qquad\qquad\qquad$ RE

si B est reconnu par M alors

$\;\;\;\; A$ $\qquad\qquad\qquad\qquad\quad$ M'

où $M'(x)$ calcule $f(x)$ puis lui

applique M

penser B "borné" $\Rightarrow$ A "borné"

A non R $\Rightarrow$ B non R

RE RE

en effet si B était R, A aussi

penser A " infini " $\Rightarrow$ B " infini "

attention

A R $\not\Rightarrow$ B R

B non R $\not\Rightarrow$ A non R

preuve d'indécidabilité

pour mq A indécidable, trouver X indécidable tq $X \leqslant_T A$

intuition : si on avait les outils pour résoudre A, on pourrait résoudre X

or X indécidable

absurde

le problème de l'acceptation

$\mathcal{A} = \{ \langle M \rangle \# x \mid M(x) \text{ accepte} \}$

est non R

on va prouver que $\mathcal{H} \leq_T \mathcal{A}$ (comme souvent)

soit M une MT et x une entrée

construire $M' = f(M)$ telle que

$M(x)$ s'arrête ssi $M'(x)$ accepte

M' va simuler M jusqu'à son point d'arrêt

puis accepter

attention : f n'a pas à être nécessairement

inversible

problème de l'équivalence

$\mathcal{E} = \{ \langle M_1 \rangle \# \langle M_2 \rangle \mid \mathcal{L}(M_1) = \mathcal{L}(M_2) \}$

est indécidable

mq $\mathcal{H} \leqslant_T \mathcal{E}$

ie que pour M MT sur l'entrée x on peut

construire M_1 et M_2 tq $M(x)$ s'arrête ssi

$\mathcal{L}(M_1) = \mathcal{L}(M_2)$

M_1 accepte tout le temps

$\mathcal{L}(M_1) = \Sigma^*$

M_2 va sur l'entrée y :

. effacer y

. écrire x

. simuler $M(x)$ jusqu'à son arrêt

. puis toujours accepter

c'est une méthode courante : insérer un problème en ignorant l'entrée

$\mathcal{L}(M_2) = \Sigma^*$ ssi $M(x)$ s'arrête

$\emptyset$ sinon

le problème de la non-vacuité

$\mathcal{L}_{ne} = \{ \langle M \rangle \mid \mathcal{L}(M) \neq \emptyset \}$

mq $\mathcal{H} \leq \mathcal{L}_{ne}$

pour M et x, trouver M' telle que

$M(x)$ s'arrête ssi $\mathcal{L}(M') \neq \emptyset$

prendre $M' = M_2$ de la preuve précédente

$$\mathcal{L}_e = \{ \langle M \rangle \mid \mathcal{L}(M) = \emptyset \}$$

est aussi

propriété

un langage P est une propriété ssi

$\forall\ M, N$ MT, $M \equiv N$, alors

$\langle M \rangle \in P$ ssi $\langle N \rangle \in P$

théorème de Rice : toute propriété non-triviale

(ie $\neq \emptyset , \Sigma^*$) est indécidable