Université de Caen

Thèse de Doctorat en Sciences

Spécialité : Fondements Mathématiques de l'Informatique

Caractérisations Logiques des Problèmes NP : Robustesse et Normalisation

Frédéric Olive

Soutenue le 2 Juillet 1996

Jury : Patrick Cegielski *Rapporteur*
Georg Gottlob *Rapporteur*
Étienne Grandjean *Directeur*
Serge Grigorieff
Jean-Jacques Hébrard
James F. Lynch *Rapporteur*
Jean-Éric Pin
Denis Richard

Caractérisations Logiques des Problèmes NP : Robustesse et Normalisation

Frédéric Olive

GREYC, Université de Caen,
Esplanade de la Paix, 14032, CAEN cedex.

email : olive@logique.jussieu.fr

*“Puisque ces mystères me dépassent,
feignons d'en être les organisateurs.”*

Jean Cocteau, Les mariés de la Tour Eiffel.

À *Carmen*, qui est le sel de la vie.
À *Anton*, petit loukoum vociférant.

Remerciements

Quelques lignes ne suffiront certe pas à donner la mesure de tout ce que cette thèse doit à Étienne Grandjean. La relation intime qu'il entretient avec la Théorie de la Complexité, les intuitions saisissantes que cette relation permet, n'ont cessé de me surprendre et de me stimuler durant plus de trois ans. Son humilité, le respect sans faille qu'il accorde à ceux qui le côtoient, rendent son expérience scientifique d'autant plus accessible et enrichissante.

Ces qualités - professionnelles et humaines - me semblent procéder d'un même choix : celui de l'Humanisme, dont le progrès scientifique est largement tributaire. En tout état de cause, elle font d'Étienne Grandjean un directeur de thèse exemplaire. Je lui suis très reconnaissant d'avoir aussi généreusement contribué à ce travail.

Je suis redevable à Serge Grigorieff et Richard Lassaigne de m'avoir présenté Étienne, lorsque j'évoquais le souhait de travailler sur la définissabilité logique des classes de complexité. Je les remercie vivement de cela et de l'attention amicale qu'ils m'ont accordée à de multiples reprises.

Je suis très honoré que Patrick Cegielski, Georg Gottlob et James F. Lynch aient accepté d'être les rapporteurs de ce mémoire. Je les en remercie.

Je remercie Serge Grigorieff, Jean-Jacques Hébrard, Jean-Éric Pin et Denis Richard, d'avoir témoigné de l'intérêt pour mon travail en acceptant de faire parti de mon jury de thèse.

Plusieurs co-thésards ont joué un rôle important dans ma capacité à affronter la situation de doctorant. Je remercie en particulier les joyeux troublions du "bureau 507", ainsi qu'Arnaud Durand, qui m'a systématiquement offert "le gîte et le couvert" lors de mes passages à Caen, et Thierry Lacoste, avec qui j'ai partagé mille doutes et quelques certitudes.

La présence de Carmen, ma compagne, est consubstantielle à l'intérêt que je porte à la recherche en particulier, et au bonheur de vivre en général. Au point qu'il serait tout à fait dérisoire de vouloir l'en remercier ici.

Avant-Propos

En proposant une interprétation des problèmes combinatoires en termes de structures finies, la *Théorie des Modèles Finis* a mis à jour des liens féconds entre la Complexité et la Logique. Au titre de ces liens notamment, les divers résultats de *caractérisation logique* établis depuis 1960, qui identifient différents échelons de la Hiérarchie Polynomiale de Stockmeyer à des fragments de la logique du second ordre.

Cette thèse consiste essentiellement en l'étude des caractérisations logiques de la complexité en temps linéaire non-déterministe.

Notons $\mathcal{L}(+)$ la classe des formules du type $\exists\overline{U}\forall\overline{x}\exists\overline{y}\phi$, où $\overline{U}$ est une suite finie de relations unaires, $\overline{x}$, $\overline{y}$ des tuples de variables, et ϕ une formule sans quantificateur dont la signature contient le symbole prédéfini $+$ (relation ternaire d'addition).

Notons $\mathcal{G}_{\wedge}^{+}(0,s)$ la classe des formules du type $\exists\overline{g}\forall x\phi$, où $\overline{g}$ est une suite finie de fonctions unaires, et ϕ une simple conjonction de formules atomiques portant sur l'unique variable x, et dont la signature contient les symbole prédéfinis 0 et s (constante *zéro* et fonction *successeur*).

Appelons enfin NLIN la classe des langages reconnus en temps $n/\log n$ sur RAM non-déterministe.

Nous prouvons que pour le codage usuel des mots par des structures relationnelles, chaque langage de NLIN est exprimable par une formule de $\mathcal{L}(+)$.

Puis nous établissons une caractérisation exacte de NLIN par la logique $\mathcal{G}_{\wedge}^{+}(0,s)$, mais cette fois-ci, pour un codage des mots en termes de *structures unaires*, c'est à dire de structures dont la signature ne comporte que des symboles de fonction unaires.

Nous en déduisons une caractérisation par des formules purement conjonctives de la classe NP, pour les structures unaires.

Enfin nous donnons une nouvelle preuve de la caractérisation de la classe des langages rudimentaires par la logique du second ordre monadique avec addition.

Table des matières

Avant-Propos **5**

Introduction **9**

1 Monadic logical definability of NP-complete problems **19**
Introduction 19
1.1 NLIN characterization and sketch of proof of the main result 22
1.2 NLIN characterization with unary functions and only one variable . . 26
1.3 From unary functional language to monadic relational language 34
1.4 Definability of arithmetical objects 39
1.5 Connection between F_w and X_w 44
1.6 From cn to n 49
Conclusion 51

2 Une Caractérisation Conjonctive des classes NTIME$_{RAM}(n^k)$ **53**
Introduction 53
2.1 Plan détaillé du chapitre 57
2.2 Une forme normale dans $\mathcal{G}^+(\mathcal{T}, 0, s)$ 61
2.3 Définissabilité dans les classes $\mathcal{G}^+_\wedge(0, s)$ et $\mathcal{G}^+(0, s)$ 64
2.4 Pseudo-réduction de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ 70
2.5 Un "pré-travail" sur $\mathcal{G}^+(\mathcal{T}, 0, s)$ 76
2.6 Réduction de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ 81
2.7 Caractérisation de NLIN et généralisation 85
2.8 Définissabilité Conjonctive de l'Arithmétique 95
Conclusion 105

3 Relations rudimentaires **107**
3.1 Rudimentary Languages and Second-Order Logic 107
3.2 Le comptage modulaire de RUD est dans $Sp(f, b)$ 114

Conclusion **123**

Bibliographie **129**

Introduction

La théorie de la complexité algorithmique se propose de donner un sens à la notion de *difficulté de résolution* des problèmes combinatoires. Après avoir formalisé de plusieurs manières les notions de *problème*, d'*algorithme* et d'*efficacité* des algorithmes, elle tente de décrire les relations qui unissent les divers modèles proposés et de classifier les problèmes combinatoires en fonction de leur difficulté intrinsèque.

Ce deuxième aspect de la théorie suscite l'apparition d'une multitude de classes, plus ou moins significatives. Certaines s'imposent comme des "pierres angulaires" de la théorie. Les classes P et NP par exemple, suscitent l'intérêt autant pour leurs propriétés mathématiques que pour leur signification intuitive : la première réunit les problèmes communément considérés comme *effectivement* décidables, tandis que les problèmes de la seconde, bien qu'apparemment proches de ceux de P, sont théoriquement décidables mais d'un traitement algorithmique inaccessible dans les faits. D'autres classes en revanche, dont "l'apparition" relève plus du hasard que de la nécessité, paraissent condamnées à un moindre rôle dans l'entreprise de classification évoquée plus haut.

Si cette prolifération sous-tend la phase d'analyse nécessaire d'une théorie en pleine expansion, il semble indispensable, en l'attente d'une approche plus synthétique, de se doter de critères permettant de juger de l'opportunité de nouvelles définitions. Bien que rien n'ait été explicité sur ce sujet, le folklore retient implicitement les trois critères suivants.

Robustesse. Ce critère est l'expression d'une exigence d'*abstraction* : une classe de complexité $\mathcal{C}$ est dite *robuste* si sa définition n'est pas liée de façon trop rigide à un modèle de calcul particulier. La classe NP, par exemple, peut être définie de manières équivalentes à partir de différents modèles de machine de Turing ou de RAM (machines à accès direct).

Existence de problèmes complets. Un problème $\mathcal{S}$ est dit complet pour une classe de complexité donnée $\mathcal{C}$ s'il est dans $\mathcal{C}$ et si tout problème de $\mathcal{C}$ se réduit à P *via* une transformation qui laisse $\mathcal{C}$ invariante. (Ainsi, $\mathcal{S}$ est NP-complet si $\mathcal{S}$ est NP et si tout problème NP se réduit à $\mathcal{S}$ *en temps déterministe polynomial*). Comme les classes de complexité sont toujours définies comme des ensembles de problèmes dont le traitement requiert *au plus* telle ou telle complexité, toute classe contient des problèmes triviaux. L'importance d'une classe est donc plutôt caractérisée par ses problèmes les plus durs. Or l'interprétation usuelle de la complétude veut précisément qu'un problème $\mathcal{C}$-complet soit au moins aussi difficile que n'importe quel problème de $\mathcal{C}$. D'où l'intérêt de cette notion.

Caractérisation logique. Ce dernier critère renvoie à la même préoccupation que le premier : l'objectif est de renforcer la robustesse d'une classe de complexité en en donnant une définition indépendante de tout modèle de calcul. L'idée est de chercher à caractériser les problèmes de cette classe par notre capacité à les *décrire* plutôt que par notre facilité à les *résoudre*.

Cette thèse se veut une contribution à l'étude des *caractérisations logiques* de la complexité en *temps linéaire*. Nous allons donc préciser ces deux notions avant de présenter les résultats obtenus dans ce travail.

Caractérisation logique des classes de complexité

Nous avons motivé la recherche de caractérisations logiques par l'exigence de robustesse des classes de complexité. Nous pourrions justifier cette recherche en termes plus généraux en l'identifiant à *l'étude des liens qui unissent la difficulté de résolution d'un problème et la structure du langage nécessaire à sa description*. Le cadre mathématique qui permet de donner un sens formel à cette étude est au croisement de deux théories : la théorie de la complexité algorithmique, déjà présentée, et la théorie des modèles finis.

Dans la première on confond la notion de *problème* que l'on cherche à résoudre et celle de *langage formel* dont on cherche à reconnaître les éléments, et on formalise la notion de *difficulté* comme la quantité de *ressources* (temps, espace ...) utilisée par un modèle de calcul pour décider de l'appartenance d'un mot donné au langage étudié.

Dans la seconde, on associe à chaque *problème* une classe de *structures finies* et on définit la *complexité descriptive* du problème comme la *structure syntaxique* de la formule caractérisant la classe de structures associée.

Ces points de vue constituent deux approches naturelles des problèmes combinatoires :

- la description d'un problème sous forme de mots est une normalisation usuelle, dès lors qu'on veut interpréter les instances de ce problème comme les entrées d'un calcul ;

– la présentation en termes de structures finies va de soit, puisque c'est celle qu'adopte "spontanément" la plupart des problèmes qui intéressent l'algorithmique, notamment les nombreux problèmes de graphes (problèmes de flots, problèmes de parcours, ...).

De plus ces points de vue sont facilement "interchangeables" :

– la représentation des structures finies sous forme de mots n'est qu'une question de choix du codage ;

– inversement, tout mot s'interprète en terme de structure grâce, par exemple, à l'identification d'un mot $w = w_0 \dots w_{n-1}$ sur l'alphabet $\{a_1, \dots, a_p\}$ à la structure finie $\langle D, U_1, \dots, U_p \rangle$, où D est le domaine $\{0, \dots, n-1\}$ et $U_1, \dots, U_p$ sont les relations unaires sur D définies par les assertions (pour $i = 1, \dots, p$) :
$$\forall x \in D,\ x \in U_i \text{ ssi } w_x = a_i.$$

Ceci étant, adoptons ce second point de vue et considérons un problème S, vu comme ensemble de structures finies de même signature $\mathcal{T}$, et une formule Φ de signature $\mathcal{T}$ (toutes les formules considérées dans ce rapport appartiendront à la *logique du second ordre*, notée SO).

On dit que Φ caractérise S *si* S *est la classe mod* (Φ) *des modèles de* Φ.

On généralise cette définition à la caractérisation d'une *classe* de problèmes par une *logique* : soient donc $\mathcal{S}$ une classe de problèmes (par exemple une classe de complexité) et $\mathcal{F}$ une classe de formules. Notons *Mod* ($\mathcal{F}$) la classe des ensembles *mod* (Φ) quand Φ parcourt $\mathcal{F}$.

On dit que la classe de formules $\mathcal{F}$ caractérise *la classe de problèmes* $\mathcal{S}$ *si* $\mathcal{S} \subseteq Mod\ (\mathcal{F})$. *On dit de plus que cette caractérisation est* exacte *si* $\mathcal{S} = Mod\ (\mathcal{F})$.

On note abusivement $\mathcal{S} \subseteq \mathcal{F}$ et $\mathcal{S} = \mathcal{F}$ plutôt que $\mathcal{S} \subseteq Mod\ (\mathcal{F})$ et $\mathcal{S} = Mod\ (\mathcal{F})$. Autrement dit :

$\mathcal{S} \subseteq \mathcal{F}$ si pour tout ensemble S de $\mathcal{T}$-structures finies,
$$S \in \mathcal{S} \Rightarrow \exists \Phi \in \mathcal{F}\ :\ S = mod\ (\Phi).$$

$\mathcal{S} = \mathcal{F}$ si pour tout ensemble S de $\mathcal{T}$-structures finies,
$$S \in \mathcal{S} \Leftrightarrow \exists \Phi \in \mathcal{F}\ :\ S = mod\ (\Phi).$$

L'étude des liens entre complexité algorithmique et complexité descriptive se ramène donc à la recherche, pour chaque classe de complexité $\mathcal{S}$, d'une classe de formules $\mathcal{F}$ qui réalise l'égalité $\mathcal{S} = \mathcal{F}$. Depuis 1960, de nombreuses contributions à cette étude ont été apportées. Nous nous contenterons d'en citer quelques unes, qui donnent une idée du pouvoir d'expression de la logique du second ordre et de ses fragments.

Quelques exemples

Si l'on ne lui impose aucune restriction, la logique du second ordre caractérise exactement la Hiérarchie Polynomiale. C'est l'égalité :

$$PH = SO$$

prouvée par Stockmeyer en 76 ([68]). Si l'on se restreint au second ordre existentiel ($\exists SO$), on "capture" précisément la classe NP. L'égalité

$$\mathrm{NP} = \exists SO$$

est établie par Fagin en 74 ([20]). Le résultat de Stockmeyer précité en est un corollaire. On peut appauvrir d'avantage la logique du second ordre en jouant sur l'arité des relations existentiellement quantifiées. En se limitant au second ordre existentiel *monadique* ($\exists SO_1$, tous les symboles de relation sont unaires), on "tombe" dans une classe de langages trop réduite pour être formulée en terme de classe de complexité. Mais dès que l'on autorise l'usage d'une relation d'ordre prédéfinie, on exprime la plus petite classe de complexité, à savoir la classe REG des langages réguliers (qui est aussi la classe des langages reconnus en temps "réel" et/ou en espace constant). Ce résultat est la première caractérisation logique de classe de complexité. Il est prouvée indépendamment par Büchi ([4], 1960) et Elgot ([18], 1961), et s'écrit :

$$REG = \exists SO_1(<).$$

La logique $\exists SO_1(+)$, obtenue en substituant l'addition à l'ordre dans la logique de Büchi, a clairement un pouvoir d'expression compris entre celui de $\exists SO_1(<)$ et celui de $\exists SO$ (l'ordre est évidemment définissable dans $\exists SO_1(+)$, et l'addition est définissable dans le second ordre existentiel d'arité au plus 2 (voir [52]), donc *a fortiori* dans $\exists SO$). On ne connaît pas le pouvoir d'expression exact de cette logique, mais Lynch a montré ([52]) qu'elle exprimait *au moins* tous les problèmes reconnaissables par une machine de Turing en temps linéaire non déterministe. C'est l'inclusion

$$\mathrm{NTIME}(n) \subseteq \exists SO_1(+),$$

que Lynch a précisée en 92 et sur laquelle nous reviendrons abondamment, puisqu'elle motive le travail dont rend compte le chapitre 1 de cette thèse.

Dans un article de 92, [28], Grädel définit de nouveaux fragments de SO par restriction sur la partie du premier ordre des formules considérées. En supposant que le préfixe du premier ordre est universel et que la matrice est une conjonction de formules de Horn, il capture exactement la classe P des problèmes reconnaissable en temps polynomial déterministe sur machine de Turing :

$$P = \exists SO(s, <)[\forall^*, Horn].$$

Pour la même restriction sur les quantificateurs du premier ordre, et en limitant les matrices à des conjonctions de 2-clauses, il obtient l'égalité :

$$NL = \exists SO(s,<)[\forall^*, 2 - clauses],$$

où NL désigne la classe de complexité en espace logarithmique non déterministe.

Dans tous ces exemples, on construit des fragments du second ordre en limitant les préfixes du premier et second ordre et/ou l'arité des relations, et/ou en enrichissant la signature de symboles prédéfinis (s, $<$, $+$). Une autre manière de définir des sous-classes de cette logique est de partir de la logique du premier ordre et de l'enrichir d'opérateurs sur les relations. Ainsi, si l'on note LFP (resp. TC) l'opérateur qui à toute relation associe son plus petit point fixe (resp. sa clôture transitive), on obtient deux nouvelles caractérisations des classes P et NL :

$$P = FO(s) + LFP \text{ et } NL = FO(s) + TC,$$

La caractérisation de P est prouvée indépendamment par Immerman ([43]) et Vardi ([72]). La caractérisation de NL est établie par Immerman dans [44].

L'exemple qui suit montre qu'une approche *sémantique* des fragments de SO peut se révéler féconde. En 94, Lautemann, Schwentick et Thérien cherchent à caractériser la classe des *langages context-free* (CFL). Il leur faut donc rechercher une logique plus riche que $\exists SO_1(<)$, pour "sortir" des langages réguliers, et moins expressive que la classe $\exists BFO(<)$ des formules du second ordre avec un unique symbole de relation d'arité 2, quantifié existentiellement, dont on sait qu'elle permet d' exprimer tous les langages context-free, mais aussi certains langages non context-free. Pour ce faire, ils restreignent cette dernière logique en "forçant" le quantificateur du second ordre à porter sur une relation binaire *de matching* (une restriction du couplage). Ils notent $\exists \text{Match} FO(<)$ la logique obtenue et prouvent ([50]) :

$$CFL = \exists \text{Match } FO(<).$$

Pour finir, signalons que les caractérisations logiques ne sont pas réservées aux classes de langages comprises entre REG et PH.

"Au dessus de PH", Vardi caractérise dans [72] la classe des langages reconnaissables en espace polynomial par l'égalité

$$PSPACE = FO(s) + PFP,$$

dans laquelle PFP désigne l'opérateur de *point fixe partiel.*

"En dessous de REG", Pin propose dans [59] un survol des caractérisations logiques de langages "sous-réguliers" par des fragments de $FO(<)$ et $FO(s)$.

Le temps linéaire

Les trois critères évoqués plus haut prennent une signification accrue pour les classes de *faible* complexité en général, de complexité *linéaire* en particulier. Bien que d'un usage courant pour les algorithmiciens, la notion de temps linéaire a longtemps résisté à la formalisation. L'ensemble des problèmes que ce terme recouvre étant par nature très sensible aux codages utilisés pour représenter les problèmes et aux modèles de calcul invoqués pour lui donner un sens précis, il semblait difficile de l'identifier à une classe de complexité robuste et bien définie. Précisons ces deux difficultés :

L'évaluation de la complexité d'un problème, dépend *a priori* du codage utilisé pour le représenter. Dire qu'une classe de graphes est reconnaissable en temps $O(n^2)$ ne signifie pas la même chose selon que le codage choisi pour un graphe $G = (V, E)$ a pour longueur $|V| + |E|$ ou $|V|^2$.

Il s'avère en fait que si C et C' sont deux codages *naturels* d'un problème fixé, leurs tailles sont polynomialement liées. Aussi, dès lors que l'on parle d'une classe de complexité suffisament large (*i.e.* au dessus de P), le choix du codage ne modifie pas le contenu de la classe.

Mais cette stabilité *modulo* le codage n'est plus respectée pour le temps linéaire : l'ensemble des entiers premiers, par exemple, est reconnaissable en temps linéaire pour un codage des entiers en "langage bâton" (qui représente un entier n par un mot de longueur n), mais il devient polynomial si l'on se rapporte à leur écriture en base 2.

La deuxième difficulté provient de la *fragilité* des définitions formelles du temps linéaire. Nous avons dit que l'on ne modifie pas la classe NP suivant les différents modèles de calcul choisis pour la définir. Ceci parce que ces modèles peuvent se simuler l'un l'autre en temps polynomial. Là encore, les choses sont différentes pour le temps linéaire. Les classes de problèmes reconnaissables en temps linéaire par une machine de Turing d'une part, par une RAM d'autre part, ne coïncident pas.

Nous verrons dans l'introduction du chapitre 1 que ce manque de robustesse apparemment intrinsèque du temps linéaire conduit certains chercheurs à récuser l'idée d'une formalisation "universelle" de cette notion (voir [65, 39, 27]). A la fin des années 80 pourtant, Grandjean définit un modèle unifié de cette complexité, dans ses versions déterministes et non déterministes, *via* les classes DLIN et NLIN.

Nous reviendrons plus loin sur la définition de ces deux classes. Disons seulement que NLIN est définie comme la classe des problèmes reconnaissables en temps $O(n)$ par une RAM non déterministe (voir la définition précise au paragraphe 1.1.1 p. 22), et que DLIN est la version déterministe de cette classe (obtenue en enlevant toute instruction non déterministe des RAM considérées).

Nous adoptons la thèse de Grandjean selon laquelle *ces deux classes offrent de "bonnes" définitions formelles des notions intuitives de temps linéaire déterministe et non déterministe.* Expliquons en quelques mots les raisons de ce choix[1] :

Le cadre déterministe est celui où se pose réellement la question de l'*adéquation de l'idée intuitive de temps linéaire et de sa formalisation* (la notion d'algorithme linéaire *concret* ne prend de sens que dans ce cadre). Or la classe DLIN semble effectivement correspondre à l'ensemble des problèmes décidables par un tel algorithme. En effet :

- DLIN n'excède pas l'ensemble de ces problèmes du fait même de sa définition, ou plus précisément, des instructions autorisées dans le modèle de calcul utilisé pour la définir. Nous verrons plus loin que le type de RAM invoqué par Grandjean ne sollicite que des opérations calculables en temps déterministe linéaire par une machine de Turing (baptisées "opérations LTTC" dans [34] et [37]).
- elle est suffisament large pour contenir tous les problèmes attendus. Dans trois articles récents ([34, 35, 36]), l'appartenance à DLIN d'un spectre large et diversifié des problèmes décidables en temps linéaire est établie.
- Enfin, la classe DLIN est robuste, en ce sens que l'on peut la définir à partir de différents types de RAM (obtenus les uns à partir des autres en modifiant les opérations LTTC auxquelles la machine peut recourir. Voir [34, 37]).

Dans le cadre non déterministe, l'argument essentiel du choix affirmé plus haut tient plutôt aux bonnes propriétés *structurelles* de la classe NLIN, c'est à dire en fait, à la satisfaction par cette classe des trois critères explicités au début de cette introduction :

- comme la classe DLIN, et pour les mêmes motifs, la classe NLIN est *robuste.*
- NLIN contient des *problèmes complets*: Grandjean a prouvé la NLIN-complétude des problèmes NP-complets $SAT_<(\mathbb{N})$[2] et RISA (Réduction d'un automate incomplètement spécifié) ([31, 32]).
- enfin NLIN admet une caractérisation logique : Grandjean a en effet montré que la classe NLIN coïncide - pour un codage bien choisi des problèmes - avec la classe *Mod* $(\mathcal{G}(<))$, où $\mathcal{G}(<)$ est la classe des formules du second ordre existentiel comportant une unique variable du premier ordre quantifiée universellement, et dont les symboles du second ordre (quantifiés ou non) sont des symboles de constante ou de fonction unaire, ou le symbole prédéfini $<$ (ordre naturel sur les entiers).

1. Pour une discussion plus détaillée de cette question, voir les introductions de [35] et [36]

2. $SAT_<(\mathbb{N})$ est le problème suivant : *Instance*: une conjonction ϕ de formules atomiques du type $f_1 \ldots f_p(a) = g_1 \ldots g_q(b)$ ou $f_1 \ldots f_p(a) < g_1 \ldots g_q(b)$, (où les f_i et les g_i sont des symboles de fonction unaires et a, b deux entiers spécifiés) ; *Question*: ϕ est-elle satisfaisable dans $\mathbb{N}$?

Il s'avère de plus que la classe NLIN contient la plupart des problèmes NP-complets naturels. Ceci nous renseigne sur la nature de ces problèmes. Dire qu'un problème est reconnaissable par un modèle de calcul non déterministe, c'est dire que sa résolution algorithmique passe par deux étapes : dans la première, on "devine" un certain nombre d'objets liés au problème ; dans la seconde, on vérifie de manière déterministe que ces objets satisfont bien diverses assertions.

Dire que le problème est dans NP, c'est affirmer que la taille des objets à deviner, ainsi que la "durée" de la vérification (au sens algorithmique) sont polynomiales en la taille de l'entrée. L'appartenance de problèmes NP-complets à NLIN indique qu'en fait la taille des objets à deviner n'excède pas celle de l'entrée, à une constante multiplicative près, et que la vérification peut se faire en temps linéaire.

Ainsi la difficulté des problèmes NP-complets semble-t-elle plus dépendre du recours nécessaire à des étapes non déterministes qu'au nombre de telles étapes effectivement utilisées dans leurs calculs.

La classe NLIN, et plus particulièrement sa caractérisation par la logique $\mathcal{G}(<)$, constitue la préoccupation principale de cette thèse.

Résultats

Le **chapitre 1** est le fruit d'un travail réalisé en commun avec Etienne Grandjean. L'objectif en est de situer la classe NLIN parmi les classes de langages définissables par un fragment de la logique du second ordre existentiel. On a vu plus haut que cette logique coïncide avec l'ensemble des langages NP, tandis que sa restriction aux formules monadiques avec l'ordre linéaire sur les entiers capture exactement la classe des langages réguliers. Nous prouvons que chaque langage de NLIN est caractérisé par une formule du second ordre existentiel monadique avec addition. Autrement dit :

$$\text{NLIN} \subseteq \exists SO_1(+).$$

Notre résultat indique qu'en conservant la restriction "monadique" de Büchi, mais en substituant l'addition à l'ordre sur les entiers, on gagne un grand pouvoir d'expression, puisque l'on arrive à définir tous les problèmes NLIN, et parmi eux les 21 problèmes NP-complets présentés par Karp ([48]) comme une sorte de "florilège" de la NP-complétude (Grandjean a prouvé que ces 21 problèmes appartiennent à NLIN). Nous précisons ce résultat en montrant que l'on conserve l'inclusion précédente en restreignant la logique $\exists SO_1(+)$ à sa sous-classe des formules dont le préfixe du premier ordre est de la forme $\forall^*\exists^*$. Ceci généralise un résultat similaire de Lynch pour la classe NTIME(n).

Ces résultats (sans le raffinement précité) ont été publiés dans les actes de la conférence CSL'94 ([38]). Une version complète est actuellement soumise au journal *Computational Complexity*. Le chapitre 1 en est la reproduction *in extenso*. Nous avons en particulier fait le choix de laisser figurer la section 1.2 (pp. 26 à 34) qui réunit diverses formulations de la caractérisation de NLIN par la logique $\mathcal{G}(<)$ évoquée plus haut. Cette section est entièrement due à Grandjean, mais nous nous y référerons à plusieurs reprises, et le souhait de présenter un mémoire qui se suffise autant que possible à lui-même nous a semblé justifier ce choix.

Dans le **chapitre 2**, nous établissons une nouvelle caractérisation exacte de NLIN. Nous avons dit plus haut que la caractérisation de cette classe prouvée par Grandjean identifie chaque problème de NLIN à l'ensemble des modèles d'une formule du type : $\exists f_1 \ldots \exists f_p \forall x \phi(x, f_1, \ldots, f_p, <)$, où les f_i sont des symboles de fonctions unaires, x est une variable du premier ordre et ϕ est une formule du premier ordre sans quantificateur. Nous avons noté :

$$\mathrm{NLIN} = \mathcal{G}(<)$$

cette caractérisation.

Nous montrerons que les matrices ϕ des formules de $\mathcal{G}(<)$ peuvent être ramenées à de simples conjonctions d'égalités, moyennant la substitution à l'ordre des symboles prédéfinis 0 et s (interprétés sur chaque domaine comme l'élément minimal et la fonction successeur). Notant $\mathcal{G}^+_\wedge(0, s)$ la classe de formules ainsi obtenues, nous établirons donc la *normalisation* suivante de la caractérisation précitée :

$$\mathrm{NLIN} = \mathcal{G}^+_\wedge(0, s).$$

Un argument de "rembourrage" permet de généraliser cette caractérisation purement conjonctive à la classe NP. Plus précisément, si l'on note $\mathcal{G}^+_\wedge[k, k'](0, s)$ la classe des formules de la forme $\exists \overline{G} \forall \overline{x} \phi$, où $\overline{G}$ est une suite de symboles de fonctions k-aires, $\overline{x}$ un k'-uplet de variables du premier ordre, et ϕ une formule du premier ordre purement conjonctive, on peut identifier chaque classe $\mathrm{NTIME}_{RAM}(n^k)$ des problèmes reconnaissables en temps non déterministe n^k $(k \in \mathbb{N}^*)$ à la logique $\mathcal{G}^+_\wedge[k, k](0, s)$. Autrement dit on prouve : sur les structures *successeurs* :

$$\mathrm{NTIME}_{RAM}(n^k) = \mathcal{G}^+_\wedge[k, k](0, s).$$

(Notons que la caractérisation de NLIN énoncée plus haut est la particularisation au cas $k = 1$ de cette dernière égalité.)

Nous présenterons ensuite un résultat de hiérarchie sur les classes $\mathcal{G}^+_\wedge[k, k'](0, s)$ déduit par Grandjean de la dernière caractérisation logique.

Finalement, nous établirons une forme de *réduction* de la logique $\mathcal{G}(0, s, <, +, \times, \exp)$ à la logique $\mathcal{G}^+_\wedge(0, s)$, en montrant la définissabilité de l'*addition*, la *multiplication* et l'*exponentielle* dans le cas où ces opérations interviennent dans des formules de $\mathcal{G}(0, s)$. Ceci permet cette fois une *extension* de la caractérisation de NLIN :

$$\mathrm{NLIN} = \mathcal{G}(0, s, +, \times, \exp).$$

Le **chapitre 3** concerne les ensembles rudimentaires, c'est-à-dire les parties de $\mathbb{N}$ définissables par une formule de l'arithmétique à quantification bornée. On sait qu'un entier peut être identifié à un mot de l'alphabet $\{1, \ldots, p\}$ par le biais de son écriture $p-$adique. La question se pose alors d'étudier la complexité descriptive des langages ainsi associés aux ensembles rudimentaires.

Nous montrons que ces langages sont exactement les langages définissables par une formule du second ordre monadique. On obtient comme corollaire une forme de représentation rudimentaire des spectres (un spectre est un ensemble d'entiers obtenus comme cardinaux des modèles d'une formule donnée). Plus précisément, nous prouvons qu'une partie S de $\mathbb{N}$ est le spectre d'une formule du second ordre d'arité k si et seulement si l'ensemble 2^{S^k} est rudimentaire.

Ces premiers résultats ont donné lieu à un article, écrit en collaboration avec Malika More et actuellement soumis au journal *Mathematical Logic Quarterly*.

Enfin, toujours à propos des spectres (dont le lien avec les ensembles rudimentaires constitue une question importante), nous utilisons un résultat d'Arnaud Durand sur les spectres de fonctions ([13]), pour prouver que les ensembles rudimentaires sont des spectres de formules du premier ordre dont la signature ne comporte qu'une bijection et une fonction, toutes deux unaires.

Ces trois chapitres peuvent être lus séparément, malgré quelques références croisées précisément spécifiées.

Chapitre 1

Monadic logical definability of NP-complete problems

Introduction

What is the relation between the computational complexity of a decision [1] problem and the logical complexity of the language required to describe it? This question was formulated first by Immerman [45] (see also [40, 41, 43, 72]).
In complexity theory any problem is identified with a language $\mathcal{L} \subseteq \Sigma^*$ (e.g. $\Sigma = \{0,1\}$). Solving the problem means deciding the associated language, *i.e.*, deciding for each word w whether it belongs to $\mathcal{L}$ or not.
Now, a word w on a finite alphabet is easily identified to a finite structure A_w. The most common encoding consists in identifying each word $w \in \{0,1\}^*$ with the finite structure $A_w = \langle n, X, succ\rangle$ defined by:

- $n = \{0, 1, \ldots, length(w) - 1\}$;
- $X \subset n$ and $\forall i \in n : X(i) \Leftrightarrow$ the i^{th} letter of w is 1 ;
- $succ$ is the successor relation on n.

Such an encoding being chosen, the purpose is the following: try to associate to every complexity class $\mathcal{C}$ a class of formulas of a given logical language, $\mathcal{F}$, in such a way that: A language $\mathcal{L} \subseteq \Sigma^*$ is in $\mathcal{C}$ iff there exists a sentence $\Phi \in \mathcal{F}$ such that:

$$\forall w \in \Sigma^* \ : \ w \in \mathcal{L} \Leftrightarrow A_w \models \Phi.$$

Among the results which were proved in that framework, let us mention those of Büchi and Fagin, often quoted in this way:

> *Regular Languages* = *monadic SO* = *monadic SO*($\exists$) ([4]) (see also [59] and [70]) and $NP = SO(\exists)$ ([20]),

1. This question is also studied for optimisation problems. See [58] or [49] for instance.

where, whenever $\mathcal{C}$ is a complexity class and $\mathcal{F}$ is a class of formulas, $\mathcal{C} = \mathcal{F}$ means that for every language $\mathcal{L}$ on Σ, $\mathcal{L}$ is in $\mathcal{C}$ iff there exists a formula $\Phi \in \mathcal{F}$ such that:

$$\forall w \in \Sigma^* \ : \ w \in \mathcal{L} \Leftrightarrow A_w \models \Phi.$$

We would expect this correspondence between computational complexity and logical definability to allow us: (i) to use the flexibility of complexity-theory tools (models of computation for instance) to get results on logical definability; (ii) to export complexity questions in the field of logical definability. An important example of the second item is the following: in order to prove that a given problem belongs to a given complexity class, it is enough to build an algorithm of that complexity which decides this problem. But of course, there is not guarantee that this algorithm is the "best" one. In other words, the difficulty is to obtain lower bounds of complexity.

In [53], Lynch focuses on this precise question. He writes: "there are hundreds of known NP-complete problems, but until recently, not one of them had a provable nontrivial lower bound." [2] This is the reason why Lynch wants to compare the class NTIME(n) (i.e. the class of languages recognized in nondeterministic linear time on multitape Turing machines) with some class of logical formulas $\mathcal{F}$. He hopes then to obtain results as "$\Pi \notin$ NTIME(n)", for some natural NP-complete problems Π, in proving their non-definability by formulas of $\mathcal{F}$. Actually, the "inclusion" NTIME$(n) \subseteq \mathcal{F}$ is enough to get such a result, because it still allows the implication "Π non-definable by $\mathcal{F} \Rightarrow \Pi \notin$ NTIME(n)".

The main result of [52, 53] may be stated as follow:

Theorem (Lynch). *Let $\mathcal{L} \subset \{0,1\}^*$ belong to NTIME(n). Then there exists a first-order formula Φ on a signature made up, in addition to the predicate symbols X, succ, plus of respective arity 1, 2, 3, of monadic predicate symbols $U_1, \ldots, U_s$, such that:*

1. *the quantifier prefix of Φ has the form $\forall^*\exists^*$;*
2. $\forall w \in \{0,1\}^n : \ w \in \mathcal{L} \Leftrightarrow \langle n, X, succ, plus\rangle \models \exists \overline{U} \Phi(X, \overline{U}, succ, plus)$,

where succ, plus et X are interpreted on $n = \{0, 1, \ldots, n-1\}$ by the predefined relations: $X(i)$ iff the i^{th} letter of w is 1; $succ(i,j)$ iff $j = i + 1$; $plus(i,j,k)$ iff $i + j = k$.

Concluding his paper ([53]), Lynch asks: "Can the (above) theorem be extended to random access models of computation, where the memory elements can be read and written in any order"? The question is justified by the specificity of his proof, which uses a discrete analogue of the Intermediate Value Theorem for the Turing machines, that is: if a tape head is at the position c at the time t and at c' at the time

2. There is a noteworthy exception with the problem "Reduction of Incompletely Specified Automata"(RISA), which has been proved non-solvable in deterministic linear time in [32].

$t' > t$, then for every position c'' between c and c', there is some time t'' between t and t' such that the head is at c'' at time t''. Of course, there is not equivalent result for RAM's.

The main purpose of this paper is to prove that Lynch's question has nevertheless a positive answer. More precisely, we show that the previous theorem remains true when replacing "NTIME(n)" by "NLIN", where NLIN is a complexity class (including NTIME(n)) elaborated by Grandjean ([34, 35]) to formalize nondeterministic linear time on RAM's. The definition of this class will be recalled in 1.1.1 and studied in section 1.2. But let us before go back over some steps of the linear time formalization story.

Although it is commonly mentioned by algorithm designers, the linear time complexity notion is especially hard to formalize, because of its great sensitivity to models of computing and to problems encodings used to describe it. This lack of robustness, apparently inherent in linear time, leads Gurevich and Shelah ([39]) and Grädel ([27]) to define two "robust closures" of linear time. Previously, Schnorr ([65]) had similarly defined "quasilinear time" (that is time $O(n(\log n)^{O(1)})$ for some Turing machine) and proved that many NP-complete problems belong to nondeterministic quasilinear time and are complete for this class. Those authors defined extensions of linear time because, as [39] explains: "It is possible that there is no universal notion of linear time and different versions of linear time are appropriate to different applications."

In [34, 35] Grandjean adopts the opposite point of view: he defines and justifies a unified, robust and powerful notion of linear time, both in deterministic and nondeterministic cases, with the classes DLIN and NLIN. Moreover, in [31, 32, 34], he gives a logical characterization of NLIN by second-order formulas written with $<$, 0 and unary function symbols. This characterization will be our main tool to prove that monadic second-order logic captures NLIN. Such a result appreciably extends the Lynch's one, since NTIME(n) $\subseteq$ NLIN, and since all the twenty one Karp's NP-complete problems ([48]) have been proved as belonging to NLIN, when only two of them (`Knapsack` and `Partition`) are known as belonging to NTIME(n).

A preliminary and abridged form of this paper appeared as a conference proceeding in [38]: neither the NLIN logical characterization with functions was proved in it, nor the precise proof that the first-order prefix of the existential monadic second-order NLIN characterization was $\forall^*\exists^*$ (main result) was given. The present paper is self contained and exhibits complete proofs of the stated results. Moreover, besides the main result which gives a logical characterization of the NLIN class of languages (*i.e.* sets of words), we also prove here new precise results (Theorems 1.4 and 1.5) which characterize the NLIN-class of first-order structure problems (e.g. graph problems) when the input is encoded as a unary structure.

This encoding and hence the associated logical characterization, may seem more natural than the similar one for NLIN languages, since it is exact (a structure problem is in NLIN *if and only if* it is definable in our logic) and since the encoding is simpler and respects the main complexity parameters of the input structure, e.g. a graph is encoded by a new unary structure of cardinality $|V| + |E|$. (Note that this is similar to the encoding of a graph in [8]).

1.1 NLIN characterization and sketch of proof of the main result

1.1.1 NLIN definition

Let $\Sigma = \{1, 2, \ldots, d\}$ be an alphabet of cardinality $d > 1$ and $\mathcal{L} \subseteq \Sigma^*$. We write $\mathcal{L} \in$ NLIN if there exists an integer $k \neq 0$ and an NRAM (i.e. non deterministic RAM) $\mathcal{R}$ such that:

1. $\mathcal{R}$ only uses the arithmetic operations $succ: \; x \mapsto x + 1$ and $pred: \; x \mapsto x - 1$.
2. $\mathcal{R}$ reads its input w by blocks of length $l = \lceil \frac{1}{k} \log(n+1) \rceil$ [3], where $n = \text{length}(w)$ ($w \in \Sigma^n$).
3. $\mathcal{R}$ recognizes $\mathcal{L}$ in time $O(n/\log n)$ (where the time is the number of executed instructions, and n is the length of the input word), or, equivalently, in time $O(m)$, where m is the number of blocks of the input.

(As a consequence of 1, 2, 3, $\mathcal{R}$ only uses words, *i.e.* addresses and contents of registers, of length $O(\log n)$.)

Remark. k only depends on Σ cardinality. For $|\Sigma|$=2, for instance, each $k > 1$ suits.

As a motivation, let us give an intuitive and rough idea why problem SAT is in NLIN (but seemingly not in NTIME(n)). On a SAT instance ϕ of length n (in a fixed alphabet Σ), an NRAM guesses truth values for its m variables identified to the integers 1, $2, \ldots,$ m, each of size $O(\log m) = O(\log n)$, and then evaluates each clause of ϕ by direct substitutions. The number of executed intructions of the whole computation is $\Theta(\textit{number of occurences of variables in } \phi) = \Theta(n/\log n)$ and each one manipulates only words of length $O(\log n)$.

Remark. NLIN is a robust class, e.g. it is equal to the class, denoted NLINEAR, of problems recognizable in time $O(n/\log n)$ (*i.e.* using $O(n/\log n)$ instructions) on NRAM's using addition, substraction and words (i.e. addresses and contents of registers) of length $O(\log n)$ (see [33, 34, 35]).

1.1.2 Reading decomposition of a word w $\in \Sigma^*$

In what follows, we assume that Σ is the dyadic alphabet ($\Sigma = \{1, 2\}$) and we denote by k an integer greater than or equal to 2. In this section, we describe a decomposition of the words of Σ^* suited to the NRAM definition given above.This

3. In this paper, $\log n$ denotes the logarithm of n in basis 2.

decomposition, called *reading decomposition*, will allow to encode every word $w \in \Sigma^*$ as a finite structure of the $\langle m, f, <, 0\rangle$ type, where f denotes a unary fonction on $m = \{0, 1, \ldots, m-1\}$ and $<$ is the usual order on the domain m. This encoding is used in an essential manner in the logical characterization of NLIN given by [34, 35].

Let $w \in \Sigma^*$ and let us denote $n = length(w)$, $l = \lceil \frac{\log(n+1)}{k} \rceil$, $m = \lceil \frac{n}{l} \rceil$. Then we can break down w into m words $w_0, w_1, \ldots, w_{m-1}$, all of which have length l exactly, except the last one, which may be of length less than or equal to l. We have $w = w_0^\frown w_1^\frown \ldots^\frown w_{m-1}$, and if w_i^d denotes the integer whose dyadic writing is w_i, it is easy to show that

- $m = \Theta(n/\log n)$ and
- $w_i^d < m$ for large enough n.

We can now identify the word $w \in \{1,2\}^*$, with reading decomposition $w = w_0^\frown w_1^\frown \ldots^\frown w_{m-1}$, with the finite structure $\langle m, f_w, <, 0\rangle$ defined by:

- $m = \{0, 1, \ldots, m-1\}$;
- $f_w : \left\{ \begin{array}{l} m \to m \\ i \mapsto f_w(i) = w_i^d \end{array} \right.$
- $<$ is the natural linear order on m and 0 is the "real" 0 of m.[4]

1.1.3 NLIN characterization and main result

Notations In the following, we denote by $FO(\forall^i\exists^j)$ the class of first-order formulas in prenex conjunctive normal form, the quantifier prefix of which has the pattern $\forall^i\exists^j$. We omit the exponent i or j when it equals 1. $FO(\forall^*\exists^*)$ is $\bigcup_{i,j} FO(\forall^i\exists^j)$. $FO(wq)$ is the class of first-order formulas without quantifier. These notations naturally extend to define any class of formulas logically equivalent to formulas of the concerned class.

With the same notations as in 1.2 , the NLIN characterization, essentially proved in [31, 32, 34], may be written:

Theorem 1.1. *A language $\mathcal{L} \subseteq \{1,2\}^*$ is in NLIN iff there exists a formula $\Psi \in FO(wq)$with only one first-order variable on signature $\{f, g_1, \ldots, g_p, <, 0\}$, where f and the $g_i^,$s denote unary functions symbols, $<$ is a binary predicate symbol and 0 is a constant symbol, such that for every word $w \in \{1,2\}^*$:*

$$(w \in \mathcal{L}) \text{ iff } (\langle m, f_w, <, 0\rangle \models \exists g_1 \ldots \exists g_p \forall x\, \Psi(x, f, \overline{g}, <, 0)) \tag{1.1}$$

where $\langle m, f_w, <, 0\rangle$ is the finite structure associated to w as above.

4. Compare this with the usual identification of a word $w \in \{0,1\}^n$ with the finite structure $\langle n, X, <\rangle$, where X is the unary relation on n such that $X(i)$ holds iff the i^{th} letter of w is 1 (see [52] or [70]).

This exact characterization of NLIN will allow to prove our main result, that is:

Theorem A. *If a language $\mathcal{L} \subseteq \{1,2\}^*$ is in NLIN, then there exists a sentence $\Theta \in FO(\forall^*\exists^*)$ on signature $\{X, U_1, \ldots, U_s, +\}$, where X and the U_i's are unary predicate symbols, $+$ is a ternary predicate symbol, such that for every word $w \in \{1,2\}^*$:*

$$[w \in \mathcal{L}] \Leftrightarrow [\langle n, X, + \rangle \models \exists U_1 \ldots \exists U_s \, \Theta(X, \overline{U}, +)] \tag{1.2}$$

where $+$ is the predefined addition on n and X is the predicate naturally coding w on n, i.e.defined by: $\forall i < n : X(i) \Leftrightarrow$ the letter of index i of w is 2

1.1.4 A simplified view of the proof of Theorem A

• We know that each positive integer x may be coded by a word of $\{0,1\}^*$ of length $\lfloor \log(x) + 1 \rfloor$ (binary representation of integers). In particular, if E is a finite set of integers, we can identify every $x \in E$ with a word $B(x) \in \{0,1\}^*$ of length $L = Max\{\lfloor \log(x)+1 \rfloor, x \in E\}$ exactly, does mean padding with 0's binary representations of the elements of E. Equality between two integers of E is equivalent to equality of their representative words. Namely, if for $x \in E$ we denote $B(x)_t$ the bit of $B(x)$ of rank t ($0 \leq t < L$ with convention: the bit of rank 0 is the less significant bit), we have:

$$\forall x, y \in E : [x = y] \text{ iff } [(\forall t < L)(B(x)_t = B(y)_t)]$$

• This encoding of integers by fixed length words can be extended to functions in the following manner: since every $x < m$ is encoded by $B(x) \in \{0,1\}^L$, where $L = \lfloor \log(m) + 1 \rfloor$, a function $f : m \to m$ may be identified with the word:

$$B^f = B(f(0))^\frown B(f(1))^\frown \ldots ^\frown B(f(m-1)) \in \{0,1\}^{mL}$$

If the t^{th} bit of B^f is denoted $(B^f)_t$, we have: $\forall f, g : m \to m$, $\forall x, y < m$:

$$f(x) = g(y) \text{ iff } (\forall t < L)(\, (B(f(x))_t = B(g(y))_t \,)$$

hence

$$f(x) = g(y) \text{ iff } (\forall t < L)(\, (B^f)_{xL+t} = (B^g)_{yL+t} \,) \quad (*)$$

Eventually, we know that a word of length mL is naturally associated with the unary predicate $X \subseteq mL$ defined by:

$$\forall t < mL : X(t) \Leftrightarrow (w)_t = 1$$

If for $h : m \to m$, the corresponding capital H is the unary predicate of mL associated in this way with the word $B^h \in \{0,1\}^{mL}$, the equivalence $(*)$ becomes:

$$\forall x, y \in m : [f(x) = g(y)] \Leftrightarrow [(\forall t < L)(F(xL + t) \leftrightarrow G(yL + t))]$$

Thus, we convert a unary functional formula into a monadic relational formula. This result may be extended to a large family of formulas on a signature made up, in addition to unary functions symbols, of the predefined symbols $<$ and 0 (natural linear order on integers and 0). This "translation", from unary functional language to the less expressive monadic relational language, has a cost: the domain of finite structures considered must be expanded (we pass from unary functions on domain $m = \Theta(n/\log n)$ to unary relations on domain mL, which is $\Theta(m \log m) = \Theta(n)$), and the predefined symbols $<$, 0 must be replaced by the much more expressive predicate $+$.

The study of this flattening of functions in unary predicates constitutes the core of this paper. Combined with the above-mentioned characterization of a language $\mathcal{L} \in$ NLIN (see (1.1)), it allows to prove that for every $\mathcal{L} \in$ NLIN, there exists $\Theta \in SO(\exists, mon, +)$ (i.e. Θ is a monadic existential second-order formula with addition), such that:

$$(w \in \mathcal{L}) \Leftrightarrow (\langle n, X, + \rangle \models \Theta),$$

where $n = length(w)$ and X is the unary predicate on n coding w.
More precisely, the paper is organized as follows:

In **section 2** we prove the logical characterization of NLIN (Theorem 1.1). We give this proof although it is essentially presented in [31, 32, 34] because we want to have a self contained paper, and because the statements and the proofs are scattered in these papers and *are not exactly* those of Theorem 1.1: NLIN definition was introduced only recently in references [34, 35]. Moreover, we give in this section three very natural versions of the same result concerning functions (Theorem 1.3) and unary structures (Theorems 1.4 and 1.5) as inputs instead of words.

In **section 3** we first specify the above-mentioned identification between a unary function $f : m \to m$ and a unary predicate $F \subseteq mL$ ($L = \lfloor \log(m) + 1 \rfloor$). Actually, for technical reasons, the predicate F is built on a domain cn (for a fixed constant c) slightly bigger than mL. We will call F "flattening of f on cn". Then, we state precisely the passage from functional language to relational language: For each formula $\Psi \in FO(\forall)$ on signature $\{f, g_1, \ldots, g_p, <, 0\}$, where the g_i's are unary function symbols, there exists $\tilde{\Psi} \in FO(\forall^*\exists^*)$, on signature $\{F, G_1, \ldots, G_q, +\}$, where F and the G_i's are monadic predicate symbols, such that: if $f : m \to m$ and if F is the flattening of f on cn:

$$\langle m, f, <, 0 \rangle \models \exists \overline{g}\, \Psi \;\Leftrightarrow\; \langle cn, F, + \rangle \models \exists \overline{G}\, \tilde{\Psi}$$

then we combine this result with the NLIN characterization of Theorem 1.1 to prove: for every $\mathcal{L} \subseteq \{1,2\}^*$ in NLIN, there exists a second-order formula $\hat{\Psi}$ such that for every $w \in \{1,2\}^n$:

$$(w \in \mathcal{L}) \Leftrightarrow (\langle cn, F, + \rangle \models \exists \overline{G} \hat{\Psi}(F, \overline{G}, +)) \tag{1.3}$$

where F is a unary predicate which codes w on cn.

