922

Ensembles récursifs, récursivement énumérables. Exemples.

On considère des fonctions définies sur un sous-ensemble de $\mathbb{N}^p$ à valeurs dans $\mathbb{N}$.

I Fonctions récursives et machines de Turing

1) Fonctions récursives primitives

[CL] p138 **Déf 1** (Fonctions de base). – Constante $O : x\in\mathbb{N} \mapsto 0\in\mathbb{N}$
– Successeur $succ : n\in\mathbb{N}\mapsto (n+1)\in\mathbb{N}$
– projections $\forall p\in\mathbb{N}, \forall i\in [\![1,p]\!], \; pr_p^i : (n_1,\dots,n_p)\in\mathbb{N}^p \mapsto n_i \in \mathbb{N}$

p138 **Déf 2** La fonction composée f de $g_1,\dots,g_p : \mathbb{N}^n \to \mathbb{N}$ et de $h : \mathbb{N}^p\to\mathbb{N}$ est définie par : $f(n_1,\dots,n_n) = h(g_1(n_1,\dots,n_n),\dots,g_p(n_1,\dots,n_n))$.

p138 **Déf 3** Le schéma de récurrence primitive associe à $g:\mathbb{N}^p\to\mathbb{N}$ et $h:\mathbb{N}^{p+2}\to\mathbb{N}$ une fonction $f:\mathbb{N}^{p+1}\to\mathbb{N}$ définie par : $f(n_1,\dots,n_p,0)=g(n_1,\dots,n_p)$ et $f(n_1,\dots,n_p,y+1)=h(n_1,\dots,n_p,y,f(n_1,\dots,n_p,y))$.

p135 **Ex 4** L'addition est définie par récurrence à partir de projections et de successeur : $x+0 = pr_1^1(x)$ et $x+(y+1) = pr_3^3(x,y,x+y)+1$

p135 **Déf 5** L'ensemble des fonctions primitives récursives est le plus petit ensemble de fonctions contenant les fonctions de base et clos par composition et récurrence primitive.

p135 **Ex 6** La somme et le produit d'un nombre fini de termes sont primitives récursives.

p137 **Contre-ex 7** Fonction d'Ackermann $A:\mathbb{N}^2\to\mathbb{N}$ définie par :
$\forall x\in\mathbb{N}, \forall y\in\mathbb{N}$ $\quad A(0,x)=x+2$; $A(1,0)=0$; $A(y,0)=1, \forall y\geq 2$
$A(y+1,x+1) = A(y, A(y+1,x))$
A n'est pas primitive récursive.

2) Fonctions récursives non-primitives

p145 **Déf 8** Une fonction partielle de $\mathbb{N}^p$ dans $\mathbb{N}$ est un couple (D,f) où $D\subseteq\mathbb{N}^p$ et $f : D\to\mathbb{N}$.
D est le domaine de définition de f.
Si $D=\mathbb{N}^p$, on dit que f est totale.

Déf 9 (Minimisation) Soit g une fonction partielle à p+1 arguments. f est définie par minimisation à partir de g lorsque p145
$$\forall (x_1,\dots,x_p)\in\mathbb{N}^p,\quad f(x_1,\dots,x_p) = \begin{cases} z \text{ si } \exists z / g(x_1,\dots,x_p,z)=0 \text{ et } \forall z'<z,\ g(x_1,\dots,x_p,z')\neq 0 \\ \text{n'est pas définie sinon} \end{cases}$$
On note : $f(x_1,\dots,x_p) = \mu z (g(x_1,\dots,x_p,z)=0)$.

Déf 10 L'ensemble des fonctions récursives partielles est le plus petit ensemble de fonctions contenant les fonctions de base, clos par composition, récurrence primitive et minimisation. p146

Ex 11 La fonction d'Ackermann est récursive. [CL] p25

3) Machines de Turing

Déf 12 Une machine de Turing (MT) est un quintuplet (Q,Σ,E,q_0,F) où Q est un ensemble fini d'états, Σ est un alphabet, $E\subset \mathcal{P}(Q\times\Sigma\times Q\times\Sigma\times\{\to,\leftarrow\})$ est un ensemble fini de transitions, $q_0\in Q$ est l'état initial et $F\subset Q$ l'ensemble des états finaux. [C] p141

Déf 13 Une étape de calcul de la machine de Turing est une paire de configurations (C,C'), notée $C\to C'$ telle que : p143
soit $C=ucpav$ et $C'=uqcbv$ et $(p,a,q,b,\leftarrow)\in E$
soit $C=upav$ et $C'=ubqv$ et $(p,a,q,b,\to)\in E$.

