9 Prenons de la hauteur pour respirer... avant de replonger

- $\exists$ des langages sans programme qui les accepte (7.2 : L_d)
- Des fonctions (même totales) ne sont pas calculables (4 et 7.2)
- $\exists$ des langages *acceptés* mais pas *décidés* (8.2 : L_u)

  $\Rightarrow$ pour programmer certaines fonctions partielles calculables...
  il faut accepter que le programme ne s'arrête pas hors de leur domaine (4 et 8.2 : leur domaine est r.e. mais non récursif)
  donc les programmes qui bouclent sont inévitables...

Problème de décision (1.3) : fonction *totale* $\{instances\} \to \{\text{oui/non}\}$

$\Leftrightarrow$ *propriété* des instances

$\Leftrightarrow$ appartenance au *langage* des *instances positives* ($\rightsquigarrow$ «oui»)
avec un codage des instances...

Exemple : «la MT m accepte (ou "s'arrête sur") l'entrée w»
n'est pas décidable (L_u non récursif)... *décidable* = *récursif*

Donc le *problème de l'arrêt* est indécidable :
il n'existe pas de programme qui prend en entrée
un programme m et une entrée w pour m (instance : (m, w)),
et qui *décide* si, oui ou non, m s'arrête sur w

10 Théorème de Rice (1951, 1953) [Henry Gordon Rice 1920–2003]

Énoncé 1 : toute propriété non triviale (triviale : toujours vraie ou toujours fausse) des langages RE est indécidable

Énoncé 2 : toute propriété non triviale des MT (des programmes), *qui ne dépend que du langage accepté par la MT (de la fonction calculée par le programme)*, est indécidable

Énoncé 3 : toute propriété non triviale des grammaires générales G, *qui ne dépend que de $L(G)$*, est indécidable

Exemples (énoncé 2 sur les MT et les langages acceptés) :

- «$L(m)$ est vide»
- «$L(m)$ contient au moins un mot (donc est non vide)»
- «$L(m)$ contient ε»
- «$L(m)$ contient exactement 42 mots»
- «$L(m)$ est fini»
- «$L(m)$ est infini»

Contre-exemples décidables (énoncé 2 sur les MT et les langages) :

- «m a 5 états»
- «$\exists w$: l'exécution de m sur w fait au moins 5 pas»

Notation : $X \in \mathcal{P} \Leftrightarrow X$ satisfait la propriété $\mathcal{P}$ (instance positive)

Preuve du théorème de Rice (énoncé 2 sur les MT et les langages)

Soit $\mathcal{P}$ une propriété en question... *Supposons-la décidable*

Note : si $L(m_1) = L(m_2)$ alors, soit m_1 et $m_2 \in \mathcal{P}$, soit m_1 et $m_2 \notin \mathcal{P}$

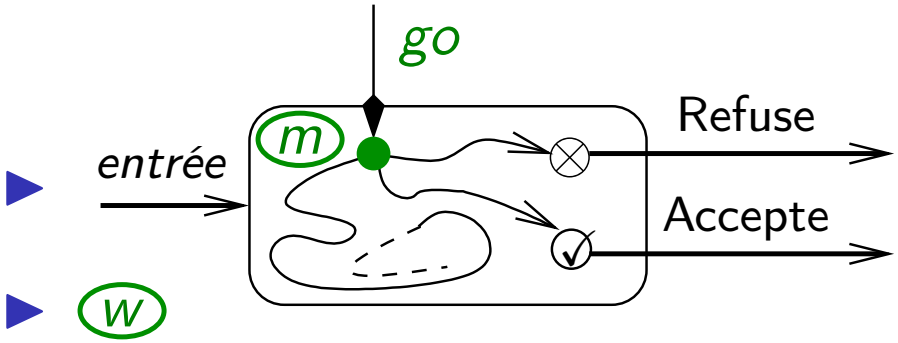
Note : $L(\varepsilon) = \emptyset$ (codage des MT dans ZOU*, 7.1)

a. Supposons $\varepsilon \notin \mathcal{P}$ Alors $\exists m^+ \in \mathcal{P}$ et forcément $L(m^+) \neq \emptyset$