Actually, the formula $\hat{\Psi}$ above mentioned is written with a set of predefined arithmetical predicates including $\{+\}$. To obtain the equivalence 1.3, we have to prove that all these predicate are definable from $+$. This is done in **section 4**, taking a specific care to the first-order part of the involved formulas, to guarantee that $\hat{\Psi}$ does belong to the Lynch's logic (*i.e.* has a $\forall^*\exists^*$ first-order prefix).

In **section 5**, we prove that the equivalence 1.3 still holds when we replace the predicate F by a predicate $\mathcal{X}$ encoding the word w in a more natural way.

Section 6 focuses on a technical result which makes it possible to provide the preceding result with a more standard form. We first prove that if $R_1, \ldots, R_p$ are predicates with respective arities $r_1, \ldots, r_p$, on a domain of the form cn, and if Φ is a first-order formula on signature $\{R_1, \ldots, R_p\}$, then there exists a division of each R_i into a series R_i^* of $r_i - ary$ predicates on n, and a first-order formula Φ^* on signature $\{R_1^*, \ldots, R_p^*\}$, such that:

$$(\langle cn, R_1, \ldots, R_p\rangle \models \Phi) \;\Leftrightarrow\; (\langle n, R_1^*, \ldots, R_p^*\rangle \models \Phi^*).$$

Moreover, the first-order prefix of Φ^* has the same form than the Φ's one, that is $\forall^*\exists^*$.

Associated to the result of section 5, this equivalence allows us to conclude: if $\mathcal{L} \subseteq \{1,2\}^*$ belongs to NLIN, then there exists $\Theta \in SO(\exists, mon, +)$ with first-order prefix $\forall^*\exists^*$ such that for every $w \in \{1,2\}^*$:

$$(w \in \mathcal{L}) \;\Leftrightarrow\; (\langle n, X, +\rangle \models \Theta) \;,$$

where X is the unary predicate naturally coding w.

1.2 NLIN characterization with unary functions and only one variable

1.2.1 NLIN definition (completed)

A language $\mathcal{L} \subseteq \{1,2\}^*$ is in $NLIN$ if there exists an NRAM $\mathcal{R}$ such that:

- $\mathcal{R}$ reads its input w (of length n) by blocks of length $l = \lceil \frac{1}{k}\log(n+1)\rceil$, i.e. at the beginning of its computation, its registers $R_0, R_1, \ldots, R_{m-1}$ respectively contain the subwords $w_0, w_1, \ldots, w_{m-1}$, all of which have length l exactly, except the last one, w_{m-1}, which may be of length less than or equal to l, and $w = w_0 \frown w_1 \frown \ldots \frown w_{m-1}$ (concatenation) (Notice: $m = \lceil n/l\rceil = \Theta(n/\log n)$).
- $\mathcal{R}$ recognizes $\mathcal{L}$ in time $O(n/\log n)$ or, equivalently, in time $O(m)$, where m is the number of blocks of the input.

Remark. More formally, at the beginning of the computation, each register R_j of address $j \geq m$ contains integer 0, and each register R_i of address $i < m$ contains the integer w_i^d whose dyadic writing is w_i.

Notice that it was proved in [34, 35] that the NLIN class does not change for many changes of the input representation and of the allowed instructions.

1.2.2 Proof of Theorem 1.1

• We let as an exercice to the reader to prove that if there exists a formula $\Psi \in FO(wq)$ such that equivalence (1.1) holds for all $w \in \{1,2\}^*$ then $\mathcal{L} \in$ NLIN.
• Let us prove the converse implication. Let $\mathcal{L} \subseteq \{1,2\}^*$ belong to NLIN. That means that $\mathcal{L}$ is recognized by an NRAM $\mathcal{R}$ in time cm, where c is a constant integer and m is defined as usual ($m = \lceil n/l \rceil$, where $l = \lceil \frac{1}{k} \log(n+1) \rceil$). Let us assume, without loss of generality, that $\mathcal{R}$ has two extra register a and b, called "accumulators" (whose initial content is 0) and that its program is a sequence of instructions $\mathcal{I}(1), \mathcal{I}(2), \ldots, \mathcal{I}(r)$ of the following types:

(1)	$a := 0$	(7)	$R_a := b$
(2)	$a := a + 1$	(8)	*goto* $\mathcal{I}(i_0)$ *or* $\mathcal{I}(i_1)$
(3)	$a := a \dot{-} 1$	(9)	*If* $a = b$ *then goto* $\mathcal{I}(i_0)$ *else goto* $\mathcal{I}(i_1)$
(4)	*guess* (a)	(10)	*accept*
(5)	$a := R_a$	(11)	*reject*
(6)	$b := a$		

Comments.

- $i \dot{-} j =_{def} max\{i - j, 0\}$.
- R_a denotes the register pointed by accumulator a (i.e. the address of R_a is the content of a).
- In fact, only a is used for accessing the sequence of registers $R_0, R_1, \ldots$ (like the control head of a Turing machine).
- (4) and (8) are nondeterministic instructions: *guess* (a) consists in storing in a any integer less than or equal to cm.
- We can assume, without loss of generality, that any integer manipulated during the computation is less than or equal to cm.

Let $F_w : cm \to cm$ be the extension of f_w defined by $F_w(i) = f_w(i)$ if $i < m$ and $F_w(i) = 0$ otherwise. Let M denote the integer cm. Notice that the following formula:

$$\phi_0 \equiv \forall x \exists y\, (succ(y) = x) \wedge (\forall x \neq\ last)(x < succ(x)) \wedge (succ(last) = 0)$$

defines the successor function $succ :\ M \to M$ such that $succ(i) = i + 1$ modulo M and the constant $last = M - 1$ on structure $\langle M, <, 0\rangle$. That means that the structure $\langle M, <, 0, succ, last\rangle$ satisfies ϕ_0 iff the function $succ$ and the constant $last$ have their right values over M.

Note that ϕ_0 is equivalent to a prenex formula with only one universal variable.

It is convenient to describe a computation of the NRAM $\mathcal{R}$ (of time cm on input w) in a first-order structure of domain $M = cm$ (notice: M is the time of the computation and is greater than any integer it involves) as we do in the following proposition similar to Theorem 1.1.

Proposition 1.2. *Let $\mathcal{L} \subseteq \{1,2\}^*$ be a language recognized by an NRAM $\mathcal{R}$ in time $M = cm$, where m is defined as above. Then there exists a formula $\Psi \in FO(wq)$ with only one first-order variable on signature $\{F, G_1, \ldots, G_p, <, 0, succ, last\}$, where F, the G_i's and $succ$ denote unary function symbols, $<$ is a binary predicate symbol and 0, $last$ are constant symbols, so that for each $w \in \{1,2\}^*$:*

$$w \in \mathcal{L} \text{ iff } \langle M, F_w, <, 0, succ, last\rangle \models \exists G_1 \ldots \exists G_p \forall x \Psi(x, F, \overline{G}, <, 0, succ, last)$$

where $\langle M, F_w, <, 0, succ, last\rangle$ is the finite structure associated to w as above.

Proof of the proposition. Let $\mathcal{I}(1)$, $\mathcal{I}(2)$, ..., $\mathcal{I}(r)$ be the program of $\mathcal{R}$ (i.e. a list of instructions of type (1) – (11)). We assume that the only instruction *accept* of $\mathcal{R}$ is $\mathcal{I}(r)$. Let I, A, B, R_A, R'_A be unary functions symbols whose intended meaning is the following:

- $I(t)$ is the index of the instruction performed at the instant t;
- $A(t)$ is the content of accumulator a at the instant t;
- $B(t)$ is the content of accumulator b at the instant t;
- $R_A(t)$ is the content of R_a at the instant t (i.e. the content of $R_{A(t)}$);
- $R'_A(t)$ is the integer stored in R_A between instants t and $t+1$ according to instruction $\mathcal{I}(I(t))$, i.e. the content of register $R_{A(t)}$ after the execution of $\mathcal{I}(I(t))$.
- Moreover, we easily see that $F_w(i)$ is the content of register R_i at instant 0 for each $i < M$.

Let $\mathcal{S}_c$ be the signature $\{F, I, A, B, R_A, R'_A, <, 0, succ, last\}$. We now give some first-order $\mathcal{S}_c$-formulas in order to describe:

- the initial configuration of $\mathcal{R}$ on input w;
- a computation step from instant t to instant $t+1$ for each $t < M-1$;
- the fact that $\mathcal{R}$ reaches the instruction *accept*.

We also use constants $1, 2, 3, \ldots$ which abbreviate $succ(0)$, $succ(succ(0))$, ...

• ϕ_1 describes the initial conditions:

$$\phi_1 \equiv I(0) = 1 \wedge A(0) = 0 \wedge B(0) = 0 \wedge R_A(0) = F(0)$$

• ϕ_2 describes a computation between any two successive instants, using for each instruction $\mathcal{I}(i)$ a formula $\Psi_i(t)$ corresponding to the type of $\mathcal{I}(i)$.

$$\phi_2 \equiv (\forall t \neq last) \bigwedge_{i=1}^{r} [I(t) = i \rightarrow \Psi_i(t)]$$

For example, if $\mathcal{I}(i)$ is the instruction "$a := a + 1$", then $\Psi_i(t)$ is the formula:

$$A(t+1) = A(t) + 1 \wedge I(t+1) = I(t) + 1 \wedge B(t+1) = B(t) \wedge R'_A(t) = R_A(t)$$

Figure 1 gives for each of the 11 types of instructions $\mathcal{I}(i)$ the corresponding formula $\Psi_i(t)$. For purpose of readability, we use some abbreviated formulas called $(same\ \ldots)(t)$, which indicate the registers among a, b, R_a that do not change their content. For instance, $(same\ b, R_a)(t)$ abbreviates the conjunction: $B(t+1) = B(t) \wedge R'_A(t) = R_A(t)$.

$\mathcal{I}(i)$	$\Psi_i(t)$
$a := a + 1$ (similarly for $a := a \dot{-} 1$ and $a := 0$)	$A(t+1) = A(t) + 1$ $\wedge\ (same\ b, R_a)(t)$ $\wedge\ I(t+1) = I(t) + 1$
guess(a)	$\exists y(A(t+1) = y)$ $\wedge\ (same\ b, R_a)(t)$ $\wedge\ I(t+1) = I(t) + 1$
$a := R_a$	$A(t+1) = R_A(t)$ $\wedge\ (same\ b, R_a)(t)$ $\wedge\ I(t+1) = I(t) + 1$
$b := a$	$B(t+1) = A(t)$ $\wedge\ (same\ a, R_a)(t)$ $\wedge\ I(t+1) = I(t) + 1$
$R_a := b$	$R'_A(t) = B(t)$ $\wedge\ (same\ a, b)(t)$ $\wedge\ I(t+1) = I(t) + 1$
goto $\mathcal{I}(i_0)$ or $\mathcal{I}(i_1)$	$[I(t+1) = i_0 \vee I(t+1) = i_1]$ $\wedge$ $(same\ a, b, R_a)(t)$
If $a = b$ then goto $\mathcal{I}(i_0)$ else goto $\mathcal{I}(i_1)$	$(same\ a, b, R_a)(t)$ $\wedge\ (A(t) = B(t) \rightarrow I(t+1) = i_0)$ $\wedge\ (A(t) \neq B(t) \rightarrow I(t+1) = i_1)$
accept (or reject)	$I(t+1) = I(t)$

figure 1

The formula ϕ_2 above is not sufficient to compell the function $R_A(t)$ to be equal to $R_{A(t)}$ for each $t < M$. We have two cases:

First case: let t' be the last instant before t when $A(t') = A(t)$. Then the content of register $R_{A(t)}$ at instant t is the integer stored in R_a between instants t' and $t'+1$, that is $R'_A(t')$.

Second case: otherwise, if the register pointed by a at instant t, *i.e.*register $R_{A(t)}$, has never been examined before instant t, then its content is the initial content. That is: $R_{A(t)} = F_w(A(t))$.

In order to express the two above conditions with a formula in $FO(\forall\exists^*)$ (which can obviously be transformed into a $FO(\forall)$-formula by Skolemization), we now introduce new unary function symbols.

Let $\mathcal{R}_A = \{(A(t), t) \mid t < M\}$, and let $\prec$ denote the lexicographical order of $\mathcal{R}_A$. *Figure 2* shows how pairs $(A(t), t)$ represented by cells are lexicographically numbered.

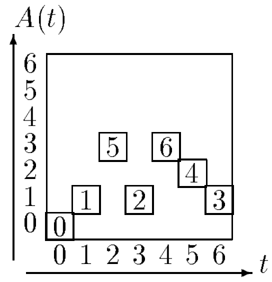

figure 2

We easily see how to rephrase the two above conditions: Let $t < M$ and $t' < M$ be such that $(A(t'), t')$ is the predecessor of $(A(t), t)$ for $\prec$. Then:

First case: if $A(t') = A(t)$ then $R_A(t) = R'_A(t')$.

Second case: if $A(t') \neq A(t)$ then $R_A(t) = F(A(t))$.

(Remark: the particular case $t = 0$ is similarly considered in formula ϕ_1.)

Let $N : M \to M \times M$ denote the lexicographical numbering of pairs $(A(t), t)$. In other words, N is the only strictly increasing function from $(M, <)$ to $(\mathcal{R}_A, \prec)$.

- N is defined by the following conjunction ϕ_3 (to be interpreted on domain M):

$$\phi_3 \equiv \forall x \exists t [N(x) = (A(t), t)] \wedge (\forall x \neq last)[N(x) \prec N(x+1)]$$

More formally, function N should be represented by its two projections $N_1, N_2 : M \to M$, and ϕ_3 should be consequently rephrased.

- Using N (i.e. N_1, N_2), the two above conditions are rewritten as follows:

$$\begin{aligned}\phi_4 \equiv (\forall t \neq 0)\, \exists t'\, \exists x \ : \ & \{N(x) = (A(t'), t') \wedge N(x+1) = (A(t), t)\} \wedge \\ & \{A(t') = A(t) \rightarrow R_A(t) = R'_A(t')\} \wedge \\ & \{A(t') \neq A(t) \rightarrow R_A(t) = F(A(t))\}\end{aligned}$$

- The fact that $\mathcal{R}$ reaches the instruction *accept* is expressed by:

$$\phi_5 \equiv \exists t\, (I(t) = r)$$

Clearly $\mathcal{R}$ accepts w (in time $M = cm$) if and only if $\langle M, F_w, <, 0, succ, last\rangle \models \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5$. Moreover, $\bigwedge_1^5 \phi_i$ can be put in form $FO(\forall\exists^*)$, and hence in form $FO(\forall)$, by Skolemization. This concludes the proof of the Proposition.

1.2.3 End of the proof of Theorem 1.1

Using the result of the Proposition 1.2, we have to encode a first-order structure $\langle cm, F_w, <, 0, G_1, \ldots, G_p\rangle$ into a structure $\langle m, f_w, <, 0, \ldots\rangle$ on domain m.

The idea is to replace each function $G : cm \rightarrow cm$ by $2c$ functions:

- c functions $G_i^0 : m \rightarrow m$ $(i < c)$
- c functions $G_i^1 : m \rightarrow c$ $(i < c)$

with $G_i^0(y) = G(im + y)$ *modulo* m and $G_i^1(y) = \lfloor G(im+y)/m \rfloor$ (euclidean division) for every $i < c$, $y < m$.

In particular, for $G = F_w$ we have $G_0^0 = f_w$ and G_i^j is zero for each $(i, j) \neq (0, 0)$.

The transformation of the formulas is similar to the one we will describe in Section 1.6 for predicate symbols instead of function symbols, so we let the details to the reader.

This concludes the proof of Theorem 1.1.

1.2.4 Complements: logical characterization of the NLIN class of unary structures and graphs

In fact, our encoding of an input $w \in \{1, 2\}^*$ by a function $f_w : m \rightarrow m - \{0\}$ may appear somewhat artificial. The natural framework seems to be functions rather than words. This gives the following variant of Theorem 1.1:

Theorem 1.3. *Let $\mathcal{F}$ be a set of functions $f : m \rightarrow m - \{0\}$ (where $m > 1$ is an integer depending on f). $\mathcal{F}$ is recognized by an NRAM $\mathcal{R}$ in time $O(m)$ ($\mathcal{R}$ is an NRAM as above, with $f(0), f(1), \ldots, f(m-1)$ in respective registers $R_0, R_1, \ldots, R_{m-1}$ at the beginning of the computation) if and only if there exists a formula $\Psi \in FO(wq)$*

with only one variable, on signature $\{f, g_1, \ldots, g_q, <, 0\}$ *as above, such that for each* $m > 1$ *and each* $f : m \to m - \{0\}$*:*

$$f \in \mathcal{F} \;\Leftrightarrow\; \langle m, f, <, 0\rangle \models \exists g_1 \ldots \exists g_q \forall x \Psi(x, f, \overline{g}, <, 0)$$

Proof. See proof of Theorem 1.1.

In fact, we can generalize Theorem 1.3 to unary structures.

DEFINITION 1.1. A signature of the form $S = \{f_1, \ldots, f_p, c_1, \ldots, c_r\}$ where the f_i's are unary functions and the c_j's are constant symbols, is called a *unary signature.*

A *unary structure* is a first-order structure of domain $m > 0$ and unary signature.

Remark. When studying the precise (computational or logical) complexity of problems, it is often natural to encode the input as a unary structure in a sparse way. For example, a directed graph $G = (V, E)$ where $V = n = \{0, \ldots, n-1\}$ and $|E| = e$ is naturally encoded by the unary structure $\langle n+e, f_1, f_2\rangle$ where $E = \{n, \ldots, n+e-1\}$ ($i.e. V \cup E = e + n$) and functions $f_1, f_2 : V \cup E \to V \cup E$ are defined by:

$f_1(x) = f_2(x) = x$ for each $x \in V$;

for each edge α, $f_1(\alpha) = x$ (resp. $f_2(\alpha) = y$) if the first endpoint (resp. the second endpoint) of edge α is x (resp. y).

More generally, any first-order structure can be similarly encoded as a unary structure.

DEFINITION 1.2. Let $\mathcal{F}$ be a set of unary S-structures as above. The set $\mathcal{F}$ is in NLIN if it is recognized by an NRAM $\mathcal{R}$ in time $O(m)$ with the following initial conditions: at the beginning of the computation, registers $R_0, R_1, \ldots, R_r$ respectively contain $m-1$ and constant values $c_1, \ldots, c_r$. Similarly, the initial content of R_{im+x} is $f_i(x)$ for each $i \leq p$ and $x < m$.

Remark. The above initial conditions (*i.e.* the way the input structure is stored) make sense only if $m > r$; we chose them to obtain an easy encoding in our proof of Theorem 1.4 that follows. In fact, our NLIN class (of sets of unary structures) is *rather insensitive to the presentation of the input*, as the similar NLIN class of languages.

Remark. Natural NP graph problems, e.g. `Hamilton circuit`, are in NLIN when they are encoded with unary structures (see Remark 1.2.4), *i.e.* are recognized on RAM's within time $O(n + e)$ where n is the number of vertices and e is the number of edges.

Theorem 1.4. *Let* $S = \{f_1, \ldots, f_p, c_1, \ldots, c_r\}$ $(p \geq 1)$ *be a unary signature. Let* $\mathcal{F}$ *be a set of* S*-structures.*

$\mathcal{F}$ *is in NLIN iff there are a unary signature* $S' = \{g_1, \ldots, g_p, c'_1, \ldots, c'_s\}$ *and a first-order formula without quantifier* Ψ *with only one variable, on signature* $S \cup S' \cup \{<\}$*, such that for each* S*-structure* $\mathcal{A} = \langle m, f_1, \ldots, f_p, c_1, \ldots, c_r\rangle$*:*

$$\mathcal{A} \in \mathcal{F} \textit{ iff } \langle \mathcal{A}, <\rangle \models \exists \overline{g}\, \exists \overline{c'}\, \forall x\, \Psi(x, \overline{f}, \overline{g}, \overline{c}, \overline{c'}, <).$$

Proof. The proof is similar to the proof of Theorem 1.1 (and Theorem 1.3). In place of function $F_w : cm \to cm$, we use a function $F_{\mathcal{A}} : cm \to cm$ that still represent precisely the way the input $\mathcal{A}$ is initially stored by the NRAM:

$F_{\mathcal{A}}(0) = m - 1$; $F_{\mathcal{A}}(i) = c_i$ for $i = 1, \ldots, r$;

$F_{\mathcal{A}}(im + x) = f_i(x)$ for each $i = 1, \ldots, p$ and $x < m$.

The encoding of the first-order structure $\langle cm, F_{\mathcal{A}}, <, 0, \ldots\rangle$ into a structure $\langle \mathcal{A}, <, \ldots\rangle$ is very similar to the one described in subsection 1.2.3. So we leave the details to the reader.

The natural graph problems (and, more generally, structure problems) are closed under isomorphism.

Definition 1.3. A set $\mathcal{F}$ of (unary) S-structures is *closed under isomorphism* if for any S-structure $\mathcal{A} = \langle m, (f_i)_{i\leq p}, (c_j)_{j\leq r}\rangle$ and any permutation $\sigma : m \to m$,

$$\mathcal{A} \in \mathcal{F} \text{ iff } \mathcal{A}^\sigma \in \mathcal{F},$$

where $\mathcal{A}^\sigma = \langle m, (f_i^\sigma)_{i\leq p}, (c_j^\sigma)_{j\leq r}\rangle$ and $f_i^\sigma(\sigma(x)) = \sigma(f_i(x))$ for all $i \leq p$ and $x < m$, and $c_j^\sigma = \sigma(c_j)$ for each $j \leq r$.

Theorem 1.5. *Let $\mathcal{F}$ be a set of unary S-structures closed under isomorhism. With the hypothesis of Theorem 1.4, we can et rid of the linear order, i.e. obtain a formula $\Psi \in FO(wq)$ with one variable, on unary signature $S \cup S'$, such that:*

$$\mathcal{A} \in \mathcal{F} \textit{ iff } \mathcal{A} \models \exists\overline{g}\,\exists\overline{c'}\,\forall x\,\Psi(x, \overline{f}, \overline{g}, \overline{c}, \overline{c'}).$$

Proof. It immediately follows from Theorem 1.4 and the fact, proved in [Gr8], that we can get rid of linear order by using new unary functions and formulas with only one first-order variable.

Remark. We easily see that we need only one constant symbol in $S \cup S'$, that is $r + s \geq 1$. More precisely, if $r \geq 1$ (*i.e.*the input structure contains some constants), then we can choose $s = 0$ (no quantification over constant symbols and we can take $s = 1$ in case $r = 0$.

Theorem 1.5 can be applied to the natural encoding (see Remark 1.2.4) of concrete NP-complete problems, since they belong to NLIN.

Example 1.4. Let `Hamilton path` (`HP`) denote the problem of the existence of an hamilton path between to fixed vertices c_1, c_2 in a (directed) graph $G = (V, E)$. Let $\mathcal{A}(G, c_1, c_2)$ denote the unary structure $\langle V \cup E, f_1, f_2, c_1, c_2\rangle$ defined as in Remark 1.2.4. There is a formula $\Psi \in FO(wq)$ with one variable, on unary signature $\{f_1, f_2, \overline{g}, c_1, c_2\}$, such that:

$$(G, c_1, c_2) \in \texttt{HP} \text{ iff } \mathcal{A}(G, c_1, c_2) \models \Psi(x, f_1, f_2, \overline{g}, c_1, c_2).$$

We let the reader contruct explicitely such a formula.

1.3 From unary functional language to monadic relational language

1.3.1 Notations

- $\mathbb{N}$ is the set of positive integers.
- $\log n$ denotes the logarithm of n in basis 2.
- For $q \in \mathbb{N}$, $B(q)$ denotes the reversed binary representation of q (e.g. $B(6) = 011$).
- For $i \in \mathbb{N}$, 0^i is the concatenation of i 0-symbols.
- If w is a word, $lg(w)$ is the length of w.
- Let q, L be positive integers such that $L \geq \lfloor \log(q) + 1 \rfloor = lg(B(q))$. We note $B_L(q) = B(q)^\frown 0^{L-lg(B(q))}$ (e.g. $B_5(6) = 01100$). $B_L(q)$ is in fact the word coding q in reversed binary which is forced to reach length L, possibly padding it with 0's on its right.
- For $m \in \mathbb{N}$, we also denote m the domain $\{0, 1, \ldots, m-1\}$. Remark that for m and $L \geq \lfloor \log(m) + 1 \rfloor$ fixed in $\mathbb{N}$, the map $B_L : m \to \{0,1\}^L$, which maps q to $B_L(q)$, is injective.
- For $m \in \mathbb{N}$, Id_m denotes the identity function on m.
- For $w = a_0 a_1 \ldots a_{n-1} \in \{0,1\}^*$ and for $t < n$, we will call "t^{th} bit of w " the symbol a_t, ie. the $(t+1)^{th}$ letter of the word w.
- Let $w = b_0 b_1 \ldots b_{n-1} \in \{0,1\}^n$. We will call " predicate naturally coding w" the unary predicate $X_w \subseteq n$ defined by $\forall t < n : X_w(t) \Leftrightarrow b_t = 1$.

  In the same way, if $w = d_0 d_1 \ldots d_{n-1} \in \{1,2\}^n$ is a word on the dyadic alphabet, the "predicate naturally coding w" will be the unary predicate X_w on n defined by $\forall t < n : X_w(t) \Leftrightarrow d_t = 2$.

Remark. In the present paper, both the binary and the dyadic[5] representation of positive integers are used. Binary representation is used when padded and uniform length notations are helpful. Dyadic one allows to identify any nonempty word of $\{1,2\}^*$ with a positive integer, in a one-one way.

5. Recall that dyadic representation consists in encoding each positive integer x by the word $D(x) = x_p \ldots x_1 x_0 \in \{1,2\}^*$, where $(x_0, x_1, \ldots, x_p)$ is the only tuple on {1,2} such that: $x = x_0 + x_1 2 + \ldots + x_p 2^p$.

1.3.2 Flattening of a function

For this whole paragraph, m, L, and m' denote three integers such that: $L = \lfloor \log(m) + 1 \rfloor$ and $m' \geq mL$. We show there that, using a simple coding, each function $h : m \to m$ can be identified with a unary relation H on m'.

Let $h : m \to m$. For every $i < m$, let us denote $b_i = B_L(h(i))$. The image of each $i < m$ being thus coded by a word of $\{0,1\}^L$, h can be coded by a word w_h, the length of which is exactly m':

$$w_h = b_0^\frown b_1^\frown \ldots^\frown b_{m-1}^\frown 0^{m'-mL}$$

Remark. If for $x < m$ and $t < L$, b_x^t denotes the t^{th} bit of b_x, then:

$$h(x) = \sum_{t=0}^{L-1} b_x^t 2^t = \sum_{\substack{t<L \ st \\ b_x^t=1}} 2^t$$

Every function from m to m is thus represented in a single way by a word of $\{0,1\}^{mL}.0^{m'-mL}$. Besides, one usually represents a word $v \in \{0,1\}^*$ of length q by the unary predicate $V \subseteq q$ defined as follows:

$$\forall i < q : \ V(i) \ \Leftrightarrow \ the \ t^{th} \ bit \ of \ v \ is \ 1.$$

Composition of this two codings makes it possible to associate to each unary function $h : m \to m$ a unary predicate $H \subseteq m'$ defined as follows:

$$\forall \alpha < m' : \ H(\alpha) \Leftrightarrow the \ \alpha^{th} \ bit \ of \ w_h \ is \ 1$$

$$i.e. \qquad H(\alpha) \Leftrightarrow \alpha = x_\alpha L + t_\alpha, \ with \ x_\alpha < m \, , \, t_\alpha < L \, , \ and \ b_{x_\alpha}^{t_\alpha} = 1,$$

which implies the following:

$$\forall x < m \, , \, \forall t < L \ : \ H(xL + t) \Leftrightarrow b_x^t = 1.$$

DEFINITION 1.5. From now on the predicate $H \subseteq m'$ thus constructed from the function $h : m \to m$ will be called the *flattening of* h *on* m'.

Remark. The previous remark immediately results in:

$$\forall x < m \ : h(x) = \sum_{\substack{t<L \ st \\ H(xL+t)}} 2^t.$$

DEFINITION 1.6. We call $Bit_m^{m'}$ the flattening of the identity function $Id_m : m \to m$ on m'.

In other words, $Bit_m^{m'}$ is the unary predicate on m' naturally associated to the word $B_L(0)^\frown B_L(1)^\frown \ldots^\frown B_L(m-1)^\frown 0^{m'-mL} \in \{0,1\}^{m'}$. Or else:

$$\forall \alpha < m' : \ Bit_m^{m'}(\alpha) \Leftrightarrow \begin{cases} \alpha = x_\alpha L + t_\alpha, \ with \ x_\alpha < m \, , \, t_\alpha < L \, , \\ and \ the \ t_\alpha^{th} \ bit \ of \ B_L(x_\alpha) \ is \ 1 \end{cases}$$

In fact, to make the notations simpler, we will note this predicate Bit, the (m, m') pair which it refers to being implicitly given by the context.

1.3.3 From functional formulas to relational formulas

Let $k > 1$ be a fixed integer. For every integer n, we note: $l = \lceil \frac{\log(n+1)}{k} \rceil$, $m = \lceil \frac{n}{l} \rceil$, $L = \lfloor \log(m) + 1 \rfloor$. Given these notations, we get: $m = \Theta(n/\log n)$ and therefore: $mL = \Theta(n)$. This guarantees the existence of an integer c such that:
$\forall n \in \mathbb{N}^*,\ mL < cn \quad (*)$

Let c be a fixed integer meeting $(*)$. For all $n \in \mathbb{N}$, the (m, cn) pair therefore verifies $mL < cn$, and it is possible to refer to the flattening of a function $h : m \to m$ on cn.

In this paragraph, we seek to associate to each language $\mathcal{L} \in$ NLIN, an existential monadic second-order formula $\Phi_{\mathcal{L}}$, in such a way that a word $w \in \{0,1\}^*$ is in $\mathcal{L}$ iff a certain finite structure, in which w is encoded by the flattening of f_w, is a model of $\Phi_{\mathcal{L}}$. This will be donne along Lemmas 1.6, 1.7, 1.8, 1.9 in using Theorem 1.1 and stating successively that:

• Flattenings are quite adapted to translate certain functional formulas, that we could call "simple (atomic) formulas", in relational formulas, provided we add to the language a series of predefined arithmetical symbols including $+$.

• Does mean changing a little its first-order quantifier prefix, the formula which characterizes a language $\mathcal{L} \in$ NLIN (by Theorem 1.1) can be required to be a boolean combination of these "simple formulas".

• The conjunction of the two preceding facts allows to associate to each $\mathcal{L} \in$ NLIN a monadic existential second-order formula whose signature contains the above mentioned arithmetical objects. Moreover, if the transformation is made carefully, the first-order part of this formula can be required to belong to the class $FO(\forall^*\exists^*)$.

• At last, we state that, excepted for $+$, all these arithmetical symbols can be removed, without changing the structure of the formula.

All this will lead to Corollary 1.10, which establishes the existence of the formula $\Phi_{\mathcal{L}}$.

Lemma 1.6. *Let f, g be two unary fonctions on m, F, G their flattenings on cn. Then, for all $x, y < m$, the assertions $m \models f(x) = g(y)$ and $m \models f(x) < g(y)$ are respectively equivalent to :*

$$cn \models (\forall t < L)(F(xL + t) \leftrightarrow G(yL + t))$$

and

$$cn \models (\exists t < L)(\forall t' < L) \left[\begin{array}{l} t' > t \to (F(xL + t') \leftrightarrow G(yL + t')\wedge \\ G(yL + t) \wedge \neg F(xL + t)) \end{array} \right]$$

Proof. We just have to recall the connection between a function $h : m \to m$, its associated word $B_L(h(0))^\frown \ldots ^\frown B_L(h(m-1)) \in \{0,1\}^*$ and its flattening on cn to see that these three equivalences express the following facts:

1. $f(x) = g(y)$ iff the words $B_L(f(x))$ and $B_L(g(y))$ are equal, *i.e.*coincide on their L successive bits.

2. $f(x) < g(y)$ iff $B_L(f(x))$ and $B_L(g(y))$ are in the same order for the reverse lexicographic order on $\{0,1\}^L$.

Lemma 1.7. *Let $\mathcal{L} \in \Sigma^*$. If $\mathcal{L} \in NLIN$, then there exists $\Psi^o \in FO(\forall^*)$ on a signature made up, beside the symbols $<, 0$, of unary function symbols $f, g_1 \ldots g_r$, such that:*

(i) $\forall w \in \Sigma^*,\ w \in \mathcal{L} \Leftrightarrow \langle m, f_w, <, 0\rangle \models \exists g_1 \ldots \exists g_r \Psi^o(f, \overline{g}, <, 0)$.

(ii) Ψ^o *has the following form:* $\Psi^o \equiv \forall \overline{x} \bigwedge_i \bigvee_j A_{ij}$, *where each* A_{ij} *can be written:* $\alpha(u) = \beta(v)$ *or* $\neg[\alpha(u) = \beta(v)]$ *or* $\neg[\alpha(u) < \beta(v)]$, *for some* $\alpha, \beta \in \{f, \overline{g}, Id\}$ *and some* $u, v \in \{0, \overline{x}\}$.

Proof. Let $\Psi \in FO(wq)$ be the formula associated to $\mathcal{L}$ by the Theorem 1.1. So we have, for all $w \in \Sigma^*$:

$$w \in \mathcal{L} \Leftrightarrow \langle m, f_w, <, 0\rangle \models \exists g_1 \ldots \exists g_p \forall x \Psi(x, f, \overline{g}, <, 0).$$

We can consider, without loss of generality, that Ψ is under conjunctive normal form. Then an atomic subformula of Ψ has the form:

$$h_1 \circ \ldots \circ h_p(u) \propto k_1 \circ \ldots \circ k_s(v),$$

with $h_i, k_i \in \{f, \overline{g}, Id\}$ (where Id is the identity function), $u, v \in \{0, x\}$ and $\propto \in \{=, \neq, <, \not<\}$. But this formula is clearly equivalent to:

$$\begin{array}{ll}\forall x & [\alpha_{p-1}(x) = h_{p-1} \circ h_p(x) \wedge \bigwedge_{i=1}^{i=p-2} \alpha_i(x) = h_i \circ \alpha_{i+1}(x) \wedge \\ & \beta_{s-1}(x) = k_{s-1} \circ k_s(x) \wedge \bigwedge_{i=1}^{i=s-2} \beta_i(x) = k_i \circ \beta_{i+1}(x) \wedge \\ & \alpha_1(u) \propto \beta_1(v)]\ ,\end{array}$$

where the α_i's and β_i's are new function symbols. Thus, if we still denote $\overline{g}$ the tuple $g_1, \ldots, g_r$ made up of $g_1, \ldots, g_p$ in addition to as many new function symbols as necessary, this allows to write Ψ in the following equivalent form:

$$\Psi \equiv \bigwedge \alpha \circ \beta(x) = \gamma(x) \wedge \bigwedge \bigvee \varepsilon(u) \propto \varepsilon'(v),$$

with $\alpha, \beta, \gamma \in \{f, \overline{g}\}$, $\varepsilon, \varepsilon' \in \{f, \overline{g}, Id\}$, $\propto \in \{=, \neq, <, \not<\}$. Besides, replacing each subformula $\alpha \circ \beta(x) = \gamma(x)$ of Ψ by $\forall y[\beta(x) = y \rightarrow \alpha(y) = \gamma(x)]$, and writing the obtained formula under prenex conjonctive normal form, we get for $\forall x \Psi$ an equivalent formula of the form:

$$\forall \overline{x} \bigwedge \bigvee \alpha(u) \propto \beta(v)$$

with $\alpha, \beta \in \{f, \overline{g}, Id\}$, $u, v \in \{0, \overline{x}\}$, $\propto \in \{=, \neq, <, \not<\}$, which can be written more precisely: $\forall \overline{x} \bigwedge_i \bigvee_j B_{ij}$, where each B_{ij} is one of the formulas $\alpha(u) = \beta(v)$, $\alpha(u) \neq \beta(v)$, $\alpha(u) < \beta(v)$, $\alpha(u) \not< \beta(v)$ for some $\alpha, \beta \in \{f, \overline{g}, Id\}$ and some $u, v \in \{0, \overline{x}\}$.

At last, we observe that the formula $\alpha(u) < \beta(v)$ can be expressed with the others (namely, those of the form $\alpha(u) \neq \beta(v)$, $\alpha(u) \not< \beta(v)$) by a universal formula using the only connectives $\wedge$ and $\vee$. Indeed we have:
$\alpha(u) < \beta(v) \Leftrightarrow \neg[\beta(v) < \alpha(u)] \wedge \neg[\beta(v) = \alpha(u)]$
The prenex conjunctive normal form of the formula produced by this last rewriting is the sought formula Ψ^o.

Lemma 1.8. *Let $\mathcal{L} \in \Sigma^*$. If $\mathcal{L} \in NLIN$, then there exists $\tilde{\Psi} \in FO(\forall^*\exists^*)$ on signature $\{F, G_1, \ldots, G_r, +, <, \times, Bit, m, L\}$, where $F, G_i, +, <, \times, Bit$ are predicate symbols of respective arities 1, 1 (for every $i \leq r$), 3, 2, 3, and m, L are constants symbols, such that, for every $w \in \Sigma^*$:*

$$w \in \mathcal{L} \;\Leftrightarrow\; \langle cn, F\rangle \models \exists G_1 \ldots \exists G_r \tilde{\Psi}(F, \overline{G}, +, <, \times, Bit, m, L),$$

where F is the flattening of f_w on cn, $+, <, \times$ are respectively the predefined addition, linear order and product on cn, Bit is the flattening of Id_m on cn, and m, L are the two predefined constants defined at the begining of this subsection.

Proof. Let $\Psi^o \equiv \forall\overline{x} \bigwedge_i \bigvee_j A_{ij}$ the formula associated to $\mathcal{L}$ by Lemma 1.7. We compute $\tilde{\Psi}$ from Ψ^o in translating each subformula A_{ij} according to the equivalences stated in Lemma 1.6. Namely, we have the following translations:

$$\begin{array}{rl}
\alpha(u) = \beta(v) \text{ becomes} & (\forall t < L)[A(uL+t) \leftrightarrow B(vL+t)]\ ; \\
\neg[\alpha(u) = \beta(v)] \text{ becomes} & (\exists t < L)\neg[A(uL+t) \leftrightarrow B(vL+t)]\ ; \\
\neg[\alpha(u) < \beta(v)] \text{ becomes} & (\forall t < L)(\exists t' < L) \\
& \neg\left[\begin{array}{l} t' > t \rightarrow (A(uL+t') \leftrightarrow B(vL+t') \wedge \\ B(vL+t) \wedge \neg A(uL+t)) \end{array}\right]
\end{array}$$

Notice that all these formulas are in $FO(\forall^*\exists^*)$. Thus, rewriting the obtained formula under prenex conjunctive normal form, we get a new formula with the following pattern:

$$(\forall\overline{x} < m)(\forall\overline{t}_1 < L)(\exists\overline{t}_2 < L) \bigwedge \bigvee \pm H(yL+t),$$

with $H \in \{F, \overline{G}, Bit\}$, $y \in \overline{x}$, $t \in \{\overline{t}_1, \overline{t}_2\}$. This is the sought formula $\tilde{\Psi}$.

We can achieve a stronger result in showing that the previous lemma still holds when using $+$ as the only arithmetical predefined symbol.

Recall that for k, c two fixed integers and for $n \in \mathbb{N}$, we denote: $l = \lceil \frac{\log(n+1)}{k} \rceil$, $m = \lceil \frac{n}{l} \rceil$, $L = \lfloor \log(m) + 1 \rfloor$. The following result, essentially due to Lynch, states that the constants k, c, n, l, m, L, as well as the predicates $<$, $\times$, Bit, are definable in cn by existential monadic second-order formulas on a signature including the predefined predicate $+$. Moreover, this definability preserves the $\forall^*\exists^*$ quantifier pattern of the formulas. More precisely, we have the:

Lemma 1.9. *Let σ_0 denote the set of the predefined symbols k, c, n, l, m, L, Bit, $\times$, $<$, $+$. Let $\overline{U}$ be a series of unary predicate symbols. Let $\Phi \in FO(\forall^*\exists^*)$ be a sentence on the signature $\sigma_0 \cup \{\overline{U}\}$. Then there exists $\Phi' \in FO(\forall^*\exists^*)$, on the signature $\{+\} \cup \{\overline{U}\} \cup \{D_1, \ldots, D_j\}$, where D_i's are new unary predicate symbols, such that:*

$$cn \models \Phi(\sigma_0, \overline{U}) \;\Leftrightarrow\; cn \models \exists D_1 \ldots \exists D_j \Phi'(\overline{U}, +, \overline{D}).$$

Proof. Definability of the predicates $<$, $\times$, Bit, has been proved in [Ly1]. We will recall the main steps of this proof and we will show the definability of the arithmetical constants k, c, n, l, m, L, taking a specific care to the quantifier prefixes of the used formulas. But as these profs are long and rather technical, we put them togather in a following section, especially devoted to definability of arithmetical predicates and constants.

Corollary 1.10. *Let $\mathcal{L}$ be a language on $\Sigma = \{1, 2\}$. If $\mathcal{L} \in NLIN$, then there exists a sentence $\tilde{\Psi} \in FO(\forall^*\exists^*)$ on signature $\{F, G_1, \ldots, G_q, +\}$, where F and the G_i's are unary predicate symbols, $+$ is a ternary predicate symbol, such that:*

$$[w \in \mathcal{L}] \Leftrightarrow [\langle cn, F, + \rangle \models \exists G_1 \ldots \exists G_q \tilde{\Psi}(F, G_1, \ldots, G_q)]$$

where F is the flattening of f_w on cn, and $+$ is the predefined addition on cn.

1.4 Definability of arithmetical objects

1.4.1 Square, exponential, product, arithmetical constants

Notations : X^2, $X^2 - X$, 2^X, $2^X - X$ denote the following unary predicates on cn:
$X^2 = \{x^2,\ 4 \leq x^2 < cn\}$
$X^2 - X = \{x^2 - x,\ 4 \leq x^2 - x < cn\}$
$2^X = \{2^x,\ 4 \leq 2^x < cn\}$
$2^X - X = \{2^x - x,\ 4 \leq 2^x < cn\}$

Lemma 1.11 (Ly1, p.134). *There exist two formulas $C(U_0, U_1)$ and $c(x, y, U_0, U_1)$, respectively belonging to $FO(\forall^3\exists)$ and $FO(\exists\forall)$, such that:*
(i) There exist $U_0, U_1 \subset cn$ s.t. $\langle cn, + \rangle \models C(U_0, U_1)$.
(ii) If $U_0, U_1 \subset cn$ and $\langle cn, + \rangle \models C(U_0, U_1)$, then for all $x, y < cn$:
$\langle cn, + \rangle \models c(x, y, U_0, U_1) \Leftrightarrow y = x^2$.

Lemma 1.12 (Ly1, p.135). *There exist two formulas $E(V_0, V_1)$ and $e(x, y, V_0, V_1)$, respectively belonging to $FO(\forall^3\exists)$ and $FO(\exists\forall)$, such that :*
(i) There exist $V_0, V_1 \subset cn$ s.t. $\langle cn, + \rangle \models E(V_0, V_1)$.
(ii) If $V_0, V_1 \subset cn$ and $\langle cn, + \rangle \models E(V_0, V_1)$, then for all $x, y < cn$:
$\langle cn, + \rangle \models e(x, y, V_0, V_1) \Leftrightarrow y = 2^x$.

Remark. $C(U_0, U_1)$ and $E(V_0, V_1)$ respectively express:
"$U_0 = X^2,\ U_1 = X^2 - X$" and "$V_0 = 2^X,\ V_1 = 2^X - X$".

Lemma 1.13. *There exist two formulas $d_0(a_0)$ and $d_1(a_0, a_1, U_0, U_1)$, respectively belonging to $FO(\forall)$ and $FO(\forall^3\exists)$, such that for every $a_0, a_1 < cn$:*
(i) $cn \models d_0(a_0)$ iff $a_0 = cn - 1$.
(ii) $cn \models C(U_0, U_1) \wedge d_0(a_0) \wedge d_1(a_0, a_1)$ iff $a_1 = \lfloor \sqrt{cn-1} \rfloor$.

Proof.

$$\begin{aligned} d_0(a_0) \equiv\ & \forall x(x \leq a_0) \\ d_1(a_0, a_1) \equiv\ & \forall x(a_1^2 = x \to x = \lfloor \sqrt{cn-1} \rfloor^2) \\ \equiv\ & \forall x(\neg c(a_1, x, U_0, U_1) \vee [U_0(x) \wedge \forall y(U_0(y) \to y \leq x)] \end{aligned}$$

Lemma 1.14 (Ly1, p.135). *There exists $p(x, y, z, a_0, a_1, U_0, U_1) \in FO(\exists^*\forall)$ s.t.: If U_0, U_1 are unary relations on cn and a_0, a_1 two integers $< cn$ such that $cn \models C(U_0, U_1) \wedge d_0(a_0) \wedge d_1(a_0, a_1)$, then for all $x, y, z < cn$: $cn \models p(x, y, z, a_0, a_1)$ iff $z = xy$.*

Lemma 1.15. *There exist $d_2(x_0, x_1, x_2)$, $d_3(x_0, x_1, x_2, x_3)$, $d_4(x_0, x_1, x_2, x_3, x_4)$, $d_5(x_0, x_1, x_2, x_3, x_4, x_5)$, formulas belonging to $FO(\forall^*\exists)$ such that for all U_0, U_1, V_0, $V_1 \subset cn$ and $a_0, a_1, \ldots, a_5 < cn$:*
If $cn \models C(U_0, U_1) \wedge E(V_0, V_1) \wedge d_0(a_0) \wedge d_1(a_0, a_1)$ then:
$cn \models d_2(a_0, a_1, a_2)$ iff $a_2 = n - 1$;
$cn \models d_3(a_0, \ldots, a_3)$ iff $a_3 = l = \lceil \frac{\log(n+1)}{k} \rceil$;
$cn \models d_4(a_0, \ldots, a_4)$ iff $a_4 = m = \lceil \frac{n}{l} \rceil$;
$cn \models d_5(a_0, \ldots, a_5)$ iff $a_5 = L = \lfloor \log(m) + 1 \rfloor$.

Proof.

$$\begin{aligned} a_2 = n - 1 \quad & \Leftrightarrow ca_2 + (c-1) = a_0 \\ a_3 = l \quad & \Leftrightarrow a_3 = \lceil \tfrac{\log(n+1)}{k} \rceil \\ & \Leftrightarrow 2^{k(a_3-1)} \leq n < 2^{ka_3} \\ & \Leftrightarrow a_3 \text{ is the greater } x < cn \text{ s.t. } 2^{k(x-1)} \leq n \\ & \Leftrightarrow \forall x(2^{k(x-1)} \leq n \to x \leq a_3) \wedge 2^{k(a_3-1)} \leq n \\ & \Leftrightarrow \left[\begin{array}{l} \forall x \forall y \forall z[(y = k(x-1) \wedge z = 2^y \wedge z \leq n) \to x \leq a_3] \\ \wedge \forall y \forall z[(y = k(a_3 - 1) \wedge z = 2^y) \to z \leq n] \end{array} \right] \\ & \Leftrightarrow \forall x \forall y \forall z \left[\begin{array}{l} [(y = k(x-1) \wedge e(y,z) \wedge z \leq a_2 + 1) \to x \leq a_3] \\ \wedge[(y = k(a_3 - 1) \wedge e(y,z)) \to z \leq a_2 + 1] \end{array} \right] \end{aligned}$$

and $e(y, x) \in FO(\exists\forall)$ implies $\neg e(y, x) \in FO(\forall\exists)$, which gives the sought result.

$$
\begin{aligned}
a_4 = m \quad & \Leftrightarrow a_4 = \lceil \tfrac{n}{l} \rceil \\
& \Leftrightarrow l(a_4 - 1) \leq n - 1 < la_4 \\
& \Leftrightarrow a_4 \text{ is the greater } x < cn \text{ s.t. } l(x-1) \leq n-1 \\
& \Leftrightarrow \forall x \forall y \ [((y = l(x-1) \wedge y \leq n-1) \rightarrow x \leq a_4) \wedge \\
& \qquad\qquad (y = l(a_4 - 1) \rightarrow y \leq n-1)] \\
& \Leftrightarrow \forall x \forall y \ [((p(a_3, x-1, y) \wedge y \leq a_2) \rightarrow x \leq a_4) \wedge \\
& \qquad\qquad (p(a_3, a_4 - 1, y) \rightarrow y \leq a_2)
\end{aligned}
$$

and the relation $p(a_3, a_4, x) \in FO(\exists^* \forall)$ gives the sought result.

$$
\begin{aligned}
a_5 = L \quad & \Leftrightarrow a_5 = \lfloor \log(m) + 1 \rfloor \\
& \Leftrightarrow a_5 \leq \log(m) + 1 < a_5 + 1 \\
& \Leftrightarrow 2^{a_5 - 1} \leq m < 2^{a_5} \\
& \Leftrightarrow 2^{a_5} \leq 2a_4 < 2.2^{a_5} \\
& \Leftrightarrow \forall x (x = 2^{a_5} \rightarrow x \leq 2a_4 < 2x) \\
& \Leftrightarrow \forall x (\neg e(a_5, x) \vee x \leq 2a_4 < 2x)
\end{aligned}
$$

but $e(a_5, x) \in FO(\exists \forall)$, hence the result.

1.4.2 BIT-definability

Preliminaries.