Déf 14 Un calcul est une suite de configurations $C_0\to\dots\to C_k$. p143
Il est acceptant si C_0 est initiale i.e. $C_0=q_0w, w\in\Sigma^*$ et C_k est finale i.e. $C_k=uqv$, $u,v\in\Sigma^*, q\in F$.

Déf 15 On dit que $w\in\Sigma^*$ est accepté par une MT M s'il existe un calcul acceptant de configuration initiale q_0w. p145
L'ensemble des mots acceptés est noté $\mathcal{L}(M)$.

Déf 16 Une fonction $f:\Sigma^*\to\Sigma^*$ est calculable lorsqu'il existe une MT qui pour toute entrée w s'arrête avec $f(w)$ sur sa bande. p160 p155 **Th17** Une fonction $f : \mathbb{N}^p \to \mathbb{N}$ est récursive ssi elle est calculable par une MT.

II Ensembles récursifs et récursivement énumérables

1) Ensembles récursifs (R)

[LR] p139 **Déf 18** Un ensemble $A \subset \mathbb{N}^p$ est récursif primitif lorsque sa fonction caractéristique est récursive primitive.

p146 **Déf 19** Un ensemble $A \subset \mathbb{N}^p$ est récursif lorsque sa fonction caractéristique est récursive.

(C) p158 **Déf 20** Un ensemble A est décidable lorsqu'il existe une MT $\mathcal{M}$ sans calcul infini tel que $A = \mathcal{L}(\mathcal{M})$

p158 **Prop 21** Les ensembles récursifs correspondent aux langages décidés par une MT.

2) Ensembles récursivement énumérables (RE)

(LR) p161 **Déf 22** Un ensemble est récursivement énumérable lorsqu'il est le domaine de définition d'une fonction récursive partielle.

(C) p63 **Prop 23** Un ensemble A est récursivement énumérable lorsqu'il est accepté par une MT $\mathcal{M}$ ie $A = \mathcal{L}(\mathcal{M})$.

Prop 24 Tout ensemble récursif est RE.

p155 **Déf 25** Un énumérateur est une MT $\mathcal{M}$ déterministe qui écrit sur une bande de sortie des mots de Σ^* séparés par un symbole $\$ \notin \Sigma$. La tête de lecture de cette bande ne se déplace jamais vers la gauche.

p155 **Prop 26** Un ensemble $A \subset \mathbb{N}^p$ est RE ssi A est l'ensemble des mots énumérés par un énumérateur.

p156 **Prop 27** Il existe des langages qui ne sont pas RE

3) Propriétés de clôture

Prop 28 $\forall p \in \mathbb{N}$, l'ensemble des sous-ensembles récursifs de $\mathbb{N}^p$ est clos par union, intersection et passage au complémentaire (CL) p142 (p13)

Prop 29 L'intersection et l'union de deux sous-ensembles RE de $\mathbb{N}^p$ sont RE. p142

Th 30 Soit $A \subset \mathbb{N}^p$, A récursif ssi A et $\mathbb{N}^p \setminus A$ sont RE. p142

Th 31 La projection d'un ensemble RE est RE p143

Th 32 Soit $A \subset \mathbb{N}$, $A \in RE$ ssi $A = \emptyset$ ou A est l'image d'une fonction primitive récursive ssi A est l'image d'une fonction récursive partielle ssi $A = pr_2^2(B)$ où B est un ensemble primitif récursif. DEV p143/44

III Exemples et applications

Déf 33 (Réduction) Soient A et B deux problèmes de langages respectifs L_A et L_B. Une réduction de A à B est une fonction f calculable telle que $w \in L_A \Leftrightarrow f(w) \in L_B$. On note $A \leq B$. (C) p160

Prop 34 Si $A \leq B$ et B est décidable alors A est décidable p161

Cor 35 Si $A \leq B$ et A est indécidable alors B est indécidable p161

1) Problèmes sur les langages

Déf 36 Langage d'acceptation $L_\epsilon = \{\langle \mathcal{M}, w\rangle \mid w \in \mathcal{L}(\mathcal{M})\}$ p159

Prop 37 L_ϵ est récursivement énumérable p159

Prop 38 L_ϵ n'est pas récursif ie il est indécidable p159

Cor 39 $\overline{L_\epsilon}$ n'est pas récursivement énumérable p160 p161 Prop 40 $L_{\neq\emptyset}=\{\langle n\rangle / \mathcal{L}(n)\neq\emptyset\}$ et $L_{\neq}=\{\langle n,n'\rangle / \mathcal{L}(n)\neq\mathcal{L}(n')\}$ sont indécidables.