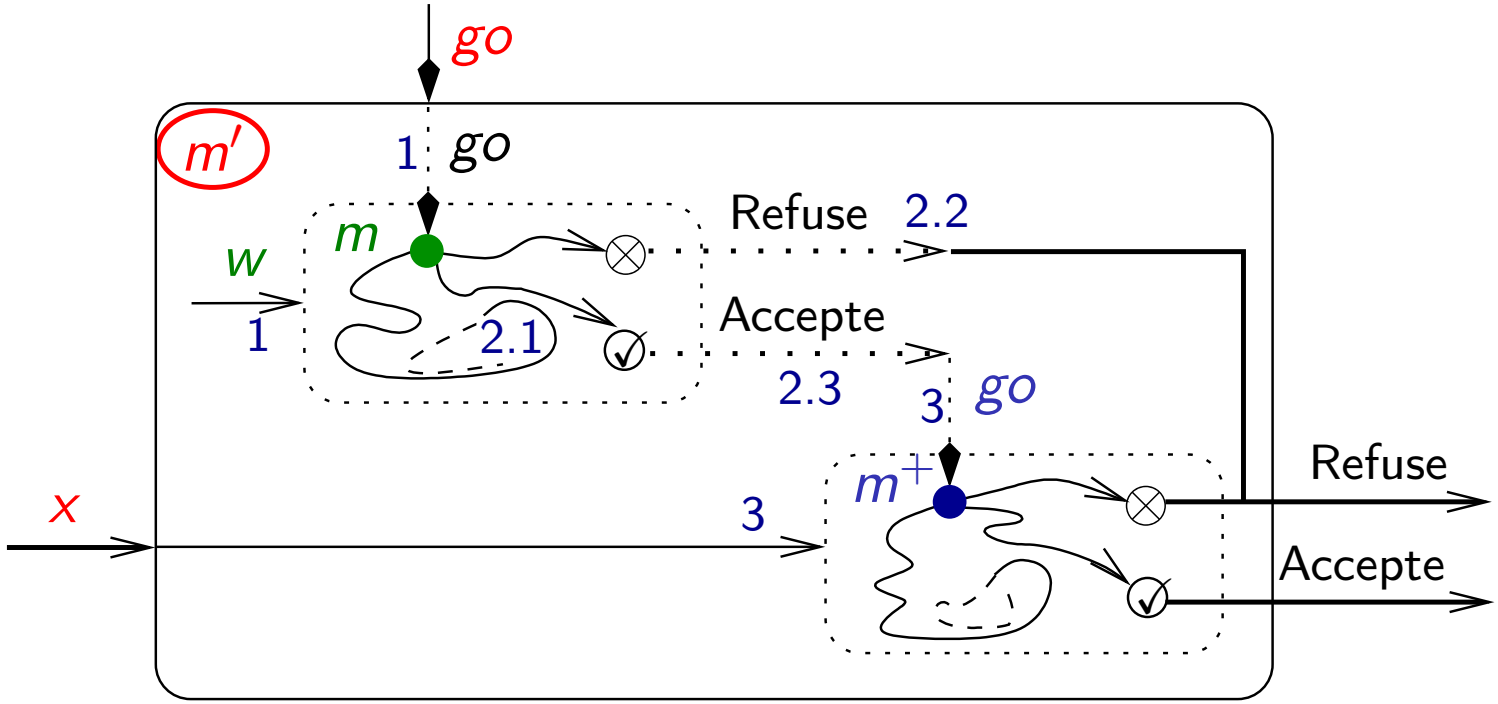
$\mathcal{P}$ décidable $\Rightarrow \exists m_{\mathcal{P}}$ qui décide $\mathcal{P} \Rightarrow$ on peut trouver une telle m^+ :
énumérer les MT jusqu'à en trouver une acceptée par $m_{\mathcal{P}}$...

- Soit m (une MT) et w (un mot) quelconques
  À partir de m et w, *construire* une MT m' (entrée : x) :
  1 *début* : comme m sur w
  2.1 m boucle sur $w \Rightarrow m'$ bouclera ($\forall$ son entrée x)
  2.2 m s'arrête sans accepter $w \rightarrow$ arrêter m' sans accepter x
  2.3 m accepte $w \rightarrow 3$ comme m^+ sur $x \rightsquigarrow$ oui / non ou boucle
- Si $w \notin L(m)$ (2.1 ou 2.2), $L(m') = \emptyset = L(\varepsilon)$ donc $m' \notin \mathcal{P}$
  Si $w \in L(m)$ (2.3 puis 3), $L(m') = L(m^+)$ donc $m' \in \mathcal{P}$
- Donc $w \in L(m) \Leftrightarrow m' \in \mathcal{P}$
- Donc, $\mathcal{P}$ décidable $\Rightarrow L_u \in$ R. Or $L_u \notin$ R... contradiction en a.

b. Supposons $\varepsilon \in \mathcal{P}$ Alors $\varepsilon \notin \overline{\mathcal{P}} \Rightarrow \overline{\mathcal{P}}$ indécidable (a. sur $\overline{\mathcal{P}}$)
$\Rightarrow \mathcal{P}$ indécidable ■



De m et w on construit m' (en utilisant m^+)



En pratique, m' a deux rubans

- le premier, initialisé avec l'entrée x, qui sert dans l'étape 3.
- le deuxième, initialisé par m' avec w, qui sert dans l'étape 1.