We have:
$\forall \alpha < cn, \ \exists! x, t < cn \ s.t. \ \alpha = xL + t \ and \ t < L$ (Euclidean division of α by L).
$\forall x < cn, \ \forall t < L, \ \exists! q, r < cn \ s.t. \ x = q2^t + r \ and \ r < 2^t$ (euclidean division of x by 2^t). Hence:

$$
\forall \alpha < cn, \ \exists! (q, x, r, t) \in (cn)^4 \text{ s.t. } \left\{ \begin{array}{l} x = q2^t + r \\ \alpha = xL + t \\ t < L; r < 2^t \end{array} \right.
$$

Which can be written, further decomposing the used operations:

$$
\forall \alpha < cn, \ \exists! (q, x, r, t, y_1, y_2, y_3) \in (cn)^7 \text{ s.t. } \left\{ \begin{array}{l} y_1 = 2^t \\ y_2 = qy_1 \\ x = y_2 + r \\ y_3 = xL \\ \alpha = y_3 + t \\ t < L \\ r < y_1 \end{array} \right. \tag{1.4}
$$

Let us call *good decomposition of* α and denote $gd(\alpha)$, the only seven-tuple $(q, x, r, t, y_1, y_2, y_3)$ of elements of cn satisfying (1.4).
Let us denote $GD(\alpha, q, x, r, t, y_1, y_2, y_3, L)$ the conjunction:

$$
\begin{aligned}
& (y_1 = 2^t) \wedge (y_2 = qy_1) \wedge (x = y_2 + r) \wedge (y_3 = xL) \wedge \\
& (\alpha = y_3 + t) \wedge (t < L) \wedge (r < y_1)
\end{aligned}
$$

$$i.e.\ GD(\alpha,q,x,r,t,y_1,y_2,y_3,L) \equiv e(t,y_1) \wedge p(q,y_1,y_2) \wedge (x = y_2 + r) \wedge \\ p(x,L,y_3) \wedge (\alpha = y_3 + t) \wedge (t < L) \wedge (r < y_1).$$

Then we have, for each $(\alpha,q,x,r,t,y_1,y_2,y_3) \in (cn)^8$:
$(q,x,r,t,y_1,y_2,y_3) = gd(\alpha) \Leftrightarrow cn \models GD(\alpha,q,x,r,t,y_1,y_2,y_3,L)$.

Remark. $GD \in FO(\exists^*\forall)$.

Definability of BIT with a formula of $FO(\forall^*\exists^*)$.

We have seen (definition 1.6) that BIT is defined on each $\alpha < cn$ by:
$BIT(\alpha) \Leftrightarrow \exists x,t < cn$ $s.t.$ $\alpha = xL + t,\ t < L,\ x < m$, and the t^{th} bit of $B_L(x)$ is 1, where $B_L(x)$ denotes the reverse binary encoding of lenght L of x.
But the t^{th} bit of $B_L(x)$ is 1 iff $x = q2^t + r$ with $r < 2^t$ and q odd. So, we have $BIT(\alpha)$ iff the 1^{st} term of $gd(\alpha)$ is odd and the 2^{nd} is $< m$. That can be easily written as:

$$BIT(\alpha) \Leftrightarrow cn \models \exists q,x,r,t,y_1,y_2,y_3[GD(\alpha,q,\ldots,y_3,L) \wedge (x < m) \wedge \\ \exists q'(q = 2q' + 1)]$$

We will denote: $BIT(\alpha) \Leftrightarrow cn \models B_1(\alpha,L,m)$
On the other hand, existence and unicity of $gd(\alpha)$ for each α allows to characterize BIT in a second way:

$$BIT(\alpha) \Leftrightarrow cn \models \forall q,x,r,t,y_1,y_2,y_3[GD(\alpha,q,\ldots,y_3,L) \to ((x < m) \wedge \\ \exists q'(q = 2q' + 1))],$$

that we will denote: $BIT(\alpha) \Leftrightarrow cn \models B_2(\alpha,L,m)$.

Remark. From $GD \in FO(\exists^*\forall)$, one immediately deduces $B_1 \in FO(\exists^*\forall)$ and $B_2 \in FO(\forall^*\exists)$.

From the two previous equivalences, we deduce a third characterization of BIT: Let $B \subseteq cn$. Then:
$B = BIT \ \Leftrightarrow\ cn \models Def_{bit}(B,m,L)$, where Def_{bit} is the following formula:
$Def_{bit}(B,m,L) \equiv \forall\alpha[(B(\alpha) \to B_2(\alpha,L,m)) \wedge (B_1(\alpha,L,m) \to B(\alpha))]$

Remark. The previous remark leads to: $Def_{bit}(B,m,L) \in FO(\forall^*\exists^*)$.

1.4.3 Elimination of arithmetical functions

Lemma 1.16. *Let σ_0 denote the set of the predefined symbols k, c, n, l, m, L, Bit, $\times$, $<$, $+$. Let $F, G_1, \ldots, G_p$ be a series of unary predicate symbols. Let Φ be a sentence on the signature $\sigma_0 \cup \{F, \overline{G}\}$ which has the form:*

$$(\forall\overline{x} < m)(\forall\overline{t}_1 < L)(\exists\overline{t}_2 < L) \bigwedge\bigvee \pm H(xL + t)$$

with $x \in \overline{x}$, $t \in \{\overline{t}_1, \overline{t}_2\}$, $H \in \{F, \overline{G}, BIT\}$.
Then there exists $\Phi' \in FO(\forall^*\exists^*)$, on the signature $\{+\} \cup \{F, \overline{G}\} \cup \{D_1, \ldots, D_j\}$, where D_i's are new unary predicate symbols, such that:

$$\langle cn, F\rangle \models \exists\overline{G}\Phi(F,\overline{G},\sigma_0,) \ \Leftrightarrow\ \langle cn, F\rangle \models \exists\overline{G}\exists\overline{D}\Phi'(F,\overline{G},\overline{D},+).$$

Proof. Let $A_0, \ldots, A_5$ be unary predicate symbols. We denote $S(A_0, \ldots, A_5)$ the following formula:

$$\begin{array}{ll} \forall x_0 \ldots \forall x_5 & [(A_0(x_0) \to d_0(x_0)) \wedge ((A_0(x_0) \wedge A_1(x_1)) \to d_1(x_0, x_1)) \\ & \wedge \ldots \wedge ((A_0(x_0) \wedge \ldots \wedge A_5(x_5)) \to d_5(x_0, \ldots, x_5))] \wedge \\ \exists a_0 \ldots \exists a_5 & [\bigwedge A_i(a_i)] \end{array}$$

which expresses: $A_0, \ldots, A_5$ are the singletons $\{cn-1\}$, $\{\lfloor\sqrt{cn-1}\rfloor\}$,$\{n-1\}$, $\{l\}$, $\{m\}$, $\{L\}$. Then the assertion:

$$cn \models \exists G_1 \ldots \exists G_p (\forall \overline{x} < m)(\forall \overline{t}_1 < L)(\exists \overline{t}_2 < L) \bigwedge \bigvee \pm H(xL + t),$$

with $x \in \overline{x}$, $t \in \{\overline{t}_1, \overline{t}_2\}$, $H \in \{F, \overline{G}, BIT\}$, is equivalent to:

$$\begin{array}{rl} cn \models & \exists G_1 \ldots \exists G_p \, \exists U_0 \exists U_1 \, \exists V_0 \exists V_1 \, \exists A_0 \ldots \exists A_5 \, \exists B : \\ & C(U_0, U_1) \wedge E(V_0, V_1) \wedge S(A_0, \ldots, A_5) \wedge \\ & \forall a_0 \ldots \forall a_5 [\bigwedge A_i(a_i) \to Def_{bit}(B, a_4, a_5)] \wedge \\ & \forall a_0 \ldots \forall a_5 \forall \overline{x} \forall \overline{t}_1 \exists \overline{t}_2 \; [(\bigwedge A_i(a_i) \wedge \overline{x} < a_4 \wedge \overline{t}_1 < a_5) \to \\ & \qquad ((\overline{t}_2 < a_5) \wedge \bigwedge \bigvee \pm H(xa_5 + t))] \end{array}$$

where H denotes B each time it was denoting BIT in the previous formula.
Let $DEF(\overline{U}, \overline{V}, \overline{A}, B)$ denote the formula:

$$C(\overline{U}) \wedge E(\overline{V}) \wedge S(\overline{A}) \wedge \forall \overline{a} [\bigwedge A_i(a_i) \to Def_{bit}(B, a_4, a_5)]$$

Let $ELIM(F, \overline{G}, \overline{U}, \overline{V}, \overline{A}, B)$ denote the formula:

$$\begin{array}{c} (\forall \overline{a}, \overline{x}, \overline{t}_1)(\exists \overline{t}_2) \\ \{[\bigwedge A_i(a_i) \wedge (\overline{x} < a_4) \wedge (\overline{t}_1 < a_5)] \to [(\overline{t}_2 < a_5) \wedge \bigwedge \bigvee \pm H(xa_5 + t)]\} \end{array}$$

where $H \in \{F, \overline{G}, B\}$, $x \in \overline{x}$ and $t \in \{\overline{t}_1, \overline{t}_2\}$.
For $\overline{U}, \overline{V}, \overline{A}, B$ fixed unary relations in cn such that $cn \models DEF(\overline{U}, \overline{V}, \overline{A}, B)$, we have:

$$cn \models ELIM(F, \overline{G}, \overline{U}, \overline{V}, \overline{A}, B)$$

iff

$$\begin{array}{rl} cn \models & (\forall \overline{a}, \overline{x}, \overline{\mathbf{u_x}}, \overline{t}_1)(\exists \overline{t}_2)\{[\bigwedge A_i(a_i) \wedge (\overline{x} < a_4) \wedge (\overline{t}_1 < a_5) \wedge \bigwedge_{x \in \overline{x}}(u_x = xa_5)] \\ & \qquad \to [(\overline{t}_2 < a_5) \wedge \bigwedge \bigvee \pm H(u_x + t)]\} \end{array}$$

where each $u_x \in \overline{u_x}$ is a new variable associated to $x \in \overline{x}$.

Let us denote:
$\overline{\alpha} = (\overline{a}, \overline{x}, \overline{u_x}, \overline{t}_1)$;
$\phi(\overline{\alpha}) \equiv \bigwedge A_i(a_i) \wedge (\overline{x} < a_4) \wedge (\overline{t}_1 < a_5) \wedge \bigwedge_{x \in \overline{x}}(u_x = xa_5)$;
$\psi(\overline{\alpha}, \overline{t}_2) \equiv (\overline{t}_2 < a_5) \wedge \bigwedge \bigvee \pm H(u_z + t)$.
As $\overline{t}_2$ does not appear in $\overline{\alpha}$, the following equivalence holds:
$cn \models \forall \overline{\alpha} \exists \overline{t}_2 \{\phi(\overline{\alpha}) \to \psi(\overline{\alpha}, \overline{t}_2)\}$
iff
$cn \models \forall \overline{\alpha} \{\phi(\overline{\alpha}) \to \exists \overline{t}_2 \psi(\overline{\alpha}, \overline{t}_2)\}$.

Whence we get:

$$
\begin{array}{ll}
cn \models & \exists\overline{G}(\forall\overline{x} < m)(\forall\overline{t}_1 < L)(\exists\overline{t}_2 < L)\bigwedge\bigvee \pm H(xL+t) \\
& \text{with } x \in \overline{x},\ t \in \{\overline{t}_1, \overline{t}_2\},\ H \in \{F, \overline{G}, BIT\} \\
\text{iff} & \\
cn \models & \exists\overline{G}\,\exists\overline{U}\,\exists\overline{V}\,\exists\overline{A}\,\exists B\,\{\, DEF(\overline{U}, \overline{V}, \overline{A}, B) \wedge \forall\overline{\alpha}[\phi(\overline{\alpha}) \to \exists\overline{t}_2\psi(\overline{\alpha}, \overline{t}_2)]\,\} \\
& \quad \text{with}: \quad \overline{\alpha} = (\overline{a}, \overline{x}, \overline{u_x}, \overline{t}_1) \\
& \qquad\qquad\quad \phi(\overline{\alpha}) \equiv s(\overline{a}, \overline{A}) \wedge (\overline{x} < a_4) \wedge (\overline{t}_1 < a_5) \wedge \bigwedge_{x\in\overline{x}}(u_x = xa_5) \\
& \qquad\qquad\quad \psi(\overline{\alpha}, \overline{t}_2) \equiv (\overline{t}_2 < a_5) \wedge \bigwedge\bigvee \pm H(u_x + t).
\end{array}
$$

To conclude, we just have to see that:
$\phi(\overline{\alpha})$ is in $FO(\exists^*\forall)$ because of the expression of the product seen above;
$\psi(\overline{\alpha}, \overline{t}_2)$ is without quantifier;
hence $\{\forall\overline{\alpha}[\phi(\overline{\alpha}) \to \exists\overline{t}_2\psi(\overline{\alpha}, \overline{t}_2)]\,\} \in FO(\forall^*\exists^*)$;
at last $DEF(\overline{U}, \overline{V}, \overline{A}, B)$, obtained by conjunctions of the $FO(\forall^*\exists^*)$-formulas $C(\overline{U})$, $E(\overline{V})$, $S(\overline{A})$ and $\forall\overline{a}[\bigwedge A_i(a_i) \to Def_{bit}(B, a_4, a_5)]$, is itself in $FO(\forall^*\exists^*)$.
Thus the formula $DEF \wedge \forall\overline{\alpha}[\phi(\overline{\alpha}) \to \exists\overline{t}_2\psi(\overline{\alpha}, \overline{t}_2)]$ is the sought formula Φ'.

Remark. From now on, we will still use subformulas like $x < m$, $Bit(yL+t)$, ... but they will have to be interpreted as the abbreviations of existential monadic formulas on $\{+\}$.

1.5 Connection between F_w and X_w

Let us now come back to Corollary 1.10. It affirms that a set of words belonging to NLIN can be characterized by an existential second-order formula whose second-order variable are monadic predicates and addition, *via* an encoding of words w into finite structures $\langle cn, F, +\rangle$. Our purpose being to get back to finite structures in which the words w are encoded by their naturally associated predicate X, we make explicit, in this section, the link between F and X.

Before stating the general result, we give the main ideas with an example. Starting with a given word of $\{1,2\}^*$, we successively compute its associated function f_w and the flattening F of f_w. Then we study the connection between F and the predicate X above mentioned. In particular, we will pass from dyadic noyations of integers to binary ones.

1.5.1 An example

Given two integers k and c, each word $w \in \{1,2\}^*$ defines a function $f_w : m \to m$, where m and f_w depend on $n = lg(w)$ and k. The problem is to determine successively:

- the function f_w;

- its flattening F on cn;

– the link between F and the predicate $X \subseteq n$ naturally associated to w, *i.e.* defined by : $\forall i < n$, $X(i)$ iff the i^{th} bit of w is 2.

Let us assume, for instance, that $k = 3$, $c = 4$, and w is the word:

w=112111221222212121211

Then we have, with the notations of 1.2:
$n = lg(w) = 21$, hence $l = \lceil \frac{\log(n+1)}{k} \rceil = 2$; $m = \lceil \frac{n}{l} \rceil = 11$; $L = \lfloor \log(m) + 1 \rfloor = 4$.

Computing f_w. The reading decomposition of w can then be written :

$$w = w_0 ^\frown w_1 ^\frown \ldots ^\frown w_{10}$$

with:

$$\begin{array}{llllll} w_0 = 11\,; & w_1 = 21\,; & w_2 = 11\,; & w_3 = 22\,; & w_4 = 12\,; & w_5 = 22\,; \\ w_6 = 21\,; & w_7 = 21\,; & w_8 = 21\,; & w_9 = 21\,; & w_{10} = 1\,. & \end{array}$$

Denoting w_i^d the integer whose reverse dyadic representation is w_i, we have :

$$\begin{array}{llllll} w_0^d = 3\,; & w_1^d = 4\,; & w_2^d = 3\,; & w_3^d = 6\,;\ v & w_4^d = 5\,; & w_5^d = 6\,; \\ w_6^d = 4\,; & w_7^d = 4\,; & w_8^d = 4\,; & w_9^d = 4\,; & w_{10}^d = 1\,. & \end{array}$$

And the function $f_w : m \to m$ defined by: $\forall i < m : f_w(i) = w_i^d$, may be represented by the following array:

i	0	1	2	3	4	5	6	7	8	9	10
$f_w(i)$	3	4	3	6	5	6	4	4	4	4	1

Computing the flattening F of f_w. Let us now describe the construction of the flattening of f_w on $cn = 4 \times 21 = 84$.
The main work consists in computing$B_L(f_w(i))$ for each $i < m$. The word obtained by the $B_L(f_w(i))$'s concatenation will immediately result in F.
Since f_w takes its values in $\{1, 3, 4, 5, 6\}$, we have just to compute:

$$B_4(1) = 1000\ ;\ B_4(3) = 1100\ ;\ B_4(4) = 0010\ ;\ B_4(5) = 1010\ ;\ B_4(6) = 0110.$$

From which we deduce for instance :

– $B_L(f_w(0)) = B_L(f_w(9)) = 1100$;

– $B_L(f_w(1)) = B_L(f_w(6)) = B_L(f_w(7)) = B_L(f_w(8)) = B_L(f_w(9)) = 0010$;

The word of $\{0, 1\}^{cn}$ encoding f_w is therefore the concatenation of the words $B_L(f_w(i))$, in the increasing order of the i's, padded on its right with as many 0's as

necessary to obtain a word of length $cn = 84$. Let us call w' this word.
We have:

$$w' = 1100\,0010\,1100\,0110\,1010\,0110\,0010\,0010\,0010\,0010\,1000{}^\frown 0^{40}.$$

Hence, F is the subset of $cn = 84$ given as follows:

$$\boldsymbol{F = \{0, 1, 6, 8, 9, 13, 14, 16, 18, 21, 22, 26, 30, 34, 38, 40\}.}$$

Connection between F and X. The predicate $X \subseteq n$ naturally associated with w is defined by:

$$X(i) \iff \textit{the } i^{th} \textit{ bit of } w \textit{ is } 2.$$

The connection between F and X is quite determined by the connection between w and w'.
Let us write these two words in the following manner:

w	11	21	11	22	12	22	21	21	21	21	1	
w'	1100	0011	1100	0110	1010	0110	0010	0010	0010	0010	1000	0^{40}

We easily see that we pass from w to w' in computing, for each $i < m$, the word $B_L(w_i^d)$.
All in all, our problem consists in the translation from the reverse dyadic representation of an integer to its reverse binary representation of fixed length L. A simple finite transducer automata can realize this translation. Rather than describing it, we just give here the result of its computing:

for every positive integer x, let us denote $D(x)$ (resp.$B(x)$) its reverse dyadic (resp. binary) representation.
It is easy to show that $lg(B(x)) = lg(D(x))$ or $lg(B(x)) = lg(D(x)) + 1$.
So for $x \neq 0$, possibly padding $B(x)$ with one 0, we can write:

$$D(x) = d_0 d_1 \ldots d_q \in \{1,2\}^*$$
$$B(x) = b_0 b_1 \ldots b_q b_{q+1} \in \{0,1\}^*$$

Then we have for every $i \leq q+1$:

$$(b_i = 1) \Leftrightarrow \left(\begin{array}{c} d_0 = d_1 = \ldots = d_i = 1 \\ \text{or} \\ (i = q+1 \text{ or } d_i = 2) \text{ and there exists } j < i \text{ s.t. } d_j = 2 \end{array} \right)$$

Examples:
For $x = 15$: $D(x) = 1111$
$B(x) = 11110$
For $x = 35$: $D(x) = 11211$
$B(x) = 110001$

For $x = 51$: $D(x) = 11212$
$B(x) = 110011$
To translate $D(x)$ to the reverse binary representation of *fixed length* L of x, we just have to pad $B(x)$ on its right with as many 0's as necessary.
This result allows to express the logical connection between F and X, in the general case, in a straightforward manner.

1.5.2 Connection between F and X: the general case

Let w be a word of $\{1,2\}^n$. Let us denote

$$w = d_0^0 \dots d_0^{l-1} d_1^0 \dots d_1^{l-1} \dots d_{m-2}^0 \dots d_{m-2}^{l-1} d_{m-1}^0 \dots d_{m-1}^{s-1},$$

its reading decomposition into m words $w_i = d_i^0 \dots d_i^{l-1}$ for $i < m-2$ and $w_{m-1} = d_{m-1}^0 \dots d_{m-1}^{s-1}$ $(0 < s \leq l)$. The function $f_w : m \to m$ is then defined by:

$$\forall i < m,\ f_w(i) = w_i^d = \begin{cases} \sum_{j=0}^{l-1} d_i^j 2^j & \text{if } i < m-1 \\ \sum_{j=0}^{s-1} d_{m-1}^j 2^j & \text{if } i = m-1 \end{cases}$$

Every integer w_i^d may be coded by its reverse binary representation of length L: $b_i^0 \dots b_i^{L-1}$. Hence f_w may be described by a binary word:

$$w' = b_0^0 \dots b_0^{L-1} \dots b_{m-1}^0 \dots b_{m-1}^{L-1} . 0^{cn-mL} \in \{0,1\}^{cn}.$$

Relationship between the words w, w' and the function f_w is therefore expressed: $\forall i < m$, $d_i^0 \dots d_i^{l-1}$ is the reverse dyadic representation of $f_w(i)$ (replace l by s for w_{m-1}) and $b_i^0 \dots b_i^{L-1}$ is the reverse binary representation of length L of $f_w(i)$. We deduce: $\forall x < m$, $\forall t < L$:

$$\begin{gathered} \{b_x^t = 1\} \\ \text{iff} \\ \left\{ \begin{array}{lll} & (x < m-1) \ \wedge & \left[\begin{array}{ll} & \{(t < l) \wedge (\forall t' \leq t)(d_x^{t'} = 1)\} \\ \vee & \{(t < l) \wedge (d_x^t = 2) \wedge (\exists t' < t)(d_x^{t'} = 2)\} \\ \vee & \{(t = l) \wedge (\exists t' < t)(d_x^{t'} = 2)\} \end{array} \right] \\ \vee & (x = m-1) \ \wedge & \text{[the same, replacing } l \text{ by } s] \end{array} \right\} \end{gathered} \tag{1.5}$$

With the choosen notations, the predicate X naturally coding w on n and the flattening F of f_w on cn are defined by:

$$\begin{aligned} &\forall x < m, \forall t < l : [n \models X(xl+t)] \Leftrightarrow [d_x^t = 2] \\ &\forall x < m, \forall t < l : [cn \models F(xL+t)] \Leftrightarrow [b_x^t = 2] \end{aligned}$$

The equivalence (1.5) can therefore be written: $\forall x < m$, $\forall t < l$:

$$F(xL+t)$$
$$\text{iff}$$
$$\left\{ \begin{array}{lll} & (x < m-1) & \wedge \left[\begin{array}{ll} & \{(t<l) \wedge (\forall t' \leq t) \neg X(xl+t')\} \\ \vee & \{(t<l) \wedge X(xl+t) \wedge (\exists t' < t) X(xl+t')\} \\ \vee & \{(t=l) \wedge (\exists t' < t) X(xl+t')\} \end{array} \right] \\ \vee & (x = m-1) & \wedge \qquad \text{[the same, replacing } l \text{ by } s] \end{array} \right\}$$

More precisely, if $\mathcal{X}$ denotes the trivial extension of X to cn (that is, $\mathcal{X} \subset cn$ and $\forall i < cn : \mathcal{X}(i) \Leftrightarrow [i < n \wedge X(i)]$), and if $\Phi(x,t,X)$ denotes the formula written between braces in the above equivalence, then we have, $\forall x < m$, $\forall t < L$:

$$[cn \models F(xL+t)] \Leftrightarrow [cn \models \Phi(x,t,X)] \, ,$$

and since $\mathcal{X}$ coincide with X on mL, which is the only part of cn concerned by the formula Φ:

$$[cn \models F(xL+t)] \Leftrightarrow [cn \models \Phi(x,t,\mathcal{X})].$$

Then, a unary predicate $F \subseteq cn$ is the flattening of f_w on cn iff $cn \models D(\mathcal{X},F)$, where $D(\mathcal{X},F)$ is the formula which makes explicit the connection between $\mathcal{X}$ and the flattening of f_w on cn.

$$i.e. \;\; D(\mathcal{X},F) \equiv \forall \alpha [F(\alpha) \leftrightarrow (\exists x < m)(\exists t < L)\{(\alpha = xL+t) \wedge \Phi(x,t,\mathcal{X})\}]$$

Since we have:

$$[w \in \mathcal{L}] \Leftrightarrow [\langle cn, F, + \rangle \models \exists G_1 \ldots \exists G_q \tilde{\Psi}(F, \overline{G})],$$
where F is the flattening of f_w on cn
(this is Corollary 1.10)

we eventually deduce:

$$[w \in \mathcal{L}] \Leftrightarrow [\langle cn, \mathcal{X}, + \rangle \models \exists F \exists G_1 \ldots \exists G_q (D(\mathcal{X},F) \wedge \tilde{\Psi}(F,\overline{G}))]$$
(since $D(\mathcal{X},F)$ characterizes the flattening F of f_w on cn.)

Whence, denoting $\hat{\Psi}(\mathcal{X},F,G_1,\ldots,G_q) \equiv D(\mathcal{X},F) \wedge \tilde{\Psi}(F,\overline{G})$:

$$[w \in \mathcal{L}] \Leftrightarrow [\langle cn, \mathcal{X}, + \rangle \models \exists F \exists G_1 \ldots \exists G_q \hat{\Psi}(\mathcal{X},F,G_1,\ldots,G_q)]$$

And at last, denoting G_0 the quantified second-order variable F, we have:

Lemma 1.17. *Let $\mathcal{L}$ be a language on $\Sigma = \{1,2\}$. If $\mathcal{L} \in NLIN$, then there exists a sentence $\hat{\Psi} \in FO(\forall^*\exists^*)$ on signature $\{\mathcal{X}, G_0, G_1, \ldots, G_q, +\}$, where $\mathcal{X}$ and the G_i's are unary predicate symbols, $+$ is a ternary predicate symbol, such that:*

$$[w \in \mathcal{L}] \Leftrightarrow [\langle cn, \mathcal{X}, + \rangle \models \exists G_0 \ldots \exists G_q \hat{\Psi}(\mathcal{X}, G_0, \ldots, G_q)]$$

when $\mathcal{X}$ is the trivial extension to cn of the predicate naturally coding w on n, and $+$ is the predefined addition on cn.

Remark. The assertion $\hat{\Psi} \in FO(\forall^*\exists^*)$ has not been justified above. Since $\hat{\Psi} \equiv D \wedge \tilde{\Psi}$, and since $\tilde{\Psi} \in FO(\forall^*\exists^*)$, it is sufficient to prove that $D \in FO(\forall^*\exists^*)$. But $D \equiv \forall\alpha[F(\alpha) \leftrightarrow (\exists x < m)(\exists t < L)\{(\alpha = xL+t) \wedge \Phi]$, can also be written: $D \equiv \forall\alpha[F(\alpha) \leftrightarrow (\forall x < m)(\forall t < L)\{(\alpha = xL+t) \rightarrow \Phi]$. Using this two writings and observing that:

- $\alpha = xL + t$ can be replaced by a formula in $FO(\forall^*\exists^*)$;
- Φ can be written with either a $\forall^*\exists^*$ or an $\exists^*\forall^*$-quantifier pattern,

the reader will easily convince himself that the result holds.

1.6 From cn to n

This part focuses on a technical result which is essentially due to Lynch (see [52] or [30]). Let us suppose that n and c are fixed in $\mathbb{N}$. Let us call *division of R on n* the set of unary predicates $R^* = \{R_0, \ldots, R_{c-1}\}$ defined on n, for all $i < c$, by:

$$\forall y < n \ : \ R_i(y) \Leftrightarrow R(in + y).$$

As each $x < cn$ may be written $x = in + y$, with $i < c$ and $y < n$, we have:

$$\begin{aligned} cn \models \forall x R(x) \Leftrightarrow \quad & cn \models (\forall y < n)[R(y) \wedge R(n+y) \wedge \ldots \wedge R((c-1)n + y)] \\ \Leftrightarrow \quad & n \models \forall y[R_0(y) \wedge \ldots \wedge R_{c-1}(y)] \end{aligned}$$

i.e.

$cn \models \forall x R(x) \Leftrightarrow \quad n \models \forall y \bigwedge_{i<c} R_i(y)$

In the same way, we have:

$cn \models \exists x R(x) \Leftrightarrow \ n \models \exists y \bigvee_{i<c} R_i(y)$

On this model, we can build, for each first-order sentence Φ on signature $\{R\}$, a first-order sentence Φ^* on signature $\{R^*\} = \{R_0, \ldots, R_{c-1}\}$, such that:

$$cn \models \Phi(R) \Leftrightarrow n \models \Phi^*(R^*).$$

This result can be generalized to any-arity predicates on cn: if R is an r-ary predicate on cn, we call "division"of R on n the set of r-ary predicates on n, $R^* = \{R_{i_1 \ldots i_r}, \ i_1, \ldots, i_r < c\}$, defined, for all $i_1, \ldots, i_r < c$, by:

$$\forall (y_1, \ldots, y_r) \in n^r \ : \ R_{i_1 \ldots i_r}(y_1, \ldots, y_r) \Leftrightarrow R(i_1 n + y_1, \ldots, i_r n + y_r).$$

We can still prove in this case that for each first-order sentence Φ on signature $\{R\}$, there exists a first-order sentence Φ^* on signature $\{R^*\}$, such that:

$$cn \models \Phi(R) \Leftrightarrow n \models \Phi^*(R^*).$$

In the same way, if Φ is a first-order sentence on signature $\{R_1, \ldots, R_s\}$, where each R_i is a r_i-ary predicate on cn, we build a first-order sentence Φ^* on signature $\{R_1^*, \ldots, R_s^*\}$, where each R_i^* is the division of R_i on n (and so, is a set of c^{r_i} r_i-ary predicates on n), such that:

$$cn \models \Phi(R_1, \ldots, R_s) \Leftrightarrow n \models \Phi^*(R_1^*, \ldots, R_s^*).$$

Finally, the bijectivity of the mapping which associates every r-ary predicate on cn to its division $R^* = \{R_{i_1 \ldots i_r},\ i_j < c\}$ on n, allows us to affirm:

Lemma 1.18. *Let Φ be a first-order sentence of signature $\{R_1, \ldots, R_s, P_1, \ldots, P_t\}$ where each R_i is a r_i-ary predicate symbol, and each P_i is a p_i-ary relation on cn. Then there exists Φ^*, first-order sentence on signature $\{R_1^*, \ldots, R_s^*, P_1^*, \ldots, P_t^*\}$, where each R_i^* is a set of c^{r_i} r_i-ary predicates symbols, and the P_i^*'s are the respective divisions of the P_i's, such that:*

$$\langle cn, P_1, \ldots, P_t\rangle \models \exists R_1 \ldots \exists R_s \Phi(R_1, \ldots, R_s, P_1, \ldots, P_t)$$
$$\textit{iff}$$
$$\langle n, P_1^*, \ldots, P_t^*\rangle \models \exists R_1^* \ldots \exists R_s^* \Phi^*(R_1^*, \ldots, R_s^*, P_1^*, \ldots, P_t^*)$$

Moreover, if $\Phi \in FO(\forall^*\exists^*)$, Φ^* can be required to be in $FO(\forall^*\exists^*)$.

Remark. The precise construction of Φ^* from Φ, as well as the proof of its membership in $FO(\forall^*\exists^*)$ when Φ has this quantifier pattern, have a tedious expression but are easy to achieve. They are left to the reader.

Theorem A immediately follows from Lemma 1.17, Lemma 1.18, and a simple study of the divisions of $+$ (addition on cn) and $\mathcal{X}$ (trivial extension of the predicate naturally coding w) on n:

Theorem A. *If a language $\mathcal{L} \subseteq \{1,2\}^*$ is in NLIN, then there exists a sentence $\Theta \in FO(\forall^*\exists^*)$ on signature $\{X, U_1, \ldots, U_s, +\}$, where X and the U_i's are unary predicate symbols, $+$ is a ternary predicate symbol, such that for every word $w \in \{1,2\}^*$:*

$$[w \in \mathcal{L}] \Leftrightarrow [\langle n, X, +\rangle \models \exists U_1 \ldots \exists U_s\, \Theta(X, \overline{U}, +)]$$

where $+$ is the predefined addition on n and X is the predicate naturally coding w on n.

Proof. Let Σ, $\mathcal{L}$, w, X and $+$ be like in the theorem hypothesis. Then, from lemma 1.17, we have:

$$[w \in \mathcal{L}] \Leftrightarrow [\langle cn, \mathcal{X}, +\rangle \models \exists G_0 \ldots \exists G_q \hat{\Psi}(\mathcal{X}, G_0, \ldots, G_q, +)]\ ,$$

where $\mathcal{X}$ is the trivial extension of X to cn. This becomes, with lemma 1.18:

$$[w \in \mathcal{L}] \Leftrightarrow [\langle n, \mathcal{X}^*, +^*\rangle \models \exists G_0^* \ldots \exists G_q^* (\hat{\Psi})^*(\mathcal{X}^*, G_0^*, \ldots, G_q^*, +^*)]\ ,$$

where $\mathcal{X}^*$ and $+^*$ are the respective divisions of $\mathcal{X}$ and $+$, and each G_i^* is a set of c unary predicate symbols. Replacing $\{G_0^*, \ldots, G_q^*\}$ by $\{U_1, \ldots, U_s\}$, with $s = (q+1)c$, and denoting Ψ^* instead of $(\hat{\Psi})^*$, we get:

$$[w \in \mathcal{L}] \Leftrightarrow [\langle n, \mathcal{X}^*, +^*\rangle \models \exists U_1 \ldots \exists U_s \Psi^*(\mathcal{X}^*, U_1, \ldots, U_s, +^*)]\ ,$$

We have just to write $+^*$ and $\mathcal{X}^*$ in a standard form to obtain the sought result.

Division of + on n. $+$ is the ternary predicate on cn defined by:

$$[cn \models +(x, y, z)] \Leftrightarrow [x + y = z \text{ in } cn].$$

$+^*$ is then the set $\{+^{ijk},\ i, j, k < c\}$ made up of c^3 ternary predicates on n, defined, for each $(i, j, k) \in c^3$, by:

$$\begin{array}{lll} n \models +^{ijk}(x, y, z) & \text{iff} & cn \models +(in + x, jn + y, kn + z) \\ & \text{iff} & (in + x) + (jn + y) = (kn + z) \text{ in } cn \\ & \text{iff} & \ \ n \models \ \ \{(i + j = k) \wedge (x + y = z)\} \vee \\ & & \qquad\quad \{(i + j + 1 = k) \wedge (x + y - z - 1 = n - 1)\} \end{array}$$

Let us call $\phi_{ijk}(x, y, z)$ this last formula, in which $+$ now denotes the addition on n. Replacing the assertions "$i + j = k$" and "$i + j + 1 = k$" by their truth value (which is perfectly defined for each triple (i, j, k)), we have thus built, for each formula $+^{ijk}(x, y, z)$, an equivalent formula $\phi_{ijk}(x, y, z)$ on n whose signature is the only predicate $+$, namely, the addition on n.

Division of $\mathcal{X}$ on n: $\mathcal{X}$ is the trivial extension of X, i.e. the unary predicate on cn defined by:

$$\forall x < cn \ \ \mathcal{X}(x) \Leftrightarrow [x < n \text{ and } X(x)].$$

So $\mathcal{X}^*$ may be denoted $\mathcal{X}^* = \{\mathcal{X}^0, \ldots, \mathcal{X}^{c-1}\}$, where each $\mathcal{X}^i$ is a unary predicate on n defined by:

$$\begin{array}{lll} n \models \mathcal{X}^i(x) & \text{iff} & cn \models \mathcal{X}(in + x) \\ & \text{iff} & in + x < n \text{ and } n \models X(in + x)] \\ & \text{iff} & i = 0 \text{ and } n \models X(x) \end{array}$$

In other words, $\mathcal{X}^0 = X$ and for every $i > 0$, $\mathcal{X}^i$ is the empty relation of n.

Let Θ be the first-order sentence obtained from Ψ^* as follows:

1. each subformula $+^{ijk}(x, y, z)$ of Ψ^* is replaced by the formula $\phi_{ijk}(x, y, z)$ described above;

2. each occurence of $\mathcal{X}^i(x)$ is replaced by $X(x)$ if $i = 0$, and by $\perp$ (the false) if $0 < i < c$.

Then Θ is the first-order sentence on the signature $\{X, U_1, \ldots, U_s, +\}$ announced in the theorem.

Conclusion and final remarks

We have proved that NLIN, and then many NP-problems (the 21 NP-complete problems of [48]), are definable in Lynch's logic, even with sentences in the class

$FO(\forall^*\exists^*)$. In our opinion, this means that the Lynch's purpose announced in [53], namely proving that a natural NP-problem is not expressible in that logic, is a considerably difficult problem, since it would imply that the concerned problem is strictly harder than each of the 21 above mentioned NP-complete problems.

A natural question is to ask what is exactly definable in Lynch's logic, or in this logic with sentences in $FO(\forall^*\exists^*)$. We cannot prove a converse of our main result, but we easily see that the problems so definable are in $\Sigma_3 TIME(n)$ (a $\Sigma_k TIME(n)$-problem is computable in linear time on an alternating Turing machine with k alternations between existential and universal steps, the first one beeing existential). This is not far from the known upper bound: NLIN $\subseteq \Sigma_2 TIME(n)$ (guess the entire linear-time bounded computation and check that it is correct; an incorrect computation is recognized within linear time on a nondeterministic Turing machine).

Our main result was proved by using the NLIN logical characterization - also proved in this paper - where the input is a unary structure. This result will be refined in the next chapter: a set of unary structures is in NLIN iff it is definable in existential second-order logic with unary functions, only one variable and a *conjonction of equalities* (*i.e. without* negation and disjonction); We will also give a similar exact characterization for each class $\mathrm{NTIME}_{RAM}(n^k)$.

Such logical normalizations are useful to prove strong completeness results, e.g. NLIN-completeness, and nonlinear lower bounds : e.g. problem RISA of [32] is NLIN-complete and consequently is not in DTIME(n) (see also [31, 34]).

Chapitre 2

Une Caractérisation Conjonctive des classes $\mathbf{NTIME}_{RAM}(n^k)$

Introduction

Dans notre présentation du lien entre complexité algorithmique et complexité descriptive, nous avons jusqu'à présent adopté le "point de vue des problèmes" : considérant un problème combinatoire naturel nous nous sommes demandé : *Est-il facile à résoudre ? Est-il simple à décrire ?* Nous proposons maintenant d'adopter le point de vue de la logique. Notre démarche consistera donc désormais à considérer une logique (*i.e.* une classe de formules), et à nous demander : *Est-elle syntaxiquement simple ? Est-elle sémantiquement riche ?* Essayons de préciser ces deux questions.

Simplicité syntaxique et richesse sémantique. La notion intuitive de *simplicité syntaxique* admet plusieurs acceptions. Pour rester informels, nous choisirons de dire qu'une logique $\mathcal{F}$ est syntaxiquement simple lorsque pour chaque formule Φ de $\mathcal{F}$ et pour chaque structure finie S, on peut répondre facilement à la question : S est-elle modèle de Φ?

Cette interprétation de la simplicité syntaxique peut être justifiée comme suit : si Φ est une formule de signature $\mathcal{T}$ et S une $\mathcal{T}$-structure finie, répondre à la question "$S \models \Phi$?", c'est regarder si les objets de S qui interprètent les symboles de $\mathcal{T}$ vérifient certaines contraintes. Or la nature de ces objets (arité, taille) et la complexité de ces contraintes dépendent précisément de la syntaxe de Φ. La complexité (algorithmique) de la question "$S \models \Phi$?" fournit donc une mesure raisonnable de la simplicité syntaxique de $\mathcal{F}$.

La notion de *richesse sémantique*, ou d'expressivité, s'impose plus naturellement. De manière toujours informelle, nous pouvons dire qu'une logique $\mathcal{F}$ est d'autant plus riche que la classe des problèmes qu'elle caractérise est vaste. Ainsi toute logique $\mathcal{F}'$ contenant $\mathcal{F}$ est plus expressive que $\mathcal{F}$. En particulier, si $\mathcal{F}'$ est construite à partir de $\mathcal{F}$ en enrichissant la syntaxe (*i.e.* les règles de production de formules) ou en élargissant la signature, alors $\mathcal{F}'$ est au moins aussi expressive que $\mathcal{F}$.

Revenons maintenant à notre problème initial. Avec les notions que nous venons d'introduire, l'assertion : *$\mathcal{F}$ caractérise exactement la classe de complexité $\mathcal{C}$* (que nous avons convenu d'abréger $\mathcal{F} = \mathcal{C}$), peut être comprise comme l'affirmation d'une forme "d'équilibre" entre deux qualités contradictoires de $\mathcal{F}$:

1. Sa *simplicité syntaxique*: les formules de $\mathcal{F}$ sont suffisament simples pour que leurs modèles soient identifiables par un algorithme dont la complexité n'excède pas $\mathcal{C}$ (c'est "l'inclusion" $\mathcal{F} \subseteq \mathcal{C}$) ;

2. Sa *richesse sémantique*: les formules de $\mathcal{F}$ sont assez expressives pour exprimer des problèmes aussi "compliqués" que ceux de $\mathcal{C}$ (c'est "l'inclusion" $\mathcal{C} \subseteq \mathcal{F}$).

Une fois l'égalité $\mathcal{F} = \mathcal{C}$ établie, une question se pose naturellement :

($\mathcal{Q}$) : *peut-on trouver une logique $\mathcal{F}'$ qui soit à la fois plus "simple" et aussi "riche" que $\mathcal{F}$?* Autrement dit, *existe-t-il une logique $\mathcal{F}'$ qui exprime les même choses que $\mathcal{F}$ plus simplement?*

Cette question conduit à la notion de *réduction d'une logique à une autre* que l'on pourrait aussi baptiser *normalisation.* Nous en donnons dans les lignes qui suivent une définition proche de celle utilisée dans [19].

La notion de réduction. Appelons *signature prédéfinie* tout ensemble de symboles logiques prédéfinis (*e.g.* l'ordre, le successeur). Si $\mathcal{T}$ est une signature quelconque et σ une signature prédéfinie, notons de plus $\langle S, \sigma\rangle$ l'expansion d'une $\mathcal{T}$-structure finie S au langage $\mathcal{T} \cup \sigma$. Soient maintenant $\mathcal{T}$ une signature quelconque, σ, σ' deux signatures prédéfinies et $\mathcal{F}$ (resp. $\mathcal{F}'$) une classe de formules de signature $(\mathcal{T}\cup\sigma)$ (resp. $(\mathcal{T}\cup\sigma')$).

DÉFINITION 2.1. On dit que $\mathcal{F}$ *se réduit à* $\mathcal{F}'$ si pour toute formule $\Phi \in \mathcal{F}$ il existe une formule $\Phi' \in \mathcal{F}'$ telle que, pour toute $\mathcal{T}$-structure S : $\langle S, \sigma\rangle \models \Phi$ ssi $\langle S, \sigma'\rangle \models \Phi'$.

La question ($\mathcal{Q}$) se reformule alors: *La logique $\mathcal{F}$ se réduit-elle à une logique $\mathcal{F}'$ syntaxiquement plus simple ?* Dans le cas où $\mathcal{F} = \mathcal{C}$, une réponse positive à cette question donne une nouvelle caractérisation exacte de C.

Ainsi la notion de réduction constitue-t-elle un outil pour optimiser des caractérisation logiques de classe de complexité par le biais de considérations purement logiques.

La réduction de la classe Σ^1_1 des formules du second ordre existentiel à la sous-classe $\Sigma^1_1\Pi^0_2$ des formules de Σ^1_1 dont le préfixe du premier ordre est de la forme $\forall^*\exists^*$, permet d'améliorer le Théorème de Fagin (NP $= \Sigma^1_1$, [20]) en NP $= \Sigma^1_1\Pi^0_2$.

La réduction plus récente de $\Sigma^1_1\Pi^0_2$ à $\Sigma^1_1\Pi^0_1(s)$ due à Leivant, [51], complète ce résultat : sur les structures successeurs, NP $= \Sigma^1_1\Pi^0_1(s)$.

Les réductions de NLIN. L'objet de ce chapitre est d'utiliser ce type de technique pour optimiser la caractérisation logique de NLIN établie par Grandjean. Rappelons que si σ désigne un ensemble de symboles prédéfini et $\mathcal{G}(\sigma)$ la classe des formules $\exists\overline{g}\forall x\phi(x)$ (où $\overline{g}$ est une suite de symboles de constante et de fonction unaires et ϕ une formule sans quantificateur dont la signature contient σ), la caractérisation en question identifie les classes de structures finies reconnaissables sur RAM en temps non déterministe linéaire, aux classes de modèles de formules de $\mathcal{G}(<)$ (voir section 1.2).

Cette caractérisation pourrait sembler minimale : le préfixe du second ordre ne comporte aucune alternance de quantificateurs ; le préfixe du premier ordre est réduit a un unique quantificateur $\forall$; la signature prédéfinie est restreinte à l'unique symbole $<$, d'un usage naturel dans ce type de caractérisations.

On peut pourtant l'améliorer : Grandjean montre dans [33] que l'ordre n'ajoute en fait aucun pouvoir d'expression à la logique $\mathcal{G}$. Autrement dit, $\mathcal{G}(<)$ se réduit à sa sous classe $\mathcal{G}(\emptyset)$, ce qui permet de caractériser NLIN par la nouvelle "égalité" : NLIN $= \mathcal{G}(\emptyset)$.

Plus récemment, Durand s'est intéressé au préfixe du second ordre des formules exprimant les problèmes NLIN. Il prouve dans [14] la réduction de $\mathcal{G}(\emptyset)$ à la classe des formules du type $\exists B\Phi$, où Φ est une formule du premier ordre dont le préfixe est de la forme $\exists^*\forall\exists^*$, et B une relation binaire de degré extérieur borné. Cela donne lieu à une nouvelle caractérisation exacte de NLIN par cette classe de formules.

Pour notre part nous nous intéresserons ici à la matrice des formules de $\mathcal{G}(<)$. Nous montrerons que l'on ne restreint pas le pouvoir d'expression de ces formules en supposant que leur matrice ϕ s'écrit sous forme purement conjonctive, et ce même si l'on remplace la signature prédéfinie $\{<\}$ par l'ensemble, en apparence moins expressif, $\{0, s\}$ (où 0 est la constante attendue et s la fonction successeur).

L'idée de ce résultat est suggérée par un article de Grandjean ([32]) dans lequel l'auteur prouve la NLIN-complétude [1] d'un problème NP-complet naturel : RISA. Pour ce faire il établit ce que l'on pourrait appeler une "pseudo-réduction" de $\mathcal{G}(<)$ à une classe de formules purement conjonctives.

L'objectif de ce chapitre est de montrer que l'on peut transformer cette pseudo-réduction en une vraie réduction. Pour présenter ce résultat de façon plus formelle, il nous faut maintenant introduire de nouvelles définitions, qui seront d'un usage constant dans la suite.

DÉFINITION 2.2. Soit $\mathcal{T}$ une signature quelconque.

- On note $\mathcal{G}(\mathcal{T})$ la classe des formules de la forme : $\exists\lambda_1 \ldots \exists\lambda_p\forall x\phi(x)$, où les λ_i sont des symboles de fonction unaires ou de constante et $\phi(x)$ est une formule du

1. Un problème P est NLIN-complet s'il est dans NLIN et si tout problème de NLIN se réduit à P *via* une réduction calculable en temps linéaire *sur machine de Turing.*

premier ordre sans quantificateur de signature $\{\overline{\lambda}\} \cup \mathcal{T}$, *avec une seule variable* x.

- On note $\mathcal{G}^+(\mathcal{T})$ (resp. $\mathcal{G}^+_\wedge(\mathcal{T})$) la sous-classe des formules de $\mathcal{G}(\mathcal{T})$ qui ne comportent pas de négation (resp. ni négation ni disjonction).
- On étend ces notations aux formules logiques équivalentes à une formule de la classe mentionnée.
- On note $\mathcal{G}$, $\mathcal{G}^+$ et $\mathcal{G}^+_\wedge$ les classes précédentes lorqu'on ne tient pas à préciser les signatures sous-jacentes.

La caractérisation de NLIN à laquelle nous nous réfèrerons désormais, ainsi que celle que nous établirons, porte sur des structures finies dont la signature ne comporte que des symboles de constante et de fonction unaires. Cela ne réduit pas la portée de ces résultats, puisque toute structure finie (notament un graphe) peut se coder de façon naturelle en une structure de ce type. Mais cela justifie l'introduction des conventions suivantes :

DÉFINITION 2.3. Une signature ne comportant que des symboles de fonction unaires et des symboles de constante est appelée *signature unaire*. On appelle alors *structure unaire* toute structure de domaine $n \in \mathbb{N}^*$ (*i.e.* de domaine $\{0, \dots, n-1\}$) et de signature unaire.

La réduction que nous prouvons utilise les seuls symboles prédéfinis 0 et s. Le premier est le symbole de constante interprété sur chaque domaine par l'élément minimal ; le second est le symbole de fonction unaire interprété par la fonction successeur *non cyclique*, c'est-à-dire :

DÉFINITION 2.4. On appelle *successeur (non cyclique)* la fonction s définie sur tout domaine fini $n = \{0, \dots, n-1\}$ par : $\forall x < n-1$, $s(x) = x+1$ et $s(n-1) = n-1$.

La pseudo-réduction de [32] peut alors se décrire comme suit : pour chaque structure unaire S et chaque entier $c > 0$, Grandjean construit une nouvelle structure cS en "multipliant" le domaine n de S par c puis en codant de manière naturelle les constantes et fonctions de S sur le nouveau domaine cn. Par ailleurs il définit, pour chaque entier $c > 0$, une signature prédéfinie σ_c contenant $c + 7$ symboles arithmétiques que nous ne détaillerons pas. Puis il prouve :

Théorème (Grandjean, [32]). *Soit $\mathcal{T}$ une signature unaire. Pour toute formule $\Phi \in \mathcal{G}(\mathcal{T}, <)$, il existe un entier $c > 0$ et une formule $\Phi' \in \mathcal{G}^+_\wedge(\mathcal{T}, \sigma_c)$ tels que pour toute $\mathcal{T}$-structure unaire on ait : $\langle S, < \rangle \models \Phi$ ssi $\langle cS, \sigma_c \rangle \models \Phi'$,* Nous ne détaillerons pas d'avantage ce résultat. Retenons simplement qu'il établit une forme de traduction des formules de $\mathcal{G}$ en formules de $\mathcal{G}^+_\wedge$, mais que cette traduction se fait au prix :

1. d'un élargissement (linéaire) du domaine des structures finies considérées ;

2. de l'utilisation d'une signature prédéfinie *ad hoc* dont le contenu exact dépend de la formule $\Phi \in \mathcal{G}$ à traduire.

Du point de vue de la complexité, ces deux conditions sont anecdotiques : le passage de la structure S à cS et les objets arithmétiques de σ_c sont calculable en temps linéaire. Les structures $\langle S, <\rangle$ et $\langle cS, \sigma_c\rangle$ sont donc algorithmiquement équivalentes, et seul ce point importe quant au résultat visé dans [32].