(W) p161 Th 41 Le problème de l'arrêt est indécidable
≈ $\{\langle n,w\rangle / n$ s'arrête sur $w\}$ n'est pas récursif

p161 Th 42 Le problème de l'arrêt existentiel : déterminer si une MT s'arrête pour au moins un mot d'entrée est indécidable

p161 Th 43 Le problème de l'arrêt universel : déterminer si une MT s'arrête pour tout mot d'entrée est indécidable.

(C) p162 Th 44 (Rice) Pour toute propriété non triviale P sur les langages RE, le problème de savoir si le langage $\mathcal{L}(M)$ d'une MT M vérifie P est indécidable

(Sip) p215 Cor 45 $L_{rat}=\{\langle n\rangle / \mathcal{L}(n)$ est rationnel$\}$ est indécidable

(C) p171 Déf 46 Une MT est linéairement bornée (MTLB) lorsqu'elle n'écrit pas en dehors de la portion de bande où est écrite l'entrée.

p171 Prop 47 $\{\langle n,w\rangle / n$ une MTLB, $w\in\mathcal{L}(n)\}$ est récursif

p172 Prop 48 $\{\langle n\rangle / n$ une MTLB, $\mathcal{L}(n)=\emptyset\}$ est indécidable.

2) Problème de correspondance de Post

(C) p163 Déf 49 (PCP) Le problème de correspondance de Post est le suivant
- une instance est une suite $(u_1,v_1),\dots,(u_m,v_m)$ de paires de mots
- une solution est une suite d'indices $i_1,\dots,i_n$ de $\{1,\dots,m\}$ telle que $u_{i_1}\dots u_{i_n}=v_{i_1}\dots v_{i_n}$
- une instance est positive si elle a au moins une solution

p165 Prop 50
- PCP n'est pas récursif
- PCP est RE

Cor 51 Pour G, G' des grammaires algébriques d'axiomes resp. S et S', les problèmes suivants ne sont pas récursifs : p167
- $\mathcal{L}_G(S)\cap\mathcal{L}_{G'}(S')=\emptyset$?
- $\mathcal{L}_G(S)=\mathcal{L}_{G'}(S')$?
- $\mathcal{L}_G(S)=\Sigma^*$?
- G ambiguë ?

3) Décidabilité de théories logiques (Le)

Déf 52 Une théorie T est récursivement axiomatisable lorsque l'ensemble des axiomes de T est récursif. p172

Déf 53 Une théorie T est décidable lorsque $\{\langle t\rangle / t$ théorème de $T\}$ est récursif p172

Prop 54 Si T est récursivement axiomatisable, $\{\langle\phi,d\rangle, \phi$ formule, d preuve de ϕ dans $T\}$ est récursif p172

Cor 55 Si T est récursivement axiomatisable, $\{\langle t\rangle / t$ théorème de $T\}$ est RE et $\{\langle\phi\rangle / \phi$ formule valide$\}$ est RE p173

Cor 56 Si T est récursivement axiomatisable et complète alors T est décidable. p173

Th 57 (Presburger) La théorie au premier ordre des entiers munis de l'addition est décidable DEV (C) p173

4) Réécriture

Th 58 La terminaison d'un système de réécriture est indécidable. (B) p99

<u>Références</u>

Lassaigne / Rougemont

Carton, <u>Langages formels</u>

Cori / Lascar II, <u>Logique mathématiques</u>

Wolper, <u>introduction à la calculabilité</u>

Baader / Nipkow, <u>Term rewriting and all that</u>

Sipser (GNS), <u>Introduction to the theory of computation</u>.

Rayan Chikhi
Dominique Lavenier

<u>Ex:</u> de prédicat réc mais pas prim. réc.

Soit $(f_n)_n$ une énumérat° des fct°s prim. réc.

$\forall n, \quad p_n : x \longmapsto \min(f_n(x), 1)$

Soit $q : n \longmapsto 1 - p_n(n)$.