11 Techniques de preuve ; réduction

Pour prouver
- qu'un langage est récursif / récursivement énumérable
- qu'un problème est décidable / *semi-décidable*
- qu'une fonction est totale calculable / (partielle) calculable

il «suffit» d'exhiber une MT qui décide ou calcule totalement / qui accepte, *semi-décide* ou calcule (partiellement)

Ça implique souvent l'utilisation de MT déjà connues...
mais pas toujours (*ex* : $L_u \in \mathrm{RE}$)

Pour prouver le contraire ($L \notin \mathrm{R}$ / $L \notin \mathrm{RE}$)...
on raisonne en général par l'absurde

Exemples :
- $L_d \notin \mathrm{RE}$ $\quad$ $L_u \notin \mathrm{R}$
- Rice pour bon nombre de langages/problèmes/fonctions...

Une technique utile à connaître est la *réduction*

Principe (exemple) : on suppose $L \in \mathrm{R}$, on en *déduit* $L' \in \mathrm{R}$
donc si on sait $L' \notin \mathrm{R}$, on a forcément $L \notin \mathrm{R}$

On dit qu'on a *réduit* L' à L

Exemples de réductions déjà rencontrées

1. réduction de L_d à $\overline{L_u}$ (cf. diapo 24)

$L_d = \{w \in \text{ZOU}^* \mid w \notin L(w)\}$ $\quad \overline{L_u} = \{(m, w) \in (\text{ZOU}^*)^2 \mid w \notin L(m)\}$

Soit $w \in \text{ZOU}^*$

(A) posons $c = (w, w)$

Alors c est une entrée pour l'hypothétique $m_{\overline{L_u}}$

et $w \in L_d \Leftrightarrow c \in \overline{L_u}$ (E)

Donc $\overline{L_u} \in \text{R} \Rightarrow L_d \in \text{R}$ et donc $L_d \notin \text{R} \Rightarrow \overline{L_u} \notin \text{R}$

On a même plus : $L_d \notin \text{RE} \Rightarrow \overline{L_u} \notin \text{RE}$

A : *algorithme* pour transformer w en c tel que E

$\{w\} = \text{ZOU}^* = \{$*instances* du problème d'appartenance à $L_d\}$

$\{c\} = \{\text{A}(w)\} \subseteq \{$*instances* du problème d'appartenance à $\overline{L_u}\}$