Du point de vue logique par contre, 1 et 2 interdisent de parler de réduction au sens défini plus haut.

Nous étudions dans ce chapitre la possibilité de surmonter ces deux difficultés. Il s'avère en définitive que l'augmentation artificielle des domaines n'est pas nécessaire à la traduction susdite. Pas plus que ne l'est le recours aux signatures σ_c, auxquelles on peut substituer les seuls symboles 0 et s. Par conséquent, la classe $\mathcal{G}(<)$ se réduit (sur les structures unaires) à la classe $\mathcal{G}^+_\wedge(0, s)$. Cette réduction constitue le résultat principal de ce chapitre. Nous l'énoncerons :

Théorème 2.11. *Soit $\mathcal{T}$ une signature unaire. Pour toute $\Psi \in \mathcal{G}(\mathcal{T}, 0, s, <)$, il existe $\Psi^+_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour toute $\mathcal{T}$-structure unaire S de domaine $M > 9$ on ait : $\langle S, 0, s, <\rangle \models \Psi$ ssi $\langle S, 0, s\rangle \models \Psi^+_\wedge$.*

- La preuve de ce théorème occupe les sections 2.4, 2.5 et 2.6 du chapitre.

- Comme premier corollaire de ce résultat, nous obtenons une nouvelle caractérisation de NLIN : NLIN $= \mathcal{G}^+_\wedge(0, s)$ (section 2.7). Dans cette même section, nous généralisons cette caractérisation aux classes $\mathrm{NTIME}_{RAM}(n^k)$. On pourra comparer ce résultat à celui de Leivant ([51]) évoqué plus haut. De plus notre généralisation donne lieu à une hiérarchie stricte non triviale de classes de définissabilité.

- Dans la section 2.8, nous généralisons le théorème 2.11 en prouvant la définissabilité dans $\mathcal{G}^+_\wedge(0, s)$ des relations d'addition, de multiplication et d'exponentielle et, par conséquent, la réduction de $\mathcal{G}(0, s, <, +, \times, \exp)$ à $\mathcal{G}^+_\wedge(0, s)$.

- La conclusion du chapitre nous permettra finalement de revenir sur la portée et le sens de ces résultats.

2.1 Plan détaillé du chapitre

Le caractère technique de ce chapitre nous conduit à en présenter un plan détaillé. Nous donnons dans les lignes qui suivent le contenu de chaque section, en énonçant explicitement les principaux théorèmes, et en donnant éventuellement une idée de leur preuve.

Section 2.2. Cette section constitue un préliminaire technique. On prouve que toute formule de $\mathcal{G}^+(\mathcal{T}, 0, s)$ peut s'écrire comme une conjonction de 3-clauses, les "atomes" des clauses étant tous de la forme $uv(t) = w(t)$, où u, v sont des symboles de fonction et t un terme de profondeur 0 (*i.e.* une variable ou une constante). Ce résultat facile est essentiellement destiné à alléger la présentation de notre developpement.

Section 2.3. On prouve ici la définissabilité dans $\mathcal{G}^+_\wedge(0, s)$ d'un ensemble d'opérations et constantes arithmétiques. La possibilité d'utiliser cet ensemble sans "sortir" de $\mathcal{G}^+_\wedge(0, s)$, sera d'un usage constant dans la suite.

Les sections 2.4 à 2.6 sont consacrées à la réduction de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$, c'est à dire à ce que nous pourrions appeler la *supression de la disjonction.*

Section 2.4. C'est une partie clé de notre développement : à défaut de la réduction visée, entre $\mathcal{G}^+(\mathcal{T}, 0, s)$ et $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$, on établit une "pseudo-réduction" entre ces deux classes, rendue possible par un élargissement du domaine des structures considérées. Plus précisément, toute fonction $f : n \to n$ étant codée par une fonction unaire $f^{(3)}$ de domaine $3n + \ell$ (où ℓ est un entier quelconque < 3) [2] , on montre :

Proposition 2.7. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires. Pour toute $\Psi \in \mathcal{G}^+(\mathcal{T}, s, 0)$, il existe une formule $\Psi_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, s, 0)$ telle que : pour tous $n \in I\!N^*$ et $\ell < 3$, et pour tout p-uplet $(f_1, \ldots, f_p)$ de fonctions unaires sur n : $\langle n, \overline{f}, s, 0\rangle \models \Psi$ ssi $\langle 3n + \ell, \overline{f^{(3)}}, s, 0\rangle \models \Psi_\wedge$.*

On verra qu'en fait l'élargissement de n a un domaine M ayant *précisément* la forme $3n + \ell$ n'a que peu d'incidence sur la preuve de ce résultat. L'essentiel ici est que le domaine $M = 3n + \ell$ considéré contienne $3n$.

Idée de preuve. Le principe qui sous-tend cette preuve est inspiré du lemme 5.2 de [32]. En voici les grandes lignes : considérons un domaine $n = \{0, \ldots, n-1\}$ et trois propriétés $\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2$ des éléments de n (*i.e.* trois parties de n). À chacune des $\mathcal{E}_i$ $(i < 3)$, on associe une propriété $\mathcal{E}_i^{(3)}$ des éléments de $3n$, de telle sorte que : (a) $\mathcal{E}_i^{(3)}$ est vraie d'un entier de $3n$ divisible par 3 si et seulement si $\mathcal{E}_i$ est vraie du quotient modulo 3 de cet entier ; (b) $\mathcal{E}_i^{(3)}$ est vraie de tous les entiers de $3n$ non-divisibles par 3. Ce que l'on peut écrire :

(a) $\forall x < n$, $\mathcal{E}_i^{(3)}(3x)$ ssi $\mathcal{E}_i(x)$;

2. En toute rigueur $f^{(3)}$ aurait du être notée, par exemple, $f_\ell^{(3)}$, puisque cette fonction dépend de ℓ. Dans les faits, le contexte permettra toujours d'identifier l'indice ℓ auquel on se réfère.

(b) $\forall x < n$, $\mathcal{E}_i^{(3)}(3x+1)$ et $\mathcal{E}_i^{(3)}(3x+2)$.

Convenons de plus d'appeler *permutation 3-stable* de $3n$ toute permutation P de $3n$ qui préserve le quotient modulo 3 (et donc telle que $\forall x < n$, $\{P(3x), P(3x+1), P(3x+2)\} = \{3x, 3x+1, 3x+2\}$).

Le lecteur se convaincra alors facilement du fait suivant :

L'assertion $\forall x : \mathcal{E}_0(x) \vee \mathcal{E}_1(x) \vee \mathcal{E}_2(x)$ est vraie dans n si et seulement si il existe une permutation 3-stable P de $3n$ telle que l'assertion $(\forall x < n) : \mathcal{E}_0^{(3)}(P(3x)) \wedge \mathcal{E}_1^{(3)}(P(3x+1)) \wedge \mathcal{E}_2^{(3)}(P(3x+2))$ est vraie dans $3n$.

Autrement dit :

$$\begin{array}{l} n \models \forall x \; : \; \mathcal{E}_0(x) \vee \mathcal{E}_1(x) \vee \mathcal{E}_2(x) \\ \text{ssi} \\ 3n \models \exists P \; : \; P \text{ bijective } \wedge \; P \text{ 3-stable } \wedge \\ \qquad\qquad (\forall x < n) \; : \; \bigwedge_{i<3} \mathcal{E}_i^{(3)}(P(3x+i)) \end{array}$$

On voit bien ainsi en quoi l'élargissement du domaine permet la "suppression" de la disjonction. On pressent de plus le rôle que jouera l'arithmétique unaire définie dans la section 2.3 pour exprimer les termes du type $3x+i$ et la 3-stabilité des permutations.

On voudrait éviter ce codage des structures de taille n en structures de taille $3n$, mais il ne semble pas possible de "simuler" la disjonction sans solliciter d'avantage de place. Néanmoins, on pourra contourner cette difficulté en travaillant "en amont" les formules de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à traduire.

Section 2.5. On cherche ici à compenser l'extension des domaines nécéssaire à la suppression de la disjonction en se ramenant préalablement à des structures dont le domaine a été restreint. Pour ce faire, on code chaque fonction unaire f de domaine M (M entier quelconque) en un tuple $f^{1/3}$ de fonctions unaires sur $\lfloor M/3 \rfloor$ et on établit :

Proposition 2.9. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires et l un symbole de constante. Soit $\Psi \in \mathcal{G}^+(\mathcal{T}, 0, s)$. Alors il existe $\Psi^{\frac{1}{3}} \in \mathcal{G}^+_\wedge(\mathcal{T}^{1/3}, 0, s, l)$ telle que pour tout $n \geq 3$, tout $\ell < 3$ et tout p-uplet $(f_1, \ldots, f_p)$ de fonctions unaires sur $3n+\ell$: $\langle 3n+\ell, \overline{f}, 0, s\rangle \models \Psi$ ssi $\langle n, \overline{f}^{1/3}, 0, s, \ell\rangle \models \Psi^{\frac{1}{3}}$. (Où l est interprété par la constante ℓ, et $\overline{f}^{1/3}$ désigne $(f_1^{1/3}, \ldots, f_p^{1/3})$).*

Idée de preuve. Soit $n \in \mathbb{N}^*$. Une fonction $f : 4n \to 4n$ se code naturellement en un 8-uplet $(q_f^0, \ldots, q_f^3, r_f^0, \ldots, r_f^3)$ de fonctions unaires sur de n dans n satisfaisant : $\forall i \leq 3$, $\forall a < n$, $f(in+a) = nq_f^i(a) + r_f^i(a)$. Si f et g sont deux fonctions unaires sur $4n$, l'assertion $4n \models \forall x : f(x) = g(x)$ se traduit alors facilement :

$$n \models \forall a \bigwedge_{i \leq 3} (q_f^i(a) = q_g^i(a) \wedge r_f^i(a) = r_g^i(a)).$$

La preuve de la proposition 2.9 étend cette idée à la traduction d'une formule quelconque de $\mathcal{G}^+(\mathcal{T}, 0, s)$ en veillant à surmonter deux difficultés :

1. dans le cas qui nous occupe, chaque fonction initiale f est définie sur une partie stricte de $4n$ (à savoir $3n+\ell$), ce qui transforme *a priori* les fonctions q_f^3 et r_f^3 évoquées plus haut en fonctions partielles sur n (puisque $f(3n+a)$ n'est alors défini que pour $a < \ell$) ;

2. les formules de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à traduire comportent des égalités du type $fg(x) = h(x)$, et cette introduction de la composition des fonctions complique sensiblement le travail de traduction.

Section 2.6. La section 2.6 fait la synthèse des deux sections précédentes. En rapprochant les propositions 2.7 et 2.9, on prouve la réduction de $\mathcal{G}^+(0, s)$ à $\mathcal{G}^+_\wedge(0, s)$. C'est à dire :

Théorème 2.10. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires. Pour toute $\Psi \in \mathcal{G}^+(\mathcal{T}, 0, s)$, il existe $\Psi_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour toute $\mathcal{T}$-structure unaire $S = \langle M, \overline{f}\rangle$ de domaine $M > 9$, $\langle S, 0, s\rangle \models \Psi$ ssi $\langle S, 0, s\rangle \models \Psi_\wedge$.*

Idée de preuve. Supposons pour simplifier que le p-uplet $\overline{f}$ est réduit à un singleton $\{f\}$. On déduit facilement des propositions 2.7 et 2.9 l'existence de deux formules $\Psi^{1/3} \in \mathcal{G}^+(\mathcal{T}, 0, s)$ et $\Psi^{\frac{3}{3}} \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telles que :

$$\begin{gathered}\langle M, f, 0, s\rangle \models \Psi \\ \text{ssi} \\ \langle \lfloor M/3 \rfloor, f^{1/3}, 0, s, \ell\rangle \models \Psi^{1/3} \text{ (prop. 2.9)} \\ \text{ssi} \\ \langle M, (f^{1/3})^{(3)}, 0, s, \ell^{(3)}\rangle \models \Psi^{3/3} \text{ (prop. 2.7)},\end{gathered}$$

où $\ell = M \mathit{mod}\ 3$, $(f^{1/3})^{(3)}$ (resp. $\ell^{(3)}$) est le tuple de fonctions (resp. la fonction) qui code $f^{1/3}$ (resp. ℓ) sur M.

On montre alors la définissabilité dans $\langle M, f, 0, s\rangle$ des fonctions de $(f^{1/3})^{(3)}$ et de la fonction $\ell^{(3)}$ par des formules de $\mathcal{G}^+_\wedge(f, 0, s)$, ce qui permet de conclure la preuve.

Le théorème principal de ce chapitre (Théorème 2.11 : réduction de $\mathcal{G}(\mathcal{T}, 0, s, <)$ à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$), déjà mentionné dans l'introduction, s'obtient facilement comme corollaire du résultat précédent.

Section 2.7. Nous pouvons désormais établir la nouvelle caractérisation de NLIN annoncée dans l'introduction. Elle se déduit sans effort de la caractérisation déjà prouvée par Grandjean, rappelée dans le Chapitre 1 (Théorème 1.4), et du résultat principal mentionné ci-dessus. On obtient donc :

Théorème 2.13. *Soit $\mathcal{S}$ un ensemble de structures unaires de signature $\mathcal{T}$. Alors $\mathcal{S}$ est dans NLIN si et seulement si il existe une formule $\Psi \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour toute $\mathcal{T}$-structure finie S : $S \in \mathcal{S}$ ssi $\langle S, 0, s\rangle \models \Psi$.*

En corollaire, on établit une caractérisation logique des classes $\mathrm{NTIME}_{RAM}(n^k)$ (pour k fixé) en terme de classes des modèles de formules du second ordre de la forme $\exists F_1 \ldots \exists F_p \forall x_1 \ldots \forall x_k \phi$, où les F_i sont des symboles de fonction d'arité k, et ϕ est une formule du premier ordre sans quantificateur dont la signature contient 0 et s.

Grandjean a montré que cette dernière caractérisation des classes $\mathrm{NTIME}_{RAM}(n^k)$ donne lieu à un résultat de hiérarchie de classes de complexité descriptive. Nous conclurons cette section en donnant l'énoncé et la preuve de ce résultat.

Section 2.8. Pour clore ce chapitre, nous abordons la définissabilité de l'addition, la multiplication et l'exponentielle (respectivement notées A, M, E), dans un cadre un peu plus général que celui adopté jusqu'à présent. Nous montrons en effet que dans toute formule du second ordre dont la signature contient $\{A, M, E\}$, on peut remplacer ces trois opérations par le successeur et la constante 0 sans modifier son pouvoir d'expression, à condition d'ajouter un nombre fixé de fonctions unaires existentiellement quantifiées (Proposition 2.30). Ce résultat donne lieu à l'extension suivante du Théorème 2.11 :

Théorème 2.31. *Soient $\mathcal{T}$ une signature unaire et $\Phi \in \mathcal{G}(\mathcal{T}, 0, s, <, +, \times, \exp)$. Il existe $\Psi \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour tout $n > 9$:*

$$\langle n, \mathcal{T}, 0, s, <, +, \times, \exp\rangle \models \Phi \textit{ ssi } \langle n, \mathcal{T}, 0, s\rangle \models \Psi$$

2.2 Une forme normale dans $\mathcal{G}^+(\mathcal{T}, 0, s)$

Le coeur de ce chapitre est consacré à la construction, pour chaque formule Ψ de $\mathcal{G}^+(\mathcal{T}, 0, s)$, d'une formule $\Psi_\wedge$ de $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$, équivalente à Ψ sur les domaines finis. Cette construction est intimement liée à la structure de Ψ. Aussi, pour permettre un traitement plus homogène des preuves à venir, nous commençons par normaliser les formules de $\mathcal{G}^+(\mathcal{T}, 0, s)$. Nous montrons en particulier que l'on ne perd pas de généralité en considérant que la matrice d'une formule quelconque de cette classe est une conjonction de 3-clauses, dont les sous-formules atomiques sont toutes de la forme $uv(t) = w(t)$ (u, v symboles de fonction, t variable ou constante). Nous énoncerons cela précisément après cette remarque, qui sera d'un usage constant dans la suite :

Remarque. Les classes $\mathcal{G}$, $\mathcal{G}^+$ et $\mathcal{G}^+_\wedge$ sont fermées pour la conjonction et pour la quantification existentielle portant sur une signature unaire. Plus précisément, soit $\mathcal{T}$ une signature quelconque :

- la conjonction de deux formules de $\mathcal{G}(\mathcal{T})$ est dans $\mathcal{G}(\mathcal{T})$ (et de même pour $\mathcal{G}^+(\mathcal{T})$ et $\mathcal{G}^+_\wedge(\mathcal{T})$) ;
- si $\mathcal{T}'$ est une signature unaire et si $\Phi \in \mathcal{G}(\mathcal{T} \cup \mathcal{T}')$, alors $\exists \mathcal{T}' \Phi$ est dans $\mathcal{G}(\mathcal{T})$ (et de même pour $\mathcal{G}^+(\mathcal{T} \cup \mathcal{T}')$ et $\mathcal{G}^+_\wedge(\mathcal{T} \cup \mathcal{T}')$).

Lemme 2.1. *Soit $\mathcal{T} = F \cup C$ une signature unaire. Toute formule de $\mathcal{G}^+(\mathcal{T}, 0, s)$ équivaut à une formule de la forme :*

$$\exists g_1 \ldots \exists g_p \forall x \bigwedge_{j<s} \bigvee_{i<3} \{uv(t) = w(t)\},$$

où les g_i sont des symboles de fonction unaires, $u, v, w \in F \cup \{\overline{g}, s\}$ et $t \in C \cup \{0, x\}$.

Preuve. Une formule $\Phi \in \mathcal{G}^+(\mathcal{T}, 0, s)$ s'écrit en toute généralité :

$$\exists g_1 \ldots \exists g_k \exists c_1 \ldots \exists c_l, \forall x \bigwedge_{i<s} \bigvee_{j<r} \{u_m \ldots u_1(t) = v_q \ldots v_1(t')\},$$

où $u_i, v_j \in F \cup \{s, \overline{g}\}$ et $t, t' \in C \cup \{0, \overline{c}, x\}$.

Pour obtenir la forme normale annoncée dans le lemme, il faut successivement :

1. remplacer les constantes existentiellement quantifiées par des fonctions unaires ;

2. prouver que l'on peut supposer les termes t et t' identiques dans chaque formule atomique $u_m \ldots u_1(t) = v_q \ldots v_1(t')$;

3. remplacer les termes du type $u_m \ldots u_1(t)$ par des termes comportant au plus deux fonctions unaires ;

4. montrer que l'on peut supposer $r = 3$.

1. Constantes existentiellement quantifiées. On obtient clairement une formule équivalente à Φ en remplaçant chaque symbole de constante c dans le préfixe de Φ par un symbole de fonction unaire $\tilde{c}$, puis en substituant le terme $\tilde{c}(0)$ à toute occurence de c dans la matrice de Φ.

Notons Φ_1 la formule ainsi obtenue. On a $\Phi_1 \equiv \exists \overline{g} \forall x\, \phi_1$, où $\overline{g}$ est un tuple de symboles de fonctions unaires et ϕ_1 une combinaison booléenne de formules atomiques du type $u_m \ldots u_1(t) = v_q \ldots v_1(t')$ ($u_i, v_i \in F \cup \{\overline{g}\}$, $t, t' \in C \cup \{0, x\}$).

2. Égalisation de t et t'. Pour chaque symbole de constante non quatifié $c \in C \cup \{0\}$, on définit à nouveau un symbole de fonction unaire $\tilde{c}$. Soit Φ_2 la formule

$$\exists \overline{g} (\exists \tilde{c})_{c \in C \cup \{0\}} \forall x \bigwedge_{c \in C \cup \{0\}} (\tilde{c}(x) = \tilde{c}s(x) \wedge \tilde{c}(c) = c) \wedge \phi'_1,$$

où ϕ'_1 est la formule sans quantificateur obtenue à partir de ϕ_1 en remplaçant chaque symbole $c \in C \cup \{0\}$ par le terme $\tilde{c}(x)$.

Alors Φ_2 équivaut à Φ_1 et s'écrit $\exists \overline{g} \forall x\, \phi_2$, où $\overline{g}$ est un nouveau tuple de symboles de fonctions unaires et ϕ_2 une combinaison booléenne de formules atomiques du type $u_m \ldots u_1(t) = v_q \ldots v_1(t)$ ($u_i, v_i \in F \cup \{\overline{g}\}$, $t \in C \cup \{0, x\}$).

3. Écriture des termes avec 2 fonctions. Soit $u_m \dots u_1(t)$ un terme de Φ_2. Alors Φ_2 est équivalente à la formule :

$$\begin{gathered}\exists \overline{g}\ \exists \alpha_2 \dots \exists \alpha_p\ \forall x\ :\\ \alpha_2(x) = u_2 u_1(x) \wedge \alpha_3(x) = u_3 \alpha_2(x) \wedge \dots \wedge \alpha_m(x) = u_m \alpha_{m-1}(x)\ \wedge\ \phi'_2,\end{gathered}$$

où ϕ'_2 est la formule obtenue à partir de ϕ_2 en remplaçant chaque occurence de $u_m \dots u_1(t)$ par $\alpha_m(t)$.

Soit Φ_3 la formule obtenue en répétant cette opération pour tous les termes de Φ_2. Alors Φ_3 équivaut à Φ_2 et ses sous-formules atomiques sont de l'une des formes : $uv(t) = w(t)$, $uv(t) = t$, $v(t) = w(t)$, $v(t) = t$, avec $u, v, w \in F \cup \{s, \overline{g}\}$ et $t \in C \cup \{0, x\}$.

Par ailleurs, la fonction identité, *Id*, étant définissable par une formule de $\mathcal{G}^+_\wedge(\emptyset)$, on ne modifie pas les classes $\mathcal{G}(\mathcal{T})$, $\mathcal{G}^+(\mathcal{T})$ et $\mathcal{G}^+_\wedge(\mathcal{T})$ (à équivalence près) en ajoutant *Id* à la signature $\mathcal{T}$.

> *Dans toute la suite, nous supposerons donc toujours, sans le rappeler, que les signatures $\mathcal{T} = F \cup C$ utilisées contiennent le symbole prédéfini Id, interprété sur tout domaine par la fonction Identité.*

De ce fait, les quatre types de formules atomiques précédents se réduisent à : $uv(t) = w(t)$, avec $u, v, w \in F \cup \{s, \overline{g}\}$ et $t \in C \cup \{0, x\}$. Par conséquent, Φ_3 s'écrit sous la forme $\exists g_1 \dots \exists g_p \forall x \bigwedge_{j<s} \bigvee_{i<r} \{uv(t) = w(t)\}$, où les g_i sont des symboles de fonction unaires, $u, v, w \in F \cup \{\overline{g}, s\}$ et $t \in C \cup \{0, x\}$. (Les indices s, r, bien que notés de la même façon qu'au début de la preuve, n'ont en fait pas les mêmes valeurs.)

4. Passage aux 3*-clauses.* Il ne reste plus qu'à montrer que l'on peut supposer $r = 3$ dans la formule précédente. Cela découlera facilement du fait suivant :

Fait. *Soit $\mathcal{T}$ une signature quelconque. Toute formule de $\mathcal{G}^+(\mathcal{T})$ est équivalente à une formule prénexe de $\mathcal{G}^+(\mathcal{T})$ dont la matrice est une conjonction de* 3*-clauses.*

Preuve. Nous avons remarqué au début de cette section que la classe $\mathcal{G}^+(\mathcal{T})$ est fermée pour la conjonction et la quantification existentielle sur des signatures unaires. Cette propriété reste clairement vérifiée par la sous classe des formules de $\mathcal{G}^+(\mathcal{T})$ dont la matrice est une conjonction de 3-clauses. Comme par ailleurs $\mathcal{G}^+(\mathcal{T})$ est la fermeture par ces deux opération de la classe des formules du premier ordre disjonctives de signature $\mathcal{T}$, il suffit de prouver que la propriété cherchée est vraie des formules de cette classe.

Soit donc $\phi \equiv \forall x : A_0(x) \vee \dots \vee A_{r-1}(x)$, où les A_i sont des formules atomiques, une telle formule. En s'inspirant de la réduction bien connue de `SAT` à 3-`SAT` on obtient facilement l'équivalence :

$$\begin{array}{c} n \models \phi \\ \text{ssi} \\ n \models \exists U_0 \dots U_{r-2} \forall x : (A_0 \vee U_0(x) = 1) \wedge \\ \bigwedge_{i<r-3} (U_i(x) = 0 \vee A_{i+1} \vee U_{i+1}(x) = 1) \wedge \\ (U_{r-3}(x) = 0 \vee A_{r-2} \vee A_{r-1}) \end{array}$$

Ainsi ϕ est bien équivalente sur tout domaine fini à une formule du second ordre de la forme:

$$\exists U_0 \dots U_l \forall x \bigwedge_{j<p} \bigvee_{i<3} B_i^j(x),$$

où les U_i sont des symboles de fonction unaires et les B_i^j des formules atomiques. Les remarques précédentes permettent d'étendre cette normalisation à toute formule de $\mathcal{G}^+(\mathcal{T})$.

Ce qui conclut la preuve du lemme.

Dans la suite, il arrivera que la présence de symboles de constante dans les signatures utilisées complique la présentation des résultats. Dans le cas (presque systématique) où les symboles en question sont prédéfinis, on contourne facilement le problème en remplaçant ces symboles par des symboles de fonction unaires.

Plus précisément, à chaque symbole de constante prédéfini c on associe le symbole de fonction prédéfini $\tilde{c}$, interprété sur tout domaine par la fonction constante en c (*i.e.* en l'interprétation de c). Il est clair alors que toute formule de signature unaire $\mathcal{T}$ est équivalente à une formule de signature $\tilde{\mathcal{T}}$, où $\tilde{\mathcal{T}}$ est obtenue à partir de $\mathcal{T}$ en remplaçant chaque symbole de constante c par le symbole de fonction $\tilde{c}$. Inversement, toute formule de signature $\tilde{\mathcal{T}}$ est équivalente à une formule de signature $\mathcal{T}$.

Aussi, toute propriété stable par équivalence est vraie d'une sous-classe de $\mathcal{G}(\mathcal{T})$ si et seulement si elle vraie de la sous-classe de $\mathcal{G}(\tilde{\mathcal{T}})$ associée.

Donc dans l'étude de propriétés stables par équivalence, on ne perd pas de généralité en considérant les formules de $\mathcal{G}(\tilde{\mathcal{T}})$ plutôt que celles de $\mathcal{G}(\mathcal{T})$.

C'est ce que nous ferons à plusieurs reprises. De plus, étant donnée la "proximité" de ces deux classes, on les notera toutes deux $\mathcal{G}(\mathcal{T})$. On s'autorisera alors par exemple à dire qu'une formule de $\mathcal{G}(\mathcal{T})$ ($\mathcal{T}$ signature unaire) est de la forme $\exists \overline{g} \forall x \bigwedge \bigvee \{uv(x) = w(x)\}$, avec $u, v, w \in \{\overline{g}\} \cup \tilde{\mathcal{T}}$.

2.3 Définissabilité dans les classes $\mathcal{G}^+_\wedge(0, s)$ et $\mathcal{G}^+(0, s)$

Cette section prend le parti inverse de la précédente: dans la construction d'une formule $\Psi_\wedge$ de $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ à partir d'une formule Ψ de $\mathcal{G}^+(\mathcal{T}, 0, s)$, on est amené à coder différents objets (notament les éléments de $\mathcal{T}$) et, pour ce faire, à utiliser des opérations arithmétiques qui ne font *a priori* pas partie du langage autorisé.

En fait ces opération n'ont qu'un rôle "intermédiaire", et l'on peut les éliminer au cours de la preuve après chacune de leurs utilisations pour se convaincre que l'on n'est pas "sorti" du langage permis. Mais ces étapes d'élimination alourdiraient considérablement la preuve et en brouilleraient le fil directeur.

Nous faisons donc le choix de prouver, dans cette section et une fois pour toutes, que l'on peut enrichir la syntaxe des classes $\mathcal{G}_\wedge^+(0,s)$ et $\mathcal{G}^+(0,s)$ sans étendre leur pouvoir d'expression (*i.e.* sans sortir de ces classes à équivalence près). Plus précisément, nous allons prouver que si c est un symbole de constante, les fonctions unaires $x \mapsto cx$, $x \mapsto c-x$, $x \mapsto \lfloor \frac{x}{c} \rfloor$, $x \mapsto x \ mod \ \ c$, ainsi qu'un certain nombre d'autres constantes et fonctions unaires (ne dépendant pas forcément de c) sont définissables par des formules de $\mathcal{G}_\wedge^+(s,0,c)$ (où s est le successeur non cyclique).

Nous procèderons comme suit : pour chaque objet à caractériser dans $\mathcal{G}_\wedge^+(s,0,c)$, on donnera successivement sa notation, sa définition arithmétique et une formule de $\mathcal{F}_\wedge^+(0,s,c)$ le définissant dans toute structure finie $\langle M, 0, s, c\rangle$. Pour les fonctions, les notations introduites sont momentanées: elles visent dans un premier temps à insister sur leur nature "unaire" plutôt que sur leur signification arithmétique (par exemple, la multiplication par c est notée $x \mapsto M_c(x)$ plutôt que $x \mapsto cx$). Nous reviendrons à des notations plus intuitives dans un second temps.

Enfin, nous notons de la même façon le symbole de constante c et son interprétation dans une structure particulière.

1. Constantes $\mathbf{max}$ et $\mathbf{max-1}$:

   Définition. max et max -1 sont respectivement interprétés par l'élément maximum de M et par son prédécesseur.*i.e.*$\max = M-1$ et $\max -1 = M-2$.

   Définissabilité. Ces deux constantes sont caractérisées par les assertions suivantes: $\max -1 \neq \max$ et $s(\max -1) = s(\max) = \max$. C'est ce qu'exprime la formule de $\mathcal{F}_\wedge^+(0,s)$:

   $$\exists \mathcal{X} \ : \ \mathcal{X}(\max) = 1 \wedge \mathcal{X}(\max -1) = 0 \wedge s(\max -1) = s(\max) = \max .$$

2. Fonction $\mathbf{p}$:

   Définition. p est le prédécesseur associé au successeur s, c'est-à-dire la fonction unaire définie par : $\forall x < M : p(x) = x-1$ si $x \geq 0$, 0 sinon.

   Définissabilité. On traduit la caractérisation suivante de p : $\forall x \neq 0,\ sp(x) = x$ et $p(0) = 0$ et $p(M-1) = M-2$, par la formule de $FO_\wedge^+(0,s,\max,\max -1)$:

   $$\forall x : sps(x) = s(x) \wedge p(0) = 0 \wedge p(\max) = \max -1.$$

3. Fonction $\mathbf{s^c}$:

Définition. C'est la fonction obtenue par c itérations de s.

Définissabilité. s^c est l'unique fonction qui commute avec s et envoit 0 sur c. Elle est donc définie par la formule de $FO^+_\wedge(0, s, c)$

$$\forall x : s^c s(x) = ss^c(x) \wedge s^c(0) = c$$

4. Constante $\mathbf{max - c}$:

   Définition. C'est la constante interprétée sur M par $M - 1 - c$

   Définissabilité. $\max - c$ est le plus petit antécédent de max pour s^c. Elle est donc définie par la formule de $FO^+_\wedge(p, \max, \max - 1, s^c)$

$$s^c(\max - c) = \max \wedge s^c p(\max - c) = \max - 1.$$

5. Fonction $\mathbf{p^c}$:

   Définition. C'est la fonction obtenue par c itérations de p.

   Définissabilité. p^c est caractérisée par les assertions suivantes : $\forall x \geq c$, $s^c p^c(x) = x$ et $p^c(0) = 0$ et $p^c(M - 1) = M - 1 - c$. Elle est donc définie par la formule de $FO^+_\wedge(0, \max, s^c, \max - c)$

$$\forall x : s^c p^c s^c(x) = s^c(x) \wedge p^c(0) = 0 \wedge p^c(\max) = \max - c.$$

6. Fonction $\boldsymbol{\Pi_c}$:

   Définition. Π_c est la fonction unaire sur M appelée *projection de M sur c* et définie par : $\forall x < M$, $\Pi_c(x) = c - x$ si $x \leq c$, 0 sinon. Autrement dit on a : $\Pi_c(0) = c$, $\Pi_c(1) = c - 1$, ..., $\Pi_c(c) = 0$ et $\forall x > c$, $\Pi_c(x) = 0$.

   Définissabilité. Π_c est caractérisée par l'assertion $\forall x < M$, $\Pi_c(x) = p^x(c)$, ou encore par la récurrence : $\Pi_c(x + 1) = \Pi_c(x) - 1$ et $\Pi_c(0) = c$. Elle est donc définie par la formule de $FO^+_\wedge(0, s, c, p)$

$$\forall x : \Pi_c s(x) = p\Pi_c(x) \wedge \Pi_c(0) = c.$$

7. Fonctions $\boldsymbol{\mathcal{X}_c}$, pour $c \neq \max$ (*i.e.* pour un symbole de constante prédéfinie dont l'interprétation sur un domaine quelconque est toujours distincte de l'élément maximal du domaine) :

   Définition. $\mathcal{X}_c$ est la fonction caractéristique du singleton $\{c\}$ sur M. Autrement dit : $\forall x < M$, $\mathcal{X}_c(x) = 0$ si $x \neq c$ et $\mathcal{X}_c(c) = 1$.

   Définissabilité. La dernière assertion peut s'écrire : $\forall x < c : \mathcal{X}_c(x) = 0$ et $\forall x > c : \mathcal{X}_c(x) = 0$ et $\mathcal{X}(c) = 1$. Des égalités ensemblistes $[0, c[= \{\Pi_c s(x),\ x < M\}$ et $]c, M - 1] = \{s^c s(x),\ x < M\}$, on tire la définissabilité de $\mathcal{X}_c$ par la formule de $FO^+_\wedge(0, s, c, s^c, \Pi_c)$

$$\forall x : \mathcal{X}_c \Pi_c s(x) = \mathcal{X}_c s^c s(x) = 0 \wedge \mathcal{X}_c(c) = 1.$$

8. Fonctions $\boldsymbol{\mathcal{X}}_{\mathbf{max}}$:

   Définition. C'est la fonction caractéristique de $\{\max\}$.

   Définissabilité. $\mathcal{X}_{\max}$ est clairement définie par la formule de $FO_\wedge^+(0, s, \max, p)$
$$\forall x : \mathcal{X}_{\max} p(x) = 0 \wedge \mathcal{X}_{\max}(\max) = 1.$$

9. Fonctions $\boldsymbol{\mathcal{X}}_{\boldsymbol{>c}}$ pour $c \neq \max$:

   Définition. $\mathcal{X}_{>c}$ est la fonction caractéristique du segment $\{c+1, \ldots, M-1\}$ sur M. Autrement dit : $\forall x < M$, $\mathcal{X}_{>c}(x) = 1$ si $x > c$ et 0 sinon.

   Définissabilité. $\mathcal{X}_{>c}$ est définie par la formule de $FO_\wedge^+(0, s, s^c, \Pi_c)$
$$\forall x : \mathcal{X}_{>c}\Pi_c(x) = 0 \wedge \mathcal{X}_{>c} s^c s(x) = 1.$$

10. Fonction $\mathbf{M_c}$:

    Définition. M_c est le produit par c sur M, *i.e.*la fonction unaire sur M définie par : $\forall x < M$, $M_c(x) = cx$ si $cx < M$ et max sinon. Où encore : $\forall x < M$, $M_c(x) = s^{cx}(0)$.

    Définissabilité. M_c est caractérisée par la relation récurrente : $M_c(x+1) = M_c(x) \dot{+} c$ et $M_c(0) = 0$. Elle est donc définie par la formule de $FO_\wedge^+(0, s, s^c)$
$$\forall x : M_c s(x) = s^c M_c(x) \wedge M_c(0) = 0.$$

11. Fonction $\mathbf{Q_c}$:

    Définition. Q_c est la fonction qui à tout $x < M$ associe son quotient dans la division euclidienne par c. *i.e.*$\forall x < M$, $Q_c(x) = \lfloor x/c \rfloor$.

    Définissabilité. On peut caractériser Q_c de la façon suivante : $\forall x < c$, $Q_c(x) = 0$ et $\forall x \leq M - 1 - c$, $Q_c(x + c) = Q_c(x) + 1$. On en déduit la définissabilité de Q_c par la formule de $FO_\wedge^+(0, s, s^c, \Pi_c, \Pi_{\max - c})$
$$\forall x \ : \ Q_c \Pi_c s(x) = 0 \wedge Q_c s^c \Pi_{\max - c}(x) = s Q_c \Pi_{\max - c}(x).$$

12. Fonction $\mathbf{R_c}$:

    Définition. R_c est la fonction qui à tout $x < M$ associe son reste dans la division euclidienne par c. *i.e.*$\forall x < M$, $R_c(x) = x - Q_c(x)$.

    Définissabilité. On peut caractériser R_c de la façon suivante : $\forall x < c$, $R_c(x) = x$ et $\forall x \leq M - 1 - c$, $R_c(x + c) = R_c(x)$. On en déduit la définissabilité de R_c par la formule de $FO_\wedge^+(0, s, s^c, \Pi_c, \Pi_{\max - c})$
$$\forall x \ : \ R_c \Pi_c s(x) = x \wedge R_c s^c \Pi_{\max - c}(x) = R_c \Pi_{\max - c}(x).$$

13. Constante $\boldsymbol{\mathcal{Q}_c}$ pour $c \geq 2$:

    Définition. $\mathcal{Q}_c$ est le quotient de M modulo c, *i.e.*le quotient de $\max +1$ modulo c. Autrement dit, $\mathcal{Q}_c = \lfloor M/c \rfloor$.

    Définissabilité. $\mathcal{Q}_c$ est caractérisée par les assertions : $c(\mathcal{Q}_c - 1) + (c-2) < M - 1$ et $c\mathcal{Q}_c + (c-2) \geq M - 1$. Ce qui se dit sur M avec la formule de $FO^+_\wedge(0, s, \max, p, s^c, M_c)$:

$$\mathcal{X}_{\max} s^{c-2} M_c p(\mathcal{Q}_c) = 0 \wedge s^{c-2} M_c(\mathcal{Q}_c) = \max$$

14. Constante $\mathbf{max} - \boldsymbol{\mathcal{R}_c}$ pour $c \leq M$:

    Définition. $\max - \mathcal{R}_c$ est l'entier $c\lfloor M/c \rfloor - 1$.

    Définissabilité. Pour éviter les problèmes de dépassement, on écrit la définition précédente sous la forme $\max - \mathcal{R}_c = c(\lfloor M/c \rfloor - 1) + (c-1)$, ce qui permet de définir $\max - \mathcal{R}_c$ avec la formule de $FO^+_\wedge(0, s, p, s^c, M_c, \mathcal{Q}_c)$:

$$\max - \mathcal{R}_c = s^{c-1} M_c p(\mathcal{Q}_c)$$

15. Constante $\boldsymbol{\mathcal{R}_c}$ et fonction $\boldsymbol{s^{\mathcal{R}_c}}$:

    Définition. $\mathcal{R}_c$ est le reste de M modulo c (*i.e.*$\mathcal{R}_c = M - c\lfloor M/c \rfloor$) et $s^{\mathcal{R}_c}$ est l'itérée du successeur associée.

    Définissabilité. $\max - \mathcal{R}_c$ est caractérisée par l'égalité : $\mathcal{R}_c = \max - (\max - \mathcal{R}_c)$. La fonction $s^{\mathcal{R}_c}$ est donc la "plus petite" itérée du successeur qui envoit l'entier $(\max - \mathcal{R}_c)$ sur max. Ces remarques permettent de définir $\mathcal{R}_c$ et $s^{\mathcal{R}_c}$ avec la formule de $FO^+_\wedge(0, s, \max, \max -1, \max - \mathcal{R}_c)$:

$$\begin{aligned} \forall x \ : \ & s^{\mathcal{R}_c} s(x) = s s^{\mathcal{R}_c}(x) \ \wedge \\ & s^{\mathcal{R}_c}(\max - \mathcal{R}_c) = \max \ \wedge \\ & s^{\mathcal{R}_c} p(\max - \mathcal{R}_c) = \max -1 \ \wedge \\ & \mathcal{R}_c = s^{\mathcal{R}_c}(0) \end{aligned}$$

On note $Arith(c)$ la suite de fonctions et constantes énumérée ci-dessus (constante c comprise). Considérons maintenant, pour chaque formule $\Phi \in \mathcal{G}^+_\wedge(0, s, Arith(c))$, la formule équivalente de signature $\{0, s, c\}$ obtenue en quantifiant existentiellement les éléments de $Arith(c)$ puis en faisant suivre ce préfixe de la conjonction des formules vues plus haut et de Φ. On montre facilement que cette formule est dans $\mathcal{G}^+_\wedge(0, s, c)$ (à équivalence près). (Ceci tient au fait suivant : pour toute signature $\mathcal{T}$ contenant au moins un symbole de constante, $\mathcal{G}^+_\wedge(\mathcal{T})$ est fermée (à équivalence près) pour la quantification existentielle sur des fonctions unaires et sur des constantes). On en déduit : *toute formule* $\Phi \in \mathcal{G}^+_\wedge(0, s, Arith(c))$ *est équivalente à une formule de* $\mathcal{G}^+_\wedge(0, s, c)$.

Ceci ce généralise au cas de plusieurs constantes. Si pour des constantes $c_1, \ldots, c_p$ on note $Arith(c_1, \ldots, c_p)$ l'ensemble $\bigcup_i Arith(c_i)$, alors :

> *Toute formule* $\Phi \in \mathcal{G}^+_\wedge(0, s, Arith(c_1, \ldots, c_p))$ *est équivalente à une formule de* $\mathcal{G}^+_\wedge(0, s, c_1, \ldots, c_p)$.

Ceci étant, nous pouvons maintenant revenir à des notations plus intuitives. Nous noterons donc respectivement $x \mapsto x \dotplus c$, $x \mapsto c \cdot x$, $x \mapsto c \dot- x$, Q_c, R_c, les fonctions unaires définies sur M par :

- $x \dotplus c = x + c$ si $x + c < M$, $M - 1$ sinon ;
- $c \cdot x = cx$ si $cx < M$, $M - 1$ sinon ;
- $c \dot- x = c - x$ si $c - x \geq 0$, 0 sinon ;
- $Q_c(x) = \lfloor \frac{x}{c} \rfloor$;
- $R_c(x) = x \ mod \ c$.

Nous allons maintenant énoncer une dernière remarque portant sur les quantifications du premier ordre des formules de $\mathcal{G}_\wedge^+(0, s, \overline{c})$.

DÉFINITION 2.5. Soit h une fonction unaire sur un domaine *Dom*. On appelle *image de Dom par f* et on note $f\langle Dom \rangle$, l'ensemble $\{f(x), x \in Dom\}$.

Soient $\mathcal{T}$ une signature, $\Phi = \exists \overline{g} \forall x \phi(x)$ une formule de $\mathcal{G}_\wedge^+(\mathcal{T})$ et h un symbole de fonction unaire. Alors la formule $\tilde{\Phi} = \exists \overline{g} (\forall x \in h\langle Dom \rangle) \phi(x)$ équivaut à $\exists \overline{g} \forall x \phi(h(x))$, qui appartient à $\mathcal{G}_\wedge^+(\mathcal{T}, h)$.

En particulier, si $\mathcal{T}$ est un tuple de symboles de fonction unaires et si $c_1, \ldots, c_p$ sont des symboles de constante, alors pour toute formule $\Phi \in \mathcal{G}_\wedge^+(\mathcal{T}, 0, s, \overline{c})$ et toute fonction $h \in \mathcal{T} \cup \{\overline{g}\} \cup Arith(\overline{c})$, $\tilde{\Phi}$ est dans $\mathcal{G}_\wedge^+(\mathcal{T}, 0, s, \overline{c})$ (à équivalence près). Et ce résultat s'étend à toute fonction h définissable par une formule de $\mathcal{G}_\wedge^+(\mathcal{T}, s, 0, Arith(c_1, \ldots, c_p))$.

Finalement, on retiendra de tout ce qui précède :

Lemme 2.2. *Soient $\mathcal{T}$ un tuple de symboles de fonction unaires et $c_1, \ldots, c_p$ des symboles de constante. Soit $\tilde{\mathcal{F}}_\wedge^+(\mathcal{T}, 0, s, Arith(\overline{c}))$ la classe de formule obtenue à partir de $\mathcal{G}_\wedge^+(\mathcal{T}, 0, s, Arith(\overline{c}))$ en autorisant dans la syntaxe la quantification du premier ordre $\forall x \in h\langle Dom \rangle$, pour toute fonction h définissable par une formule de $\mathcal{G}_\wedge^+(\mathcal{T}, 0, s, Arith(\overline{c}))$. Alors toute formule de $\tilde{\mathcal{F}}_\wedge^+(\mathcal{T}, 0, s, Arith(\overline{c}))$ est équivalente à une formule de $\mathcal{G}_\wedge^+(\mathcal{T}, 0, s, \overline{c})$.*

Par conséquent, une classe de formule se réduit à $\mathcal{G}_\wedge^+(\mathcal{T}, 0, s, \overline{c})$ si et seulement si elle se réduit à $\tilde{\mathcal{F}}_\wedge^+(\mathcal{T}, 0, s, Arith(\overline{c}))$.

Remarque. Comme tous les résultats qui suivent sont vrais "à équivalence près", nous confondrons désormais les classes $\mathcal{G}_\wedge^+(\mathcal{T}, 0, s, \overline{c})$ et $\tilde{\mathcal{F}}_\wedge^+(\mathcal{T}, 0, s, Arith(\overline{c}))$.

Remarque. La possibilité d'utiliser des préfixes de la forme $\forall x \in h\langle Dom \rangle$ sera essentiellement utilisée pour introduire les quantifications $\forall x < c$ et $\forall x \in 3(c)$ pour une constante c donnée (où $3(c)$ dénote l'ensemble $\{3x, x < c\}$). Cet usage est légitime puisque pour tout domaine $n \in \mathbb{N}^*$ et toute constante $c \in n$, les ensembles c et $3(c)$ sont respectivement les images du domaine par $x \mapsto c \dot- x$ et $x \mapsto 3 \cdot (c \dot- x)$, fonctions définissables par des formules de $\mathcal{G}_\wedge^+(0, s, c)$.

Remarque. Il sera aussi fréquement fait usage de la fermeture de $\mathcal{G}^+_\wedge(\mathcal{T})$ ($\mathcal{T}$ signature quelconque) pour la conjonction et la quantification existentielle portant sur des symboles de fonction unaires.

Supposons maintenant qu'une constante c_0 soit définissable par une formule de $\mathcal{G}^+_\wedge(\mathcal{T})$ (*i.e.* qu'il existe une formule $\Phi(c) \in \mathcal{G}^+_\wedge(\mathcal{T}, c)$ telle que pour tout $n \in \mathbb{N}^*$ et tout $c < n$, $\langle n, c \rangle \models \Phi(c)$ ssi $c = c_0$). Alors toute formule $\Psi(c_0) \in \mathcal{G}^+_\wedge(\mathcal{T}, c_0)$ équivaut à $\exists c : \Phi(c) \wedge \Psi(c)$ qui appartient à $\mathcal{G}^+_\wedge(\mathcal{T})$. Ceci se généralise à plusieurs constantes $c_0, \ldots, c_q$ et donne lieu, grâce au lemme 2.2, au résultat suivant :

Lemme 2.3. *Soient $\mathcal{T}$ un tuple de symboles de fonction unaires et $c_1, \ldots, c_p$ des symboles de constante* définissables par des formules de $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$. *Alors toute formule de $\tilde{\mathcal{F}}^+_\wedge(\mathcal{T}, 0, s, Arith(\overline{c}))$ est équivalente à une formule de $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$.*

Par conséquent, une classe de formule se réduit à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ si et seulement si elle se réduit à $\tilde{\mathcal{F}}^+_\wedge(\mathcal{T}, 0, s, Arith(\overline{c}))$.

Remarque. Toutes les démontrations de cette section sont faites dans le cadre syntaxique le plus restrictif, celui de $\mathcal{G}^+_\wedge$. On vérifie facilement qu'elles restent valables dans $\mathcal{G}^+$. Autrement dit :

les lemmes 2.2 et 2.3 restent vrais si l'on remplace partout $\mathcal{G}^+_\wedge$ par $\mathcal{G}^+$.

2.4 Pseudo-réduction de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$

On montre ici qu'une formule $\Phi \in \mathcal{G}^+(0, s)$ portant sur un tuple $\overline{f}$ de fonctions unaires de domaine $n \in \mathbb{N}^*$ peut être "traduite" en une formule de $\mathcal{G}^+_\wedge(0, s)$ portant sur un tuple $\overline{f}^{(3)}$ de fonctions unaires codant $\overline{f}$, à condition que le domaine M de ces dernières fonctions contienne $3n$ (*i.e.* $M \geq 3n$).

La proposition 2.6 formalisera ce résultat. Mais auparavant, on isole la "clef" du raisonnement dans le lemme 2.4. Plus précisément, on prouve le résultat visé dans le cas très particulier des formules de $\mathcal{G}^+(0, s)$ de la forme :

$$\forall a \bigvee_{i<3} u_i v_i(a) = w_i(a).$$

On étend ensuite cette traduction aux formules de $\mathcal{G}^+(0, s)$ du premier ordre (Lemme 2.5) avant d'établir la proposition 2.6 dans toute sa généralité.

Enfin, on donne une version de cette proposition plus adaptée à l'usage que l'on veut en faire dans les sections suivantes (Proposition 2.7).

DÉFINITIONS 2.6.

- Soient $M \in \mathbb{N}^*$ et $P : M \to M$. On dit que P est 3-*stable* si elle préserve le quotient modulo 3, autrement dit si pour tout $x < M$, $\lfloor P(x)/3 \rfloor = \lfloor x/3 \rfloor$.

– Soient $n \in \mathbb{N}^*$ et $M \geq 3n$. On note $3(n)$ la partie de M : $\{3a,\ a < n\}$ et pour $f : n \to n$, on appelle *3-dispersion de f sur M* la fonction $f^{(3)} : M \to M$ définie par :

$$\forall x < M,\ f^{(3)}(x) = \begin{cases} 3f(\frac{x}{3}) \text{ si } x \in 3(n)\ ; \\ 1 \text{ sinon.} \end{cases}$$

Lemme 2.4. *Soient $n \in \mathbb{N}^*$ et $M \geq 3n$. Soient $f_1, \ldots, f_p$ des fonctions unaires sur n. Soient $(u_i)_{i<3}$, $(v_i)_{i<3}$, $(w_i)_{i<3}$, trois triplets à valeurs dans $\{f_1, \ldots, f_p\}$. Alors les assertions suivantes sont équivalentes :*

1. $\langle n, \overline{f}\rangle \models \forall a \bigvee_{i<3} u_i v_i(a) = w_i(a)$;
2. *il existe une fonction $P : M \to M$, bijective, 3-stable, et telle que :* $\langle M, \overline{f^{(3)}}, s, 0, n, P\rangle \models (\forall x < n) \bigwedge_{i<3} u_i^{(3)} v_i^{(3)} P(3x+i) = w_i^{(3)} P(3x+i)$.