Si q prim. réc. : $\exists k, q = f_k$

$q(x) = \min(q(x), 1) = \min(f_k(x), 1) = p_k$
(car $\mathrm{Im}\, q \subset \{0,1\}$)

$\Rightarrow q(k) = 1 - p_k(k)$ et $q(k) = p_k(k) \longrightarrow \boxed{\text{NON}}$.

q est un <u>prédicat primitif récursif</u>.

Pb de pavages – pavages de Wang – Dynamic logic

Caractérisation des ensembles récursivement énumérables (RE)

Réfs : [illegible] Cori/Lascar II (Carnet [illegible]) → Dév de Théo [illegible]

Th Soit $A \subseteq \mathbb{N}$, 1) $A \in RE$ $\iff$ 2) $A = \emptyset$ ou $A = \mathrm{Im} f$ avec f primitive récursive

$\iff$ 3) $A = \mathrm{Im}\, g$ avec g récursive partielle

$\iff$ 4) $A = \pi_2^2(B)$ où $B \subseteq \mathbb{N}^2$ est primitif récursif et π_2^2 est la projection dans $\mathbb{N}^2$ par rapport à la seconde variable.

---

dém 1) $\Rightarrow$ 2) Soit $A \in RE$, on suppose $A \neq \emptyset$. Soit $y \in A$.

Comme $A \in RE$, il existe une MT $\mathcal{M}$ qui accepte A :

Pour $\alpha \in \mathbb{N}$, $\mathcal{M}$ prend en entrée un codage de α (par ex binaire) et accepte ssi $\alpha \in A$.
→ Dans la suite, on confond α et son codage.

On définit : – $B : \mathbb{N}^2 \to \{0,1\}$
$(t,\alpha) \mapsto \begin{cases} 1 & \text{si } \mathcal{M} \text{ s'arrête en } t \text{ étapes à partir de } \alpha \\ 0 & \text{sinon} \end{cases}$ donc $B(t,\alpha) = \mathrm{stop}(\mathrm{config}(t,\alpha))$

où $\mathrm{stop}(c) = \begin{cases} 1 & \text{si } c \text{ est finale} \\ 0 & \text{sinon} \end{cases}$ et $\mathrm{config}(t,\alpha)$ renvoie la configuration de $\mathcal{M}$ après t étapes de calcul en partant de α. Comme stop et config sont primitives récursives, B est primitive récursive (par composée)

– $\varphi : \mathbb{N} \to \mathbb{N}^2$, $k \mapsto (t_k, \alpha_k)$ bijection primitive récursive (ex arg. diagonal) ← regarder proprement.

On considère l'ensemble $C = \{k \in \mathbb{N} / B(t_k,\alpha_k) = 1\}$, par définition C est primitif récursif (car B l'est.)

On pose $f : \mathbb{N} \to \mathbb{N}$
$k \mapsto \begin{cases} \alpha_k & \text{si } B(t_k,\alpha_k) = 1 \\ y & \text{sinon} \end{cases}$ donc $f = \pi_2^2(\varphi)\chi(C) + y\,\chi(\mathbb{N}\setminus C)$ (où χ = fct caractéristique)

Ainsi f est primitive récursive (somme, pdt de fct prim. réc).

Ainsi $\alpha \in A \iff \exists t \in \mathbb{N} / B(t,\alpha) = 1 \underset{\varphi \text{ bij}}{\iff} \exists k \in \mathbb{N} / \alpha = \alpha_k$ et $B(t_k,\alpha_k) = 1 \underset{\text{déf de } f \text{ et } y \in A}{\iff} \exists k \in \mathbb{N} / f(k) = \alpha \iff \alpha \in \mathrm{Im} f$

D'où $A = \mathrm{Im} f$ et f primitive récursive.

2) $\Rightarrow$ 3) • On suppose que $A = \mathrm{Im} f$ avec f primitive récursive. Comme f est récursive partielle, on a le résultat.

• On suppose que $A = \emptyset$ alors $A = \mathrm{Im}\, g$ avec $g : \alpha \in \mathbb{N} \mapsto \mu y (y+1=0)$ récursive partielle.

Ainsi dans les deux cas, $A = \mathrm{Im}\, g$ avec g récursive partielle.

3) $\Rightarrow$ 4) On suppose que $A = \mathrm{Im}\, g$ avec g récursive partielle.