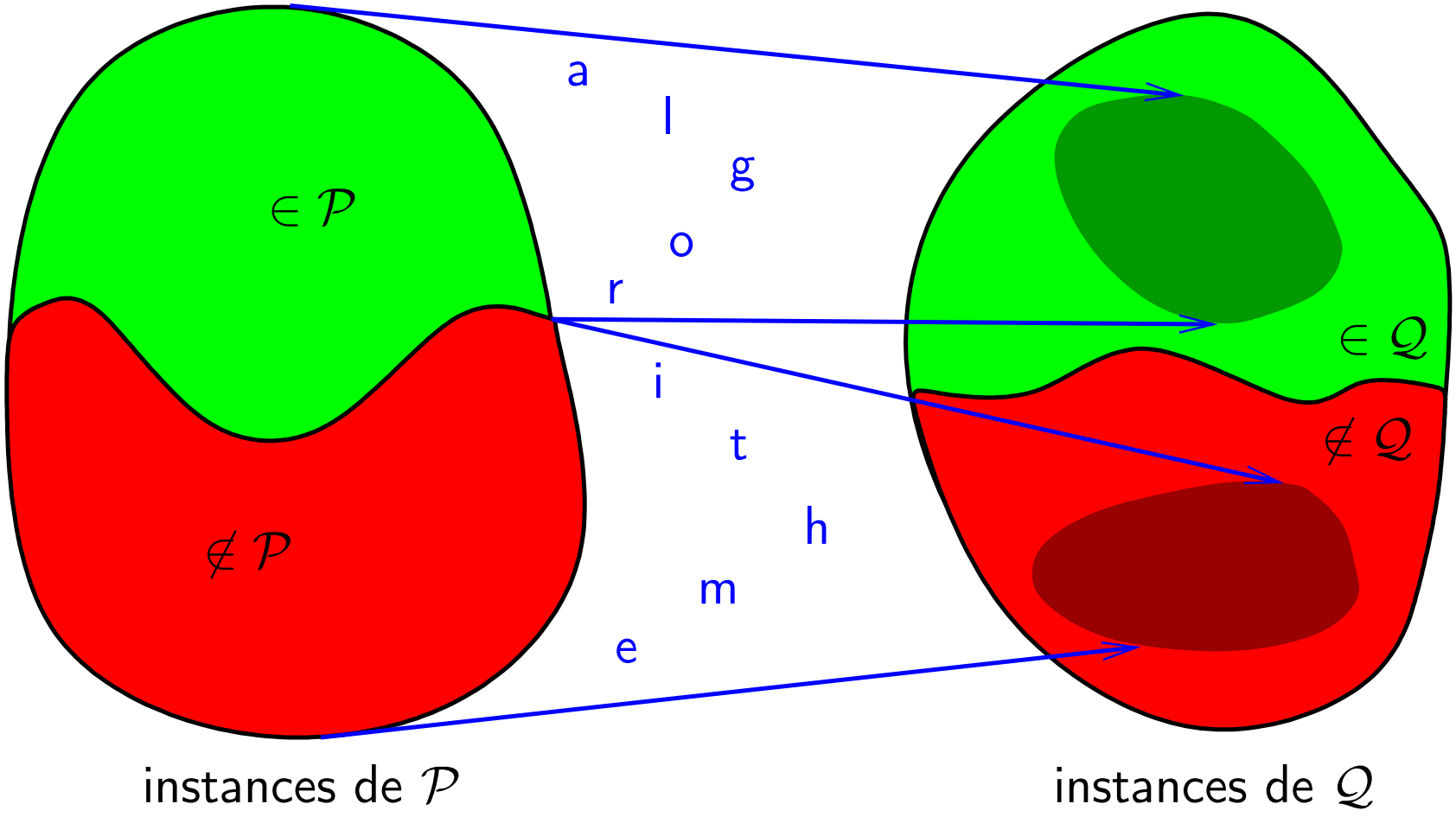
L'algo «transforme» les *instances positives* (*les* w éléments de L_d)

en *instances positives* (*des* c éléments de $\overline{L_u}$)

et les *instances négatives* (*les* w non éléments de L_d)

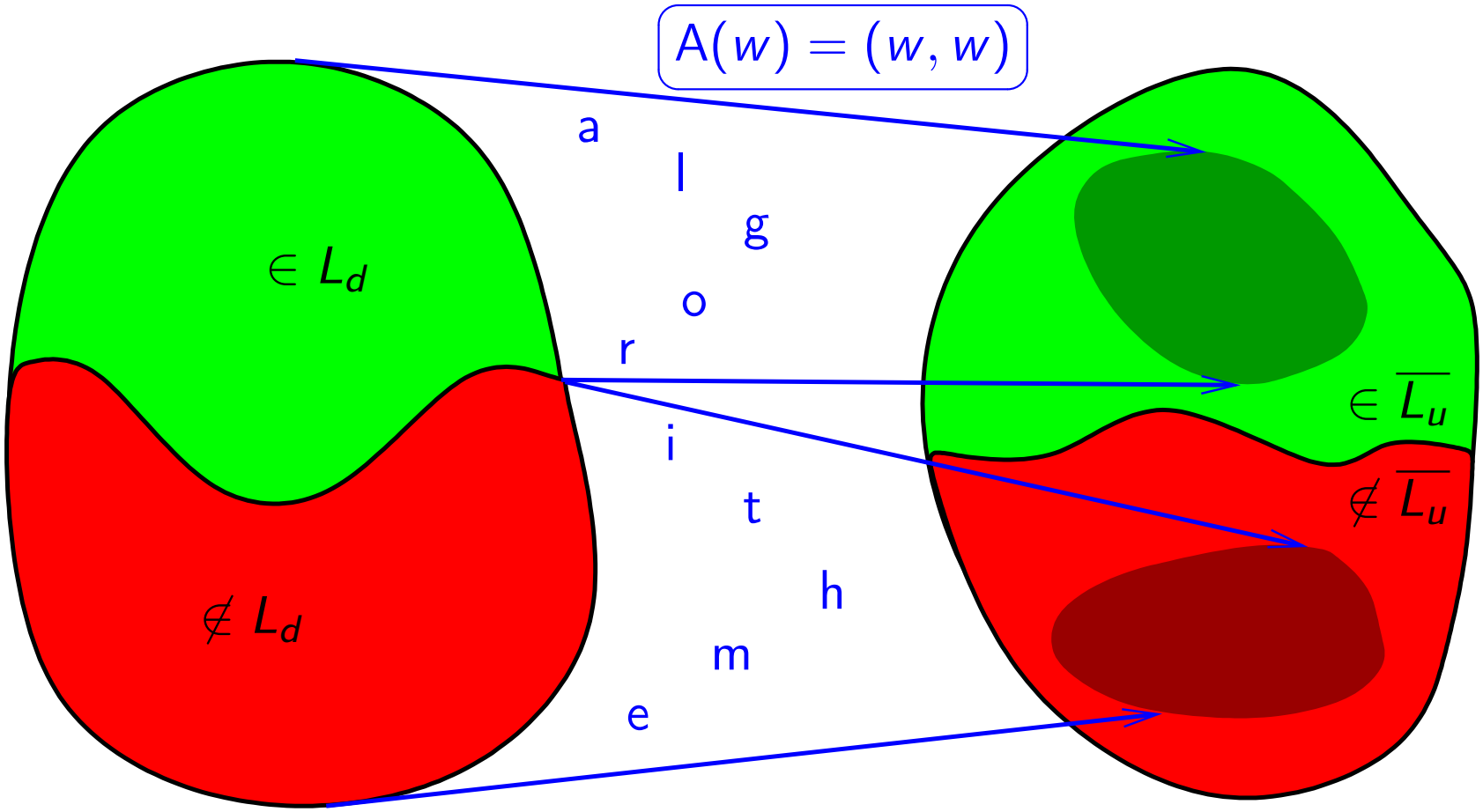
en *instances négatives* (*des* c non éléments de $\overline{L_u}$)