Preuve.

$$\boxed{1 \Rightarrow 2}$$

Pour chaque $a < n$ on note p_a la transposition de l'ensemble $\{0, 1, 2\}$ qui échange 0 et le plus petit $i < 3$ tel que $n \models u_i v_i(a) = w_i(a)$ (l'existence de cet entier étant garantie par l'hypothèse *1*).

On vérifie sans peine que la fonction $P : M \to M$ définie par : $\forall a < n$, $\forall i < 3$, $P(3a+i) = 3a + p_a(i)$ et $\forall x \geq 3n$, $P(x) = x$, est une permutation 3-stable de M. Montrons que l'on a alors :

$$M \models (\forall x < n) \bigwedge_{i<3} u_i^{(3)} v_i^{(3)} P(3x+i) = w_i^{(3)} P(3x+i).$$

Soient donc $x < n$ et $i < 3$. Alors par définition de P :

$$P(3x+i) = 3x + p_x(i) \tag{2.1}$$

Deux cas de figure se présentent :

1. $p_x(i) \neq 0$.

   Dans ce cas, l'équation (2.1) d'une part, la définition des 3-dispersions d'autre part, permettent d'écrire pour chaque $j < p$:

   $$f_j^{(3)} P(3x+i) = 1$$

   En particulier, puisque v_i et w_i sont dans $\{f_1, \ldots, f_p\}$: $v_i^{(3)} P(3x+i) = 1 = w_i^{(3)} P(3x+i)$. Et comme $u_i^{(3)}(1) = 1$, on a finalement, pour tout entier $i < 3$ tel que $p_x(i) \neq 0$:

   $$M \models u_i^{(3)} v_i^{(3)} P(3x+i) = w_i^{(3)} P(3x+i)$$

2. $p_x(i) = 0$.

   Alors par (2.1) : $u_i^{(3)}v_i^{(3)}P(3x+i) = u_i^{(3)}v_i^{(3)}(3x)$.

   De l'identité $u^{(3)}v^{(3)} = (uv)^{(3)}$, vraie (et facilement vérifiable) pour toutes fonctions u, v sur n, on tire : $u_i^{(3)}v_i^{(3)}(3x) = (u_iv_i)^{(3)}(3x) = 3u_iv_i(x)$ (par définition de la 3-dispersion). D'où finalement :

   $$u_i^{(3)}v_i^{(3)}P(3x+i) = 3u_iv_i(x).$$

   On montre encore plus simplement :

   $$w_i^{(3)}P(3x+i) = 3w_i(x).$$

   Or l'égalité $p_x(i) = 0$ entraine précisement, par définition de la transposition p_x : $u_iv_i(x) = w_i(x)$.

   Ainsi, pour l'entier $i < 3$ antécédent de 0 par la transposition p_x, on a à nouveau :

   $$M \models u_i^{(3)}v_i^{(3)}P(3x+i) = w_i^{(3)}P(3x+i)$$

Cette dernière assertion étant maintenant prouvée pour tout $x < n$ et tout $i < 3$, on peut conclure :

$$M \models (\forall x < n) \bigwedge_{i<3} u_i^{(3)}v_i^{(3)}P(3x+i) = w_i^{(3)}P(3x+i) \tag{2.2}$$

$\boxed{2 \Rightarrow 1}$

Soit P une permutation 3-stable de M telle que :

$$M \models (\forall x < n) \bigwedge_{i<3} u_i^{(3)}v_i^{(3)}P(3x+i) = w_i^{(3)}P(3x+i) \tag{2.3}$$

Soit $a < n$. Considérons l'entier $i < 3$ tel que $P(3a+i) = 3a$ (dont l'existence est assurée par la bijectivité et la stabilité de P). On obtient alors, grâce à l'assertion (2.3) :

$$M \models u_i^{(3)}v_i^{(3)}(3a) = w_i^{(3)}(3a)$$

Or :

$$u_i^{(3)}v_i^{(3)}(3a) = (u_iv_i)^{(3)}(3a) = 3u_iv_i(a) \text{ et } w_i^{(3)}(3a) = 3w_i(a).$$

Ce prouve bien que pour tout $a < n$ il existe un $i < 3$ tel que $\langle n, \overline{f}\rangle \models u_iv_i(a) = w_i(a)$. C'est le résultat cherché.

Lemme 2.5. *Soient $n \in I\!N^*$ et $M \geq 3n$. Soient $f_1, \ldots, f_p$ des fonctions unaires sur n. Soient $s \in I\!N$ et (u_{ij}), (v_{ij}), (w_{ij}), $i < 3$, $j < s$, trois $3s$-uplets à valeurs dans $\{f_1, \ldots, f_p\}$. Alors il existe une formule $\psi_\wedge \in \mathcal{F}^+_\wedge(\overline{f^{(3)}}, 0, s, n)$ telle que les assertions suivantes soient équivalentes :*

1. $\langle n, \overline{f} \rangle \models \forall a \bigwedge_{j<s} \bigvee_{i<3} u_{ij}v_{ij}(a) = w_{ij}(a)$;

2. $\langle M, \overline{f^{(3)}}, s, 0, n \rangle \models \psi_\wedge$.

Preuve. Tout d'abord, grâce au lemme 2.4, l'hypothèse 1 équivaut clairement à :

$$\text{pour tout } j < s, \text{ il existe une permutation 3-stable } P_j \text{ telle que}$$
$$M \models (\forall x < n) \bigwedge_{i<3} u^{(3)}_{ij} v^{(3)}_{ij} P_j(3x+i) = w^{(3)}_{ij} P_j(3x+i),$$

c'est-à-dire à :

$$\text{il existe } P_0, \ldots, P_{s-1}, \text{ permutations 3-stables telles que}$$
$$M \models (\forall x < n) \bigwedge_{j<s} \bigwedge_{i<3} u^{(3)}_{ij} v^{(3)}_{ij} P_j(3x+i) = w^{(3)}_{ij} P_j(3x+i).$$

On note $\phi(0, s, n, \overline{P})$ cette dernière formule. Il reste à constater qu'une fonction unaire P sur M est une permutation 3-stable si et seulement si elle satisfait la formule de $\mathcal{F}^+_\wedge(Q_3, P)$:

$$Persta(P) \equiv \exists P' \forall x \,:\, PP'(x) = x \wedge Q_3P(x) = Q_3(x).$$

En notant $Persta(\overline{P}) \equiv \bigwedge_{j<s} Persta(P_j)$, on conclut maintenant facilement que la formule $\psi_\wedge$ cherchée est obtenue par :

$$\psi_\wedge \equiv \exists P_0 \ldots \exists P_{s-1} : Persta(\overline{P}) \wedge \phi(0, s, n, \overline{P}).$$

Proposition 2.6. *Soit $\mathcal{T}$ une suite de p symboles de fonction unaires et c un symbole de constante. Pour toute $\Psi \in \mathcal{G}^+(\mathcal{T}, s, 0)$, il existe une formule $\Psi_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, s, 0, c)$ telle que : pour tous $n \in I\!N^*$ et $M \geq 3n$, et pour tout p-uplet $(f_1, \ldots, f_p)$ de fonctions unaires sur n :*

$$\langle n, \overline{f}, s, 0 \rangle \models \Psi \text{ ssi } \langle M, \overline{f^{(3)}}, s, 0, n \rangle \models \Psi_\wedge$$

Preuve. Considérons l'assertion

$$\langle n, \overline{f}, s, 0 \rangle \models \Psi. \tag{2.4}$$

Compte tenu de de la signature de Ψ, on peut supposer sans perte de généralité que cette formule se réécrit, en notant z la fonction constante en 0 :

$$\Psi(\overline{f}, s, z) \equiv \exists \overline{g} \forall x \bigwedge_{j<s} \bigvee_{i<3} u_{ij}v_{ij}(x) = w_{ij}(x),$$

avec $u_{ij}, v_{ij}, w_{ij} \in \{\overline{f}, s, z\}$. L'assertion (2.4) équivaut alors à :

$$\exists g_1, \ldots, g_q : n \to n \ : \ \langle n, \overline{f}, \overline{g}, s, z\rangle \models \forall x \bigwedge_{j<s} \bigvee_{i<3} u_{ij}v_{ij}(x) = w_{ij}(x)$$

C'est à dire à :

$$\exists g_1, \ldots, g_q : n \to n \ : \ \langle M, \overline{f^{(3)}}, \overline{g^{(3)}}, s^{(3)}, z^{(3)}, s, 0, n\rangle \models \psi_\wedge, \tag{2.5}$$

où $\psi_\wedge$ est la formule de $\mathcal{G}^+_\wedge(\overline{f^{(3)}}, \overline{g^{(3)}}, s^{(3)}, z^{(3)}, s, 0, n)$ construite comme dans le lemme précédent. Partant de cette assertion, on accède au résultat annoncé dans le lemme en trois étapes :

1. on commence par prouver que l'on peut "rentrer les fonctions g_i existentiellement quantifiées à l'intérieur de la formule $\psi_\wedge$". Il s'agit en fait de montrer que la propriété "être une 3-dispersion" est définissable par une formule de $\mathcal{G}^+_\wedge(s, 0)$.

2. On montre ensuite que la fonction $s^{(3)}$ est elle aussi définissable sur M par une formule de $\mathcal{G}^+_\wedge$ à partir des seuls symboles prédéfinis 0, s et n.

3. On fait de même pour la fonction $z^{(3)}$.

Définissabilité des dispersions. L'assertion (2.5) peut s'écrire de manière équivalente :
il existe q fonctions unaires sur M, $h_1, \ldots, h_q$, telles que :

(a) chaque h_i est la 3-dispersion d'une fonction $g_i : n \to n$;

(b) $\langle M, \overline{f^{(3)}}, \overline{h}, s^{(3)}, z^{(3)}, s, 0\rangle \models \psi_\wedge$.

La définition des 3-dispersions mène à la formulation équivalente de (a) (pour chaque i) : $\forall x \in 3(n) : h_i(x) \in 3(n)$ et $\forall x \in M \setminus 3(n) : h_i(x) = 1$. Ou encore à :

$$\begin{array}{l}(\forall x \in 3(n))(\exists y \in 3(n)) : h_i(x) = y \ \wedge \\ (\forall x \in 3(n)) : h_i(x+1) = h_i(x+2) = 1 \ \wedge \\ (\forall x \geq 3n) : h_i(x) = 1.\end{array}$$

Soit, en tenant compte des égalités ensemblistes : $3(n) = \{3\cdot(n \dot{-} x), x < M\}$ et $\{3n, \ldots, M-1\} = \{3\cdot n \dot{+} x, x < M\}$, puis en skolémisant la variable y :

$$\begin{array}{ll}\exists Y \forall x : & h_i(3\cdot(n \dot{-} x)) = 3\cdot(n \dot{-} Y(x)) \ \wedge \\ & h_i(3\cdot(n \dot{-} x)+1) = h_i(3\cdot(n \dot{-} x)+1) = h_i(3\cdot n \dot{+} x) = 1.\end{array}$$

On note *Disp*(h_i) cette dernière formule et *Disp*$(h_1, \ldots, h_q) \equiv \bigwedge_i$ *Disp*(h_i). Les remarques de la section 2.3 garantissent l'appartenance de *Disp* à $\mathcal{G}^+_\wedge(\overline{h}, s, 0, n)$. La formule $\Psi^*_\wedge \equiv \exists \overline{h} : \mathit{Disp}(\overline{h}) \wedge \psi_\wedge$ appartient donc à $\mathcal{G}^+_\wedge(\overline{f^{(3)}}, s^{(3)}, z^{(3)}s, 0, n)$ et l'on a par construction :

$$\langle n, \overline{f}, s, 0\rangle \models \Psi \text{ ssi } \langle M, \overline{f^{(3)}}, s^{(3)}, z^{(3)}s, 0, n\rangle \models \Psi^*_\wedge.$$

Il nous reste pour conclure à prouver la définissabilité sur M de $s^{(3)}$ et de $z^{(3)}$ par des formules $d_1(s^{(3)}), d_2(z^{(3)}) \in \mathcal{G}^+_\wedge$.

C'est ce que l'on fait dans les lignes qui suivent en differenciant momentanément, par souci de lisibilité, les successeurs sur n et sur M, respectivement notés s et S.

Définissabilité de $s^{(3)}$. Pour tout $x < M$, $s^{(3)}(x)$ est défini comme suit :

$$s^{(3)}(x) = \begin{cases} 3s(x/3) = x + 3 = S^3(x) \text{ si } x \in 3(n) \\ 1 \text{ sinon} \end{cases},$$

où S^3 désigne la fonction successeur composée 3 fois. Par conséquent $s^{(3)}$ est définie par la formule :

$$\begin{aligned} d_1(s^{(3)}) \equiv \ \forall x : \ & s^{(3)}(3(n-s(x))) = S^3(3(n-s(x))) \wedge \\ & s^{(3)}(3(n-s(x)))+1 = s^{(3)}(3(n-s(x)))+2 = s^{(3)}(3 \cdot n+x) = 1. \end{aligned}$$

Définissabilité de $z^{(3)}$. Pour tout $x < M$, $z^{(3)}(x)$ est défini comme suit :

$$z^{(3)}(x) = \begin{cases} 3z(x/3) = 0 \text{ si } x \in 3(n) \\ 1 \text{ sinon} \end{cases}.$$

Par conséquence, $z^{(3)}$ est définie par la formule :

$$\begin{aligned} d_2(z^{(3)}) \equiv \ \forall x : \ & z^{(3)}(3(n-s(x))) = 0 \wedge \\ & z^{(3)}(3(n-s(x))+1 = z^{(3)}(3(n-s(x))+2 = z^{(3)}(3 \cdot n+x) = 1. \end{aligned}$$

Les formules $d_1(s^{(3)})$ et $d_2(z^{(3)})$ sont dans $\mathcal{G}^+_\wedge(s, 0, n)$ et la formule $\Psi_\wedge$ annoncée s'écrit simplement : $\Psi_\wedge(\overline{f^{(3)}}, s, 0, n) \equiv \exists s^{(3)} \exists z^{(3)} \ : \ d_1(s^{(3)}) \wedge d_2(z^{(3)}) \wedge \Psi^*_\wedge$.

En fait, ce résultat nous sera particulièrement utile dans le cas où les entiers n et M sont liés par la relation $n = \lfloor M/3 \rfloor$ (*i.e.* sont tels que $M = 3n + \ell$ avec $\ell \in \{0, 1, 2\}$). Dans ce cas en effet, n est définissable dans M par une formule de $\mathcal{G}^+_\wedge(0, s)$ (voir section 2.3, paragraphe 13 : *définissabilité de $\mathcal{Q}_c$*). Par conséquent, on peut supposer dans ce cas précis que la formule $\Psi_\wedge$ appartient à $\mathcal{G}^+_\wedge(\mathcal{T}, s, 0)$. Nous retiendrons donc :

Proposition 2.7. *Soit $\mathcal{T}$ une suite de p symboles de fonction unaires. Pour toute $\Psi \in \mathcal{G}^+(\mathcal{T}, s, 0)$, il existe une formule $\Psi_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, s, 0)$ telle que : pour tous $n \in I\!N^*$ et $\ell < 3$, et pour tout p-uplet $(f_1, \ldots, f_p)$ de fonctions unaires sur n :*

$$\langle n, \overline{f}, s, 0\rangle \models \Psi \ \textit{ssi} \ \langle 3n + \ell, \overline{f^{(3)}}, s, 0\rangle \models \Psi_\wedge$$

Remarque. Dans la preuve de la proposition 2.6, nous avons utilisé en le prouvant (dans la construction des formules *Disp* et d_1, d_2) le résultat suivant :

> *On ne modifie pas $\mathcal{G}^+_\wedge(0,s,c)$ en autorisant dans la syntaxe la quantification $\forall x \notin 3(c)$.*

Ce résultat reste clairement vrai si l'on remplace $\mathcal{G}^+_\wedge(0,s,c)$ par $\mathcal{G}^+(0,s,c)$. Nous le considèrerons désormais comme aquis, et nous en ferons à nouveau usage dans les sections suivantes.

2.5 Un "pré-travail" sur $\mathcal{G}^+(\mathcal{T},0,s)$

Notations. Soit $M > 9$. On suppose que M s'écrit $M = 3n + \ell$, avec $0 \leq \ell < 3$ (et $n > 3$).

- On dit qu'un couple d'entiers (i,a) appartient à $bf(3n+\ell)$ s'il existe un entier $x < 3n+\ell$ tel que $Q_n(x) = i$ et $R_n(x) = a$. L'ensemble $bf(3n+\ell)$ est donc exactement constitué des couples d'entiers (i,a) tels que $(a < n \wedge i < 3) \vee (i = 3 \wedge a < \ell)$.

- Soit $f : M \to M$ une fonction unaire. L'égalité $f(in+a) = nQ_nf(in+a) + R_nf(in+a)$ vraie pour tout $(i,a) \in bf(3n+\ell)$ permet de définir, pour tout $i < 3$, les fonctions q^i_f et r^i_f : $n \to n$, définies pour tout $a < n$ par : $f(in+a) = nq^i_f(a) + r^i_f(a)$. Pour $i = 3$, on peut encore définir $q^3_f(a)$ et $r^3_f(a)$ tels que $f(3n+a) = nq^3_f(a) + r^3_f(a)$ lorsque $a < \ell$. On prolonge alors ces deux fonctions au domaine n en posant : $\forall a \in [\ell, n[$, $q^3_f(a) = 3$ et $r^3_f(a) = \ell$. En résumé on a :

$$\begin{array}{rcl} \forall (i,a) \in bf(3n+\ell) & : & q^i_f(a) = Q_nf(in+a) \text{ et } r^i_f(a) = R_nf(in+a) \\ \forall a \geq \ell & : & q^3_f(a) = 3 \text{ et } r^3_f(a) = \ell \end{array} \tag{2.6}$$

- Soit $T = \{f_1, \ldots, f_p\}$ un p-uplet de fonctions unaires sur $3n+\ell$. On note $T^{1/3}$ le $8p$-uplet de fonctions unaires sur n : $(q^i_{f_1}, r^i_{f_1}, \ldots, q^i_{f_p}, r^i_{f_p})_{i \leq 3}$.

- Si T est réduit au singleton f, on notera $f^{1/3}$ plutôt que $\{f\}^{1/3}$ le 8-uplet $(q^i_f, r^i_f)_{i<3}$.

- Quand le p-uplet T est noté $\overline{f}$, on notera parfois $\overline{f}^{1/3}$ plutôt que $\{f_1, \ldots, f_p\}^{1/3}$.

- On adopte les mêmes conventions pour les p-uplets de symboles de fonction unaires (si $\mathcal{T} = \{f_1, \ldots, f_p\}$ est un tel p-uplet, $\mathcal{T}^{1/3}$ désigne un $8p$-uplet de symboles de fonction unaires, $(q^i_f, r^i_f)_{i \leq 3}$).

Lemme 2.8. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires. Soit ψ une formule du premier ordre de $\mathcal{G}^+(\mathcal{T})$ de la forme : $\psi \equiv \forall x \bigwedge \bigvee uv(x) = w(x)$, avec $u, v, w \in \mathcal{T}$.*

Alors il existe $\psi^{1/3} \in \mathcal{G}^+(\mathcal{T}^{1/3}, 0, s)$ *telle que pour tout* $n > 3$, *tout* $\ell < 3$ *et tout p-uplet* $(f_1, \ldots, f_p)$ *de fonctions unaires sur* $3n + \ell$ *:*

$$\langle 3n + \ell, \overline{f} \rangle \models \psi \textit{ ssi } \langle n, \overline{f}^{1/3}, 0, s \rangle \models \psi^{1/3}$$

Preuve. La preuve se fait en trois étapes.
1. Montrons d'abord l'équivalence suivante :

$$\begin{array}{c} \langle 3n + \ell, \overline{f} \rangle \models \forall x \bigwedge \bigvee \{uv(x) = w(x)\} \\ \text{ssi} \\ \langle n, \overline{f}^{1/3}, 0, s \rangle \models \forall a (\forall i \leq 3) \bigwedge \bigvee \{q^i_{uv}(a) = q^i_w(a) \wedge r^i_{uv}(a) = r^i_w(a)\} \end{array} \tag{2.7}$$

La condition nécéssaire provient des deux faits suivants : Soient $i \leq 3$ et $a < n$.

- Si $(i, a) \in bf(3n + \ell)$, alors l'entier $in + a$ est dans $3n + \ell$ et on a pour toutes fonctions $u, v : 3n + \ell \to 3n + \ell$: $3n + \ell \models u(in + a) = v(in + a)$ ssi $n \models q^i_u(a) = q^i_v(a) \wedge r^i_u(a) = r^i_v(a)$. De l'hypothèse : $\langle 3n+\ell, \overline{f} \rangle \models \bigwedge \bigvee \{uv(in+a) = w(in+a)\}$, on tire alors facilement : $\langle n, \overline{f}^{1/3}, 0, s \rangle \models \bigwedge \bigvee \{q^i_{uv}(a) = q^i_w(a) \wedge r^i_{uv}(a) = r^i_w(a)\}$.
- Si $i = 3$ et $a \geq \ell$, alors pour toute fonction $u : 3n + \ell \to 3n + \ell$ on a $q^i_u(a) = 3$ et $r^i_u(a) = \ell$. Cela garantie bien à nouveau : $\langle n, \overline{f}^{1/3}, 0, s \rangle \models \bigwedge \bigvee \{q^i_{uv}(a) = 3 = q^i_w(a) \wedge r^i_{uv}(a) = \ell = r^i_w(a)\}$.

La condition suffisante se déduit facilement de la définition des fonctions q_i et r_i.
2. Or on a par ailleurs, pour toutes $u, v : 3n + \ell \to 3n + \ell$ et pour tous $i \leq 3$, $a < n$:

$$q^i_{uv}(a) = q_u^{q^i_v(a)}(r^i_v(a)) \text{ et } r^i_{uv}(a) = r_u^{q^i_v(a)}(r^i_v(a)). \tag{2.8}$$

En effet :

- Si $(i, a) \in bf(3n+\ell)$, alors $uv(in+a) = u(q^i_v(a)n + r^i_v(a))$, et comme $(q^i_v(a), r^i_v(a))$ est nécéssairement dans $bf(3n + \ell)$, on a aussi : $uv(in + a) = q_u^{q^i_v(a)}(r^i_v(a))n + r_u^{q^i_v(a)}(r^i_v(a))$. Par ailleurs, puisque $(i, a) \in bf(3n + \ell)$, on a aussi : $uv(in + a) = q^i_{uv}(a)n + r^i_{uv}(a)$. L'égalité cherchée s'en déduit.
- Si $i = 3$ et $a \geq \ell$, alors on a : $q^i_v(a) = 3$, $r^i_v(a) = \ell$ et donc $q_u^{q^i_v(a)}(r^i_v(a)) = q^3_u(\ell) = 3 = q^i_{uv}(a)$ et $r_u^{q^i_v(a)}(r^i_v(a)) = r^3_u(\ell) = \ell = r^i_{uv}(a)$.

3. Les égalités de (2.8) donnent lieu à l'équivalence suivante, vraie pour toutes $u, v : 3n + \ell \to 3n + \ell$ et pour tous $i \leq 3$, $a < n$:

$$\begin{array}{c} n \models q^i_{uv}(a) = q^i_w(a) \wedge r^i_{uv}(a) = r^i_w(a) \\ \text{ssi} \\ n \models \bigvee_{j \leq 3} (q^i_v(a) = j \wedge q^j_u r^i_v(a) = q^i_w(a) \wedge r^j_u r^i_v(a) = r^i_w(a)). \end{array} \tag{2.9}$$

Pour chaque $i \leq 3$, notons $Comp_i(a, u^{1/3}, v^{1/3}, w^{1/3})$ la dernière formule obtenue. Des équivalences (2.7) et (2.9) on tire :

$$\langle 3n+\ell, \overline{f}\rangle \models \forall x \bigwedge\bigvee\{uv(x) = w(x)\}$$
$$\text{ssi}$$
$$\langle n, \overline{f}^{1/3}, 0, s\rangle \models \forall a \bigwedge_{i\leq 3} \bigwedge\bigvee Comp_i(a, u^{1/3}, v^{1/3}, w^{1/3})$$

et $\psi^{1/3}(\mathcal{T}, 0, s) \equiv \forall a \bigwedge_{i\leq 3} \bigwedge\bigvee Comp_i(a, u^{1/3}, v^{1/3}, w^{1/3})$ est la formule attendue.

Proposition 2.9. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires et l un symbole de constante. Soit $\Psi \in \mathcal{F}^+(\mathcal{T}, 0, s)$. Alors il existe $\Psi^{\frac{1}{3}} \in \mathcal{F}^+(\mathcal{T}^{1/3}, 0, s, l)$ telle que pour tout $n > 3$, tout $\ell < 3$ et tout p-uplet $(f_1, \ldots, f_p)$ de fonctions unaires sur $3n+\ell$:*

$$\langle 3n+\ell, \overline{f}, 0, s\rangle \models \Psi \text{ ssi } \langle n, \overline{f}^{1/3}, 0, s, \ell\rangle \models \Psi^{\frac{1}{3}}.$$

(Où l est interprété par la constante ℓ).

Preuve. Compte tenu de la signature de Ψ, on peut supposer sans perte de généralité que cette formule s'écrit : $\Psi \equiv \exists g_1 \ldots \exists g_p \psi(\overline{f}, \overline{g}, s, z)$, où ψ est elle même de la forme $\psi \equiv \forall x \bigwedge\bigvee uv(x) = w(x)$, avec $u, v, w \in \mathcal{T} \cup \{\overline{g}, s, z\}$.

L'assertion $\langle 3n+\ell, \overline{f}, 0, s\rangle \models \Psi$ équivaut alors à :

$$\exists g_1, \ldots, g_p : 3n+\ell \to 3n+\ell \ : \ \langle 3n+\ell, \overline{f}, \overline{g}, 0, s\rangle \models \psi(\overline{f}, \overline{g}, s, z).$$

C'est à dire, grâce au lemme 2.8, à :

$$\exists g_1, \ldots, g_p : 3n+\ell \to 3n+\ell \ : \ \langle n, \overline{f}^{1/3}, \overline{g}^{1/3}, s^{1/3}, z^{1/3}, 0, s\rangle \models \psi^{\frac{1}{3}}.$$

On se retrouve alors dans une situation très semblable à celle rencontrée dans la proposition 2.6. L'essentiel de la preuve consiste à prouver la définissabilité par une formule de $\mathcal{G}^+(0, s, \ell)$:

(a) de la propriété, pour un 8-uplet de fonctions unaires sur n, d'être de la forme $g^{1/3}$, où g est une fonction unaire sur $3n+\ell$;

(b) du 8-uplet $s^{1/3}$;

(c) du 8-uplet $z^{1/3}$.

Définissabilité des 8-uplets $g^{1/3}$. Un 8-uplet $(q^0, r^0, \ldots, q^3, r^3)$ de fonctions unaires sur n est de la forme $g^{1/3}$ (avec $g : 3n+\ell \to 3n+\ell$) si et seulement si il vérifie les conditions :

– $\forall (i, a) \in bf(3n+\ell) : q^i(a) < 3 \vee (q^i(a) = 3 \wedge r^i(a) < \ell)$;

– $\forall a \geq \ell$: $q^3(a) = 3$ et $r^3(a) = \ell$.

Cela revient à dire que le 8-uplet satisfait dans n la conjonction des formules :

– $\forall a \ : \ \bigwedge_{i<3} \bigvee_{j<3} (q^i(a) = j) \vee (q^i(a) = 3 \wedge r^i(a) < \ell)$;

– $\forall a < \ell \ : \ \bigvee_{j<3} (q^3(a) = j) \vee (q^3(a) = 3 \wedge r^3(a) < \ell)$;

– $\forall a \geq \ell \ : \ (q^3(a) = 3 \wedge r^3(a) = \ell)$.

qui équivaut à la conjonction des formules :

– $\forall a \exists y_0, y_1, y_2 \ : \ \bigwedge_{i<3} \bigvee_{j<3} (q^i(a) = j) \vee (q^i(a) = 3 \wedge r^i(a) = \ell \dot{-} s(y_i))$;

– $\forall a \exists y_3 \ : \ \bigvee_{j<3} (q^3(\ell \dot{-} s(a)) = j) \vee (q^3(\ell \dot{-} s(a)) = 3 \wedge r^3(\ell \dot{-} s(a)) = \ell \dot{-} s(y_3))$;

– $\forall a \ : \ q^3(a+\ell) = 3 \wedge r^3(a + \ell = \ell)$.

On note $BF(q^i, r^i)$ la formule obtenue en skolémisant les y_i dans cette dernière conjonction. Par construction, $BF(q^i, r^i)$ appartient à $\mathcal{F}^+(q^i, r^i, 0, s, \ell)$. et impose au tuple $(q_i, r_i)_{i \leq 3}$ de s'écrire $g^{1/3}$ pour une certaine $g : 3n + \ell \to 3n + \ell$. Ceci permet d'affirmer :

$$\langle 3n + \ell, \overline{f}, 0, s \rangle \models \exists g_1, \ldots, g_s \psi(\overline{f}, \overline{g})$$
$$\text{ssi}$$
$$\langle n, \overline{f}^{1/3}, s^{1/3}, z^{1/3}, 0, s \rangle \models \Psi_*^{\frac{1}{3}},$$

où $\Psi_*^{\frac{1}{3}}$ est la formule de $\mathcal{G}^+(\overline{f}^{1/3}, s^{1/3}, z^{1/3}, 0, s, \ell)$:

$$(\exists q_1^i, r_1^i, \ldots, q_s^i, r_s^i)_{i \leq 3} \ : \ \bigwedge_{j<s} BF(q_j^i, r_j^i) \wedge \psi^{1/3}(\overline{f}^{1/3}, q_j^i, r_j^i).$$

On comprend aisément alors que l'existence de la formule $\Psi^{\frac{1}{3}} \in \mathcal{G}^+(\overline{f}^{1/3}, 0, s, \ell)$ annoncée se déduit de la définissabilité des 8-uplets $s^{1/3}$ et $z^{1/3}$ par des formules D_1 et D_2 de $\mathcal{G}^+(f^{1/3}, 0, s, \ell)$.

Définissabilité de $s^{1/3}$. Rappelons que pour toute fonction $f : 3n + \ell \to 3n + \ell$, le 8-uplet $f^{1/3} = (q_f^i, r_f^i)_{i \leq 3}$ de fonctions unaires sur n est défini de la façon suivante :

$$\begin{aligned} \forall (i, a) \in bf(3n + \ell) \ &: \ \ q_f^i(a) = Q_n f(in + a) \text{ et } r_f^i(a) = R_n f(in + a) \\ \forall a \geq \ell \ &: \ \ q_f^3(a) = 3 \text{ et } r_f^3(a) = \ell \end{aligned}$$

Le calcul de $q_s^i(a)$ et $r_s^i(a)$ pour $i \leq 3$ et $a < n$ va donc dépendre de i, de a et de la valeur de $s(in + a)$. Il apparait plus simple de traiter séparément les cas $i = 3$, $i = 2$ et $i < 2$.

$i = 3$. Si $a \geq \ell$, alors $(i, a) \notin bf(3n + \ell)$ et $q_s^i(a) = 3$, $r_s^i(a) = \ell$.

Si $a < \ell$, alors $(i,a) \in bf(3n+\ell)$ et $q_s^i(a) = Q_n s(3n+a)$, $r_s^i(a) = R_n s(3n+a)$.

Or $s(3n+a) = \begin{cases} 3n+a \text{ si } a = \ell - 1 \\ 3n + s(a) \text{ si } a < \ell - 1 \end{cases}$

Donc : $q_s^3(a) = 3$ si $a < \ell$ et $r_s^3(a) = \begin{cases} a \text{ si } a = \ell - 1 \\ s(a) \text{ si } a < \ell - 1 \end{cases}$

Finalement, pour tout $a < n$:

$$q_s^3(a) = 3 \text{ et } r_s^3(a) = \begin{cases} \ell \text{ si } a \geq \ell \\ \ell - 1 \text{ si } a = \ell - 1 \\ s(a) \text{ si } a < \ell - 1 \end{cases}$$

$i = 2$. Dans ce cas $(i,a) \in bf(3n+\ell)$ pour tout $a < n$ et :

$s(in+a) = s(2n+a) = \begin{cases} 2n + s(a) \text{ si } a < n-1 \\ 3n \text{ si } a = n-1 \text{ et } \ell \neq 0 \\ 2n + a \text{ si } a = n-1 \text{ et } \ell = 0 \end{cases}$

D'où :

$$q_s^2(a) = \begin{cases} 2 \text{ si } a < n-1 \\ 3 \text{ si } a = n-1 \text{ et } \ell \neq 0 \\ 2 \text{ si } a = n-1 \text{ et } \ell = 0 \end{cases}, \quad r_s^2(a) = \begin{cases} s(a) \text{ si } a < n-1 \\ 0 \text{ si } a = n-1 \text{ et } \ell \neq 0 \\ n-1 \text{ si } a = n-1 \text{ et } \ell = 0 \end{cases}$$

Ce qui s'écrit finalement, pour tout $a < n$:

$$q_s^2(a) = \begin{cases} 2 \text{ si } a < n-1 \\ 2 + \mathcal{X}_{\neq 0}(\ell) \text{ si } a = n-1 \end{cases} \quad \text{et } r_s^2(a) = \begin{cases} s(a) \text{ si } a < n-1 \\ \mathcal{X}_0(\ell) \cdot (n-1) \text{ si } a = n-1 \end{cases}$$

$i < 2$. Dans ce cas $(i,a) \in bf(3n+\ell)$ pour tout $a < n$ et :

$s(in+a) = \begin{cases} in + s(a) \text{ si } a < n-1 \\ (i+1)n \text{ si } a = n-1 \end{cases}$

D'où, pour tout $a < n$:

$$q_s^i(a) = \begin{cases} i \text{ si } a < n-1 \\ (i+1) \text{ si } a = n-1 \end{cases} \quad \text{et} \quad r_s^i(a) = \begin{cases} s(a) \text{ si } a < n-1 \\ 0 \text{ si } a = n-1 \end{cases}$$

On se convaincra maintenant facilement que les fonctions q_s^i et r_s^i $(i \leq 3)$ sont définies par la conjonction des formules suivantes :

- $\forall a : q_s^3(a) = 3$
- $\forall a : r_s^3(a \dot{+} \ell) = \ell \wedge r_s^3(\ell - 1) = \ell - 1 \wedge r_s^3(\ell \dot{-} s(a)) = s(\ell \dot{-} s(a))$
- $\forall a : q_s^2(\max \dot{-} s(a)) = 2 \wedge q_s^2(\max) = 2 + \mathcal{X}_{\neq 0}(\ell)$

- $\forall a : r_s^2(\max - s(a)) = s(\max - s(a)) \wedge r_s^2(\max) = \mathcal{X}_0(\ell)\cdot(n-1)$
- $\forall a : \bigwedge_{i<2} q_s^i(\max - s(a)) = i \wedge q_s^i(\max) = i+1$
- $\forall a : \bigwedge_{i<2} r_s^i(\max - s(a)) = s(\max - s(a)) \wedge r_s^i(\max) = 0$

Soit $D_1(s^{1/3})$ cette conjonction. Pour toutes les raisons décrites dans la section 2.3, D_1 appartient à $\mathcal{G}^+(s^{1/3},0,s,\ell)$.

Définissabilité de $z^{1/3}$. Pour ce 8-uplet de fonctions, le calcul est beaucoup plus rapide. On a en effet :

$$\begin{array}{rcl} \forall (i,a) \in bf(3n+\ell) & : & q_z^i(a) = Q_n z(in+a) = Q_n(0) = 0 \text{ et} \\ & & r_z^i(a) = R_n z(in+a) = R_n(0) = 0 \\ \forall a \geq \ell & : & q_z^3(a) = 3 \text{ et } r_z^3(a) = \ell \end{array}$$

On en déduit que $z^{1/3}$ est caractérisé par la conjonction des formules :

- $\forall a \bigwedge_{i<3} q_z^i(a) = r_z^i(a) = 0$
- $\forall a : q_z^3(\ell - s(a)) = r_z^3(\ell - s(a)) = 0 \ \wedge \ q_z^3(\ell + a) = 3 \wedge r_z^3(\ell + a) = \ell$

Soit $D_2(z^{1/3})$ cette conjonction. Pour les mêmes raisons que précédement, la formule $\exists q_z^0, \ldots, q_z^3, r_z^0, \ldots, r_z^3 D_1(z^{1/3})$ est dans $\mathcal{G}^+(\overline{f}^{1/3}, s, 0, \ell)$.

Par conséquent la formule

$$\exists \overline{q}_s, \overline{r}_s, \overline{q}_z, \overline{r}_z : D_1(\overline{q}_s, \overline{r}_s) \wedge D_2(\overline{q}_z, \overline{r}_z) \wedge \Psi_*^{1/3}$$

est dans $\mathcal{G}^+(\overline{f}^{1/3}, s, 0, \ell)$. C'est la formule $\Psi^{1/3}$ annoncée.

2.6 Réduction de $\mathcal{G}^+(\mathcal{T},0,s)$ à $\mathcal{G}^+_\wedge(\mathcal{T},0,s)$

Nous allons maintenant réunir les résultats des sections 2.4 et 2.5 pour prouver la réduction de la classe $\mathcal{G}^+(\mathcal{T},0,s)$ à la classe $\mathcal{G}^+_\wedge(\mathcal{T},0,s)$. Très grossièrement, l'idée est la suivante :

- Dans la section 2.5, on préserve la caractérisation par une formule de $\mathcal{G}^+(\mathcal{T},0,s)$ en passant de structures finies de domaine M à des structures finies de domaine $\lfloor M/3 \rfloor$;

- Dans la section 2.4, on substitue à la caractérisation par une formule de $\mathcal{G}^+(\mathcal{T},0,s)$ une nouvelle caractérisation par une formule de $\mathcal{G}^+_\wedge(\mathcal{T},0,s)$ en passant de structures finies de domaine n à des structures finies de domaine $M = 3n+\ell$ ($\ell < 3$).

Il parait naturel alors de mettre bout à bout ces deux transformations pour passer de $\mathcal{G}^+(\mathcal{T}, 0, s)$ à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ sans modifier le domaine M des structures finies considérées. Cette opération est rendue délicate par les codages successifs des fonctions d'entrée. Car dans les deux transformations précitées, les fonctions unaires ont du être codées pour transporter l'information d'un domaine à l'autre (codage de f en $f^{1/3}$ pour passer du domaine M au domaine n, et codage de f en $f^{(3)}$ pour passer du domaine n au domaine M). Et la composition de ces deux codages appliquée à une fonction $f : M \to M$ donne un 8-uplet F de fonctions unaires sur M évidement distinct de f.

On verra néanmoins que ce 8-uplet F est définissable dans M par une formule de $\mathcal{G}^+_\wedge(f, 0, s)$, ce qui, finalement, rendra possible le résultat espéré.

Avant d'énoncer précisément ce résultat, introduisons une nouvelle notation : À chaque fonction $f : M \to M$ on a associé un 8-uplet $f^{1/3}$ de fonctions unaires sur $\lfloor M/3 \rfloor$. Par ailleurs, chacune des fonctions u de $f^{1/3}$ définit une fonction $u^{(3)} : M \to M$, à savoir sa 3-dispersion sur M. On notera $f^{3/3}$ le 8-uplet de fonctions unaires sur M constitué des 3-dispersions des éléments de $f^{1/3}$. Si $\overline{f} = (f_1, \ldots, f_p)$ est un p-uplet de fonctions unaires sur M, on notera $\overline{f}^{3/3}$ le tuple $(f_1^{3/3}, \ldots, f_p^{3/3})$.

Théorème 2.10. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires. Pour toute $\Psi \in \mathcal{G}^+(\mathcal{T}, 0, s)$, il existe $\Psi_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour tout $M > 9$ et tout p-uplet $\overline{f}$ de fonctions unaires sur M on ait :*

$$\langle M, \overline{f}, 0, s\rangle \models \Psi \textit{ ssi } \langle M, \overline{f}, 0, s\rangle \models \Psi_\wedge.$$

Preuve. Étant donné $\Psi \in \mathcal{F}^+(0, s, \mathcal{T})$, on sait, par la proposition 2.9, qu'il existe $\Psi^{\frac{1}{3}} \in \mathcal{F}^+(0, s, \mathcal{T}^{1/3}, l)$ (l symbole de constante) telle que, pour tous $n > 3$, $\ell < 3$ et pour tout p-uplet $\overline{f}$ de fonctions unaires sur $3n + \ell$:

$$\langle 3n + \ell, \overline{f}, 0, s\rangle \models \Psi \text{ ssi } \langle n, \overline{f}^{1/3}, 0, s, \ell\rangle \models \Psi^{\frac{1}{3}} \tag{2.10}$$

On considère alors la formule obtenue à partir de $\Psi^{\frac{1}{3}}$ en remplaçant chaque occurence de ℓ par le terme $\tilde{\ell}(x)$, où $\tilde{\ell}$ est le symbole de fonction prédéfini interprété par la fonction constante en ℓ sur tout domaine. Notons encore $\Psi^{\frac{1}{3}}$ la formule ainsi obtenue. L'équivalence (2.10) s'écrit alors :

$$\langle 3n + \ell, \overline{f}, 0, s\rangle \models \Psi \text{ ssi } \langle n, \overline{f}^{1/3}, 0, s, \tilde{\ell}\rangle \models \Psi^{\frac{1}{3}}.$$

La proposition 2.5 entraine alors l'existence d'une formule $\Psi^{\frac{3}{3}} \in \mathcal{F}^+_\wedge(f^{3/3}, 0, s, \tilde{\ell}^{(3)})$ telle que pour tout $\ell' < 3$,

$$\langle 3n + \ell, \overline{f}, 0, s\rangle \models \Psi \text{ ssi } \langle 3n + \ell', \overline{f}^{3/3}, 0, s, \tilde{\ell}^{(3)}\rangle \models \Psi^{\frac{3}{3}}$$

En particulier, en se limitant au cas ou $\ell' = \ell$ on obtient :

> *Soit $\Psi \in \mathcal{F}^+(\mathcal{T},0,s)$. Il existe $\Psi^{\frac{3}{3}} \in \mathcal{F}^+_\wedge(\mathcal{T}^{3/3},0,s,L)$ telle que, pour tous $n \geq 3$, $\ell < 3$ et pour tout p-uplet $\overline{f}$ de fonctions unaires sur $3n+\ell$:*
>
> $$\langle 3n+\ell,\overline{f},0,s\rangle \models \Psi \text{ ssi } \langle 3n+\ell,\overline{f}^{3/3},0,s,\tilde{\ell}^{(3)}\rangle \models \Psi^{\frac{3}{3}} \tag{2.11}$$
>
> *(Où L est un symbole de fonction unaire interprété par $\tilde{\ell}^{(3)}$ dans (2.11).)*

On voit bien désormais que prouver le résultat attendu revient en fait à montrer la définissabilité des fonctions de $\overline{f}^{3/3}$ et de la fonction $\tilde{\ell}^{(3)}$ dans la structure $\langle 3n+\ell,\overline{f},0,s\rangle$ par des formules de $\mathcal{F}^+_\wedge(0,s,\overline{f})$.

On ne perd aucune généralité dans le raisonnement qui va suivre en supposant que le p-uple $\overline{f}$ est réduit à une unique fonction unaire f sur $3n+\ell$. Pour simplifier les notations, on peut noter alors q^i, r^i plutôt que q^i_f, r^i_f $(i \leq 3)$ les éléments du 8-uplet $f^{1/3}$. On a donc pour tous $(i,a) \in bf(3n+\ell)$, $f(in+a) = nq^i(a) + r^i(a)$ et $\forall a \geq \ell$, $q^3(a) = 3$ et $r^3(a) = \ell$.

Définissabilité des fonctions de $f^{3/3}$. On considère séparément les cas $i < 3$ et $i = 3$.

Soit donc $i < 3$ fixé. Pour tout $a < n$, le couple (i,a) est dans $bf(3n+\ell)$ et donc :

$$q^i(a) = Q_n f(in+a) \text{ et } r^i(a) = R_n f(in+a). \tag{2.12}$$

Par ailleurs, la définition des 3-dispersions appliquée à q^i et r^i donne : $\forall x < 3n+\ell$,

$$q^{i(3)}(x) = \begin{cases} 3q^i(\frac{x}{3}) \text{ si } x \in 3(n) \\ 1 \text{ sinon} \end{cases} \quad \text{et } r^{i(3)}(x) = \begin{cases} 3r^i(\frac{x}{3}) \text{ si } x \in 3(n) \\ 1 \text{ sinon} \end{cases}$$

C'est à dire, grace à (2.12) :

$$q^{i(3)}(x) = \begin{cases} 3Q_n f(in + Q_3(x)) \text{ si } x \in 3(n) \\ 1 \text{ sinon} \end{cases}$$

et

$$r^{i(3)}(x) = \begin{cases} 3R_n f(in + Q_3(x)) \text{ si } x \in 3(n) \\ 1 \text{ sinon} \end{cases}$$

Pour chaque $i < 3$ le couple de fonctions $(q^{i(3)}, r^{i(3)})$ est donc définissable par la conjonction des formules :

- $\forall x \in 3(n) : q^{i(3)}(x) = 3Q_n f(in + Q_3(x)) \wedge r^{i(3)}(x) = 3R_n f(in + Q_3(x))$
- $\forall x \notin 3(n) : q^{i(3)}(x) = r^{i(3)}(x) = 1$

On note $\Delta_i(q^{i(3)}, r^{i(3)})$ cette conjonction. Les résultats de la section 2.3 d'une part, la remarque qui clôt la section 2.5 d'autre part, permettent d'affirmer l'appartenance de Δ_i à $\mathcal{G}^+_\wedge(0,s,f,q^{i(3)},r^{i(3)})$.

Supposons maintenant $i = 3$. Pour tout $a < n$ on a :

$$q^3(a) = \begin{cases} Q_n f(3n + a) \text{ si } a < \ell \\ 3 \text{ si } a \in [\ell, n[\end{cases} \quad \text{et} \quad r^3(a) = \begin{cases} R_n f(3n + a) \text{ si } a < \ell \\ \ell \text{ si } a \in [\ell, n[\end{cases}$$

On obtient donc, pour les 3-dispersions de ces fonctions :

$$q^{3(3)}(x) = \begin{cases} 3Q_n f(3n{+}Q_3(x)) \text{ si } x \in 3(\ell) \\ 9 \text{ si } 3|x \text{ et } x \in [3\ell, 3n[\\ 1 \text{ sinon} \end{cases}$$

et

$$r^{3(3)}(x) = \begin{cases} 3R_n f(3n{+}Q_3(x)) \text{ si } x \in 3(\ell) \\ 3\ell \text{ si } 3|x \text{ et } x \in [3\ell, 3n[\\ 1 \text{ sinon} \end{cases}$$

On se convaincra facilement que ces deux égalités équivalent à la conjonction des formules suivantes :

- $\forall x : \; q^{3(3)}(3(\ell{-}s(x))) = 3Q_n f(3n{+}(\ell{-}s(x)))\wedge$
  $r^{3(3)}(3(\ell{-}s(x))) = 3R_n f(3n{+}(\ell{-}s(x)))$
- $\forall x : q^{3(3)}(3(\ell{+}((n{-}\ell){-}s(x)))) = 9 \wedge r^{3(3)}(3(\ell{+}((n{-}\ell){-}s(x)))) = 3\ell$
- $\forall x : \bigwedge_{0<j<3} q^{3(3)}(3{\cdot}x{+}j) = r^{3(3)}(3{\cdot}x{+}j) = 1$

On note $\Delta_3(q^{3(3)}, r^{3(3)})$ cette conjonction. Pour les mêmes raisons que précédement, Δ_3 appartient à $\mathcal{G}_\wedge^+(0, s, f, q^{3(3)}, r^{3(3)})$. La formules $\Delta \equiv \bigwedge_{i\leq 3} \Delta_i$ appartient donc à $\mathcal{G}_\wedge^+(0, s, f, q^{0(3)}, r^{0(3)}, \ldots, q^{3(3)}, r^{3(3)})$ et la formule

$$\Psi_\wedge^* \equiv \exists (q_i, r_i)_{i\leq 3} \;:\; \Delta(\overline{q}, \overline{r}) \wedge \Psi^{\frac{3}{3}}$$

est dans $\mathcal{G}_\wedge^+(0, s, f, \tilde{\ell}^3)$.

Définissabilité de la fonction $\tilde{\ell}^{(3)}$. On a, par définition de $\tilde{\ell}$ et de la 3-dispersion : $\forall x < 3n + \ell$,

$$\tilde{\ell}^{(3)}(x) = \begin{cases} 3\tilde{\ell}(x/3) = 3\ell \text{ si } x \in 3(n) \;; \\ 1 \text{ sinon }. \end{cases}$$

La fonction $\tilde{\ell}^{(3)}$ est donc définie par la formule

$$\Omega(\tilde{\ell}^{(3)}) \equiv (\forall x \in 3(n))(\tilde{\ell}^{(3)}(x) = 3\ell) \wedge (\forall x \notin 3(n))(\tilde{\ell}^{(3)}(x) = 1).$$

Cette formule appartient à $\mathcal{G}_\wedge^+(0, s, n, \ell)$, et donc, puisque n et ℓ sont définissables dans $3n + \ell$ par des formules de $\mathcal{G}_\wedge^+(0, s)$: $\Omega \in \mathcal{G}_\wedge^+(0, s)$. Par conséquent, la formule $\exists L(\Omega(L) \wedge \Psi_\wedge^*(L))$ est dans $\mathcal{G}_\wedge^+(0, s)$; c'est la formule $\Psi_\wedge$ cherchée.

Théorème 2.11. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires. Pour toute $\Psi \in \mathcal{G}(\mathcal{T}, 0, s, <)$, il existe $\Psi^+_\wedge \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour tout $M > 9$ et tout p-uplet $\overline{f}$ de fonctions unaires sur M on ait :*

$$\langle M, \overline{f}, 0, s, <\rangle \models \Psi \textit{ ssi } \langle M, \overline{f}, 0, s\rangle \models \Psi^+_\wedge.$$

Preuve. Compte tenu du théorème 2.10, il suffit de prouver que $\mathcal{G}(\mathcal{T}, 0, s, <)$ se réduit à $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$, *i.e.* de construire, pour chaque formule $\Psi \in \mathcal{G}(\mathcal{T}, 0, s, <)$, une formule $\Psi^+ \in \mathcal{G}^+(\mathcal{T}, 0, s)$ telle que $\langle M, \overline{f}, 0, s, <\rangle \models \Psi$ ssi $\langle M, \overline{f}, 0, s\rangle \models \Psi^+$.