Comme g est récursive partielle, il existe une MT $\mathcal{M}$ qui calcule g.

On définit $\mathcal{C} : \mathbb{N}^2 \to \mathbb{N}$
$(t,\alpha) \mapsto a$ = le nombre calculé par la MT $\mathcal{M}$ à partir de α en t étapes.

donc $\mathcal{C}(t,\alpha) = \mathrm{sortie}\left[\mathrm{config}\left(\alpha, \mu(i \le t)\left(\mathrm{stop}(\mathrm{config}(\alpha,i))\right)\right)\right]$ où $\mathrm{sortie}(c)$ renvoie le nombre écrit sur le ruban dans la configuration c. Comme sortie, stop, config et μ (minimisation bornée) sont primitives récursives, $\mathcal{C}$ est primitive récursive (par composition).

On pose $G = \{(k,a) \mid e(k, \ldots) = a\}$: G est primitif récursif. ($(k, \ldots) = g(k)$ avec ... (1 ⇒ 2)).

Alors $a \in A \Leftrightarrow \exists x \in \mathbb{N} / a = g(x) \Leftrightarrow \exists (t,x) \in \mathbb{N}^2 / e(t,x) = a \Leftrightarrow \exists k \in \mathbb{N} / e(k, \ldots) = a \Leftrightarrow \exists k \in \mathbb{N} / (k,a) \in G$

$A = \mathrm{Im}\, g$ — déf de g — déf de φ

D'où $\underline{A = \pi_2^2(G)}$.

(3 ⇒ 1) On suppose que $A = \pi_2^2(B)$ où $B \subset \mathbb{N}^2$ primitif récursif (i.e. $x \in A$ ssi $\exists y \in \mathbb{N} / (y,x) \in B$)

On définit une MT M qui accepte A : soit M définie par $M(x)$:

[pour $k = 0$ à ∞ :
  [si $x = \pi(\varphi(k))$ et si $\varphi(k) = (k, x) \in B$
    accepter]
]

Alors M accepte uniquement les éléments de A.

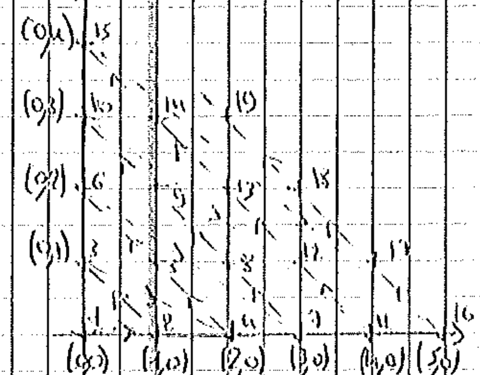
D'où $\underline{A \in RE}$. □

[ou bien on calcule la fonction récursive $x \mapsto \mu y((x,y) \in B)$]

Construction d'une bijection primitive récursive

On énumère les couples de $\mathbb{N}^2$ selon les diagonales :

$f : (t,x) \in \mathbb{N}^2 \mapsto \frac{(x+t)(x+t+1)}{2} + t \in \mathbb{N}$



f a bien ses valeurs dans $\mathbb{N}$ car $(x+t)(x+t+1)$ est paire.

. f récursive primitive (car addition (récursive), produit et quotient le sont)

. f bijective et $f^{-1} : k \in \mathbb{N} \mapsto (\pi_2^1(k), \pi_2^2(k))$ où $\begin{cases} \pi_2^1(k) = \mu(t \le k)(\exists x \le k, f(t,x) = k) \\ \pi_2^2(k) = \mu(x \le k)(\exists t \le k, f(t,x) = k) \end{cases}$.

Soit M une MT, une configuration c de M est (avant, état, reste)

. stop : $c \mapsto \begin{cases} 1 & \text{si } c \text{ est finale} \\ 0 & \text{sinon} \end{cases}$ primitive récursive : car c finale ssi état $\pi_3^2(c)$ est final et il y a un nb fini d'états finaux (...) donc c'est un ens. primitif récursif.

. suite de $c \mapsto$ avant : primitive récursive (car $\pi_3^1(c)$ = avant)

. config(t,c) primitive récursive car config suiv(c), qui donne la configuration donnée après une étape de calcul de M sur c, est primitive récursive — elle est définie par cas sur les transitions possibles.