Schéma général de réduction de $\mathcal{P}$ à $\mathcal{Q}$



Si on peut accepter/décider $\mathcal{Q}$, alors on peut accepter/décider $\mathcal{P}$: transformer par l'algorithme l'instance de $\mathcal{P}$ en instance de $\mathcal{Q}$, puis accepter/décider...

Exemple : réduction de L_d à $\overline{L_u}$



ZOU*
(instances de «$\in L_d$?»)

ZOU$^* \times$ ZOU*
(instances de «$\in \overline{L_u}$?»)

Exemples de réductions déjà rencontrées

2. preuve du théorème de Rice (cf. diapo 27)

a. : réduction (compliquée) de L_u à $\mathcal{P}$

Dans l'algorithme A :

- m^+ est une constante pré-calculée
  sous l'hypothèse que $\mathcal{P}$ est décidable
  $\rightsquigarrow$ A n'existe que si $\mathcal{P}$ est décidable
  Pas un problème, la réduction ne sert que si $\mathcal{P}$ est décidable
- (m, w) est le paramètre (instance de «$\in L_u$?»)
- m' est le résultat (instance de $\mathcal{P}$)

Et on a montré E : $(m, w) \in L_u \Leftrightarrow m' \in \mathcal{P}$

b. : a. sur $\overline{\mathcal{P}}$ (réduction de L_u à $\overline{\mathcal{P}}$)

puis propriété des complémentaires

12 Problème de Correspondance de Post (PCP, 1946) [E. Post 1897–1954]

Problème de décision sans rapport avec les lang. RE ou les MT...

Instance : couple de listes L et M de même longueur $k > 0$
chaque liste : k mots *non vides* sur un vocabulaire V
$L = w_1, ...w_k$ $M = x_1, ...x_k$ instance : (V, L, M)

Instance positive : $\exists m \in \mathbb{N}, \exists i_0, ...i_m \in [1, k]^{m+1} : w_{i_0}...w_{i_m} = x_{i_0}...x_{i_m}$
$i_0, ...i_m$: *solution* de l'instance

Exemples : ① $L = a, abaaa, ab$ $M = aaa, ab, b$
une solution : $2, 1, 1, 3 \rightarrow abaaa\,a\,a\,ab = ab\,aaa\,aaa\,b$

② $L = ab, baa, aba$ $M = aba, aa, baa$
... pas de solution (exercice : le démontrer...)

Théorème : PCP est indécidable

Preuve : passe par le Problème de Post Modifié (PPM)