Soit donc $\Psi \in \mathcal{G}(\mathcal{T}, 0, s, <)$. Grâce aux équivalence $\neg(t < t') \leftrightarrow (t = t' \vee t' < t)$ et $\neg(t = t') \leftrightarrow (t < t' \vee t' < t)$, on peut supposer sans perte de généralité que Ψ ne comporte aucune négation (*i.e.* appartient à $\mathcal{G}^+(\mathcal{T}, 0, s, <)$). Par conséquent, Ψ est de la forme :

$$\Psi \equiv \exists\overline{g}\forall x \bigwedge\bigvee\{\sigma(x) = \tau(x), \sigma(x) < \tau(x)\},$$

où $\sigma(x)$ et $\tau(x)$ désigne des termes du langages, c'est à dire des mots de la forme $u_1 \ldots u_q(x)$, avec $u_1, \ldots, u_q \in \{\mathcal{T}, \overline{g}, z, s\}$.

Or chaque formule $\sigma(x) < \tau(x)$ équivaut à $\exists A : \forall y(As(y) = sA(y) \wedge A(0) \neq 0) \wedge \sigma(x) \neq \max \wedge \tau(x) = A(\sigma(x))$ (puisqu'une fonction A commute avec le successeur si et seulement si elle est de la forme $x \mapsto x + A(0)$). Donc Ψ équivaut à :

$$\begin{aligned}\exists\overline{g}\exists\overline{A}\forall x \ : \ & \bigwedge_{A \in \overline{A}} (As(x) = sA(x) \wedge A(0) \neq 0) \wedge \\ & \bigwedge\bigvee\{\sigma(x) = \tau(x), (\sigma(x) \neq \max \wedge \tau(x) = A(\sigma(x)))\}.\end{aligned}$$

où $\overline{A}$ est une suite contenant autant de symboles de fonction unaires qu'il y a de formule atomique du type $\sigma(x) < \tau(x)$ dans Ψ.

Cette dernière formule est la formule Ψ^+ cherchée (rappelons que les formules négatives $A(0) \neq 0$ et $\sigma(x) \neq \max$ s'écrivent positivement $\mathcal{X}_0(A(0)) = 0$ et $\mathcal{X}_{\max}(\sigma(x)) = 0$).

2.7 Caractérisation de NLIN et généralisation

Nous disposons maintenant des outils nécessaires à la présentation de la caractérisation de NLIN mentionnée dans l'introduction. Nous l'énoncerons dans le théorème 2.13 de la sous-section 2.7.1.

Dans la sous-section 2.7.2, nous rappellerons le théorème de Fagin, qui établit la caractérisation de NP par des formules du second ordre existentiel relationnel [20]. Nous montrerons que, moyennant l'introduction de symboles de fonction, du zéro et du successeur, cette première caractérisation en entraine une seconde, par des formules ne comportant ni négation ni disjonction (Corollaire 2.15).

Ce résultat peut être sensiblement précisé, à condition de le reconstruire à partir de notre caractérisation de NLIN plutôt que du théorème de Fagin. Le théorème 2.13 se généralise en effet à chaque classe $\mathrm{NTIME}_{RAM}(n^k)$, $k \geq 1$, et donne ainsi lieu à une caractérisation de NP $= \bigcup_{k>0}$ NTIME$_{RAM}(n^k)$ plus précise que celle évoquée plus haut. Ce sera l'objet de la proposition 2.17.

Dans la sous-section 2.7.3 enfin, nous présenterons un corollaire de cette dernière proposition, établi par Grandjean. Il s'agit d'un résultat de hiérarchie entre diverses classes de complexité descriptive.

2.7.1 Caractérisation de NLIN

DÉFINITION 2.7. Soit $\mathcal{S}$ un ensemble de structures unaires de signature $\mathcal{T} = \{f_1, \ldots, f_p, c_1, \ldots, c_r\}$. On dit que $\mathcal{S}$ est dans NLIN s'il existe une NRAM $\mathcal{R}$ telle que :

1. $\mathcal{R}$ utilise comme seules opérations arithmétiques les fonctions $succ : x \mapsto x+1$ et $pred : x \mapsto x-1$.

2. A l'instant $t=0$, une $\mathcal{T}$-structure S de domaine $n \in \mathbb{N}^*$ est stockée dans $\mathcal{R}$ de la façon suivante : les registres $R_0, R_1, \ldots, R_r$ contiennent respectivement $n-1$ et les constantes $c_1, \ldots, c_r$; pour tous $i \in \{1, \ldots, p\}$, $x < n$, le registre R_{in+x} contient $f_i(x)$.

3. $\mathcal{R}$ reconnait S en temps $O(n)$ (*i.e.* en exécutant $O(n)$ instructions).

Théorème 2.12 (Grandjean). *Soit $\mathcal{S}$ un ensemble de structures unaires de signature $\mathcal{T}$. Alors $\mathcal{S}$ est dans NLIN si et seulement si il existe une formule $\Phi \in \mathcal{G}(\mathcal{T}, 0, <)$ telle que pour toute $\mathcal{T}$-structure unaire S :*

$$S \in \mathcal{S} \text{ ssi } \langle S, 0, < \rangle \models \Phi.$$

Notre résultat principal se déduit sans effort des théorèmes 2.11 et 2.12:

Théorème 2.13. *Soit $\mathcal{S}$ un ensemble de structures unaires de signature $\mathcal{T}$. Alors $\mathcal{S}$ est dans NLIN si et seulement si il existe une formule $\Psi \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour toute $\mathcal{T}$-structure unaire S :*

$$S \in \mathcal{S} \text{ ssi } \langle S, 0, s \rangle \models \Psi.$$

Commentaires.

– Grandjean a prouvé que les propriétés de graphes de NP les plus usuelles (e.g. la connexité, l'hamiltonicité) sont dans NLIN. Si l'on adopte le codage fonctionnel des graphes (avec la signature $\sigma = \{f_1, f_2\}$, voir la remarque qui suit la définition 1.1 du chapitre 1, p.32), ces propriétés sont donc définissable par des formules de $\mathcal{G}^+_\wedge(f_1, f_2, 0, s)$.

– Le théorème 2.13 est valable pour toute signature unaire $\mathcal{T}$. En particulier, si $\mathcal{T}$ est vide on obtient :

> *Une partie $\mathcal{S}$ de $I\!N$ est dans $NTIME_{RAM}(n)$ (où n est la valeur de l'entier donné en entrée) si et seulement si elle est spectre d'une formule de $\mathcal{G}^+_\wedge(0,s)$.*

2.7.2 Le théorème de Fagin et sa normalisation

On sait qu'une classe de structures finies est NP si et seulement si elle est l'ensemble des modèles d'une formule de $\Sigma^1_1\Pi^2_0$, *i.e.* de la forme $\exists\overline{R}\forall\overline{x}\exists\overline{y}\phi$, où $\overline{R}$ est un tuple de symboles de relation d'arités quelconques et ϕ une formule sans quantificateur. Il va de soit que la logique obtenue à partir de la classe de formules précédente en remplaçant les symboles de relation par des symboles de fonction de même arité capture encore la classe NP et permet de se restreindre à un préfixe de quantificateurs du premier ordre de la forme $\forall^*$ (par Skolémisation). Ce qui est plus surprenant, c'est que l'addition des symboles prédéfinis 0 et s au langage considéré permet de plus de supposer la matrice ϕ sans négation ni disjonction.

C'est ce corollaire du théorème de Fagin que nous allons prouver dans cette sous-section. Mais pour faciliter sa formalisation, il, nous faut d'abord définir de nouvelles classe de formules - les classes $\mathcal{G}^+_\wedge[k]$ - qui généralisent la classe $\mathcal{G}^+_\wedge$ au second ordre fonctionnel d'arité $k \geq 1$.

Définition 2.8. Soit k un entier non nul et $\mathcal{T}$ une signature quelconque. On note $\mathcal{G}^+[k](\mathcal{T})$ (resp. $\mathcal{G}^+_\wedge[k](\mathcal{T})$) la classe des formules de la forme :

$$\exists G_1 \ldots \exists G_q \, \forall x_1 \ldots \forall x_k \, \phi,$$

où les G_i sont des symboles de fonction d'arité au plus k et ϕ est une formule du premier ordre sans quantificateur de signature $\mathcal{T} \cup \{\overline{G}\}$, ne comportant pas de négation (resp. ni négation ni disjonction).

Propriétés de la classe $\mathcal{G}^+_\wedge[k](\mathcal{T})$.

– On a clairement l'égalité $\mathcal{G}^+_\wedge(\mathcal{T}) = \mathcal{G}^+_\wedge[1](\mathcal{T})$ et les inclusions $\mathcal{G}^+_\wedge[1](\mathcal{T}) \subseteq \mathcal{G}^+_\wedge[2](\mathcal{T}) \subseteq \mathcal{G}^+_\wedge[3](\mathcal{T}) \subseteq \ldots$, où l'écriture $A \subseteq B$ dénote ici le fait que chaque formule de A est équivalente à une formule de B.

– En particulier, toutes les constantes et fonctions définissables dans $\mathcal{G}^+_\wedge(\mathcal{T})$ sont définissables dans $\mathcal{G}^+_\wedge[k](\mathcal{T})$, pour tout $k \geq 1$.

– Soit $k \geq 2$. L'addition (resp la multiplication) est définissable par la formule du premier ordre de $\mathcal{G}^+_\wedge[k](0,s)$ $\forall x_1 \ldots \forall x_k : x_1 + 0 = x_1 \wedge x_1 + s(x_2) = s(x_1 + x_2)$ (resp $\forall x_1 \ldots \forall x_k : x_1 \times 0 = 0 \wedge x_1 \times s(x_2) = x_1 \times x_2 + x_1$).

– Soient $k \geq 2$ et $k' \leq k$. Si c est une constante, la fonction caractéristique du k'-uplet $(c,\ldots,c)$ est définissable dans $\mathcal{G}^+_\wedge[k](0,s,c)$. En effet, si $\mathcal{X}^{k'}_c$ dénote la fonction k'-aire définie sur tout domaine fini n par : $\forall x_1,\ldots,x_{k'} < n$, $\mathcal{X}^{k'}_c(x_1,\ldots,x_{k'}) =$ 1 si $x_1 = \ldots = x_{k'} = c$ et 0 sinon, alors $\mathcal{X}^{k'}_c$ est définie par la formule : $\forall x_1 \ldots \forall x_k : \mathcal{X}^{k'}_c(x_1,\ldots,x_{k'}) = \prod_{1\leq i\leq k'} \mathcal{X}_c(x_i)$

– La fonction caractéristique du complémentaire de $\{(c,\ldots,c)\}$ (k'-uplet) est notée $\mathcal{X}^{k'}_{\neq c}$. Elle est définissable par la formule $\forall x_1 \ldots \forall x_k : \mathcal{X}^{k'}_{\neq c}(x_1,\ldots,x_{k'}) = 1 - \mathcal{X}^{k'}_c(x_1,\ldots,x_{k'})$.

On déduit de ces remarques et des résultats de la section 2.3 la réduction de la classe

$$\mathcal{G}^+_\wedge[k](0,s,Arith(c),+,\times,(\mathcal{X}^{k'}_c)_{k'\leq k},(\mathcal{X}^{k'}_{\neq c})_{k'\leq k})$$

à la classe $\mathcal{G}^+_\wedge[k](0,s,c)$. En particulier, si c est définissable par une formule de $\mathcal{G}^+_\wedge[k](0,s)$, toute formule de $\mathcal{G}^+_\wedge[k](0,s,Arith(c),+,\times,(\mathcal{X}^{k'}_c)_{k'\leq k},(\mathcal{X}^{k'}_{\neq c})_{k'\leq k})$ équivaut à une formule de $\mathcal{G}^+_\wedge[k](0,s)$.

Remarque. Dans la suite, on ne différenciera pas les fonctions $\mathcal{X}^{k'}_c$, $k' \leq k$, que l'on notera toutes $\mathcal{X}_c$, le contexte indiquant à quel entier k' on se réfère (et de même pour $\mathcal{X}^{k'}_{\neq c}$).

Théorème 2.14 (Fagin). *Soit $\mathcal{T}$ une signature unaire. Une classe $\mathcal{S}$ de $\mathcal{T}$-structures est dans NP si et seulement si il existe une formule Φ de $\Sigma^1_1\Pi^2_0(\mathcal{T})$ telle que pour toute $\mathcal{T}$-structure S :*

$$S \in \mathcal{S} \text{ ssi } S \models \Phi.$$

Corollaire 2.15. *Soit $\mathcal{T}$ une signature unaire. Une classe $\mathcal{S}$ de $\mathcal{T}$-structures est dans NP si et seulement si il existe une formule Ψ de $\bigcup_k \mathcal{G}^+_\wedge[k](\mathcal{T},0,s)$ telle que pour toute $\mathcal{T}$-structure S :*

$$S \in \mathcal{S} \text{ ssi } \langle S,0,s\rangle \models \Psi.$$

Preuve. Soient $\mathcal{S}$ une classe de structures unaires de signature $\mathcal{T}$ appartenant à NP et $\Phi \equiv \exists\overline{R}\forall\overline{x}\exists\overline{y}\phi$ la formule associée à $\mathcal{S}$ par le théorème 2.14 ($\overline{R}$ tuple de symboles de relation, ϕ formule du premier ordre sans quantificateur).

Soit Φ_1 la formule obtenue à partir de Φ en remplaçant chaque symbole $R \in \overline{R}$ par un symbole de fonction F de même arité, en traduisant chaque formule atomique $R(\overline{t})$ par $F(\overline{t}) = 1$, en laissant inchangées les sous-formule de ϕ du type $\sigma = \tau$ (où σ et τ sont des termes du langage $\mathcal{T}$), puis en Skolémisant le tuple $\overline{y}$ dans le préfixe de Φ.

Alors Φ_1 est équivalente à Φ et sa matrice ϕ_1 est de la forme $\bigwedge\bigvee\{\pm(F(\overline{t}) = 1), \pm(\sigma = \tau)\}$.

Soit Φ_2 la formule obtenue à partir de Φ_1 en traduisant chaque formule atomique $F(\overline{t}) = 1$ (resp. $\sigma = \tau$) par $F(\overline{t}) - 1 = 0 \wedge 1 - F(\overline{t}) = 0$ (resp. $\sigma - \tau = 0 \wedge \tau - \sigma = 0$). (Rappelons que $x - y = \max(0, x - y)$ est défini pour tous $x, y < n$.)

Alors Φ_2 est équivalente à Φ_1 et sa matrice ϕ_2 est de la forme $\bigwedge_i \bigvee_j \pm(t_{ij} = 0)$, où les t_{ij} sont des termes écrit à partir des variables quantifiées de Φ_2, des symboles de constante et de fonction unaires de $\mathcal{T}$, des symboles de fonction quantifiés de Φ_2, des constantes 0 et 1 et de la soustraction.

Soit Φ_3 la formule obtenue à partir de Φ_2 en remplaçant chaque formule $t \neq 0$ par $\mathcal{X}_0(t) = 0$. Alors Φ_3 est équivalente à Φ_2 (et donc à Φ) et peut s'écrire sous la forme :

$$\Phi_3 \equiv \exists \overline{F} \, \forall \overline{x} \bigwedge_{i<s} \bigvee_{j<r} t_{ij} = 0,$$

où $\overline{F}$ est un tuple de symbole de fonction et les t_{ij} des termes du langage $\mathcal{T} \cup \{\overline{x}, 0, 1, \overline{F}, -, \mathcal{X}_0\}$.

Finalement la formule Ψ cherchée est donnée par :

$$\Psi \equiv \exists \overline{F} \, \forall \overline{x} \bigwedge_{i<s} \prod_{j<r} t_{ij} = 0.$$

L'appartenance de Ψ à $\mathcal{G}_\wedge^+[k](\mathcal{T}, 0, s)$ se déduit de la définissabilité de l'addition, la multiplication, la soustraction et $\mathcal{X}_0$ dans $\mathcal{G}_\wedge^+[k](0, s)$ dès que $k \geq 2$.

2.7.3 Caractérisation de NTIME$_{RAM}(n^k)$

On sait que NP $= \bigcup_{k>0}$ NTIME$_{RAM}(n^k)$. On peut donc légitimement se demander si le corollaire 2.15 peut être précisé relativement à cette décomposition de NP. Plus précisément, l'écriture de NP sous la forme $\bigcup_{k>0}$ NTIME$_{RAM}(n^k)$ permet de reformuler ce corollaire de la façon suivante :

Soit $\mathcal{S}$ une classe de structures unaires de signature $\mathcal{T}$. Alors :

$$\begin{array}{c} \exists k > 0 \text{ t.q. } \mathcal{S} \in \textit{NTIME}_{RAM}(n^k) \\ \textit{ssi} \\ \exists k' > 0 \text{ t.q. } \mathcal{S} \textit{ est caractérisée par une formule de } \mathcal{G}_\wedge^+[k'](\mathcal{T}, 0, s). \end{array} \tag{2.13}$$

(En convenant qu'une classe $\mathcal{S}$ est caractérisée par une formule Ψ de $\mathcal{G}_\wedge^+[k'](\mathcal{T}, 0, s)$ si $\mathcal{S}$ est exactement l'ensemble des modèles de Ψ.) La question précédente s'énonce alors : *peut-on exhiber une dépendance fonctionnelle entre k et k' ?*

Or la preuve du théorème de Fagin, pas plus que celle du corollaire 2.15, ne permet de répondre à cette question. Tout au plus peut-on garantir les majorations $k' \leq 2k$ (pour k fixé) et $k \leq 2k'$ (pour k' fixé).

Nous prouverons pour notre part dans les lignes qui suivent que pour tout $k > 0$, la classee NTIME$_{RAM}(n^k)$ coïncide avec la classe des ensembles $\mathcal{S}$ de structures unaires caractérisables dans $\mathcal{G}_\wedge^+[k'](\mathcal{T}, 0, s)$. Autrement dit, on peut supposer $k = k'$ dans l'équivalence (2.13) ci-dessus.

L'idée de la preuve est la suivante : considérons un ensemble $\mathcal{S}$ de structures unaires de signature $\mathcal{T}$. Dire que $\mathcal{S}$ est dans NTIME$_{RAM}(n^k)$ ($k > 0$ fixé), c'est affirmer qu'il existe une RAM non déterministe $\mathcal{R}$ qui décide de l'appartenance d'une $\mathcal{T}$-structure S de taille n à la classe $\mathcal{S}$ en temps n^k.

Or, il est généralement facile de coder une $\mathcal{T}$-structure S de domaine n sur une nouvelle $\mathcal{T}$-structure $S^{[k]}$, ne contenant pas plus "d'informations" que S, mais dont le domaine a pour cardinal n^k (en rembourrant l'écriture de la structure initiale). Si ce codage reste naturel, le temps pris par la machine $\mathcal{R}$ pour reconnaitre que $S^{[k]}$ code une structure de $\mathcal{S}$ sera le même que celui utilisé pour décider de l'appartenance de S à $\mathcal{S}$. Mais ce temps, qui était polynomial en la taille de S devient linéaire en la taille de $S^{[k]}$.

Ainsi, on peut associer à la classe $\mathcal{S}$ une classe $\mathcal{S}^{[k]}$ telle que

$$\mathcal{S} \in \text{NTIME}_{RAM}(n^k) \text{ ssi } \mathcal{S}^{[k]} \in \text{NLIN}.$$

Le lien ainsi construit entre les classes NTIME$_{RAM}(n^k)$ et NLIN, associé à la caractérisation de NLIN prouvée plus haut, permet d'établir une nouvelle caractérisation de NTIME$_{RAM}(n^k)$.

Avant de formaliser ce résultat, il nous faut introduire un nouveau codage et démontrer un petit lemme technique à son propos.

Soit $k \geq 1$. Soit $n \in \mathbb{N}^*$. Tout $x < n^k$ s'écrit $x = x_0 + x_1 n + \ldots + x_{k-1} n^{k-1}$, avec $x_0, \ldots, x_{k-1} < n$. Ceci permet d'associer à toute fonction $f : n^k \to n^k$ le k-uplet $(F_0, \ldots, F_{k-1})$ de fonction k-aires sur n défini par l'égalité :

$$\forall x_0, \ldots x_k < n \ : \ f(\sum_{i<k} x_i n^i) = \sum_{i<k} F_i(\overline{x}) n^i$$

et cette association constitue une bijection de l'ensemble des fonctions unaires sur n^k vers l'ensemble des k-uplet de fonctions k-aires sur n.

Nous noterons ξ cette bijection. Et si $\overline{f} = (f_1, \ldots, f_p)$ est un p-uplet de fonctions unaires, nous noterons $\xi(\overline{f})$ le pk-uplet de fonctions k-aires $(\xi(f_1), \ldots, \xi(f_p))$.

Lemme 2.16. *Soit $\mathcal{T}$ un p-uplet de symboles de fonction unaires. Soit $k \geq 1$. Soit $\Phi \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ une formule de la forme $\exists g_1 \ldots \exists g_q \forall x \bigwedge uv(x) = w(x)$ $(u, v, w \in \mathcal{T} \cup \{\overline{g}, z, s\}$. Alors il existe $\Phi^* \in \mathcal{G}^+_\wedge[k](\mathcal{T}^{[k]}, 0, s)$ telle que pour tout $n \in \mathbb{N}^*$ et tout p-uplet $\overline{f}$ de fonctions unaires sur n :*

$$\langle n^k, \overline{f}, 0, s\rangle \models \Phi \textit{ ssi } \langle n, \xi(\overline{f}), 0, s\rangle \models \Phi^*$$

Preuve. Considérons tout d'abord une sous formule atomique de Φ, *i.e.* une égalité $uv(x) = w(x)$, pour $u, v, w \in \{\overline{f}, z, s\}$ et notons $\xi(u) = (U_0, \ldots, U_{k-1})$, $\xi(v) = (V_0, \ldots, V_{k-1})$, $\xi(w) = (W_0, \ldots, W_{k-1})$. Alors on a pour tous $\overline{x} = (x_0, \ldots, x_{k-1}) \in n^k$:

$$uv(\sum_{i<k} x_i n^i) = u(\sum_{i<k} V_i(\overline{x}) n^i) = \sum_{i<k} U_i(V_0(\overline{x}), \ldots, V_{k-1}(\overline{x})) n^i$$

On en déduit :

$$uv(\sum_{i<k} x_i n^i) = w(\sum_{i<k} x_i n^i) \text{ ssi } \forall i < k,\ U_i(V_0(\overline{x}), \ldots, V_{k-1}(\overline{x})) = W_i(\overline{x}).$$

Ceci mène facilement, grâce à la bijection canonique de n^k sur $(n)^k$, à :

$$\begin{gathered}\langle n^k, u, v, w\rangle \models \forall x \ :\ uv(x) = w(x) \\ \text{ssi} \\ \langle n, \xi(u), \xi(v), \xi(w)\rangle \models \forall x_0 \ldots \forall x_{k-1} \ :\ \textstyle\bigwedge_{i<k} U_i(V_0(\overline{x}), \ldots, V_{k-1}(\overline{x})) = W_i(\overline{x}).\end{gathered}$$

Si $\overline{f}$ et $\overline{g}$ sont deux tuples de fonctions unaires sur n, on étend immédiatement l'équivalence précédente à :

$$\begin{gathered}\langle n^k, \overline{f}, 0, s, \overline{g}\rangle \models \forall x \ :\ \textstyle\bigwedge_{j<r}\{uv(x) = w(x)\} \\ \text{ssi} \\ \langle n, \xi(\overline{f}), \xi(z), \xi(s), \xi(\overline{g})\rangle \models \forall x_0 \ldots \forall x_{k-1} \ : \\ \textstyle\bigwedge_{j<r}\{\bigwedge_{i<k} U_i(V_0(\overline{x}), \ldots, V_{k-1}(\overline{x})) = W_i(\overline{x})\},\end{gathered}$$

où $\overline{U}, \overline{V}, \overline{W}$ désignent les tuples $\xi(u), \xi(v), \xi(w)$ associés aux fonctions $u, v, w \in \{\overline{f}, \overline{g}\}$.

La bijectivité de ξ permet alors d'écrire :

$$\begin{gathered}\langle n^k, \overline{f}, 0, s\rangle \models \exists \overline{g} \forall x \ :\ \textstyle\bigwedge_{j<r}\{uv(x) = w(x)\} \\ \text{ssi} \\ \langle n, \xi(\overline{f}), \xi(z), \xi(s)\rangle \models \exists \overline{G} \forall x_0 \ldots \forall x_{k-1} \ : \\ \textstyle\bigwedge_{j<r}\{\bigwedge_{i<k} U_i(V_0(\overline{x}), \ldots, V_{k-1}(\overline{x})) = W_i(\overline{x})\},\end{gathered}$$

où $\overline{G}$ dénote un qk-uplet de symboles de fonction k-aires (qui code, intuitivement, le q-uplet $\overline{g}$ de fonctions unaires).

Il ne reste, pour conclure, qu'à prouver la définissabilité de $\xi(z)$ et $\xi(s)$ dans $\mathcal{G}^+_\wedge[k](0, s)$.

Définissabilité de $\xi(z)$. Le k-uplet $\xi(z) = (Z_0, \ldots, Z_{k-1})$ est défini par l'égalité :

$$\forall x_0, \ldots x_k < n \ :\ 0 = z(\sum_{i<k} x_i n^i) = \sum_{i<k} Z_i(\overline{x}) n^i.$$

On en déduit triviallement l'égalité de chaque fonction Z_i avec la fonction k-aire identiquement nulle.

Définissabilité de $\xi(s)$. On note $\xi(s) = (S_0, \ldots, S_{k-1})$.

Soit $i < k$. On a, pour tout $(x_0, \ldots, x_{k-1}) \in n^k$:

$$S_i(\overline{x}) = \begin{cases} x_i \text{ si } (x_0, \ldots, x_{i-1}) \neq (\max, \ldots, \max) \\ x_i + 1 \text{ si } (x_0, \ldots, x_{i-1}) = (\max, \ldots, \max) \text{ et } x_i \neq \max \\ 0 \text{ si } \ (x_0, \ldots, x_{i-1}, x_i) = (\max, \ldots, \max) \text{ et} \\ \qquad (x_{i+1}, \ldots, x_{k-1}) \neq (\max, \ldots, \max) \\ x_i \text{ si } (x_0, \ldots, x_{k-1}) = (\max, \ldots, \max) \end{cases}$$

Ce qui s'écrit, dans $\mathcal{G}^+_\wedge[k](0,s)$:

$$\begin{aligned} S_i(\overline{x}) = \; & x_i(1 - \mathcal{X}_{\max}(x_0,\ldots,x_{i-1})) + \\ & (x_i+1)\mathcal{X}_{\max}(x_0,\ldots,x_{i-1})(1-\mathcal{X}_{\max}(x_i)) + \\ & x_i(\mathcal{X}_{\max}(x_0,\ldots,x_{k-1})). \end{aligned}$$

Proposition 2.17. *Soient k un entier non nul et $\mathcal{T}$ un tuple de symboles de fonction unaires. Une classe $\mathcal{S}$ de $\mathcal{T}$-structures unaires est dans NTIME$_{RAM}(n^k)$ si et seulement si il existe $\Psi \in \mathcal{G}^+_\wedge[k](\mathcal{T},0,s)$ (resp. $\mathcal{G}^+[k](\mathcal{T},0,s)$) telle que, pour toute $\mathcal{T}$-structure unaire S de domaine $n > 9$:*

$$S \in \mathcal{S} \textit{ ssi } \langle S, 0, s\rangle \models \Psi.$$

Preuve. Soit $n > 9$. A toute fonction $f : n \to n$ on associe la fonction $f^{[k]} : n^k \to n^k$ définie par:

$$\forall x < n^k,\ f^{[k]}(x) = \begin{cases} f(x) \text{ si } x < n \\ 0 \text{ sinon} \end{cases}$$

Si S est une structure finie de la forme $\langle n, f_1, \ldots, f_p\rangle$ (où les f_i sont des fonctions unaires sur le domaine n), on note $S^{[k]}$ la structure $\langle n^k, f_1^{[k]}, \ldots, f_p^{[k]}\rangle$. On note enfin $\mathcal{S}^{[k]}$ la classe des structures ainsi associées aux structures de la classe $\mathcal{S}$. On a alors l'équivalence immédiate:

$$\mathcal{S} \in \text{NTIME}_{RAM}(n^k) \text{ ssi } \mathcal{S}^{[k]} \in \text{NLIN}$$

Compte tenu de la caractérisation de NLIN vue plus haut, on peut donc écrire:

$\mathcal{S}$ est dans NTIME$_{RAM}(n^k)$ si et seulement si il existe une formule $\Phi \in \mathcal{G}^+_\wedge(\mathcal{T},0,s)$ telle que: pour toute $\mathcal{T}$-structure unaire S,

$$S \in \mathcal{S} \text{ ssi } \langle S^{[k]}, 0, s\rangle \models \Phi.$$

Soit donc $S = \langle n, f_1, \ldots, f_p\rangle$ une $\mathcal{T}$-structure de taille $n > 9$. On a, d'après ce qui précède:

$$S \in \mathcal{S} \text{ ssi } \langle n^k, f_1^{[k]}, \ldots, f_p^{[k]}, z, s\rangle \models \Phi,$$

et donc, grâce au lemme 2.16:

$$S \in \mathcal{S} \text{ ssi } \langle n, \xi(f_1^{[k]}), \ldots, \xi(f_p^{[k]}), z, s)\rangle \models \Phi^*$$

Pour prouver le résultat cherché, il ne reste donc qu'à établir la définissabilité des k-uplets $\xi(f_i^{[k]})$, $1 \le i \le p$, par des formules de $\mathcal{G}^+_\wedge[k](\overline{f},0,s)$.

Soit donc $f \in \{f_1,\ldots,f_p\}$. Notons $(F_0^{[k]},\ldots,F_{k-1}^{[k]})$ le k-uplets $\xi(f^{[k]})$.

Par définition de $f^{[k]}$ on a, pour tout $\overline{x} = (x_0,\ldots,x_{k-1}) \in (n)^k$:

$$f^{[k]}(\sum_{i<k} x_i n^i) = \begin{cases} f(x_0) \text{ si } x_1 = \ldots = x_{k-1} = 0 \\ 0 \text{ sinon} \end{cases}$$

Or $f^{[k]}(\sum_{i<k} x_i n^i) = \sum_{i<k} F_i^{[k]}(\overline{x}) n^i$, ce qui donne :

- pour $0 < i < k$: $\forall \overline{x} \in (n)^k : F_i^{[k]}(\overline{x}) = 0$
- pour $i = 0$: $\forall \overline{x} \in (n)^k : F_0^{[k]}(\overline{x}) = \begin{cases} f(x_0) \text{ si } x_1 = \ldots = x_{k-1} = 0 \\ 0 \text{ sinon} \end{cases}$

Les fonctions $F_i^{[k]}$, $i < k$, sont donc définies par les formules de $\mathcal{G}_\wedge^+[k](f, 0, s)$:

- si $0 < i < k$: $\forall x_0 \ldots \forall x_{k-1} : F_i^{[k]}(\overline{x}) = 0$;
- si $i = 0$: $\forall x_0 \ldots \forall x_{k-1} : F_0^{[k]}(\overline{x}) = \mathcal{X}_0(x_1, \ldots, x_{k-1}) f(x_0)$

Notons $DEF^{[k]}(F_0, \ldots, F_{k-1}, f)$ la conjonction

$$\forall \overline{x} \ : \ F_0(\overline{x}) = \mathcal{X}_0(x_1, \ldots, x_{k-1}) f(x_0) \wedge \bigwedge_{0<i<k} F_i(\overline{x}) = 0.$$

Alors tout ce qui précède permet d'affirmer :

$$S \in \mathcal{S} \text{ ssi } \langle n, f_1, \ldots, f_p, z, s \rangle \models \Psi,$$

avec

$$\Psi \equiv \exists \overline{F_1}, \ldots, \overline{F_p} \ : \ \bigwedge_{1 \leq i \leq p} DEF^{[k]}(\overline{F_i}, f_i) \wedge \Phi^*,$$

où chaque $\overline{F_i}$ désigne un k-uplet de symboles de fonction k-aires.

Ψ appartient bien à $\mathcal{G}_\wedge^+[k](\mathcal{T}, 0, s)$: c'est la formule cherchée.

2.7.4 Hiérarchie

Définition 2.9.

- On note $\mathcal{G}[k', k](\mathcal{T})$ la classe des formules du second ordre existentiel de la forme : $\exists \overline{G} \forall x_1 \ldots \forall x_k \, \phi$, où $\overline{G}$ est un tuple de symbole de fonction d'arité k' et ϕ une formule sans quantificateur de signature $\mathcal{T} \cup \{\overline{G}\}$.
- On note $\mathcal{G}^+[k', k](\mathcal{T})$ (resp. $\mathcal{G}_\wedge^+[k', k](\mathcal{T})$) la sous classe de $\mathcal{G}[k', k](\mathcal{T})$ des formules dont la matrice ϕ ne contient pas de négation (resp. ni négation ni disjonction).
- On note $\mathcal{G}[\omega, k](\mathcal{T})$ la classe $\bigcup_{k'>0} \mathcal{G}[k', k](\mathcal{T})$.

Remarque. $\mathcal{G}[k](\mathcal{T}) = \mathcal{G}[k, k](\mathcal{T})$ (et de même pour les classes $\mathcal{G}^+$ et $\mathcal{G}_\wedge^+$.

Soient c (resp. f) un symbole de constante (resp. de fonction unaire) et $\mathcal{T}$ une signature unaire quelconque. Soit Ψ une formule de signature $\mathcal{T} \cup \{c, f\}$. On note $\Psi(0, s)$ la formule obtenue à partir de Ψ en remplaçant partout le symbole c (resp. f) par le symbole prédéfini 0 (resp. s). La proposition 2.17 se reformule alors comme suit :

Proposition 2.18. *Soient k un entier non nul et $\mathcal{T}$ un tuple de symboles de fonction unaires. Une classe $\mathcal{S}$ de $\mathcal{T}$-structures unaires est dans NTIME$_{RAM}(n^k)$ si et seulement si il existe $\Psi(c, f) \in \mathcal{G}^+_\wedge[k](\mathcal{T}, c, f)$ (resp. $\mathcal{G}^+[k](\mathcal{T}, c, f)$) telle que, pour toute $\mathcal{T}$-structure finie S de domaine entier $n > 9$:*

$$S \in \mathcal{S} \text{ ssi } \langle S, 0, s\rangle \models \Psi(0, s).$$

Proposition 2.19 (Grandjean). *Soit $\mathcal{S}$ une classe de structures unaires de signature (éventuellement vide) $\mathcal{T}$. Soit k un entier non nul. Alors $\mathcal{S}$ est dans la classe NTIME$_{RAM}(n^k)$ si et seulement si il existe une formule $\Psi \in \mathcal{G}[\omega, k](\mathcal{T}, 0, s)$ telle que pour toute $\mathcal{T}$-structure unaire S : $S \in \mathcal{S}$ ssi $\langle S, 0, s\rangle \models \Psi$;*

La proposition suivante est une reformulation du théorème de hiérarchie pour le temps non déterministe (voir [7]).

Proposition 2.20. *Soient k, k' deux entiers tels que $1 \leq k < k'$. Pour toute signature unaire (éventuellement vide) $\mathcal{T}$, il existe une classe de $\mathcal{T}$-structures unaires $\mathcal{S}$ telle que : $\mathcal{S} \in$ NTIME$_{RAM}(n^{k'}) -$ NTIME$_{RAM}(n^k)$.*

Corollaire 2.21 (Grandjean). *Soit $\mathcal{T} \cup \{c, f\}$ une signature unaire contenant au moins un symbole de constante (c) et un symbole de fonction unaire (f). Soient $1 \leq k < k'$ deux entiers. Il existe une classe de structures unaires $\mathcal{S}$ de signature $\mathcal{T} \cup \{f, c\}$, telle que : $\mathcal{S} \in \mathcal{G}^+_\wedge[k', k'](\mathcal{T}, f, c) - \mathcal{G}[\omega, k](\mathcal{T}, f, c)$.*

Preuve. Considérons un ensemble $\mathcal{S}$ de $\mathcal{T}$-structure unaires appartenant à la classe NTIME$_{RAM}(n^{k'})$ et n'appartenant pas à NTIME$_{RAM}(n^k)$. (L'existence de $\mathcal{S}$ est garantie par la proposition 2.20).

Par la proposition 2.18, il existe une formule $\Psi \in \mathcal{G}^+_\wedge[k', k'](\mathcal{T}, c, f)$ telle que pour toute $\mathcal{T}$-structure unaire S : $S \in \mathcal{S}$ ssi $\langle S, 0, s\rangle \models \Psi(0, s)$.

Soit $\mathcal{S}''$ la classe des $\mathcal{T} \cup \{c, f\}$-structures unaires modèles de Ψ. Par définition, $\mathcal{S}''$ est dans $\mathcal{G}^+_\wedge[k', k'](\mathcal{T}, f, c)$. Montrons que $\mathcal{S}'' \notin \mathcal{G}[\omega, k](\mathcal{T}, f, c)$.

Soit $\mathcal{S}'$ la classe des $\mathcal{T} \cup \{0, s\}$-structures unaires $\{\langle S, 0, s\rangle, S \in \mathcal{S}\}$. On a clairement l'implication $\mathcal{S}' \in$ NTIME$_{RAM}(n^k) \Rightarrow \mathcal{S} \in$ NTIME$_{RAM}(n^k)$.

De plus, pour toute $\mathcal{T} \cup \{c, f\}$-structure unaire S : $S \in \mathcal{S}'$ ssi ($S \in \mathcal{S}''$ et $(c_S, f_S) = (0, s)$). Comme on peut déterminer en temps linéaire si une structure $\langle n, c, f\rangle$ coïncide avec $\langle n, 0, s\rangle$, on en déduit : $\mathcal{S}'' \in$ NTIME$_{RAM}(n^k) \Rightarrow \mathcal{S}' \in$ NTIME$_{RAM}(n^k)$.

Par conséquent : $\mathcal{S}'' \in$ NTIME$_{RAM}(n^k) \Rightarrow \mathcal{S} \in$ NTIME$_{RAM}(n^k)$. Et donc, puisque $\mathcal{S} \notin$ NTIME$_{RAM}(n^k)$: $\mathcal{S}'' \notin$ NTIME$_{RAM}(n^k)$. D'où finalement, avec la proposition 2.19 : $\mathcal{S}'' \notin \mathcal{G}[\omega, k](\mathcal{T}, f, c)$.

Corollaire 2.22 (Grandjean). *Soient $1 \leq k < k'$ deux entiers. Alors les inclusions $\mathcal{G}^+_\wedge[k,k] \subset \mathcal{G}^+_\wedge[k',k']$, $\mathcal{G}^+_\wedge[\omega,k] \subset \mathcal{G}^+_\wedge[\omega,k']$ et $\mathcal{G}[\omega,k] \subset \mathcal{G}[\omega,k']$ sont strictes.*

Preuve. Les inclusions larges sont claires. Le fait qu'elles soient strictes découle facilement des chaines d'inclusions :

$$\begin{array}{rcccl} \mathcal{G}^+_\wedge[k,k] & \subseteq & \mathcal{G}^+_\wedge[\omega,k] & \subseteq & \mathcal{G}[\omega,k], \\ \mathcal{G}^+_\wedge[k',k'] & \subseteq & \mathcal{G}^+_\wedge[\omega,k'] & \subseteq & \mathcal{G}[\omega,k'], \end{array}$$

et de l'assertion $\mathcal{G}^+_\wedge[k',k'] - \mathcal{G}[\omega,k] \neq \emptyset$ prouvée dans le corollaire 2.21.

2.8 Définissabilité Conjonctive de l'Arithmétique

Dans toute cette section, $\mathcal{T}$ désigne une signature quelconque. On note $+$, $\times$, exp les relations ternaires d'addition, de multiplication et d'exponentielle. Pour chaque formule du premier ordre Φ de signature $\mathcal{T} \cup \{+, \times, \exp\}$, on construit une formule Θ du second ordre existentiel fonctionnel unaire de signature $\mathcal{T} \cup \{0, s\}$, équivalente à Φ sur tout domaine fini $n > 0$.

Pour ce faire, on commence par montrer que l'on ne perd pas de pouvoir d'expression en substituant à $+, \times, \exp$ des relations unaires A, M, E, "accompagnées" d'un ensemble $\mathcal{T}_{ar}$ d'opérations arithmétiques prédéfinies (Corollaire 2.26).

Puis on montre la définissabilité de A, M, E et des opérations de $\mathcal{T}_{ar}$ par des formules de $\mathcal{G}(0,s)$ (Lemmes 2.27 et 2.28).

Ceci permet de construire la formule Θ évoquée plus haut, dans le cas où Φ est une formule du premier ordre (Corollaire 2.29), puis dans le cas où Φ est une formule quelconque (Proposition 2.30).

Ce dernier résultat donne lieu à une généralisation du théorème 2.11. Il entraine en effet la réduction de la classe $\mathcal{F}(\mathcal{T}, 0, s, <, +, \times, \exp)$ à la classe $\mathcal{F}^+_\wedge(\mathcal{T}, 0, s)$ (Théorème 2.31).

A titre d'application, on montre enfin la définissabilité de l'ensemble des entiers pairs (resp. premiers, parfaits, de Fibonacci) par une formule de $\mathcal{F}^+_\wedge(0, s)$ (Corollaire 2.32).

2.8.1 De $+, \times, \exp$ à A, M, E

DÉFINITION 2.10.

- On note m, π_0, π_1, π_2 le symbole de constante et les trois symboles de fonction unaires prédéfinis, dont l'interprétation sur tout domaine $n > 0$ est caractérisée par les assertions : (i) $m = \lfloor \sqrt{n} \rfloor$, (ii) pour tout $x < n$, $x = \pi_0(x) + \pi_1(x)m + \pi_2(x)m^2$ et $\pi_0(x), \pi_1(x), \pi_2(x) < m$.
- On note π le triplet de symboles (π_0, π_1, π_2) et on abrège par $\pi(x) = (x_0, x_1, x_2)$ la formule $\bigwedge_{i<3} \pi_i(x) = x_i$.

- On note A, M, E les trois symboles de fonction unaires prédéfinis interprétés comme suit sur tout domaine $n > 0$: pour tout $x < n$, $A(x) = \pi_0(x) + \pi_1(x)$, $M(x) = \pi_0(x)\pi_1(x)$ et $E(x) = \pi_0(x)^{\pi_1(x)}$.

  Noter qu'en conséquence, ces trois fonctions satisfont : $\forall x, y < m$, $A(x+ym) = x+y$, $M(x+ym) = xy$, $E(x+ym) = x^y$ (en convenant que $x^0 = 1$, même pour $x = 0$).

- On note $PO(\mathcal{T})$ l'ensemble des formules du premier ordre de signature $\mathcal{T}$.

Lemme 2.23. *Pour toute formule $\Phi \in PO(\mathcal{T}, +)$ il existe $\Phi_1 \in PO(\mathcal{T}, 0, s, \pi, s^m, A)$ telle que :*

$$\langle n, \mathcal{T}, + \rangle \models \Phi \text{ ssi } \langle n, \mathcal{T}, 0, s, \pi, s^m, A \rangle \models \Phi_1.$$

De plus, si $\mathbf{q}$ *est le préfixe de Φ, alors le préfixe de Φ_1 peut être supposé de la forme* $\mathbf{q}\exists^*$.

Preuve. Soient $x_0, x_1, x_2, y_0, y_1, y_2$ des entiers strictement inférieurs à m. Puisque la somme des entiers $x_0 + x_1 m + x_2 m^2$ et $y_0 + y_1 m + y_2 m^2$ est strictement inférieure à m^4, elle s'écrit : $(b_0 + b_1 m + b_2 m^2 + b_3 m^3)$, où les b_i sont des entiers $< m$ entièrement déterminés par la formule suivante, écrite sous forme arborescente :

$$x_0 + y_0 = b_0 \wedge \begin{cases} x_1 + y_1 = b_1 \wedge \begin{cases} x_2 + y_2 = b_2 \wedge b_3 = 0 \\ \vee \\ x_2 + y_2 = b_2 + m \wedge b_3 = 1 \end{cases} \\ \vee \\ x_1 + y_1 = b_1 + m \wedge \begin{cases} x_2 + y_2 + 1 = b_2 \wedge b_3 = 0 \\ \vee \\ x_2 + y_2 + 1 = b_2 + m \wedge b_3 = 1 \end{cases} \end{cases}$$
$$\vee$$
$$x_0 + y_0 = b_0 + m \wedge \begin{cases} x_1 + y_1 + 1 = b_1 \wedge \begin{cases} x_2 + y_2 = b_2 \wedge b_3 = 0 \\ \vee \\ x_2 + y_2 = b_2 + m \wedge b_3 = 1 \end{cases} \\ \vee \\ x_1 + y_1 + 1 = b_1 + m \wedge \begin{cases} x_2 + y_2 + 1 = b_2 \wedge b_3 = 0 \\ \vee \\ x_2 + y_2 + 1 = b_2 + m \wedge b_3 = 1 \end{cases} \end{cases}$$

Si de plus on considère des entiers t_0, t_1, t_2, que l'on peut choisir inférieurs à m^2, tels que $\pi_0(t_i) = x_i$ et $\pi_1(t_i) = y_i$ pour $i < 3$, on peut écrire la formule précédente comme suit :

$$
\begin{array}{l}
A(t_0) = b_0 \wedge \left\{ \begin{array}{l}
A(t_1) = b_1 \wedge \left\{ \begin{array}{l} A(t_2) = b_2 \wedge b_3 = 0 \\ \vee \\ A(t_2) = s^m(b_2) \wedge b_3 = 1 \end{array} \right. \\
\vee \\
A(t_1) = s^m(b_1) \wedge \left\{ \begin{array}{l} sA(t_2) = b_2 \wedge b_3 = 0 \\ \vee \\ sA(t_2) = s^m(b_2) \wedge b_3 = 1 \end{array} \right.
\end{array} \right. \\
\vee \\
A(t_0) = s^m(b_0) \wedge \left\{ \begin{array}{l}
sA(t_1) = b_1 \wedge \left\{ \begin{array}{l} A(t_2) = b_2 \wedge b_3 = 0 \\ \vee \\ A(t_2) = s^m(b_2) \wedge b_3 = 1 \end{array} \right. \\
\vee \\
sA(t_1) = s^m(b_1) \wedge \left\{ \begin{array}{l} sA(t_2) = b_2 \wedge b_3 = 0 \\ \vee \\ sA(t_2) = s^m(b_2) \wedge b_3 = 1 \end{array} \right.
\end{array} \right.
\end{array}
$$

Notons ψ la formule ainsi obtenue et $add(\overline{x}, \overline{y}, \overline{b})$ la formule $\exists t_1, t_2, t_3 \psi(\overline{x}, \overline{y}, \overline{b}, \overline{t})$. Alors :

- $add(\overline{x}, \overline{y}, \overline{b})$ est une formule du premier ordre existentielle de signature $\{0, s, s^m, A\}$;
- pour tous tuples (x_0, x_1, x_2), (y_0, y_1, y_2), $(b_0, \ldots, b_4)$ d'entiers $< m$,

$$
\begin{array}{c}
(x_0 + x_1 m + x_2 m^2) + (y_0 + y_1 m + y_2 m^2) = (b_0 + b_1 m + \ldots + b_4 m^4) \\
\text{ssi} \\
m^3 \models add(\overline{x}, \overline{y}, \overline{b})
\end{array}
$$

Soient maintenant $x, y, z < n$. On a clairement :

$$
\begin{array}{rl}
x + y = z \text{ ssi } n \models (\exists \overline{b} < m) : & add(\pi(x), \pi(y), \overline{b}) \wedge \\
& (b_0, b_1, b_2) = \pi(z) \wedge b_3 = 0
\end{array}
$$

et

$$
\begin{array}{rl}
x + y \neq z \text{ ssi } n \models (\exists \overline{b} < m) : & add(\pi(x), \pi(y), \overline{b}) \wedge \\
& \{(b_0, b_1, b_2) \neq \pi(z) \vee b_3 \neq 0\}
\end{array}
$$

On peut écrire, de manière équivalente :

$$
x + y = z \text{ ssi } n \models add(\pi(x), \pi(y), \pi(z), 0)
$$

et

$$
\begin{array}{rl}
x + y \neq z \text{ ssi } n \models (\exists u, v) : & add(\pi(x), \pi(y), \pi(u), \pi_0(v)) \wedge \\
& \{u \neq z \vee \pi_0(v) \neq 0\}
\end{array}
$$

On note respectivement $\pi(x) \oplus \pi(y) = \pi(z)$ et $\pi(x) \oplus \pi(y) \neq \pi(z)$ ces dernières formules. La formule Φ_1 annoncée dans l'énoncé du lemme s'obtient alors à partir de Φ en remplaçant chaque sous-formule de la forme $x + y = z$ (resp $x + y \neq z$) par $\pi(x) \oplus \pi(y) = \pi(z)$ (resp $\pi(x) \oplus \pi(y) \neq \pi(z)$).

Remarque. Les formules $\pi(x) \oplus \pi(y) = \pi(z)$ et $\pi(x) \oplus \pi(y) \neq \pi(z)$ sont des formules du premier ordre existentiel de signature $\{0, s, s^m, \pi, A\}$.

Lemme 2.24. *Pour toute formule $\Phi \in PO(\mathcal{T}, \times)$ il existe $\Phi_2 \in PO(\mathcal{T}, 0, s, \pi, M, A)$ telle que :*

$$\langle n, \mathcal{T}, \times \rangle \models \Phi \ \textit{ssi} \ \langle n, \mathcal{T}, 0, s, \pi, M, A \rangle \models \Phi_2.$$

De plus, si $\mathbf{q}$ *est le préfixe de Φ, alors le préfixe de Φ_2 peut être supposé de la forme* $\mathbf{q}\exists^*$.