$\mathcal{L} = \varphi ::= (x = y) \mid (x+y=z) \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x \varphi$ ($\wedge$ et $\forall$ sont obtenus avec des négations)

Décidabilité de l'arithmétique de Presburger

Manon Ruffini

Théorème 1 (*Presburger*)

La théorie au premier ordre des entiers munis de l'addition et de l'égalité est décidable.

Grammaire ?

On veut montrer que pour toute formule φ close de la logique de Presburger, on peut décider si $\mathbb{N} \models \varphi$. On peut supposer que φ est sous forme prénexe : $\varphi = Q_1x_1...Q_nx_n\psi$, où les Q_i sont des quantificateurs.
Pour tout entier k tel que $0 \leqslant k \leqslant n$, soit $\varphi_k = Q_{k+1}x_{k+1}...Q_nx_n\psi$; avec les conventions $\varphi_0 = \varphi$ et $\varphi_n = \psi$. Pour $0 \leqslant k \leqslant n$, les variables $x_1, ..., x_k$ sont exactement les variables libres dans la formule φ_k, et on peut écrire $\varphi_k(x_1, ..., x_k)$. ← codages ⇒ définir le codage avant
On définit l'ensemble $X_k := \{(n_1, ..., n_k) \mid (\mathbb{N}, \{x_i := n_i\}_{1\leqslant i \leqslant k}) \models \varphi_k(x_1, ..., x_k)\}$.
On va montrer par récurrence sur $n - k$ que l'ensemble X_k est un ensemble rationnel.
D'abord, on définit un codage qui permet d'écrire les k-uplets d'entiers :
- Chaque entier est écrit en binaire sur l'alphabet $\Sigma = \{0, 1\}$.
- Un k-uplet d'entier est écrit sur l'alphabet $\Sigma^k = \{0, 1\}^k$, en ajoutant éventuellement des zéros en tête des écritures pour les rendre de la même longueur.

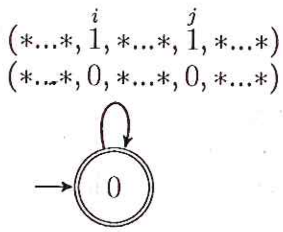
  Si $\forall 1 \leqslant i \leqslant k, x_i$ est codé $x_i^1x_i^2...x_i^r$, le k-uplet est codé $(x_1^1x_2^1...x_k^1)(x_1^2...x_k^2)...(x_1^k...x_k^k)$

Exemple 1

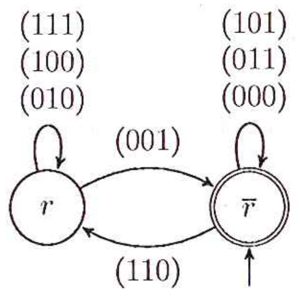
2, 3 et 7 sont codés respectivement sur Σ : (010), (011) *et* (111) ; *et le triplet* (2, 3, 7) *est codé* (001)(111)(011)

Pour tout $1 \leqslant k \leqslant n$, on construit un automate $\mathcal{A}_k$ qui accepte les écritures sur Σ^k des éléments de X_k. On le fait par récurrence sur $n - k$.
- Initialisation : construisons un automate $\mathcal{A}_n$ qui accepte exactement X_n, l'ensemble des n-uplets qui satisfont ψ. On le fait par induction sur la formule ψ.

  Si $\psi = (x_i = x_j)$: On construit facilement l'automate pour l'égalité :



  Si $\psi = (x_i + x_j = x_k)$: On construit un automate comme s'il lisait de la droite vers la gauche. L'état r correspond à l'état avec retenue et l'état $\overline{r}$ correspond à l'état sans retenue. On n'écrit pas les $*$, pour plus de clarté (on suppose i, j, k distincts).



On en déduit facilement les automates pour i, j, k quelconques.

Si $\psi = \neg\psi_1$: Soit $\mathcal{A}_\psi$ l'automate qui accepte exactement les n-uplets satisfaisant ψ. On construit son complémentaire : il accepte exactement les n-uplets d'entiers qui satisfont $\neg\psi$.