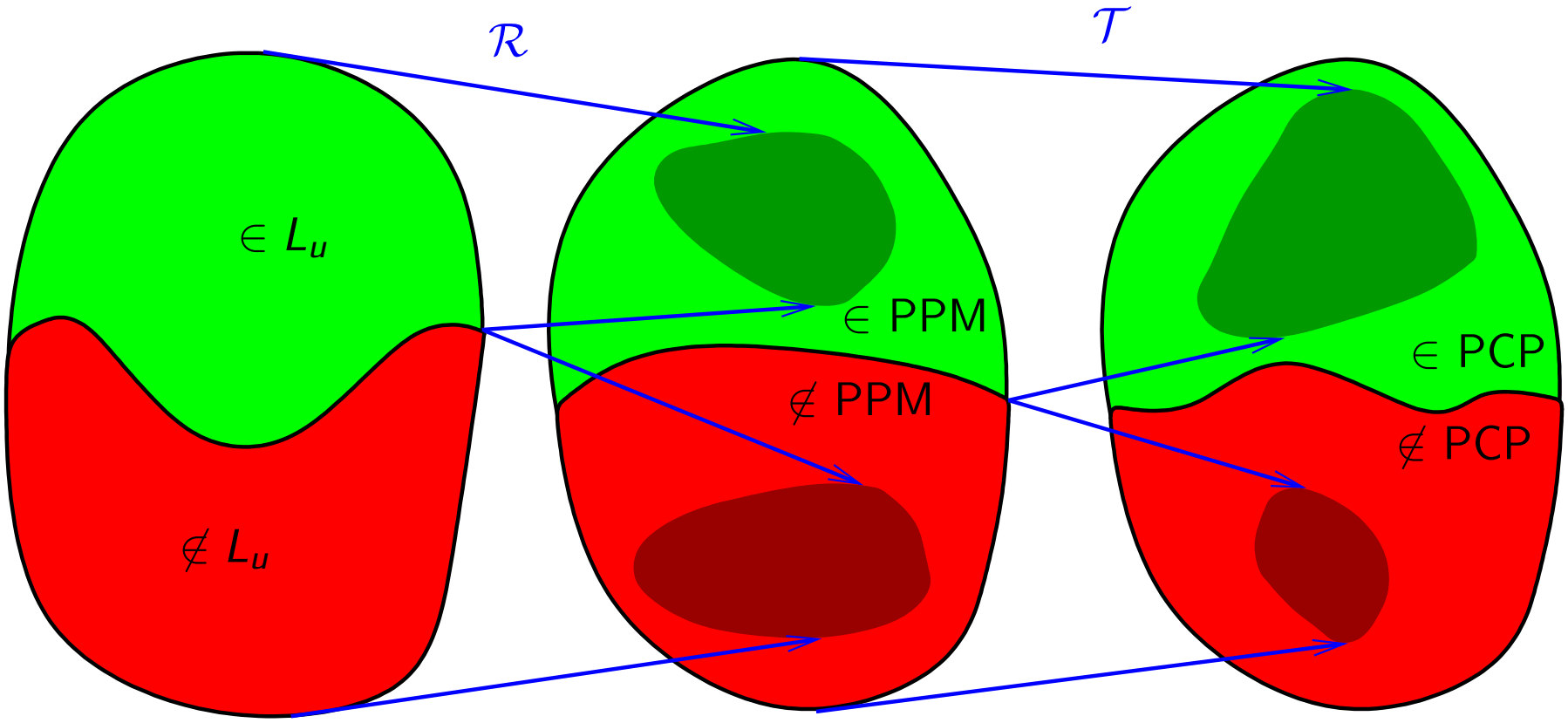
PPM : idem PCP, mais impose $i_0 = 1$

Instance positive : $\exists m \in \mathbb{N}, \exists i_1, ...i_m \in [1, k]^m : w_1 w_{i_1}...w_{i_m} = x_1 x_{i_1}...x_{i_m}$
$i_1, ...i_m$: *solution* de l'instance

Exemples : ② n'a toujours pas de solution, mais ① non plus

Double réduction : L_u à PPM et PPM à PCP

Schéma :



instances de «$\in L_u$?»

(m, w)

$m = (Q, \Gamma, \Sigma, B, q_0, F, \delta)$

$w \in \Sigma^*$

instances de PPM

(V, L, M)

$L \in (V^+)^+$

$M \in (V^+)^+$

$|L| = |M|$

instances de PCP

(V', C, D)

$C \in (V'^+)^+$

$D \in (V'^+)^+$

$|C| = |D|$

a. Réduction de PPM à PCP

À partir d'une instance $(V,\ L = w_1, ...w_k,\ M = x_1, ...x_k)$ de PPM on construit une instance $\mathcal{T}(V, L, M) = (V', C, D)$ de PCP t.q. (V, L, M) positive dans PPM $\Leftrightarrow$ (V', C, D) positive dans PCP

$V' = V \cup \{\diamondsuit, \heartsuit\}$: vocabulaire de C et D $(\diamondsuit, \heartsuit \notin V)$