Preuve. Soient $x_o, x_1, x_2, y_o, y_1, y_2$ des entiers strictement plus petits que m. Le produit des entiers $x = x_o + x_1 m + x_2 m^2$ et $y = y_o + y_1 m + y_2 m^2$ peut s'écrire, suivant le formalisme bien connu :

$$\begin{array}{cccccc}
 & & & x_2 & x_1 & x_0 \\
 & & \times & y_2 & y_1 & y_0 \\
\hline
 & & \beta_{03} & \beta_{02} & \beta_{01} & \beta_{00} \\
 & \beta_{13} & \beta_{12} & \beta_{11} & \beta_{10} & \\
\beta_{23} & \beta_{22} & \beta_{21} & \beta_{20} & & \\
\hline
b_5 & b_4 & b_3 & b_2 & b_1 & b_0
\end{array}$$

En explicitant cette écriture, et en particulier en exhibant les retenues, on obtient la formulation équivalente : les entiers $b_0, \ldots, b_5 < m$ sont les composantes en base m du produit de $x_o + x_1 m + x_2 m^2$ par $y_o + y_1 m + y_2 m^2$ si et seulement si il existe trois suites finies, $(\alpha_{ij})_{i,j<3}$, $(\beta_{ij})_{i<3,j<4}$, $(a_i)_{0<i<5}$, à valeurs dans m, qui satisfont les conjonctions suivantes :

$$\left\{\begin{array}{rrcl}
 & y_i x_0 & = & \alpha_{i0} m + \beta_{i0} \\
\wedge & y_i x_1 + \alpha_{i0} & = & \alpha_{i1} m + \beta_{i1} \\
\wedge & y_i x_2 + \alpha_{i1} & = & \alpha_{i2} m + \beta_{i2} \\
\wedge & \alpha_{i2} & = & \beta_{i3}
\end{array}\right. \quad \text{(pour } i = 0, 1, 2)$$

et

$$\left\{\begin{array}{rrcl}
 & \beta_{00} & = & b_0 \\
\wedge & \beta_{01} + \beta_{10} & = & a_1 m + b_1 \\
\wedge & \beta_{02} + \beta_{11} + \beta_{20} + a_1 & = & a_2 m + b_2 \\
\wedge & \beta_{03} + \beta_{12} + \beta_{21} + a_2 & = & a_3 m + b_3 \\
\wedge & \beta_{13} + \beta_{22} + a_3 & = & a_4 m + b_4 \\
\wedge & \beta_{23} + a_4 & = & b_5
\end{array}\right.$$

On peut écrire, de façon équivalente : Il existe des suites $(\alpha_{ij})_{i,j<3}$, $(\beta_{ij})_{i<3,j<4}$ et $(a_i)_{0<i<5}$ à valeurs dans m, et $(t_{ij})_{i,j<3}$ à valeurs dans n telles que :

$$\bigwedge_{i,j<3} \pi_0(t_{ij}) = y_i \wedge \pi_1(t_{ij}) = x_j$$

et

$$\bigwedge_{i<3} \begin{cases} \pi_0 M(t_{i0}) = \beta_{i0} \ \wedge \ \pi_1 M(t_{i0}) = \alpha_{i0} \\ \pi_0(M(t_{i1}) + \alpha_{i0}) = \beta_{i1} \ \wedge \ \pi_1(M(t_{i1}) + \alpha_{i0}) = \alpha_{i1} \\ \pi_0(M(t_{i2}) + \alpha_{i1}) = \beta_{i2} \ \wedge \ \pi_1(M(t_{i2}) + \alpha_{i1}) = \alpha_{i2} \\ \alpha_{i2} = \beta_{i3} \end{cases}$$

et

$$\begin{cases} \beta_{00} = b_0 \\ \pi_0(\beta_{01} + \beta_{10}) = b_1 \wedge \pi_1(\beta_{01} + \beta_{10}) = a_1 \\ \pi_0(\beta_{02} + \beta_{11} + \beta_{20} + a_1) = b_2 \wedge \pi_1(\beta_{02} + \beta_{11} + \beta_{20} + a_1) = a_2 \\ \pi_0(\beta_{03} + \beta_{12} + \beta_{21} + a_2) = b_3 \wedge \pi_1(\beta_{03} + \beta_{12} + \beta_{21} + a_2) = a_3 \\ \pi_0(\beta_{13} + \beta_{22} + a_3) = b_4 \wedge \pi_1(\beta_{13} + \beta_{22} + a_3) = b_4 \\ \beta_{23} + a_4 = b_5 \end{cases}$$

Finalement, en intoduisant de nouvelles variables existentiellement quantifiées pour supprimer les termes de la forme $\pi_i(a+b)$ incompatibles avec la nature relationnelle de l'addition, puis en remplaçant toute sous formule du type $\lambda_1+\lambda_2 = \lambda_3$ (resp $\lambda_1+\lambda_2 \neq \lambda_3$) de la formule ainsi obtenue par $\pi(\lambda_1)\oplus\pi(\lambda_2) = \pi(\lambda_3)$ (resp $\pi(\lambda_1)\oplus\pi(\lambda_2) \neq \pi(\lambda_3)$), on obtient une formule $mult(\overline{x}, \overline{y}, \overline{b})$ telle que :

- $mult(\overline{x}, \overline{y}, \overline{b})$ est une formule du premier ordre existentiel de signature $\{\pi_0, \pi_1, M, A\}$;
- pour tous tuples (x_0, x_1, x_2), (y_0, y_1, y_2), $(b_0, \ldots, b_5)$ d'entiers $< m$,

$$(x_0 + x_1 m + x_2 m^2)(y_0 + y_1 m + y_2 m^2) = (b_0 + b_1 m + \ldots + b_5 m^5)$$
$$\text{ssi}$$
$$m^3 \models mult(\overline{x}, \overline{y}, \overline{b})$$

Soient maintenant $x, y, z < n$. On a clairement :

$$xy = z \text{ ssi } n \models (\exists \overline{b} < m) : \ mult(\pi(x), \pi(y), \overline{b}) \wedge (b_0, b_1, b_2) = \pi(z) \wedge b_3 = b_4 = b_5 = 0$$

et

$$xy \neq z \text{ ssi } n \models (\exists \overline{b} < m) : \ mult(\pi(x), \pi(y), \overline{b}) \wedge \{(b_0, b_1, b_2) \neq \pi(z) \vee b_3 + b_4 + b_5 \neq 0\}$$

Ce que l'on peut aussi écrire :

$$xy = z \text{ ssi } n \models (\exists u, v) : \ mult(\pi(x), \pi(y), \pi(u), \pi(v)) \wedge u = z \wedge v = 0$$

et

$$xy \neq z \text{ ssi } n \models (\exists u, v) : \begin{array}{l} mult(\pi(x), \pi(y), \pi(u), \pi(v)) \wedge \\ \{u \neq z \vee v \neq 0\} \end{array}$$

On note respectivement $\pi(x) \otimes \pi(y) = \pi(z)$ et $\pi(x) \otimes \pi(y) \neq \pi(z)$ ces dernières formules. La formule Φ_2 annoncée dans l'énoncé du lemme s'obtient alors à partir de Φ en remplaçant chaque sous-formule de la forme $xy = z$ (resp $xy \neq z$) par $\pi(x) \otimes \pi(y) = \pi(z)$ (resp $\pi(x) \otimes \pi(y) \neq \pi(z)$).

Remarque. Les formules $\pi(x) \otimes \pi(y) = \pi(z)$ et $\pi(x) \otimes \pi(y) \neq \pi(z)$ sont des formules du premier ordre existentiel de signature $\{0, s, s^m, \overline{\pi}, A, M\}$.

Lemme 2.25. *Pour toute formule $\Phi \in PO(\mathcal{T}, \exp)$ il existe $\Phi_3 \in PO(\mathcal{T}, 0, s, \pi, E)$ telle que :*

$$\langle n, \mathcal{T}, \exp\rangle \models \Phi \textit{ ssi } \langle n, \mathcal{T}, 0, s, \pi, E\rangle \models \Phi_3.$$

De plus, si $\mathbf{q}$ *est le préfixe de* Φ*, alors le préfixe de* Φ_3 *peut être supposé de la forme* $\mathbf{q}\exists^*$.

Preuve. Considérons trois entiers $x, y, z < n$. L'égalité $x^y = z$ impose évidement $x^y < n$, et donc ($x < 2$ ou $y < 2$) ou ($x < \lfloor\sqrt{n}\rfloor = m$ et $y < \log_2(n) \leq m$). Compte tenu de la définition de E on a, plus précisément :

$$z = x^y \text{ ssi } \left[\begin{array}{llllll} & \{x < m & \wedge & y < m & \wedge & z = E(x + ym)\} \\ \vee & \{x = 0 & \wedge & y \geq m & \wedge & z = 0\} \\ \vee & \{x = 1 & \wedge & y \geq m & \wedge & z = 1\} \\ \vee & \{x \geq m & \wedge & y = 0 & \wedge & z = 1\} \\ \vee & \{x \geq m & \wedge & y = 1 & \wedge & z = x\} \\ \vee & \{x = m & \wedge & y = 2 & \wedge & z = m^2\} \end{array}\right]$$

Soit alors $EXP(x, y, z)$ et $\sim EXP(x, y, z)$ les formules suivantes :

$$EXP \equiv \exists a : \left[\begin{array}{cccccc} & \{x = \pi_0(a) & \wedge & y = \pi_1(a) & \wedge & z = E(a)\} \\ \vee & \{x = 0 & \wedge & y \geq m & \wedge & z = 0\} \\ \vee & \{x = 1 & \wedge & y \geq m & \wedge & z = 1\} \\ \vee & \{x \geq m & \wedge & y = 0 & \wedge & z = 1\} \\ \vee & \{x \geq m & \wedge & y = 1 & \wedge & z = x\} \\ \vee & \{x = m & \wedge & y = 2 & \wedge & \pi(z) = (0, 0, 1)\} \end{array}\right]$$

$$\sim EXP \equiv \exists a : \left[\begin{array}{cccccc} & \{x = \pi_0(a) & \wedge & y = \pi_1(a) & \wedge & z \neq E(a)\} \\ \vee & \{x = 0 & \wedge & y \geq m & \wedge & z \neq 0\} \\ \vee & \{x = 1 & \wedge & y \geq m & \wedge & z \neq 1\} \\ \vee & \{x \geq m & \wedge & y = 0 & \wedge & z \neq 1\} \\ \vee & \{x \geq m & \wedge & y = 1 & \wedge & z \neq x\} \\ \vee & \{x = m & \wedge & y = 2 & \wedge & \pi(z) \neq (0, 0, 1)\} \\ \vee & \{x > m & \wedge & y > 1\} & & \\ \vee & \{x = m & \wedge & y > 2\} & & \end{array}\right]$$

dans lesquelles les expressions $x \geq m$, $x > m$, $x = m$ abrègent, respectivement, les formules : $\pi_1(x) \neq 0 \vee \pi_2(x) \neq 0$; $\pi_1(x) \neq 0 \vee \pi_2(x) \neq 0 \wedge \pi(x) \neq (0,1,0)$; $\pi(x) = (0,1,0)$. Alors on a clairement, pour tous $x, y, z < n$:

$\langle n, \exp\rangle \models \exp(x,y,z)$ ssi $\langle n, E, \times, \pi, m\rangle \models EXP(x,y,z)$ et

$\langle n, \exp\rangle \models \neg\exp(x,y,z)$ ssi $\langle n, E, \times, \pi, m\rangle \models \sim EXP(x,y,z)$

En conséquence, Φ est équivalente sur n à la formule Φ_3 obtenue en remplaçant chaque formule $\exp(t_1,t_2,t_3)$ (resp $\neg\exp(t_1,t_2,t_3)$) de Φ (où t_1, t_2, t_3 sont des termes du langage $\mathcal{T}$) par la formule $EXP(t_1,t_2,t_3)$ (resp $\sim EXP(t_1,t_2,t_3)$).
Comme de plus les formules EXP et $\sim EXP$ sont des formules du premier ordre existentiel de signature $\{0, s, \pi, E\}$, la formule Φ_3 a bien le préfixe de quantificateurs et la signature annoncés.

Corollaire 2.26. *Pour toute formule* $\Phi \in PO(\mathcal{T}, +, \times, \exp)$ *il existe une formule* $\Psi \in PO(\mathcal{T}, 0, s, \pi, s^m, E, M, A)$ *telle que :*

$$\langle n, \mathcal{T}, +, \times, \exp\rangle \models \Phi \text{ ssi } \langle n, \mathcal{T}, 0, s, \pi, s^m, E, M, A\rangle \models \Psi.$$

De plus, si $\mathbf{q}$ *est le préfixe de* Φ*, alors le préfixe de* Ψ *peut être supposé de la forme* $\mathbf{q}\exists^*$*.*

2.8.2 Définissabilité de π, m, s^m, E, M, A

Lemme 2.27 (définissabilité de E, M, A, s^m***).*** *Soient* f_E, f_M, f_A, σ *quatre symboles de fonction unaires. Il existe une formule du premier ordre* $\mathcal{D}$ *de signature* $(s, 0, \pi, m, f_E, f_M, f_A, \sigma)$ *telle que pour tout* $n \in I\!N$ *et pour tout quadruplet de fonctions unaires sur* n*,* (f_E, f_M, f_A, σ) *:*

$$\langle n, f_E, f_M, f_A, \sigma\rangle \models \mathcal{D} \text{ ssi } f_E = E \text{ et } f_M = M \text{ et } f_A = A \text{ et } \sigma = s^m.$$

De plus, le préfixe de $\mathcal{D}$ *est de la forme* $\forall\exists^*$*.*

Preuve. Rappelons que pour tous $a, b < m$, $E(a+bm) = a^b$, $M(a+bm) = a.b$ et$A(a+bm) = a+b$. Ceci permet de caractérisér les fonctions s^m, E, M, A par les assertions suivantes :

$\forall x: \ s^m s(x) = ss^m(x) \wedge s^m(0) = m$

$$\begin{aligned}
\forall x: \ & \pi_1(x) = 0 \rightarrow E(x) = 1 \ \wedge \\
& \pi_1(x) < m-1 \rightarrow E(s^m(x)) = \pi_0(x)E(x) \\
\forall x: \ & \pi_1(x) = 0 \rightarrow M(x) = 0 \ \wedge \\
& \pi_1(x) < m-1 \rightarrow M(s^m(x)) = \pi_0(x) + M(x) \\
\forall x: \ & \pi_1(x) = 0 \rightarrow A(x) = \pi_0(x) \ \wedge \\
& \pi_1(x) < m-1 \rightarrow A(s^m(x)) = sA(x)
\end{aligned}$$

Soient D_{s^m}, D_E, D_M, D_A les formules du premier ordre de signature $\{0, s, \pi, m, \sigma, f_E, f_M, f_A\}$:

- $D_{s^m}(\sigma) \equiv \forall x: \ \sigma s(x) = s\sigma(x) \wedge \sigma(0) = m$

– $D_E(f_E, f_M, f_A) \equiv \ \forall x : \ \ \pi_1(x) = 0 \to f_E(x) = 1 \ \wedge$
$s\pi_1(x) \neq m \to \pi(f_E\sigma(x)) = \pi(\pi_0(x)) \otimes \pi(f_E(x))$

– $D_M(f_M, f_A) \equiv \ \forall x : \ \ \pi_1(x) = 0 \to f_M(x) = 0 \ \wedge$
$s\pi_1(x) \neq m \to \pi(f_M\sigma(x)) = \pi(\pi_0(x)) \oplus \pi(f_M(x))$

– $D_A(f_A) \equiv \ \forall x : \ \ \pi_1(x) = 0 \to f_A(x) = \pi_0(x) \ \wedge$
$s\pi_1(x) \neq m \to f_A\sigma(x) = sf_A(x)$

Alors la conjonction $D_{s^m}(\sigma) \wedge D_A(f_A) \wedge D_M(f_M, f_A) \wedge D_E(f_E, f_M, f_A)$ est la formule $\mathcal{D}(f_E, f_M, f_A, \sigma)$ annoncée.

Remarque. Le préfixe de la formule $\mathcal{D}$ est bien de la forme $\forall\exists^*$: les quantificateurs existentiels sont introduits par les sous formules $\pi(\pi_0(x)) \otimes \pi(f_M(x))$ et $\pi(\pi_0(x)) \oplus \pi(f_M(x))$ qui interviennent positivement dans $\mathcal{D}$.

Lemme 2.28 (définissabilité de m, π_0, π_1, π_2***).*** *Soit $\mathcal{T}$ une signature contenant un symbole de constante et trois symboles de fonction unaires. Il existe une formule $\mathcal{D}' \in PO(\mathcal{T}, 0, s)$ telle que pour tout entier $n \geq 4$, pour tout $b < n$ et pour tout triplet (p_0, p_1, p_2) de fonctions unaires sur n :*

$$\langle n, b, p_0, p_1, p_2, 0, s\rangle \models \mathcal{D}' \text{ ssi } b = m \text{ et } (p_0, p_1, p_2) = (\pi_0, \pi_1, \pi_2)$$

De plus, le préfixe de $\mathcal{D}'$ est de la forme $\forall$.

Preuve. Abrégeons par $p(x)$ le triplet $(p_0(x), p_1(x), p_2(x))$ et par $p(x) = (x_0, x_1, x_2)$ la formule $\bigwedge_{i<3} p_i(x) = x_i$. Considérons la formule :

$$\begin{array}{l} p(0) = (0,0,0) \wedge \forall x \ \ : \\ x = \max \ \vee \\ \{p_0(x) \neq b-1 \wedge p(s(x)) = (s(p_0(x)), p_1(x), p_2(x))\} \ \vee \\ \{p_0(x) = b-1 \wedge p_1(x) \neq b-1 \wedge p(s(x)) = (0, s(p_1(x)), p_2(x))\} \ \vee \\ \{p_0(x) = b-1 \wedge p_1(x) = b-1 \wedge p_2(x) \neq b-1 \ \wedge \\ \qquad\qquad p(s(x)) = (0, 0, s(p_2(x)))\}. \end{array} \tag{2.14}$$

On montre facilement par récurrence sur x que la formule (2.14) exprime le fait suivant : *pour tout $x < n$, $x = p_0(x) + p_1(x)b + p_2(x)b^2$, et $\bigwedge_{i<3} p_i(x) < b$. En particulier,* $\max < b^3$*, donc $n \leq b^3$.*

Il reste donc, pour forcer l'égalité $p = \pi$, à exprimer l'égalité $b = m (= \lfloor\sqrt{n}\rfloor)$, c'est à dire les inégalités : $b^2 \leq n < (b+1)^2$.

Mais ces inégalités équivalent à $(b-1) + (b-1)b \leq n - 1 = \max \leq (b-1) + b + b^2$, c'est à dire à $(b-1, b-1, 0) \preceq p(\max) \preceq (b-1, 1, 1)$, où $\preceq$ est l'ordre lexicographique sur les triplets de $(n)^3$. L'égalité $b = m$ s'exprime donc par la formule :

$$p(\max) = (b-1, b-1, 0) \vee [p_2(\max) = 1 \wedge (p_1(\max) = 0 \vee p_1(\max) = 1)]. \tag{2.15}$$

Finalement, la conjonction des formules (2.14) et (2.15) implique les égalités $(p_0, p_1, p_2) = (\pi_0, \pi_1, \pi_2)$ et $b = m$. La formule $\mathcal{D}'$ cherchée s'obtient à partir de cette conjonction en explicitant la constante max (grâce à l'équivalence $x = \max$ ssi $s(x) = x$), ce qui se fait sans modifier le préfixe obtenu ($\forall$).

Remarque. Pour coder tout entier $x < n$ en un triplet (x_0, x_1, x_2) d'entiers strictement inférieurs à $m = \lfloor\sqrt{n}\rfloor$, il faut que $n \leq m^3$. Il suffit pour cela que $(m+1)^2 - 1 \leq m^3$ puisque par définition de m, $n < (m+1)^2$. Or cela impose $m \geq 2$. D'où la contrainte "$n \geq 4$" dans l'énoncé du lemme précédent.

Corollaire 2.29. *Soient $\mathcal{T}$ une signature quelconque et $\Phi \in PO(\mathcal{T}, +, \times, \exp)$. Alors il existe une formule du second ordre Θ de signature $\mathcal{T} \cup \{0, s\}$ telle que :*

$$\langle n, \mathcal{T}, +, \times, \exp\rangle \models \Phi \text{ ssi } \langle n, \mathcal{T}, 0, s\rangle \models \Theta.$$

De plus si le préfixe $\mathbf{q}$ de Φ contient au moins un $\forall$, Θ peut s'écrire sous la forme :

$$\Theta \equiv \exists f_1, \ldots, f_8 \mathbf{q}\overline{x}\ \exists\overline{y}\ :\ \psi(\overline{x}, \overline{y}), \tag{2.16}$$

où les f_i sont des symboles de fonctions unaires et ψ une formule du premier ordre sans quantificateur de signature $\mathcal{T} \cup \{0, s, \overline{f}\}$.

Preuve. Soit Ψ la formule de $PO(\mathcal{T}, 0, s, \pi, s^m, A, M, E)$ construite à partir de Φ dans le corollaire 2.26. Soient $\mathcal{D}(0, s, \pi, f_A, f_M, f_E, \sigma)$ et $\mathcal{D}'(0, s, p, b)$ les formules construites dans les lemmes 2.27 et 2.28. Alors Φ équivaut à :

$$\begin{gathered}\exists p_0, p_1, p_2, b, f_A, f_M, f_E, \sigma\ :\\ \mathcal{D}'(0, s, p, b) \wedge \mathcal{D}(0, s, p, f_A, f_M, f_E, \sigma) \wedge \Psi(\mathcal{T}, 0, s, p, \sigma, f_A, f_M, f_E).\end{gathered}$$

C'est la formule Θ annoncée. On montre facilement, à l'aide des remarques faites sur les préfixes de $\mathcal{D}'$, $\mathcal{D}$ et Ψ dans les lemmes précédents, que l'on peut se ramener à la forme (2.16) en renommant eventuellement les variables.

Ce résultat se généralise facilement au cas des formules du second ordre de la façon suivante :

Proposition 2.30. *Soient $\mathcal{T}$ une signature quelconque et Φ une formule du second ordre de signature $\{\mathcal{T}, +, \times, \exp\}$. Alors il existe une formule du second ordre Θ de signature $\mathcal{T} \cup \{0, s\}$ telle que :*

$$\langle n, \mathcal{T}, +, \times, \exp\rangle \models \Phi \text{ ssi } \langle n, \mathcal{T}, 0, s\rangle \models \Theta.$$

De plus si Φ est de la forme $\mathbf{Q}\overline{R}\ \mathbf{q}\overline{x}\ \phi$, où $\overline{R}$ est une signature quelconque, $\mathbf{q}$ un préfixe contenant au moins un $\forall$, et ϕ une formule sans quantificateur de signature $\mathcal{T} \cup \{\overline{R}, +, \times, \exp\}$, alors Θ peut s'écrire sous la forme :

$$\Theta \equiv \mathbf{Q}\overline{R}\exists f_1, \ldots, f_8 \mathbf{q}\overline{x}\ \exists\overline{y}\ :\ \psi(\overline{x}, \overline{y}), \tag{2.17}$$

où les f_i sont des symboles de fonctions unaires et ψ une formule du premier ordre sans quantificateur de signature $\mathcal{T} \cup \{0, s\} \cup \{\overline{f}, \overline{R}\}$

Preuve. Immédiat.

De la proposition 2.30 et du théorème 2.11 on déduit :

Théorème 2.31. *Soit $\mathcal{T}$ une signature unaire. Pour toute $\Phi \in \mathcal{G}(\mathcal{T}, 0, s, <, +, \times, \exp)$ il existe $\Psi \in \mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ telle que pour tout $n > 9$:*

$$\langle n, \mathcal{T}, 0, s, <, +, \times, \exp\rangle \models \Phi \text{ ssi } \langle n, \mathcal{T}, 0, s\rangle \models \Psi$$

Preuve. En appliquant la proposition 2.30 au cas où $\mathbf{Q} \in \{\exists\}^*$; les R_i sont des fonctions unaires ; $\mathbf{q} = \forall$, on passe de Φ à une formule Φ' du type : $\exists\overline{g}\exists\overline{f}\forall x\exists\overline{y}\ \psi(x, \overline{y})$, où $\overline{g}$ et $\overline{f}$ sont des suites de symboles de fonction unaires et ψ une formule sans quantificateur. Puis on passe de Φ' à une formule Ψ' de $\mathcal{G}(0, s)$ par Skolémisation du tuple $\overline{y}$. Enfin Ψ' se réduit à une formule Ψ de $\mathcal{G}^+_\wedge(\mathcal{T}, 0, s)$ grâce au théorème 2.11.

Remarque. Il est facile de voir que ce résultat reste vrai si l'on remplace les *relations* $+, \times, \exp$ par les *fonctions* correspondantes (*i.e.* définies par $z = x{*}y$ ssi $*(x, y, z) \vee z = \max$, pour $* \in \{+, \times, \exp\}$).

DÉFINITION 2.11. On note PAIRS l'ensemble des entiers pairs strictement supérieur à 9. On définit de la même façon les ensembles FIBONACCI, PREMIERS et PARFAITS.

Corollaire 2.32. *Les ensembles* PAIRS, FIBONACCI, PREMIERS *et* PARFAITS *sont dans spectre*$(\mathcal{G}^+_\wedge(0, s))$.

Preuve. Compte tenu du théorème 2.31, il suffit de montrer que ces ensembles sont définissables par des formules de $\mathcal{G}(\mathcal{T}, 0, s, <, +, \times, \exp)$.

PAIRS *:* Cet ensemble est clairement le spectre de la formule :

$$\mathcal{R}_2 = 0,$$

où $\mathcal{R}_2$ est le symbole de constante prédéfini interprété sur tout domaine $n > 0$ par $n \bmod 2$. Or on sait que cette constante est définissable dans $\mathcal{G}^+_\wedge(0, s)$ (voir Section 2.3, paragraphe 15, p. 68).

FIBONACCI *:* La suite de Fibonacci est définie par la relation récurrente : $f(0) = 0$, $f(1) = 1$ et $\forall x \in \mathbb{N},\ f(x + 2) = f(x + 1) + f(x)$. Ceci permet de caractériser ses éléments non nuls par la formule :

$$\begin{aligned} \exists f \exists p \forall x\ :\ f(0) = 0 \wedge f(1) = 1 \wedge f(x{+}2) = f(x{+}1) + f(x) \\ \wedge\ \max - 1 = f(p - 1) + (f(p) - 2). \end{aligned}$$

PREMIERS : Notons q et r les fonctions unaires définies sur chaque domaine $n > 9$ par les assertions : $\forall 2 \leq x < n$, $n = q(x)x + r(x)$ et $q(i) = r(i) = n - 1$ pour $i = 1, 2$. Alors q et r sont définissables sur n par la formule :

$$\begin{array}{rl} D(q,r) \equiv \forall x: & q(0) = q(1) = r(0) = r(1) = \max \ \wedge \\ & \left\{ \begin{array}{l} (r(x) = 0 \ \wedge \ \max -1 = (q(x) - 1)x + (x - 2)) \vee \\ (r(x) = 1 \ \wedge \ \max = (q(x) - 1)x + (x - 1)) \vee \\ (r(x) \geq 2 \ \wedge \ \max -1 = q(x)x + (r(x) - 2)) \end{array} \right\} \end{array}$$

Ceci permet de caractériser PREMIERS par la formule :

$$\exists q, r \ : \ D(q,r) \wedge \ \forall x (r(x) \neq 0).$$

PARFAITS : Un entier n est parfait s'il est la somme de ses diviseurs stricts.

Notons σ la fonction $n \rightarrow n$ qui à tout $x < n$ associe la somme des diviseurs de n inférieurs ou égaux à x. Considérant les fonctions q et r évoquées plus haut, on définit σ par la formule :

$$\begin{array}{rl} D'(\sigma) \equiv & \sigma(0) = 0 \ \wedge \ \sigma(1) = 1 \ \wedge \\ & \forall x \neq 0 : \sigma(x+1) = \sigma(x) + \mathcal{X}_0(r(x+1))(x+1), \end{array}$$

où $\mathcal{X}_0$ désigne la fonction caractéristique du singleton $\{0\}$.

L'ensemble PARFAITS est alors clairement caractérisé par la formule :

$$\exists q, r, \sigma \ : D(q,r) \wedge D'(\sigma) \wedge \sigma(\max) - 2 = \max -1.$$

Conclusion

La réduction de $\mathcal{G}(0, s, <)$ à $\mathcal{G}^+_\wedge(0, s)$ est un résultat de *normalisation* de la classe $\mathcal{G}(0, s, <)$ qui peut être compris comme une "skolémisation forte" des formules du type

$$\exists g_1 \ldots \exists g_p \forall x \exists y_1 \ldots \exists y_q \ : \ \phi(x, \overline{y}, \overline{g}, 0, s, <).$$

Pour de telles formules en effet, on peut éliminer "toutes formes" de disjonction : celle liée à la quantification existentielle au premier ordre, grâce à la skolémisation usuelle ; celle apparaissant dans la matrice de la formule, grâce à notre réduction.

De plus cette normalisation peut être qualifiée de *minimale*, en ce sens que la signature prédéfinie préservée dans la classe normalisée ($\{0, s\}$) parait irréductible. Nous reviendrons sur cet aspect dans la conclusion générale.

La réduction de $\mathcal{G}(0, s, <, +, \times, \exp)$ à $\mathcal{G}(0, s, <)$ est un résultat d'*extension* de la classe $\mathcal{G}(0, s, <)$: on ne modifie pas le pouvoir d'expression de cette logique en ajoutant toute la puissance des opérateurs arithmétique usuels (puissance toutefois limitée par le fait que ces opérateurs agissent sur une unique variable).

Notons que cette extension permet de retrouver très simplement, *via* la caractérisation exacte NLIN $= \mathcal{G}(0, s, <)$, la calculabilité en temps linéaire des ensembles PAIRS, PREMIERS, PARFAITS et FIBONACCI, pour un codage des entiers *en langage bâton.*

Enfin les égalités $\mathcal{G}(0, s, <) = \mathcal{G}_{\wedge}^{+}(0, s) = \mathcal{G}(0, s, <, +, \times, \exp)$, auxquelles s'ajoute la réduction $\mathcal{G}(0, s, <) = \mathcal{G}(\emptyset)$ prouvée par Grandjean dans [33], témoigne de la *très grande stabilité* de la logique $\mathcal{G}(0, s, <)$.

Du fait de l'égalité NLIN $= \mathcal{G}(0, s, <)$, cette stabilité "s'étend" à NLIN et confirme la *robustesse* de cette classe. Conformément à la discussion menée dans l'introduction générale, ceci renforce la "légitimité" de la modélisation du temps linéaire non déterministe proposée par Grandjean.

Chapitre 3

Relations rudimentaires

3.1 Rudimentary Languages and Second-Order Logic

3.1.1 Introduction

This paper contains two related parts. The first part gives an arithmetical proof of a descriptive complexity result about rudimentary languages. The second part is devoted to a representation result of second-order logic into rudimentary sets.

Descriptive complexity is interested in providing logical characterizations for various complexity classes of languages, in the following sense: a class B of formulas is said to be a logical characterization of a class A of languages when any element of A corresponds to the set of all finite models of a formula in B and conversely. For instance, the class of languages in NP is logically characterized by the class of existential second-order formulas (see [20]). Similarly, the class of regular languages corresponds to monadic existential second-order logic with linear order (see [4]), the class of non-deterministic linear-time languages is captured by the existential monadic second-order logic with addition (see [52]) and the languages in Stockmeyer's polynomial hierarchy correspond to second-order logic (see [68] and [44]).

The *Rudimentary relations* are the relations over words (over some finite alphabet) obtained from concatenation using boolean operations, bounded quantifications and explicit transformations (see [61] and [66]). The class of all rudimentary relations, seen as relations over $\mathbb{N}$ (using the p-adic representation of integers) is independant of the alphabet (see [3]). Also, it was proved (see [66], [3] and [42]) that a relation over integers is rudimentary iff it is defined by a bounded arithmetical formula, i.e. a formula of signature $\{+, \times\}$, in which all quantifications are bounded by some polynomial of the free variables. For instance, the set of prime numbers is defined by the following bounded arithmetical formula : x is prime if and only if $x > 1 \wedge [\forall y < x \; \forall z < x \; \neg(x = y \times z)]$

In this paper, we show that the class of rudimentary languages (i.e. unary rudimentary relations) is logically characterized by monadic second-order logic with addition. Our proof uses arithmetical tools, but the result could also be obtained in an indirect way using complexity tools. Indeed, rudimentary languages also have a complexity characterization (see [75]), and in turn the corresponding complexity class can be proved to correspond to monadic second-order logic with addition, using results of [52] and [44].

The *spectrum* of a first-order sentence is the set of cardinalities of all finite models of this formula. It is known that rudimentary sets are spectra and that equality would imply $NP \neq co-NP$ and the collapsing of Stockmeyer's hierarchy (see [74]). On the other hand, a converse representation result is known: we say that a set A of sets of integers is *represented* in a set B of sets of integers if there is a one-to-one function from $\mathbb{N}$ to $\mathbb{N}$ transforming any element of A into an element of B. If A is the spectrum of a sentence of k-ary signature then 2^{A^k} is a rudimentary set and the replacement of any of the exponential functions (for $k \geq 2$) by some polynomial function would imply that all spectra are rudimentary sets (see [56]).

In this paper, we generalize the representation theorem above to a larger class of formulas and prove a converse result. The *(generalized) spectrum* of a second order formula is the set of the cardinalities of all finite models of this formula. We show that a set of integers A is the spectrum of a second-order formula involving quantified relation symbols of arity $k \geq 2$ at most (with no extra-predicate) if and only if 2^{A^k} is a rudimentary set.

Finally, we list some consequences of the two previous results, concerning complexity theory and bounded arithmetics.

3.1.2 Definitions and Theorems

The purpose of this paragraph is to set up the main definitions formalizing the connexion between integers, words and finite structures. So doing, we give a precise meaning to the notion of *logical definability* of a language and we precisely settle our results.

Except in last section, we only consider languages over the dyadic alphabet.

Definition 3.1. Let σ be a signature, an let k be a non-zero integer. We call $SO_k(\sigma)$ the set of second-order formulas of arity k and of signature σ. Hence, the formulas in $SO_k(\sigma)$ are of the form $Q_1R_1 \dots Q_pR_p\Psi$, where $Q_i \in \{\forall, \exists\}$, R_i is a k-ary relation symbol, and Ψ is a first-order formula of signature $\sigma \cup \{R_1, \dots, R_p\}$.

Definition 3.2. Let n be a non-zero integer. We denote by $ld(n)$ its *dyadic length*, i.e. we have $n = \sum_{i=0}^{ld(n)-1} n_i.2^i$ with $n_i \in \{1,2\}$. Note that $ld(n) \geq 1$ because n is non-zero. We denote by w_n the *dyadic notation for n*, i.e. the word $n_0n_1 \dots n_{ld(n)-1}$.

Definition 3.3. Let w be a finite word over $\{1,2\}$ and n be a non-zero integer such that $w = w_n$. We denote by $\mathcal{2}_w$ the unary relation over $\{0, \dots ld(n)-1\}$ defined by:

$i \in \mathit{2}_w$ iff the digit of weight 2^i of w is 2. Let σ be a finite set of predefined relations. We denote by $\mathcal{R}_w(\sigma)$ the structure $\langle ld(n), \mathit{2}_w, \sigma\rangle$.

Definition 3.4. Let k be a non-zero integer and let σ be a finite set of predefined relations. We say that a language $L \subseteq \{1,2\}^*$ is *definable* in $SO_k(\sigma)$ when there exists a formula Θ in $SO_k(\sigma \cup X)$ (where X is a unary relation symbol) such that: $w \in L$ iff $\mathcal{R}_w(\sigma) \models \Theta$.

Definition 3.5. For each set of integers $A \subseteq \mathbb{N}$, we call *dyadic language of A* the set of dyadic notations of elements of A (denoted by $L_2(A)$).

Definition 3.6. Let us denote by RUD the class of unary rudimentary relations over integers. The class of *rudimentary languages*, denoted by $\mathcal{L}_2(\mathrm{RUD})$, is the class of the dyadic languages of the rudimentary sets.

Remark. In the sequel, order (denoted by $<$) and addition (denoted by $+$) are seen as predefined relations over the sets of the form $\{0,\ldots,n-1\}$.

Let us state our main theorem:

Theorem 3.1. *Let L be a language over $\{1,2\}$. L is definable in $SO_1(+)$ iff $L \in \mathcal{L}_2(RUD)$.*

Then, theorem 3.1 is used to prove theorem 3.2. First, we need two more definitions.

Definition 3.7. If F denotes a set of second-order formulas, let us denote by $Sp(F)$ the class of spectra of formulas in this class.

Remark. First-order spectra correspond to $Sp(\exists SO(\emptyset))$.

Definition 3.8. Let $f : \mathbb{N} \to \mathbb{N}$ and $S \subseteq \mathbb{N}$. We denote by $f(S) = \{f(s); s \in S\}$ the set of images under f of elements of S.

Theorem 3.2. *For any set of integers S, the following equivalences hold:*

1. *for any integer $k \geq 1$, $S \in Sp(SO_k(+))$ iff $2^{S^k} \in RUD$;*
2. *for any integer $k \geq 2$, $S \in Sp(SO_k(\emptyset))$ iff $2^{S^k} \in RUD$.*

3.1.3 Languages

In this section, theorem 3.1 is proved along lemmas 3.3 and 3.4.

Lemma 3.3. *Let A be a set of integers. If $L_2(A)$ is definable in $SO_1(+)$, then $A \in RUD$.*

Proof. The hypothesis of the lemma asserts that there exists a $SO_1(+)$-formula $\Theta(X,+) \equiv Q_1U_1 \dots Q_pU_pq_1x_1 \dots q_sx_s\psi(\overline{x},\overline{U},X,+)$ such that $w \in L_2(A)$ iff $\mathcal{R}_w(+) \models \Theta$. From the very definition of w_n, we can translate this hypothesis by: $n \in A$ iff $\langle lg(n), \mathcal{2}_n, +\rangle \models \Theta$.

In using Bennett's characterization of rudimentary sets, we have to construct a bounded arithmetical formula $\Phi(n)$ with one free variable such that $n \in A$ iff $\mathbb{N} \models \Phi(n)$. The main idea to proceed is to encode each unary relation U over $ld(n)$ as an integer u smaller than n. Then we have to translate assertions about U (actually $U(x)$, for a given $x < ld(n)$) into equivalent arithmetical assertions about u. At last, we can build the formula Φ, taking a specific care to the bounds of its quantifications. Let us describe briefly these three steps:

• *Encoding.* Let n be an integer and $l = ld(n)$. Each subset U of l defines a single word w of length l such that $\mathcal{2}_w = U$. Now, consider the integer u such that $w_u = w$. It is easy to check that the encoding thus obtained is a bijection from the set of unary relations over l onto the set of integers $[2^l - 1; 2^{l+1} - 2]$ (note that these integers are smaller than $2n$).

• *Translation.* Let U and u be defined as above. Let $x < l$. Refering to the definition of u from U, observe that: $U(x)$ holds in n iff $(\exists a,b,c < 2^{l+1})(u = a+b+cb \wedge 2^x - 1 \leq a < 2^{x+1} - 1 \wedge b = 2^{x+1} \wedge c \neq 0)$ is true in $\mathbb{N}$. We denote by $digit(u,x)$ this last formula. Now we are ready to built the bounded arithmetical formula $\Phi(n)$.

• *Construction of* Φ. The atomic formulas in Θ have the following four possible forms: $U_i(x_j)$, $X(x_j)$, $x_j = x_k$ and $x_j + x_k = x_\ell$ and $\Phi(n)$ is obtained from Θ by the following substitutions:

Q_iU_i	becomes $Q_iu_i \in [2^l - 1; 2^{l+1} - 2]$
q_jx_j	becomes $q_jx_j < l$
$U_i(x_j)$	becomes $digit(u_i, x_j)$
$X(x_j)$	becomes $digit(n, x_j)$
$x_j = x_k$	remains unchanged
$x_j + x_k = x_\ell$	remains unchanged

It remains to deal with l, which appears up to now like a predefined variable, and with exponential, which must not occur in a bounded arithmetical formula. First, note that the relation $l = ld(n)$ is defined by the sentence: $(\exists L < 2n)(L = 2^l - 1 \wedge L \leq n \leq 2L)$. Finally, we get rid of the exponential using the fact that the graph of exponentiation is rudimentary (see [3]).

Lemma 3.4. *If a set of integers A is in RUD, then $L_2(A)$ is definable in $SO_1(+)$.*

Proof. This proof has the same pattern than the previous one but now we use Smullyan's definition of rudimentary sets instead of Bennett's one. Indeed, we interpret the hypothesis of lemma 3.4 as follows: there exists a bounded $\{*\}$-formula $\Phi(n)$, such that $n \in A$ iff $\mathbb{N} \models \Phi(n)$.

• *Encoding.* Let $n \in \mathbb{N}$ and $l = ld(n)$. To each integer u of dyadic length smaller than l, we associate the couple of unary relations $(\mathit{1}_u, \mathit{2}_u)$ over l defined by: $\forall x < l$, $\mathit{1}_u(x)$ (resp. $\mathit{2}_u(x)$) holds iff the letter of index x of w_u is 1 (resp. 2). Remark that the map $u \mapsto (\mathit{1}_u, \mathit{2}_u)$ is a bijection from $[0, 2^{l+1} - 2]$ onto the set of couples (U, V) such that $U, V \subseteq l$, $U \cap V = \emptyset$, and $U \cup V$ is an initial segment of l.

• *Translation.* Now we have to translate the atomic formulas which constitute the formula $\Phi(n)$. Actually, we have to interpret the subformulas $u_i < n$ and $u_i * u_j = u_k$ (some of these u_i's may be n). To proceed, we fist translate without proof some intermediate assertions.

- There exists a formula $WECO_{<max}(U, V)$ such that for every couple (U, V) of unary relations over a domain n we have: $n \models WECO_{<max}(U, V)$ iff there exists an $x < n$ such that $(\mathit{1}_x, \mathit{2}_x) = (U, V)$.

- There exists a formula $(U, U') * (V, V') = (T, T')$ such that for every l, n such that $l = lg(n)$, for every $u, v, t < l$, we have $l \models u * v = t$ iff $n \models (\mathit{1}_u, \mathit{2}_u) * (\mathit{1}_v, \mathit{2}_v) = (\mathit{1}_t, \mathit{2}_t)$.

• *Construction of* Θ. Now, let $\tilde{\Theta}(\mathit{1}_n, \mathit{2}_n, <, +)$ be the second-order formula obtained from Φ by the following substitutions:

$$\begin{array}{rl} \forall u_i < n & \text{becomes } \forall \mathit{1}_i \forall \mathit{2}_i\ [WECO_{<n}(\mathit{1}_i, \mathit{2}_i) \rightarrow \\ \exists u_i < n & \text{becomes } \exists \mathit{1}_i \exists \mathit{2}_i\ [WECO_{<n}(\mathit{1}_i, \mathit{2}_i) \wedge \\ u_i * u_j = u_k & \text{becomes } (\mathit{1}_i, \mathit{2}_i) * (\mathit{1}_j, \mathit{2}_j) = (\mathit{1}_k, \mathit{2}_k) \end{array}$$

Note that, although it doesn't appear explicitly in our notations, the linear order $<$ and the relations $\mathit{1}_n$ and $\mathit{2}_n$ are in the signature of $\tilde{\Theta}$: the relation $<$ is hidden in the formulas $WECO_{<n}$ and $\mathit{1}_n$ and $\mathit{2}_n$ appear with the translations of the subformulas $u_i * u_j = u_k$ in wich some of the variables are n. But $<$ is trivially expressed with addition, and $\mathit{2}_n$ can be replaced by $\neg \mathit{1}_n$. All in all, the sought formula Θ is obtained from $\tilde{\Theta}$ in expliciting $<$ with $+$ and in replacing $\mathit{2}_n$ by X and $\mathit{1}_n$ by $\neg X$.

3.1.4 Exponentiations of spectra and rudimentary sets

In this section, we prove theorem 3.2. Precisely, we first prove the first item and then deduce the second one. The proof consists in two lemmas respectively corresponding to each implication.

Lemma 3.5. *If* $S \in Sp(SO_k(+))$, *then* $2^{S^k} \in RUD$.

Proof. The main idea is to encode each R_i into an integer c_i smaller than $2^{n^k+1} - 2$, and to replace second-order quantification over R_i by a bounded first-order quantification over c_i. We do not give a detailed proof of this lemma, because it is easy and similar to the proof given in [56].

Lemma 3.6. *If* $L_2(2^{S^k})$ *is definable in* $SO_1(+)$, *then* $S \in Sp(SO_k(+))$.

Proof. Let S be a set of integers such that $L_2(2^{S^k})$ is definable in $SO_1(+)$. Let $\Psi \equiv Q_1 V_1 \ldots Q_p V_p q_1 y_1 \ldots q_r y_r \Phi$ be a $SO_1(+)$-formula (where Φ is a first-order open formula of signature $\{=,+,X,V_1,\dots,V_p\}$ and variables $y_1\dots y_r$) such that $w_m \in L_2(2^{S^k})$ iff $\langle ld(m),=,2_{w_m},+\rangle \models \Psi$.

Since $m \in 2^{S^k}$, there exists $n \in S$ such that $m = 2^{n^k}$. Consequently, w_m is $1^{n^k-1}2$. Hence, we have to change a structure of size $ld(m) = n^k$ into a structure of size n, to cancel the second-order unary free variable X and to replace the addition over $\{0,\dots,n^k-1\}$ by the addition over $\{0,\dots,n-1\}$. Formally, we have to construct a $SO_k(+)$-formula Δ such that $n \in S$ iff $\langle n,=,+\rangle \models \Delta$.

To change the cardinality of the domain, we encode each integer $t < n^k$ into a k-tuple of integers $t_i < n$ (notation in basis n), and thus naturally translate every unary relation V_i over n^k into a k-ary relation R_i over n. We cancel the free variable X (interpreted by $2_{w_m} = \{n^k-1\}$) by introducing a new second-order k-ary variable R_0 satisfying the condition $VALR_0$ expressing the fact that $n^k - 1 = \sum_{i=0}^{k-1}(n-1).n^i$. We replace $u+v=w$ (addition over $\{0,\dots,n^k-1\}$) by $ADD'(u_0,\dots,u_{k-1},v_0,\dots,v_{k-1},w_0,\dots,w_{k-1})$ (k additions over $\{0,\dots,n-1\}$), i.e. we compute in basis n, and we deal by hand with the carries.

Then we apply on the formula Ψ the following transformations:

• we add the second-order quantification $\exists R_0$ over a k-ary relation;

• each second-order quantification over a unary relation Q_iV_i is replaced by the same second-order quantification Q_iR_i over a k-ary relation;

• each first-order quantification q_iy_i is replaced by the k first-order quantifications $q_iy_{i,1} \cdots q_iy_{i,k}$;

• the first-order open formula Φ is changed into a first-order formula of signature $\{=,+,R_0,R_1,\dots,R_p\}$ denoted by Γ and obtained by replacing $y_i = y_j$ by $\bigwedge_{s=1}^{k}(y_{i,s} = y_{j,s})$, $y_i + y_j = y_l$ by $ADD'(y_{i,1},\dots,y_{i,k},y_{j,1},\dots,y_{j,k},y_{l,1},\dots,y_{l,k})$, $X(y_j)$ by $R_0(y_{j,1},\dots,y_{j,k})$, and $V_i(y_j)$ by $R_i(y_{j,1},\dots,y_{j,k})$.

Finally, the formula Δ is $\exists R_0\ Q_1R_1\dots Q_pR_p\ q_1y_{1,0}\dots q_ry_{r,k-1}(VALR_0 \wedge \Gamma)$.

The proof of the first item of theorem 3.2 is now obtained via theorem 3.1. In order to prove the second item, the remaining task is to eliminate the predefined addition. Nothing changes in the proof of lemma 3.5 if one is interested into spectra in $Sp(SO_k(\emptyset))$ instead of $Sp(SO_k(+))$. But, in the proof of lemma 3.6, we actually use the addition over the structure $\langle n,=,+\rangle$. To overcome this difficulty, we just have to invoke the following result:

Theorem 3.7 (Lynch, [52]). *The addition is first-order definable with binary relations.*

3.1.5 Consequences

Let us state first a corollary in the field of complexity.

Corollary 3.8. *$\mathcal{L}_2(RUD)$ contains many natural NP-complete languages.*

Proof. Many natural NP-complete languages, in which the twenty-one listed by [48], are known to belong to NLIN, a complexity class introduced by E. Grandjean to formalize linear time complexity on RAM's. But it is also proved in [38] that all languages in NLIN are definable in $\exists SO_1(+)$, so that the above mentioned languages are in $\mathcal{L}_2$(RUD).

Now, let us turn to descriptive complexity. The particular role played by the dyadic representation of integers in theorem 3.1 can easily be avoided. First, let us note that it is almost straightforward to adjust our definitions to the p-adic case. We only precise the following one.

DEFINITION 3.9. For each word w over $\{1, 2, \ldots, p\}$, we still denote by $\mathcal{R}_w$ the finite structure $\langle l, \mathit{1}_w, \ldots, p_w \rangle$, where l is the length of w and $\mathit{1}_w, \ldots, p_w$ are the unary relations over the domain $l = \{0, \ldots, l-1\}$ defined by: for all $s = 1$ to p, for all $x < l$, $s_w(x)$ holds iff the letter of index x of w is s.

Now, we say that a language $L \subseteq \{1, \ldots, p\}^*$ is *definable in* $SO_1(+)$ if there exists a formula Θ (with p free unary second-order variables $X_1, \ldots, X_p$) in $SO_1(+)$ such that for all $w \in \{1, \ldots, p\}^*$ we have: $w \in L$ iff $\mathcal{R}_w(+) \models \Theta(X_1, \ldots, X_p)$ (where X_s is interpreted by s_w in $\mathcal{R}_w(+)$). Let us note that for all integers $p \geq 2$ and $n \geq 1$, there is a bijection between the integers of p-adic length $\leq n$ and the p-tuples of unary relations providing a partition of an initial segment of $\{0, \ldots, n-1\}$. Then, the proof of the following result strickingly follows the one of theorem 3.1.

Corollary 3.9. *For any integers $p \geq 2$ and $q \geq 2$,*

1. *a language $L \subseteq \{1, \ldots, p\}^*$ is definable in $SO_1(+)$ iff $L \in \mathcal{L}_p(RUD)$.*
2. *Let $A \subseteq I\!N$. Then $L_p(A)$ is definable in $SO_1(+)$ iff $L_q(A)$ isdefinable in $SO_1(+)$.*

Theorem 3.2 leads to a similar result in the scope of spectra. The proof is straightforward, since RUD is closed under inverse ranges by polynomials.

Corollary 3.10. *For any integers $p \geq 2$ and $k \geq 1$, for any set of integers A, the following equivalence holds: $A \in Sp(SO_k(+))$ iff $p^{A^k} \in RUD$.*

Finally, let us note that, according to theorem 3.2, generalized Fagin's question (see [21]):

does $Sp(SO_k(\emptyset)) = Sp(SO_2(\emptyset))$ for all $k \geq 2$?

is equivalent to the following question :

for all $k \geq 2$, for all set S of integers, does $2^{S^k} \in RUD$ iff $2^{S^2} \in RUD$?

Remark that we know no such equivalence for Fagin's question : does $Sp(\exists SO_k(\emptyset)) = Sp(\exists SO_2(\emptyset))$ for all $k \geq 2$?