Si $\psi = \psi_1 \wedge \psi_2$: On contruit l'automate de l'intersection de $\mathcal{A}_{\psi_1}$ et $\mathcal{A}_{\psi_2}$

Si $\psi = \psi_1 \vee \psi_2$: On contruit l'automate de l'union de $\mathcal{A}_{\psi_1}$ et $\mathcal{A}_{\psi_2}$

– Pour le passage de $\mathcal{A}_{k+1}$ à $\mathcal{A}_k$: Soit $1 \leqslant k < n$. On suppose construit l'automate $\mathcal{A}_{k+1}$. On a : $\varphi_k = Q_{k+1}x_{k+1}\varphi_{k+1}$, où Q_{k+1} est le quantificateur $\forall$ ou $\exists$. Dans le cas où Q_{k+1} est $\forall$, on utilise : $\varphi_k = \neg\exists x_{k+1}\neg\varphi_{k+1}$ et la stabilité des langages rationnels par complémentation. Il reste à traiter le cas $Q_{k+1} = \exists$.

On note : $\mathcal{A}_k = (M_k, \Sigma^k, M_k^0, F_k, E_k)$, et $\mathcal{A}_k = (M_{k+1}, \Sigma^{k+1}, M_{k+1}^0, F_{k+1}, E_{k+1})$, où : M_i aux états, Σ^i à l'alphabet, M_i^0 aux états initiaux, F_i aux états finaux et $E_i \subset M_i \times \Sigma^i \times M_i$ aux transitions. Par hypothèse de récurrence, l'automate $\mathcal{A}_{k+1}$ est construit ; il accepte exactement X_{k+1}.

On définit $\mathcal{A}_k$ par :

$M_k = M_{k+1}$

$(p, (x_1...x_k), q) \in E_k$ si et seulement si $\exists x_{k+1}, (p, (x_1...x_k x_k + 1), q) \in E_{k+1}$

$F_k = F_{k+1}$

$M_k^0 = \{q \in M_k | q$ soit accessible depuis un état de M_{k+1}^0 en lisant $(0...0) \in \Sigma^k\}$, ce qui permet de prendre en compte les x_{k+1} ayant une écriture binaire plus longue que celle des $x_1, ..., x_k$.

Par construction, l'automate $\mathcal{A}_k$ accepte exactement X_k.

Ainsi, on peut construire l'automate $\mathcal{A}_1$ qui accepte exactement $X_1 = \{n_1 \in \Sigma | (\mathbb{N}, x_1 := n_1) \models Q_1 n_1 \varphi_1(x_1)\}$.

Pour conclure :
- Si $\varphi = \forall x_1 \varphi_1(x_1)$: $\mathbb{N} \models \varphi$ ssi $\mathcal{L}(\mathcal{A}_1) = \{0,1\}^*$
- Si $\varphi = \exists x_1 \varphi_1(x_1)$: $\mathbb{N} \models \varphi$ ssi $\mathcal{L}(\mathcal{A}_1) \neq \emptyset$

Quelle est la complexité ? // Presburger ~ quelle complexité est ...

φ close, $\mathcal{P}(\varphi)$
$\mathbb{N} \models \varphi$ ssi $\mathcal{L}(\mathcal{A}_{\varphi(\cdot)}) = \{\cdot\}$ ← codage du tuple à 0 élément
($\mathcal{L}(\mathcal{A}_{\varphi(\cdot)}) = \emptyset$ ssi $\mathbb{N} \not\models \varphi$)
ou alors : sur-ensemble des variables libres

On prend φ une formule quelconque. On montre par induction

$\mathcal{P}(\varphi)$ = "Si $\{x_1 \dots x_k\}$ est un sur-ensemble des variables libres de φ, alors il existe un automate $\mathcal{A}_\varphi$ / $\mathcal{L}(\mathcal{A}_\varphi) = \{c \,/\, c$ codage de $(n_1, \dots, n_k)$ et $\mathbb{N}, \{x_i := n_i\}_{i=1..k} \models \varphi\}$"

Pb: Entrée : $\varphi \in \mathcal{L}$ close
Sortie : oui lorsque $\mathbb{N} \models \varphi$

~ $\mathcal{P}(x_1 = x_2)$
~ $\mathcal{P}(x_1 + x_2 = x_3)$
~ Soit $\varphi(x_1 \dots x_n)$ tq $\mathcal{P}(\varphi)$
Mque $(\exists x_n \varphi)$
...

*Pb : ds le $\forall$: si on ajoute des variables ~ Il faut prendre un $\boxed{\text{sur-ensemble}}$ des variables libres (on garde les $(* \dots *)$)
... $\mathcal{L}(\mathcal{A}_{\varphi(x_1 \dots x_k)})$.