$C = y_1, ..., y_{k+2} \quad D = z_1, ...z_{k+2}$

$\forall i \in [1, k]$, si $w_i = a_1...a_p$ alors $y_i = a_1\diamondsuit...a_p\diamondsuit$

si $x_i = b_1...b_q$ alors $z_i = \diamondsuit b_1...\diamondsuit b_q$

$y_{k+1} = \diamondsuit y_1 = \diamondsuit a_1\diamondsuit...a_p\diamondsuit \qquad z_{k+1} = z_1 = \diamondsuit b_1...\diamondsuit b_q$

$y_{k+2} = \heartsuit \qquad z_{k+2} = \diamondsuit\heartsuit$

Exemple : ① comme instance de PPM donne

i	1	2	3
w_i (L)	a	$abaaa$	ab
x_i (M)	aaa	ab	b

i	1	2	3	4	5
y_i (C)	$a\diamondsuit$	$a\diamondsuit b\diamondsuit a\diamondsuit a\diamondsuit a\diamondsuit$	$a\diamondsuit b\diamondsuit$	$\diamondsuit a\diamondsuit$	$\heartsuit$
z_i (D)	$\diamondsuit a\diamondsuit a\diamondsuit a$	$\diamondsuit a\diamondsuit b$	$\diamondsuit b$	$\diamondsuit a\diamondsuit a\diamondsuit a$	$\diamondsuit\heartsuit$

1. (V, L, M) a une sol. dans PPM $\Rightarrow \mathcal{T}(V, L, M)$ a une sol. dans PCP

Soit $i_1, \ldots i_m$ solution de (V, L, M) : $w_1 w_{i_1} \ldots w_{i_m} = x_1 x_{i_1} \ldots x_{i_m}$
Alors $k+1, i_1, \ldots i_m, k+2$ est une solution de $\mathcal{T}(V, L, M)$:

$y_{k+1} y_{i_1} \ldots y_{i_m} y_{k+2} = z_{k+1} z_{i_1} \ldots z_{i_m} z_{k+2}$

Ex. : $L = a, ab, bb \quad M = abb, a, b$ Une sol. : 3,3 ($\rightarrow abbbb$)
$C = a\diamondsuit, a\diamondsuit b\diamondsuit, b\diamondsuit b\diamondsuit, \diamondsuit a\diamondsuit, \heartsuit \quad D = \diamondsuit a\diamondsuit b\diamondsuit b, \diamondsuit a, \diamondsuit b, \diamondsuit a\diamondsuit b\diamondsuit b, \diamondsuit\heartsuit$
Solution correspondante : 4, 3, 3, 5 ($\rightarrow \diamondsuit a\diamondsuit b\diamondsuit b\diamondsuit b\diamondsuit b\diamondsuit\heartsuit$)

2. $\mathcal{T}(V, L, M)$ a une sol. S dans PCP $\Rightarrow (V, L, M)$ a une sol. dans PPM

S ne peut commencer que par $k+1$ et terminer par $k+2$
donc $S = k+1, i_1, \ldots i_m, k+2 \quad (m \in \mathbb{N}$ et les $i_j \in [1, k])$

Si on enlève les $\diamondsuit$ et le $\heartsuit$ de $y_{k+1} y_{i_1} \ldots y_{i_m} y_{k+2}$
on obtient exactement $w_1 w_{i_1} \ldots w_{i_m}$
De même en ôtant les $\diamondsuit$ et le $\heartsuit$ de $z_{k+1} z_{i_1} \ldots z_{i_m} z_{k+2}$
on obtient exactement $x_1 x_{i_1} \ldots x_{i_m}$

Donc $y_{k+1} y_{i_1} \ldots y_{i_m} y_{k+2} = z_{k+1} z_{i_1} \ldots z_{i_m} z_{k+2}$
$\Rightarrow w_1 w_{i_1} \ldots w_{i_m} = x_1 x_{i_1} \ldots x_{i_m}$
et donc $i_1, \ldots i_m$ est une solution de (V, L, M) ■

b. Réduction de L_u à PPM

Objectif : de (m, w) produire $\mathcal{R}(m, w) = (V, L, M)$ tel que
m accepte $w \Leftrightarrow (V, L, M)$ a une solution (dans PPM)

Rappel : on peut supposer m avec un seul état final f, sans transition

Rappel : on peut supposer que m n'écrit jamais B

Note : on peut supposer que m ne reste jamais *S*tationnaire (exo...)