3.2 Le comptage modulaire de RUD est dans $Sp(f,b)$

Nous avons vu que la nature du rapport entre RUD et SPECTRES constitue une question ouverte et difficile. Nous l'avons abordée dans un premier temps en *représentant* chaque spectre dans RUD. Une deuxième approche consiste à chercher des candidats à la séparation de ces deux classes. Dans ce but nous allons introduire la classe $\mathrm{RUD}_{\#\mathrm{RUD}}$, inspirée de la notion de *relation de comptage*. Commençons par préciser quelques définitions :

DÉFINITIONS 3.10. Soient A, B deux parties de $\mathbb{N}$ et n un entier non nul.

- si A est fini, $|A|$ dénote le cardinal de A ;
- A_n est l'ensemble $A \cap n$;
- si $n \in A$, on appelle *rang de n (dans A)*, et on note $rg(n)$, l'entier $|A \cap n|$. En particulier, si $a_0, a_1, a_2, \ldots$ est une énumération strictement croissante des éléments de A, alors pour tout $i \in \mathbb{N}$, $rg(a_i) = i$;
- on note A_B l'ensemble des éléments de A dont le rang est dans B. *i.e.* $A_B = \{n \in A \text{ t.q. } |A \cap n| \in B\}$;
- si $\mathcal{A}$ et $\mathcal{B}$ sont deux ensembles de parties de $\mathbb{N}$, on note $\mathcal{A}_{\#\mathcal{B}} = \{A_B,\ A \in \mathcal{A},\ B \in \mathcal{B}\}$. En particulier $\mathrm{RUD}_{\#\mathrm{RUD}} = \{A_B,\ A, B \in \mathrm{RUD}\}$.

On a clairement les inclusions $\mathrm{RUD} \subseteq \mathrm{RUD}_{\#\mathrm{RUD}} \subseteq$ SPECTRES et l'on peut être tenté d'étudier la séparation de RUD et SPECTRES "au niveau" de l'inclusion $\mathrm{RUD} \subseteq \mathrm{RUD}_{\#\mathrm{RUD}}$. Nous nous contenterons ici de montrer que la sous-classe de $\mathrm{RUD}_{\#\mathrm{RUD}}$ des ensembles A_B, où $A \in \mathrm{RUD}$ et B est un ensemble *modulaire* (en un sens que nous préciserons), peut être conjecturée *très proche* de RUD.

3.2.1 Définissabilité du comptage

DÉFINITION 3.11.

- Soit σ une signature. On note $\exists SO_1(\sigma)$ la classe des formules du second ordre existentielles relationnelles monadiques de signature σ, et $PO(\sigma)$ la classe des formules du premier ordre de signature σ.
- Soit $\mathcal{F}$ un ensemble de formules closes inclus dans $\exists SO_1$. On note $Sp(\mathcal{F})$ l'ensemble des cardinaux des modèles des formules de $\mathcal{F}$.

DÉFINITION 3.12. Soient A une partie de $\mathbb{N}$ et r, c deux entiers.

- On note $\mathbb{N}_{r[c]}$ l'ensemble $\{n \in \mathbb{N} \ :\ n \equiv r\,[c]\}$.

– On note Mod la classe des ensembles $\mathbb{N}_{r[c]}$ quand c et $r < c$ varient dans $\mathbb{N}$. Les éléments de Mod sont appelés *ensembles modulaires.*

Le lemme suivant est un corollaire immédiat des résultats de Lynch sur la définissabilité de l'arithmétique dans $\exists SO_1(+)$ (voir section 1.4, page 39).

Lemme 3.11. $RUD \subseteq Sp(\exists SO_1(+))$.

Nous allons maintenant prouver que cette inclusion reste valide si l'on remplace RUD par $\text{RUD}_{\#Mod}$.

DÉFINITION 3.13. Soit $\Theta \equiv \exists\overline{U}\overline{qx}\theta$ une formule de $\exists SO_1(\sigma)$ (σ signature quelconque, θ formule sans quantificateur). Soit X un symbole de relation monadique. On appelle *relativisée de* Θ *à* X la formule de $\exists SO_1(\sigma, X, <)$ notée Θ^X :

$$\Theta^X \equiv (\exists\overline{U} \subseteq X)(\overline{qx} \in X)\theta.$$

Lemme 3.12. *Soit $A \subseteq \mathbb{N}$. Si A est dans RUD, alors il existe une formule $SI_A \in \exists SO_1(X,+)$ (SI pour "segment initial") telle que : pour tout $n \in \mathbb{N}$ et tout $U \subseteq n$,*

$$U = A_n \text{ ssi } \langle n, U, +\rangle \models SI_A(U,+)$$

Preuve. Soit $\phi \in PO(+,\times,<)$ telle que $\forall n \in \mathbb{N}$, $n \in A$ ssi $n \models \phi$. On a les équivalences :

$$\begin{array}{rl} U = A_n \text{ ssi} & \forall x < n,\ x \in U \Leftrightarrow x \in A \\ \text{ssi} & \forall x < n,\ \langle n, U\rangle \models U(x) \Leftrightarrow \langle x, +, \times, <\rangle \models \phi \\ \text{ssi} & \forall x < n,\ \langle n, U\rangle \models U(x) \Leftrightarrow \langle n, +, \times, <, x\rangle \models \phi^x \\ \text{ssi} & \langle n, +, \times, <, U\rangle \models \forall x : U(x) \leftrightarrow \phi^x \end{array}$$

Comme la formule $\psi \equiv \forall x : U(x) \leftrightarrow \phi^x$ appartient à $PO(+,\times,<,U)$, on sait qu'il existe une formule de $\exists SO_1(U,+)$ équivalente à ψ sur les domaines finis. C'est la formule SI_A cherchée.

Lemme 3.13. *Soient A une partie finie de $\mathbb{N}$ et $n > \max(A)$. Soit $\Theta \in \exists SO_1(<)$. Alors*

$$\langle |A|, <\rangle \models \Theta \text{ ssi } \langle n, <, A\rangle \models \Theta^A$$

Preuve. $\langle |A|, <\rangle$ et $\langle A, <\rangle$ sont deux $\{<\}$-structures isomorphes. On a donc

$$\langle |A|, <\rangle \models \Theta \text{ ssi } \langle A, <\rangle \models \Theta$$

Comme de plus A est inclus dans n, on a clairement

$$\langle A, <\rangle \models \Theta \text{ ssi } \langle n, <, A\rangle \models \Theta^A$$

D'où le résultat.

Lemme 3.14. *Soient $A \in RUD$ et $B \in Sp(\exists SO_1(<))$. Alors $A_B \in Sp(\exists SO_1(+))$. En d'autre termes : $RUD_{\# Sp(\exists SO_1(<))} \subseteq Sp(\exists SO_1(+))$.*

Preuve. Soit Θ_A (resp Θ_B) la formule de $\exists SO_1(+)$ (resp $\exists SO_1(<)$) qui caractérise A (resp B). Soit SI_A la formule de $\exists SO_1(X, +)$ telle que pour tout $n \in \mathbb{N}$ et tout $U \subseteq n$: $U = A_n$ ssi $\langle n, U, + \rangle \models SI_A(X, +)$ (voir Lemme 3.12). Alors pour tout entier n :

$$\begin{array}{rl} n \in A_B \text{ ssi} & n \in A \text{ et } |A_n| \in B \\ \text{ssi} & n \in A \text{ et } \exists U \subseteq n : U = A_n \text{ et } |U| \in B \\ \text{ssi} & \langle n, + \rangle \models \Theta_A \text{ et } \exists U \subseteq n : \langle n, U, + \rangle \models SI_A \text{ et } \langle |U|, < \rangle \models \Theta_B \\ \text{ssi} & \langle n, + \rangle \models \Theta_A \text{ et } \exists U \subseteq n : \langle n, U, + \rangle \models SI_A \text{ et } \langle n, U, < \rangle \models \Theta_B^U \\ \text{ssi} & \langle n, +, < \rangle \models \exists U : \Theta_A \wedge SI_A \wedge \Theta_B^U \end{array}$$

Soit Θ la formule obtenue en écrivant $\exists U : \Theta_A \wedge SI_A \wedge \Theta_B^U$ sous forme prenexe, puis en exprimant l'ordre en fonction de l'addition. Alors $\Theta \in \exists SO_1(+)$ et pour tout $n \in \mathbb{N}$, $n \in A_B$ ssi $\langle n, + \rangle \models \Theta$. Θ est donc la formule annoncée.

Corollaire 3.15. *Soient $A \in RUD$ et $r < p$ deux entiers. Alors l'ensemble des éléments de A dont le rang est congru à r modulo p est caractérisable dans $\exists SO_1(+)$. En particulier, l'ensemble des entiers premiers de rang pair est dans $Sp(\exists SO_1(+))$.*

Preuve. Il s'agit d'établir l'inclusion : $\mathrm{RUD}_{\#\mathbb{N}_{r[p]}} \subseteq Sp(\exists SO_1(+))$. Compte tenu du lemme 3.14, il suffit pour cela de prouver que $\mathbb{N}_{r[p]}$ est dans $Sp(\exists SO_1(<))$. Or $\mathbb{N}_{r[p]}$ est caractérisable par la formule

$$\exists U : U(0) \wedge \bigwedge_{i=1}^{r-1} \neg U(i) \wedge \bigwedge_{i=1}^{p-1} \neg U(n-i) \wedge \forall x (U(x) \leftrightarrow U(x+r)),$$

elle même équivalente à une formule de $\exists SO_1(<)$.

La remarque sur les entiers premiers de rang pairs se déduit alors immédiatement de la rudimentarité de l'ensemble des premiers.

Nous avons rappelé dans la section précédente l'équivalence des égalités RUD = BIN et RUD = SPECTRES (où BIN désigne la classe des spectres de graphes) prouvée par Woods ([74]). Nous avons dit de plus que la validité de ces égalités entrainerait l'effondrement de la hiérarchie polynomiale.

Nous établissons maintenant une version plus fine de cette équivalence, obtenue comme corollaire des inclusions : RUD $\subseteq Sp(\exists SO_1(+)) \subseteq$ SPECTRES.

Corollaire 3.16. *$RUD = Sp(\exists SO_1(+))$ ssi $RUD =$* SPECTRES.

Preuve. Le contenu significatif de cet énoncé peut s'écrire :

$$Sp(\exists SO_1(+)) \subseteq \text{RUD entraîne SPECTRES} \subseteq \text{RUD}.$$

Cette implication découle des faits suivants, que nous énonçons sans preuve : pour tout entier $k \geq 2$ et toute partie A de $\mathbb{N}$,

$$A \in \text{RUD ssi } A^k \in \text{RUD et}$$
$$A \in Sp(\exists SO_k) \text{ ssi } A^k \in Sp(\exists SO_1(+)).$$

On a par conséquent :

$\forall A \in$ SPECTRES, $\exists k \in \mathbb{N}$ tel que $A \in Sp(\exists SO_k)$.

D'où $A^k \in Sp(\exists SO_1(+)$, donc : $A^k \in$ RUD.

Et finalement : $A \in$ RUD.

3.2.2 $Sp(\exists SO_1(+))$ est dans *sp(f, b)*

Nous montrons ici que RUD et $\text{RUD}_{\#Mod}$ sont inclues dans une classe de spectres du premier ordre. Ceci découlera de l'inclusion plus générale de la classe $Sp(\exists SO_1(+))$ - dont nous venons de voir qu'elle contient $\text{RUD}_{\#Mod}$ - dans la classe *sp(f, b)* des spectres de formules du premier ordre dont la signature comprend exactement une fonction et une bijection.

Cette nouvelle "majoration" de RUD et $\text{RUD}_{\#Mod}$, quoiqu'apparemment moins précise que la précédente, permet la formulation naturelle et intuitive d'une conjecture sur la "proximité" des classes RUD et $\text{RUD}_{\#Mod}$.

Notations. Pour tous $A \subseteq \mathbb{N}$, $c \in \mathbb{N}^*$, $r < c$, on note :

- $A_{r[c]} = \{a \in A \,:\, a \equiv r\,[c]\}$
- $A - r = \{a - r,\ a \in A\}$
- $A + r = \{a + r,\ a \in A\}$
- $A/c = \{n \in \mathbb{N} \,:\, cn \in A\}$
- $cA = \{ca,\ a \in A\}$

On note de plus *sp(k fcts)* (resp. *sp(f, b)*) l'ensemble des spectres de formules du premier ordre dont la signature contient exactement k fonctions unaires (resp. une fonction unaire et une bijection unaire).

Lemme 3.17 (Durand, [13]). *Soient k un entier et A une partie de $\mathbb{N}$. Si $A \in$ sp(k fcts), alors $(k+2)A \in$ sp(f, b).*

Lemme 3.18. *Soient $k \in \mathbb{N}$, $c = k+2$ et $\mathcal{C} \subseteq$ sp(k fcts) telle que pour tout $A \subseteq \mathbb{N}$ on ait :*

1. *$A \in \mathcal{C} \Rightarrow \forall i < c,\ A_{i[c]} \in \mathcal{C}$;*

2. $A \in \mathcal{C} \Rightarrow \forall i < c,\ A - i \in \mathcal{C}$ *;*

3. $cA \in \mathcal{C} \Rightarrow A \in \mathcal{C}$*.*

Alors $\mathcal{C} \subseteq sp(f,b)$

Preuve. Soient $A \in \mathcal{C}$ et $i < c$. Des hypothèses (1), (2) et (3) du lemme on tire successivement :

$$A_{i[c]} \in \mathcal{C},\ A_{i[c]} - i \in \mathcal{C} \text{ et } \frac{A_{i[c]} - i}{c} \in \mathcal{C}.$$

Et donc, puisque $\mathcal{C} \subseteq sp(k\ fcts)$:

$$\frac{A_{i[c]} - i}{c} \in sp(k\ fcts).$$

Le lemme 3.17 entraine alors :

$$c(\frac{A_{i[c]} - i}{c}) \in sp(f,b).$$

Par ailleurs, la classe $sp(f,b)$ vérifie clairement les deux propriétés de fermeture suivante :

$$\forall A \subseteq \mathbb{N},\ \forall r \in \mathbb{N}\ :\ A \in sp(f,b) \Rightarrow A + r \in sp(f,b);$$
$$\forall A, B \subseteq \mathbb{N}\ :\ A, B \in sp(f,b) \Rightarrow A \cup B \in sp(f,b).$$

De la première on tire :

$$\forall i < c,\ c(\frac{A_{i[c]} - i}{c}) + i \in sp(f,b).$$

Ce qui entraine, avec la seconde :

$$\bigcup_{i<c} [c(\frac{A_{i[c]} - i}{c}) + i] \in sp(f,b).$$

La conclusion se déduit alors de l'identité :

$$A = \bigcup_{i<c} [c(\frac{A_{i[c]} - i}{c}) + i].$$

Proposition 3.19. $Sp(\exists SO_1(+)) \subseteq sp(f,b)$*.*

Preuve. Dans le but d'appliquer le lemme 3.18 à la classe $\mathcal{C} = Sp(\exists SO_1(+))$, commençons par prouver qu'il existe un $c_0 \in \mathbb{N}$ tel que $Sp(\exists SO_1(+)) \subseteq sp(c_0\ fcts)$. Soit $\Phi \equiv \exists U_1 \ldots \exists U_p \phi(\overline{U}, +)$ une formule de $\exists SO_1(+)$ (ϕ formule du premier ordre).

De la proposition 2.30, on tire facilement l'existence d'une formule Θ de la forme :

$$\Theta \equiv \exists \overline{U} \exists f_1 \ldots \exists f_8 \phi'(\overline{U}, \overline{f}, 0, s),$$

où les f_i sont des symboles de fonction unaires et ϕ' une formule du premier ordre, telle que :

$$sp(\Phi) = sp(\Theta).$$

Par ailleurs, le p-uplet de relations unaires $\overline{U}$ est codable par une unique fonction unaire. Plutôt que de décrire cette "simulation" dans le cas général, construisons la dans le cas où $p = 2$:

À tout couple $\overline{U} = (U_1, U_2)$ de relations unaires sur un domaine $n \geq 4$, on associe la fonction unaire $f_0 : n \rightarrow n$ définie comme suit :

$\forall x < n,\ f_0(x) \in \{0, 1, 2, 3\}$ et :

$$\begin{aligned}
f_0(x) = 0 &\text{ ssi } \neg U_1(x) \wedge \neg U_2(x)\ ;\\
f_0(x) = 1 &\text{ ssi } U_1(x) \wedge \neg U_2(x)\ ;\\
f_0(x) = 2 &\text{ ssi } \neg U_1(x) \wedge U_2(x)\ ;\\
f_0(x) = 3 &\text{ ssi } U_1(x) \wedge U_2(x).
\end{aligned}$$

La correspondance ainsi définie, entre les couples de relations unaires sur n et les fonctions unaires sur n à valeurs dans $\{0, 1, 2, 3\}$, est clairement bijective. Par conséquent, la formule $\Theta \equiv \exists U_1, U_2 \exists \overline{f} \phi'$ équivaut sur tout domaine $n \geq 4$ à la formule $\Psi \equiv \exists f_0 \exists \overline{f}\ :\ \forall x \bigvee_{i<4} f_0(x) = i \wedge \phi''$, où ϕ'' est la formule du premier ordre obtenue à partir de ϕ' par les substitutions suivantes :

$$\begin{aligned}
U_1(x) &\text{ devient } f_0(x) = 1 \vee f_0(x) = 3\ ;\\
U_2(x) &\text{ devient } f_0(x) = 2 \vee f_0(x) = 3\ ;\\
\neg U_1(x) &\text{ devient } f_0(x) = 0 \vee f_0(x) = 2\ ;\\
\neg U_2(x) &\text{ devient } f_0(x) = 0 \vee f_0(x) = 1.
\end{aligned}$$

La formule Ψ est de la forme $\exists f_0 \exists f_1 \ldots \exists f_8 \psi(\overline{f}, 0, s)$ (ψ formule du premier ordre). Il ne reste pour conclure, qu'à invoquer le théorème suivant, prouvé par Grandjean (cité ici sous forme affaiblie) :

Théorème (Grandjean, [31]). *Toute formule Ψ du second ordre existentiel fonctionnel unaire de signature prédéfinie $\{0, s\}$ est équivalente sur les structures unaires à une formule Ψ' de même structure mais de signature prédéfinie vide.*

Comme de plus le passage de Ψ à Ψ' nécessite l'addition d'un nombre fixé (indépendant de ψ) de fonctions existentiellement quantifiées, on peut conclure à l'existence d'un entier c_0 tel que :

$\forall \Phi \in \exists SO_1(+),\ \exists \Psi' \in \exists SO_1^f(c_0)$ (où $\exists SO_1^f(c_0)$ désigne la classe des formules du second ordre existentiel fonctionnel unaire dont le préfixe du second ordre est un c_0-uplet de fonctions unaires existentiellement quantifiées), telle que :

$$\forall n \in \mathbb{N}\ :\ n \models \Phi \text{ ssi } n \models \Psi'. \tag{3.1}$$

Remarque. Nous avons vu, dans le cas ou le préfixe du second ordre de Φ comporte 2 relations unaires, que le codage de ces relations par une unique fonction unaire ne peut se faire que sur un domaine $n \geq 4$.

D'une façon générale (pour un p-uplet de relations), l'assertion (3.1) n'est valable *a priori* que pour $n \geq 2^p$. En fait, cette assertion se généralise à tout n en traitant "à la main" le cas des entiers $< 2^p$.

Ceci permet finalement d'affirmer l'existence d'un entier c_0 fixé tel que :

$$Sp(\exists SO_1(+) \subseteq sp(c_0 \textit{ fcts}).$$

Il suffit maintenant de prouver que la classe $\mathcal{C} = Sp(\exists SO_1(+))$ satisfait les conditions *1*, *2*, *3* du lemme 3.18 pour $c = c_0 + 2$. En fait nous allons prouver ce résultat pour tout $c \in \mathbb{N}^*$, en considérant les conditions susdites dans l'ordre *1*, *3*, *2*.

$A \in \mathcal{C} \Rightarrow \forall i < c, \ A_{i[c]} \in \mathcal{C}$.

Soient $A \in \mathcal{C}$ et Θ_A la formule de $\exists SO_1(+)$ qui définit A. Soit n un entier. On a les équivalences : $n \in A_{i[c]}$ ssi $n \in A$ et $\exists q < n$ t.q. $n = qc + i$ ssi $n \models \Theta$, où Θ est la formule : $\Theta_A \wedge \exists q(n-1 = (q-1)c + (c-1) + i)$. L'ensemble $A_{i[c]}$ est ainsi définissable dans $\exists SO_1(+, \times)$, donc dans $\exists SO_1(+)$.

$cA \in \mathcal{C} \Rightarrow A \in \mathcal{C}$.

Cette implication a déjà été prouvée dans la section 1.6 du chapitre 1 (p. 49).

$A \in \mathcal{C} \Rightarrow \forall i < c, \ A - i \in \mathcal{C}$.

Soient $A \in \mathcal{C}$ et Θ_A la formule de $\exists SO_1(+)$ qui définit A. Soit n un entier. Clairement, $n \in A - i$ équivaut à $n + i \in A$, c'est à dire à : $n + i \models \Theta_A$.

Comme par ailleurs l'entier $n+i$ est définissable par une formule du premier ordre dans le domaine $2n$ ($n-1$ est défini dans $2n$ par la formule $\forall x(\exists y : x = 2y \leftrightarrow x \leq n-1)$, et $n + i$ est évidemment donné par : $n + i = (n-1) + (i+1)$), l'assertion $n + i \models \Theta_A$ peut-être écrite de manière équivalente sous la forme $2n \models \Theta'_A$, où Θ'_A est encore une formule de $\exists SO_1(+)$ (Θ'_A est à peu près la relativisée de Θ_A à $n+i$).

Enfin, toujours grâce à la la section 1.6 du chapitre 1, nous savons qu'il existe une formule $\Theta \in \exists SO_1(+)$ telle que $2n \models \Theta'_A$ ssi $n \models \Theta$. En résumé, nous avons prouvé les équivalences suivantes :

$n \in A - i$ ssi $n + i \models \Theta_A$ ssi $2n \models \Theta'_A$ ssi $n \models \Theta$.

C'est le résultat cherché.

Nous avons prouvé la chaine d'inclusions :

$$\mathrm{RUD} \subseteq \mathrm{RUD}_{\#Mod} \subseteq Sp(\exists SO_1(+) \subseteq sp(f, b).$$

La nature de l'inclusion RUD $\subseteq$ $\mathrm{RUD}_{\#Mod}$ ne nous est pas connue. Nous conjecturons néanmoins que ces deux classes sont "très proches sur l'échelle des spectres du premier ordre". Plus précisément, on peut raisonnablement penser :

Conjecture. *$sp(f) \subseteq RUD \subseteq RUD_{\#Mod} \subseteq sp(f, b)$. (Où $sp(f)$ désigne la classe des spectres de formules du premier ordre dont la signature ne comporte qu'un seul symbole de fonction unaire.)*

La nature hypothétique de cette assertion tient au fait que, autant que nous sachions, l'inclusion $sp(f) \subseteq$ RUD n'est pas prouvée. Mais la classe $sp(f)$ est communémment considérée comme très inexpressive, du fait de la décidabilité de la théorie d'une seule fonction. En fait, nous conjecturons que cette classe coïncide avec la classe des parties de $\mathbb{N}$ obtenues comme unions finies d'ensemble modulaires (voir définition 3.12). Elle serait alors évidemment strictement inclue dans RUD.

Conclusion

Pour conclure cette thèse, nous nous efforçons de motiver et formuler quelques unes des nombreuses questions que soulèvent nos résultats.

Le pouvoir d'expression de la logique de Lynch

Notons $\mathcal{L}(+)$ la logique de Lynch, c'est à dire la classe des formules de $\exists SO_1(+)$ dont le préfixe du premier ordre est de la forme $\forall^*\exists^*$. Nous avons prouvé l'inclusion :

$$\text{NLIN} \subseteq \mathcal{L}(+). \tag{3.2}$$

La logique $\mathcal{L}(+)$ est "stable" : nous avons vu qu'on peut y définir la multiplication et l'exponentielle de base 2 (et plus généralement, de base fixée) et le prédicat *BIT* naturellement associé aux codages de mots sur un alphabet binaire.

Ceci justifie que l'on s'intéresse à la complexité algorithmique des formules de $\mathcal{L}(+)$ (*i.e.* de la question "$S \in mod\ \Phi$?" pour Φ fixée dans $\mathcal{L}(+)$). Plus précisément, la question se pose de trouver une réciproque du résultat (3.2).

> $\mathcal{Q}_1$: *Peut-on définir de manière uniforme une classe de complexité $\mathcal{C}$ telle que $\mathcal{C} = \mathcal{L}(+)$?*

Nous avons vu dans la conclusion du chapitre 1 qu'une telle classe serait à chercher "en dessous" de $\Sigma_3 TIME(n)$, puisque tous les problèmes définissables dans $\mathcal{L}(+)$ sont dans cette classe de complexité.

NLIN : caractérisation relationnelle *vs* caractérisation fonctionelle

La définition de NLIN sous forme de classe de langages présente un avantage et un inconvénient. *L'avantage* : la notion de mot est universelle. La définition *via* les langages est la manière la plus générale de présenter les classes de complexité. *L'inconvénient*: la caractérisation de NLIN par la logique $\mathcal{G}(<)$ exige, dans ce formalisme, de revenir à un codage des mots sous forme de structures finies peu naturel (nécessitant entre autre un scindage des mots w en sous-mots de longueur $\lceil \frac{\log(|w|+1)}{k} \rceil$ pour un certain $k \geq 2$).

Ce deuxième aspect rend difficile la comparaison de la caractérisation de la classe *de langages* NLIN aux autres caractérisations, qui se réfèrent généralement à un codage plus naturel des mots.

De ce point de vue, le résultat principal du chapitre 1 (Théorème A) peut être vu comme une forme de normalisation. Précisons cette remarque :

Pour chaque mot w de longueur n sur un alphabet binaire (disons $\{1, 2\}$ pour fixer les idées), notons $\mathcal{F}_w$ et $\mathcal{R}_w$ les structures finies suivantes :

- $\mathcal{F}_w = \langle m, f_w \rangle$, avec $m = \lceil \frac{n}{l} \rceil$ pour $l = \lceil \frac{\log(n+1)}{k} \rceil$, et en notant f_w la fonction unaire sur m qui associe à $x < m$ l'entier (inférieur à m) dont l'écriture dyadique est le $x^{\text{ème}}$ sous-mot de longueur l de w (voir la sous-section 1.1.2, p. 22).

- $\mathcal{R}_w = \langle n, X_w \rangle$, où X_w est l'ensemble des "places" du symbole 2 dans w (voir l'introduction du chapitre 1, p. 19).

On note de plus $\mathcal{F}_w(\sigma)$ (resp. $\mathcal{R}_w(\sigma)$) l'expansion de ces structures à une signature prédéfinie σ.

Avec ces conventions, la caractérisation de NLIN s'énonce comme suit : un langage L est dans NLIN si et seulement si il existe une formule $\Phi \in \mathcal{G}(<)$ telle que :

$$\forall w \in \{1, 2\}^* \ : \ w \in L \text{ ssi } \mathcal{F}_w(<) \models \Phi.$$

Le résultat principal du chapitre 1 caractérise NLIN relativement au codage relationnel, plus naturel, des mots : si un langage L est dans NLIN alors il existe une formule $\Psi \in \mathcal{L}(+)$ telle que :

$$\forall w \in \{1, 2\}^* \ : \ w \in L \text{ ssi } \mathcal{R}_w(+) \models \Psi.$$

La démonstration se fait dans un cadre purement logique en construisant, pour chaque formule $\Phi \in \mathcal{G}(<)$, une formule $\Psi \in \mathcal{L}(+)$ telle que :

$$\forall w \in \{1, 2\}^*, \ \mathcal{F}_w(<) \models \Phi \text{ ssi } \mathcal{R}_w(+) \models \Psi. \tag{3.3}$$

Il serait intéressant (mais probablement difficile) d'établir une forme de réciproque de ce résultat, c'est à dire de prouver l'existence, pour chaque formule Ψ d'une sous classe de $\mathcal{L}(+)$ à déterminer, d'une formule $\Phi \in \mathcal{G}(<)$ telle que l'assertion 3.3 soit vérifiée. Une telle réciproque permettrait d'espérer une caractérisation exacte de NLIN *pour le codage relationnel des mots.*

> $\mathcal{Q}_2$: *Existe-t-il une restriction de la logique de Lynch caractérisant exactement NLIN pour le codage relationnel des mots?*

Minimalité de la caractérisation conjonctive de NLIN

Dans le chapitre 2, nous avons prouvé la caractérisation exacte de NLIN par la classe de formules purement conjonctives $\mathcal{G}^+_\wedge(0,s)$. Cette caractérisation peut être raisonnablement jugée minimale, du fait de la structure de ces formule - quantifications du second ordre réduites à la quantification existentielle, symboles du second ordre limitées à des fonctions d'arité 1, préfixe du premier ordre restreint à une unique variable, matrice ne faisant intervenir ni négation, ni disjonction - et du fait de la signature prédéfinie utilisée. On peut en effet montrer par des considérations modèles théoriques que l'omission des symboles 0 et s restreindrait strictement le pouvoir d'expression des formules de $\mathcal{G}^+_\wedge(0,s)$ (il suffit de remarquer que la logique $\mathcal{G}^+_\wedge(\emptyset)$ est close par produit de modèles). De plus, nous conjecturons que l'omission du seul symbole 0, ou la substitution du successeur *cyclique* S (définit sur chaque $n \in \mathbb{N}^*$ par : $\forall x < n-1$, $S(x) = x+1$ et $S(n-1) = 0$) au successeur non cyclique utilisé dans notre preuve, affaiblirait aussi le pouvoir d'expression de notre logique. En d'autres termes, nous conjecturons :

> $\mathcal{C}_1$: *La logique $\mathcal{G}^+_\wedge(0,s)$ est strictement plus expressive que les logiques $\mathcal{G}^+_\wedge(s)$ et $\mathcal{G}^+_\wedge(0,S)$.*

Interprétations des formules de $\mathcal{G}^+_\wedge$

Si $\overline{f}$ est un q-uplet de fonctions unaires, une formule de $\mathcal{G}^+_\wedge(\overline{f},0,s)$ est une formule du type :

$$\Phi \equiv \exists g_1 \ldots \exists g_p \forall x \bigwedge_{i<r} \sigma_i(x) = \tau_i(x),$$

où les g_i sont des fonctions unaires, et les σ_i, τ_i des composées de fonctions prises parmi $\{\overline{f}, \overline{g}, z, s\}$. Puisque deux fonctions sur un domaine n sont égales si et seulement si elles coïncident sur tout $x \in n$, la formule Φ peut "s'écrire" plus simplement :

$$\Phi \equiv \exists g_1 \ldots \exists g_p \bigwedge_{i<r} \sigma_i = \tau_i. \tag{3.4}$$

Il nous semblerait intéressant de tirer les conséquences de cette remarque selon trois directions :

Caractérisation algébrique de NLIN. En écrivant sous la forme 3.4 les formules qui caractérisent les classes de structures unaires NLIN, on doit pouvoir reformuler notre caractérisation de cette classe (Théorème 2.13, p. 86) sous la forme (non aboutie) suivante :

> *Une classe S de q-uplet de fonctions unaires est dans NLIN si et seulement si il existe un système d'équations fonctionnelles paramètré $\mathcal{E}(\alpha_1, \ldots, \alpha_{q+2})$*

(les paramètres $\alpha_1, \ldots, \alpha_{q+2}$ sont des symboles de fonction unaires), telle que pour tout q-uplet de fonctions unaires $(f_1, \ldots, f_q)$:

$$\overline{f} \in S \text{ ssi } \mathcal{E}(\overline{f}, z, s) \text{ a une solution.}$$

(Où z et s désignent respectivement la fonction identiquement nulle et la fonction successeur, et en convenant que ces deux fonctions remplacent respectivement les symboles α_{q+1} et α_{q+2} dans $\mathcal{E}$.)

Notons que l'opération implicitement invoquée ici dans la notion de *système d'équations fonctionnelles* est celle de *composition* des fonctions. Ainsi les égalités :

$$\begin{cases} g_1 \circ \alpha_1 \circ g_2 = \alpha_3 \circ g_2 \circ \alpha_4 \\ \alpha_1 \circ g_3 \circ g_5 \circ g_6 = \alpha_2 \circ g_1 \\ g_4 \circ \alpha_1 = g_7 \end{cases}$$

constituent-elles un système $\mathcal{E}(\alpha_1, \ldots, \alpha_4)$ de paramètre $\alpha_1, \ldots, \alpha_4$ et d'inconnues $g_1, \ldots, g_7$. L'ensemble défini par $\mathcal{E}$ est alors exactement l'ensemble des couples de fonctions (f_1, f_2) tels que le système :

$$\begin{cases} g_1 \circ f_1 \circ g_2 = z \circ g_2 \circ s \\ f_1 \circ g_3 \circ g_5 \circ g_6 = f_2 \circ g_1 \\ g_4 \circ f_1 = g_7 \end{cases}$$

ait une solution.

Une telle caractérisation de NLIN, sous réserve d'une formulation plus précise, pourrait présenter des analogies avec la caractérisation des ensembles récursivement énumérables établie par Matijassevitch ([55]).

Caractérisation au premier ordre de NLIN. Soit n un entier. Notons ξ la bijection canonique de l'ensemble $(n)^{(n)}$ des applications unaires sur n sur l'ensemble $n^n = \{0, \ldots, n^n - 1\}$ (*i.e.* $\forall f : n \to n$, $\xi(f) = \sum_{i<n} f(i)n^i$). Soit $*$ le symbole de fonction binaire prédéfini interprété sur tout domaine n^n par $\xi(\circ)$ (où $\circ$ est la composition des fonctions unaires sur n (*i.e.* $\forall x, y, z < n^n : x * y = z$ ssi $\xi^{-1}(x) \circ \xi^{-1}(y) = \xi^{-1}(z)$).

Alors la satisfaction par n de la formule du second ordre

$$\Phi \equiv \exists g_1 \ldots \exists g_p \, \forall x \bigwedge_{i<r} \sigma_i(x) = \tau_i(x)$$

équivaut à la satisfaction sur le domaine n^n de la formule *du premier ordre* :

$$\phi \equiv \exists y_1 \ldots \exists y_p \bigwedge_{i<r} \alpha_i = \beta_i,$$

où les α_i et β_i sont des termes construits comme on le devine à partir des σ_i et τ_i (par exemple, si $\sigma_1(x) = g_2 g_4(x)$, alors $\alpha_1 = y_2 * y_4$).

Ceci permet par exemple de caractériser les ensembles d'entiers de la forme $A^A = \{n^n,\ n \in A\}$, pour les parties $A \subseteq \mathbb{N}$ qui appartiennent à NLIN, par des formules du premier ordre existentiel de signature prédéfinie $\{*, 0, N\}$ ($\tilde{0}$ et $\tilde{s}$ sont deux constantes prédéfinies interprétées par $\xi(0)$ et $\xi(s)$). En dehors de leur signature prédéfinie peu naturelle, les formules qui caractérisent la classe $\{A^A,\ A$ ensemble d'entiers appartenant à NLIN$\}$ sont donc d'une très grande simplicité.

Il s'avère de plus que les symboles $*$, $\tilde{0}$ et $\tilde{s}$ sont définissables au premier ordre sur n^n dans le langage de l'arithmétique. Ceci permet de prouver facilement que pour chaque ensemble d'entiers A :

$$A \in \text{NLIN entraine } A^A \in \text{RUD}.$$

Il est même probablement très accessible de préciser ce résultat en l'écrivant :

$$A \in \text{NLIN entraine } A^A \in \text{RUD}_{lin}, \tag{3.5}$$

où RUD_{lin} est une sous classe de RUD à déterminer, obtenue par restriction sur les préfixes des formules qui définissent les ensembles d'entiers rudimentaires.

Il serait de plus intéressant de déterminer RUD_{lin} de telle sorte que la réciproque l'implication (3.5) soit vraie. Autrement dit :

> $\mathcal{Q}_3$: *Peut-on définir, par restriction sur la syntaxe des formules qui définissent les ensembles d'entiers rudimentaires, la sous-classe RUD_{lin} de RUD telle que $\forall A \subseteq \mathbb{N}$:*
>
> $$A \in NLIN \text{ ssi } A^A \in RUD_{lin}\ ?$$

Prouver des résultats de NLIN-complétude. Nous avons dit dans l'introduction de ce travail qu'un l'intérêt majeur des caractérisations logiques de classe de complexité est de permettre l'obtention de problèmes complets. C'est d'ailleurs en partant des caractérisations de NLIN établies par lui même que Grandjean a prouvé la NLIN-complétude de *RISA* et $SAT_{\mathbb{N}}$.

A priori, plus la caractérisation est simple, plus les preuves de complétude sont faciles. On peut donc légitimement se demander si notre caractérisation conjonctive de NLIN facilite la détermination de *nouveaux* problèmes NLIN-complets.

Si nous n'avons pas de réponse à cette question, du moins pouvons nous avancer que notre caractérisation permet de simplifier sensiblement les preuves de NLIN-complétude déjà établies. Il nous semble en particulier que la complétude de *RISA* devient un corollaire presqu'immédiat de l'égalité $\text{NLIN} = \mathcal{G}^+_\wedge(0, s)$.

Un problème "clef" : la DLIN-complétude

Évoquons pour finir une dernière question, probablement plus déterminante, mais aussi plus difficile, que les précédentes.

Nous avons évoqué l'importance de l'existence de problèmes complets pour une classe de complexité donnée. Nous avons discuté par ailleurs le rôle et la légitimité de la classe DLIN relativement à la formalisation de la notion usuelle d'*algorithme linéaire*.

Si l'on s'accorde sur l'idée que la classe DLIN capture bien cette notion, on comprend la portée qu'aurait la production d'un problème DLIN-complet (*via* des réductions calculables en temps linéaire déterministe *sur machine de Turing*) : elle fournirait une sorte d'algorithme de référence pour tous les problèmes linéaires. D'où l'importance de la question :

Q_4 : *Peut-on exhiber un problème DLIN-complet ?*

Comme nous l'avons précisé plus haut, cette question est fortement liée à l'existence d'une caractérisation logique de la classe DLIN (en présence d'un ordre total ou d'une relation *successeur*). D'où la question :

Q_5 : *Peut-on caractériser exactement la classe DLIN par une logique "naturelle" ?*

Cette question, initialement posée par Etienne Grandjean dans différents articles, est jugée par son auteur comme très difficile. En particulier, le codage du calcul d'une RAM fonctionnant en temps linéaire *non déterministe* (codage établi par Grandjean et décrit dans le premier chapitre de cette thèse), ne semble pas pouvoir s'adapter au cas *déterministe*

Bibliographie

[1] **A.V. Aho, J.E. Hopcroft and J. D. Ullman**, *The design and Analysis of Computer Algorithms,* Addison-Wesley, Reading, 1974.

[2] **J.L. Balcázar, J. Díaz et J. Gabaró**, *Structural Complexity, vol. 1 et 2*, EATCS, Monographs on Theoritical Computer Science, 1988, Springer-Verlag, New-York-Berlin-Heidelberg.

[3] **J.H. Bennett**, *On Spectra,* Doctoral Dissertation, Princeton University, Princeton, N.J., 1962.

[4] **J.R. Büchi**, *Weak second order arithmetic and finite automata*, Z.Math. Logik Grundlagen Math. 6 (1960), pp. 66-92.

[5] **Cobham**, *Intrinsec difficulty of functions*, Logic, Methodology and Philosophy of Sciences Colloquium, Proc. of the 1964 International congress (1965), North Holland.

[6] **S.A. Cook**, *The complexity of theorem proving procedures*, 3rd STOC (1971), pp. 151-158.

[7] **S.A. Cook**, *A hierarchy for nondeterministic time complexity*, J. Comput. Systems Sci. 7 (1973), pp. 343-353.

[8] **B. Courcelle**, *The expression of graph properties and graph transformations in monadic second-order logic*, In G. Rozenberg, editor, Handbook of graph grammars, vol. 1: Foundations. World Scientific Publisher, 1996.

[9] **N. Creignou**, *Temps linéaire et problèmes NP-complets*, thèse de doctorat, Université de Caen, France, 1993.

[10] **N. Creignou**, *Problems linearly equivalent to satisfiability or a uniform method to prove NP-completness*, CSL'92, Lecture Notes in Comput. Sci., 702 (1993), pp. 115-133.

[11] **M. De Rougemont**, *Second-order and inductive definability on finite structures*, Zeitschrift für Mathematische Logik and Grundlagen der Mathematik, 33 (1987), pp. 47-63.

[12] **C. Dimitracopoulos**, *Matijasevic's Theorem and Fragments of Arithmetic.* Ph. D. Thesis, University of Manchester, 1980.

[13] **A. Durand**, Les spectres de k fonctions sont spectres d'une fonction et d'une bijection. Communication personnelle.

[14] **A. Durand**, Binary-NP, linear time and the power of one first-order quantifier. Rapport technique du GREYC (1996) Université de Caen.

[15] **A. Durand, Y. Ranaivoson**, *First-order spectra with one binary predicate*, Theoretical Computer Sciences, to appear in 1996.

[16] **H.D Ebbinghaus, J. Flum**, *Finite Model Theory*, Berlin, NY, Springer, 1995.

[17] **J. Edmonds**, *Pathes, trees and flowers*, Canadian J. of Math, 1965.

[18] **C. Elgot**, *Decision problems of finite-automata design and related arithmetics*, Transaction of AMS, 98 (1961), pp. 21-51.

[19] **T. Eiter, G. Gottlob et Y. Gurevich**, *Normal Form for Second-Order Logic over Finite Structures, and Classification of NP Optimization Problems.* Technical report CD-TR-95/77, Vienna University of Technology (Technische Universität), 1995.

[20] **R. Fagin**, *Generalized first-order spectra and polynomial-time recognizable sets*, in Complexity of Computations, R.Karp, ed., SIAM-AMS Proc. 7, 1974, pp. 43-73.

[21] **R. Fagin**, *A Spectrum Hierarchy*, Math. Logik Grundlagen Math., 21, 1975, pp. 123-134.

[22] **R. Fagin**, *Monadic generalized spectra*, Math. Logik Grundlagen Math., 21, 1975, pp. 89-96.

[23] **R. Fagin**, *Finite model theory: a personal perspective*, Theoritical Computer Science, 116 (1993), pp. 3-31.

[24] **M.R. Garey, D.S. Johnson**, *Computers and intractibility: a guide to the theory of NP-completness*, Freeman, San Francisco (1979).

[25] **G. Gottlob**, *Relativized Logspace and Generalized Quantifiers over Finite Structures*, LICS'95.

[26] **E. Grädel**, *On logical descriptions of some concepts in structural complexity theory*, in E.Börger, H. Kleine Buning and M.M. Richter, editors. CSL'89: 3rd Workshop on Computer Science Logic, volume 440 of LNCS, pages 163-175, Springer Verlag, 1990.

[27] **E. Grädel**, *On the notion of linear time computability*, International J. of Foundations of Computer Science, 1 (1990), pp. 295-307.

[28] **E. Grädel**, *Capturing complexity classes by fragment of second-order logic*, TCS, 101 (1992), pp. 35-57.

[29] **E. Grandjean**, *The spectra of first-order sentence and computational complexity*, SIAM J.Comput., 13 (1984), pp. 356-373.

[30] **E. Grandjean**, *Universal quantifiers and time complexity of random access machines*, Math. Systems Theory, 18 (1985), pp. 171-187.

[31] **E. Grandjean**, *A natural NP-complete problem with a nontrivial lower bound*, SIAM J.Comput.,17 (1988), pp. 786-809.

[32] **E. Grandjean**, *A nontrivial lower bound for an NP problem on automata*, SIAM J. Comput., 19 (1990), pp. 438-451.

[33] **E. Grandjean**, *First-order spectra with one variable*, J. of Comput. and System Sci., 40 (1990), pp. 136-153.

[34] **E. Grandjean**, *Invariance properties of RAM's and linear time*, Computational Complexity, 4 (1994), pp. 62-106.

[35] **E. Grandjean**, *Linear time algorithms and NP-complete problems*, SIAM J. Comput., 23 (1994), pp. 573-597.

[36] **E. Grandjean**, *Sorting, linear time and the satisfiability problem*, to appear in special issue of Annals of Math. and Artificial Intelligence, 1996.

[37] **E. Grandjean, J.M. Robson**, *Ram with compact memory : a realistic and robust model of computation*, proc. CSL 90, Lect. Note Comput. Sci., 533 (1991), pp. 195-233.

[38] **E. Grandjean, F.Olive**, *Monadic Logical Definability of NP-Complete Problems*, CSL'94 Proc. LNCS 933 (1995), pp. 190-204.

[39] **Y. Gurevich, S. Shelah**, *Nearly linear time*, Lect.notes Comput.Sci. 363 (1989), Springer-Verlag, pp. 108-118.

[40] **Y. Gurevich**, *Toward Logic Tailored for Computational Complexity*, Computation and Proof Theory, (M.M.Richter et. al., eds.).Springer-Verlag Lecture Notes in Math. Vol.1104, pp. 175-216, Springer-Verlag, New York/Berlin, 1984.

[41] **Y. Gurevich**, *Logic and the challenge of computer science*, Current Trends in Theorical Computer Science, (E.Boerger Ed.), pp. 1-55, Computer science, Rockville, MD, 1986.

[42] **K. Harrow**, *Sub-elementary Classes of Functions and Relations.* Doctoral Dissertation, New-York University, Department of Mathematics, 1973.

[43] **N. Immerman**, *Relational Queries Computable in Polynomial Time*, 14th ACM STOC Symp., 1982, pp.147-152. Also appeared in revised form in Information and Control, 68 (1986), pp. 86-104.

[44] **N. Immerman**, *Languages that Capture Complexity Classes.* SIAM J. Comput., vol.16, No.4, 1987, pp. 760-777.

[45] **N. Immerman**, *Descriptive and Computational Complexity*, in J. Hartmanis ed., Computational Complexity Theory, Proc. of AMS Symposia in Appl. Math. 38 (1989), pp. 75-91.

[46] **N. Immerman**, *Descriptive complexity: a logician approach to computation*, Notice of AMS, 42 (1995), pp. 1127-1132.

[47] **N.G. Jones, A.L. Selmans**, *Turing Machines and the spectra of first-order formulas*, J. Symbolic Logic, 30 (1974), pp. 139-150.

[48] **R.M. Karp**, *Reducibility among combinatorial problems*, IBM Symp.1972, Complexity of Computers Computations, Plenum Press, New York, 1972.

[49] **P.G. Kolaitis, M.N. Thakur**, *Logical definability of NP-optimization problems*, Technical report UCSC-CRL-90-48, Computer and Information Sciences, University of California, Santa-Cruz, 1990.

[50] **C. Lautemann, T. Schwentick et D. Thérien**, *Logic for Context-Free Languages*, Technical report, Informatik-Bericht Nr.4/94, Johannes Gutemberg - Universität Mainz.

[51] **D. Leivant**, *Descriptive Characterizations of Computational Complexity*, Journal of Computer and System Sciences, 39 (1989), pp. 51-83.

[52] **J.F. Lynch**, *Complexity classes and theories of finite models*, Math. Systems Theory, 15 (1982), pp. 127-144.

[53] **J.F. Lynch**, *The quantifier structure of sentences that characterize nondeterministic time complexity*, in Comput. Complexity, 2 (1992), pp. 40-66.

[54] **L. Lovász, P. Gásc**, *Some remarks on generalized spectra*, Z. Math. Logik. Grundlag. Math., 23 (1977), pp. 547-553.

[55] **Y. Matijassevitch**, *Diophantine representation of recursively enumerable predicates*, Proceedings International Congress of Mathematics, pp. 234-238.

[56] **M. More**, *Rudimentary representations of spectra.* prépublication du LLAIC1, Numéro 40, 1994.

[57] **C. Papadimitriou**, *Computational Complexity*, Addison-Wesley, 1994

[58] **C. Papadimitriou, M. Yannakakis**, *Optimization, Approximation and Complexity*, in J.Comput.System Sci., 43 (1991), pp. 425-440.

[59] **J.E. Pin**, *Logic, semigroups and automata on words*, to appear in special issue of Annals of Math. and Artificial Intelligence, 1995.

[60] **P. Pudlàk**, *The observational predicate calculus and complexity of computations*, Comment. Math. Univ. Carolin., 16 (1975), pp. 395-398.

[61] **W. Quine**, *Concatenation as a Basis for Arithmetic from a Logical Point of View.* J. Symb. Logic, 11, 1946, p. 105-114. Selected Logic Papers, ch V, Random House, 1966, pp. 70-82.

[62] **Y. Ranaivoson**, *Bornes inférieures non triviales pour des problèmes naturels en théorie des graphes et des automates*, thèse de doctorat, Université de Caen, France, 1990.

[63] **Y. Ranaivoson**, *Nontrivial lower bounds for some NP-complete problems on directed graphs*, CSL 90, Lecture Notes in Computer Sci., 533 (1991), pp. 318-339.

[64] **D. Richard**, *Equivalence of some questions in mathematical logic with some conjectures in number theory*, Number theory and applications, R. Mollin edition.

[65] **C.P. Schnorr**, *Satisfiability is quasilinear complete in NQL*, J. ACM, 25 (1978), pp. 136-145.

[66] **R. Smullyan**, *Theory of Formal Systems.* Annals of Math. Studies, 47 (1961), Princeton University Press, Princeton, N.J..

[67] **J. Stern**, *Fondements Mathématiques de l'Informatique*, Mac Graw-Hill ed., 1990.

[68] **L.J. Stockmeyer**, *The polynomial time hierarchy*, Theor. Comput. Sci. 3, 1976, pp. 1-22.

[69] **R. E. Tarjan**, *Depth-first search and linear time algorithms*, SIAM J. Comput., 1 (1972), pp. 146-160.

[70] **W. Thomas**, *Classifying regular events in symbolic logic*, J. Comput. System Sci. 25 (1982), pp. 360-376.

[71] **J. van Benthem, K. Doets**, *Higher Order Logic*, in D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic* (1983), Vol. 1, chapter 1.4, pp. 275-329, D. Reidel Publishing Company.

[72] **M. Vardi**, *Complexity of Relational Query Languages*, 14th ACM Symp. on Theory of Computation, 1982, pp. 137-146.

[73] **K. Wagner, G. Weichsung**, *Computational Complexity*, Reidel, Berlin, 1986.

[74] **A. Woods**, *Some Problems in Logic and Number Theory, and their Connections.* Doctoral Dissertation, University of Manchester, 1981.

[75] **C. Wrathall**, *Rudimentary predicates and relative computation.* SIAM J. Comput. vol. 7 no.2, 1978, pp. 194-209.