Note : on peut supposer que m ne va jamais à gauche
de sa position initiale (exo...)

m accepte $w \Leftrightarrow q_0 w = c_1 \vdash c_2 ... \vdash c_n = \alpha f \beta \not\vdash$

Préliminaire : *solution partielle* pour (V, L, M) : séquence telle que
$w_1 w_{i_1} ... w_{i_r}$ «partie L» est un préfixe strict de $x_1 x_{i_1} ... x_{i_r}$ «partie M»

Idée :

Faire en sorte que les solutions partielles soient les *préfixes*
de la «séquence de configurations» $\sharp c_1 \sharp c_2 \sharp c_3 \sharp ...$ ($\sharp \notin Q \cup \Gamma$)

Construire L et M de sorte que dans la solution partielle,
la «partie M» ait toujours «un pas d'avance» sur la «partie L»

Permettre à la «partie L» de «rattraper» la «partie M»
une fois l'état final atteint

On aura donc $V = Q \cup \Gamma \cup \{\sharp\}$

On démarre avec la config. initiale (un pas d'avance dans M) :

w_1 (L)	x_1 (M)
$\sharp$	$\sharp q_0 w \sharp$

(début obligatoire de solution dans PPM)

Simulation des transitions : autres couples (w_i, x_i)

transition	w_i (L)	x_i (M)	commentaire
$\delta(q, X) = (p, Y, D)$	qX	Yp	
$\delta(q, X) = (p, Y, G)$	ZqX	pZY	$\forall Z \in \Gamma$
$\delta(q, B) = (p, Y, D)$	$q\sharp$	$Yp\sharp$	
$\delta(q, B) = (p, Y, G)$	$Zq\sharp$	$pZY\sharp$	$\forall Z \in \Gamma$

Exemple : $\delta(q_0, a) = (q_1, b, D)$

on a donc un couple $w_i = q_0 a \qquad x_i = b q_1$

donc si $w = aba$ on peut prolonger le début (w_1, x_1) par :

dans L : $\sharp q_0 a \qquad$ dans M : $\sharp q_0 aba \sharp b q_1$

$w = aba$ dans L : $\sharp q_0 a$ dans M : $\sharp q_0 aba \sharp bq_1$

On ajoute des couples pour «compléter» les configurations :

w_i (L)	x_i (M)	commentaire
X	X	$\forall X \in \Gamma$
$\sharp$	$\sharp$	

On peut ainsi prolonger jusqu'à $\sharp c_1 \sharp$ (L) et $\sharp c_1 \sharp c_2 \sharp$ (M) :
dans L : $\sharp q_0 aba \sharp$ dans M : $\sharp q_0 aba \sharp bq_1 ba \sharp$

Supposons $\delta(q_1, b) = (f, a, D)$
on a donc un couple $w_i = q_1 b$ $x_i = af$

On peut alors prolonger jusqu'à $\sharp c_1 \sharp c_2 \sharp$ (L) et $\sharp c_1 \sharp c_2 \sharp c_3 \sharp$ (M) :
dans L : $\sharp q_0 aba \sharp bq_1 ba \sharp$ dans M : $\sharp q_0 aba \sharp bq_1 ba \sharp bafa \sharp$

Ici on a «atteint» l'état final (en c_3), donc $w \in L(m)$!

dans L : $\sharp q_0 aba \sharp bq_1 ba \sharp$ dans M : $\sharp q_0 aba \sharp bq_1 ba \sharp baf a \sharp$

Reste à permettre à L de «rattraper» M quand f est atteint :

w_i (L)	x_i (M)	commentaire
Xf	f	$\forall X \in \Gamma$
fX	f	$\forall X \in \Gamma$

On peut alors prolonger (plusieurs façons de faire) :

dans L : $\sharp q_0 aba \sharp bq_1 ba \sharp baf a \sharp bf a \sharp f a \sharp$

dans M : $\sharp q_0 aba \sharp bq_1 ba \sharp baf a \sharp bf a \sharp f a \sharp f \sharp$

Pour finir : ajouter le couple $(f\sharp\sharp, \sharp)$ qui nous amène à :

dans L : $\sharp q_0 aba \sharp bq_1 ba \sharp baf a \sharp bf a \sharp f a \sharp f \sharp\sharp$

dans M : $\sharp q_0 aba \sharp bq_1 ba \sharp baf a \sharp bf a \sharp f a \sharp f \sharp\sharp$

Donc m accepte $w \Rightarrow \mathcal{R}(m, w) = (V, L, M)$ a une solution

$\mathcal{R}(m, w) = (V, L, M)$ a une solution $\Rightarrow$ m accepte w :

Une solution démarre forcément avec $(w_1, x_1) = (\sharp, \sharp q_0 w \sharp)$ ...PPM...

Informellement : les couples (w_i, x_i) «simulent» les transitions

seul le dernier groupe (avec f) permet d'avoir une solution

donc solution $\Rightarrow$ état f atteint ■