



50376
1993
302



Laboratoire d'Informatique
Fondamentale de Lille



50376
1993
302

N. Ordre 1225

Thèse

présentée à

L'Université des Sciences et Technologies de Lille

pour l'obtention du titre de

Docteur en Informatique

par

Yves ANDRE

Equivalence de Transducteurs Lettre à Lettre d'Arbres



Soutenue le 16 décembre 1993 devant le jury:

Jean-Claude RAOULT,	Président.
Christian CHOFFRUT,	
Helmut SEIDL,	Rapporteurs.
Francis BOSSUT,	
Max DAUCHET,	
Sophie TISON,	Examineurs.

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M. H. LEFEBVRE, M. PARREAU

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT
ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PARREAU, J. LOMBARD, M. MIGEON, J. CORTOIS, A. DUBRULLE

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

M. P. LOUIS

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CHAMLEY Hervé
M. CONSTANT Eugène
M. ESCAIG Bertrand
M. FOURET René
M. GABILLARD Robert
M. LABLACHE COMBIER Alain
M. LOMBARD Jacques
M. MACKÉ Bruno

Géotechnique
Electronique
Physique du solide
Physique du solide
Electronique
Chimie
Sociologie
Physique moléculaire et rayonnements atmosphériques

M. MIGEON Michel
M. MONTREUIL Jean
M. PARREAU Michel
M. TRIDOT Gabriel

EUDIL
Biochimie
Analyse
Chimie appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre
M. BIAYS Pierre
M. BILLARD Jean
M. BOLLLY Bénoni
M. BONNELLE Jean Pierre
M. BOSCO Denis
M. BOUGHON Pierre
M. BOURIQUET Robert
M. BRASSELET Jean Paul
M. BREZINSKI Claude
M. BRIDOUX Michel
M. BRUYELLE Pierre
M. CARREZ Christian
M. CELET Paul
M. COEURE Gérard
M. CORDONNIER Vincent
M. CROSNIER Yves
Mme DACHARRY Monique
M. DAUCHET Max
M. DEBOURSE Jean Pierre
M. DEBRABANT Pierre
M. DECLERCQ Roger
M. DEGAUQUE Pierre
M. DESCHEPPER Joseph
Mme DESSAUX Odile
M. DHAINAUT André
Mme DHAINAUT Nicole
M. DJAFARI Rouhani
M. DORMARD Serge
M. DOUKHAN Jean Claude
M. DUBRULLE Alain
M. DUPOUY Jean Paul
M. DYMENT Arthur
M. FOCT Jacques Jacques
M. FOUQUART Yves
M. FOURNET Bernard
M. FRONTIER Serge
M. GLORIEUX Pierre
M. GOSSELIN Gabriel
M. GOUDMAND Pierre
M. GRANELLE Jean Jacques
M. GRUSON Laurent
M. GUILBAULT Pierre
M. GUILLAUME Jean
M. HECTOR Joseph
M. HENRY Jean Pierre
M. HERMAN Maurice
M. LACOSTE Louis
M. LANGRAND Claude

Astronomie
Géographie
Physique du Solide
Biologie
Chimie-Physique
Probabilités
Algèbre
Biologie Végétale
Géométrie et topologie
Analyse numérique
Chimie Physique
Géographie
Informatique
Géologie générale
Analyse
Informatique
Electronique
Géographie
Informatique
Gestion des entreprises
Géologie appliquée
Sciences de gestion
Electronique
Sciences de gestion
Spectroscopie de la réactivité chimique
Biologie animale
Biologie animale
Physique
Sciences Economiques
Physique du solide
Spectroscopie hertzienne
Biologie
Mécanique
Métallurgie
Optique atmosphérique
Biochimie structurale
Ecologie numérique
Physique moléculaire et rayonnements atmosphériques
Sociologie
Chimie-Physique
Sciences Economiques
Algèbre
Physiologie animale
Microbiologie
Géométrie
Génie mécanique
Physique spatiale
Biologie Végétale
Probabilités et statistiques

M. LATTEUX Michel	Informatique
M. LAVEINE Jean Pierre	Paléontologie
Mme LECLERCQ Ginette	Catalyse
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique atomique et moléculaire
M. LEROY Jean Marie	Spectrochimie
M. LHENAFF René	Géographie
M. LHOMME Jean	Chimie organique biologique
M. LOUAGE François	Electronique
M. LOUCHEUX Claude	Chimie-Physique
M. LUCQUIN Michel	Chimie physique
M. MAILLET Pierre	Sciences Economiques
M. MAROUF Nadir	Sociologie
M. MICHEAU Pierre	Mécanique des fluides
M. PAQUET Jacques	Géologie générale
M. PASZKOWSKI Stéfan	Mathématiques
M. PETIT Francis	Chimie organique
M. PORCHET Maurice	Biologie animale
M. POUZET Pierre	Modélisation - calcul scientifique
M. POVY Lucien	Automatique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. RAMAN Jean Pierre	Sciences de gestion
M. SALMER Georges	Electronique
M. SCHAMPS Joël	Spectroscopie moléculaire
Mme SCHWARZBACH Yvette	Géométrie
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
M. SLIWA Henri	Chimie organique
M. SOMME Jean	Géographie
Melle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. THIEBAULT François	Sciences de la Terre
M. THOMAS Jean Claude	Géométrie - Topologie
M. THUMERELLE Pierre	Démographie - Géographie humaine
M. TILLIEU Jacques	Physique théorique
M. TOULOTTE Jean Marc	Automatique
M. TREANTON Jean René	Sociologie du travail
M. TURRELL Georges	Spectrochimie infrarouge et raman
M. VANECCLOO Nicolas	Sciences Economiques
M. VAST Pierre	Chimie inorganique
M. VERBERT André	Biochimie
M. VERNET Philippe	Génétique
M. VIDAL Pierre	Automatique
M. WALLART François	Spectrochimie infrarouge et raman
M. WEINSTEIN Olivier	Analyse économique de la recherche et développement
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ABRAHAM Francis	Composants électroniques
M. ALLAMANDO Etienne	Biologie des organismes
M. ANDRIES Jean Claude	Analyse
M. ANTOINE Philippe	Génétique
M. BALL Steven	Biologie animale
M. BART André	Génie des procédés et réactions chimiques
M. BASSERY Louis	Géographie
Mme BATTIAU Yvonne	Systèmes électroniques
M. BAUSIERE Robert	Mécanique
M. BEGUIN Paul	Physique atomique et moléculaire
M. BELLET Jean	Physique atomique, moléculaire et du rayonnement
M. BERNAGE Pascal	Sciences Economiques
M. BERTHOUD Arnaud	Sciences Economiques
M. BERTRAND Hugues	Analyse
M. BERZIN Robert	Physique de l'état condensé et cristallographie
M. BISKUPSKI Gérard	Algèbre
M. BKOUCHE Rudolphe	Biologie végétale
M. BODARD Marcel	Biochimie métabolique et cellulaire
M. BOHIN Jean Pierre	Mécanique
M. BOIS Pierre	Génie civil
M. BOISSIER Daniel	Spectrochimie
M. BOIVIN Jean Claude	Physique
M. BOUCHER Daniel	Biologie appliquée aux enzymes
M. BOUQUELET Stéphane	Gestion
M. BOUQUIN Henri	Chimie
M. BROCARD Jacques	Paléontologie
Mme BROUSMICHE Claudine	Mécanique
M. BUISINE Daniel	Biologie animale
M. CAPURON Alfred	Géographie humaine
M. CARRE François	Chimie organique
M. CATTEAU Jean Pierre	Sciences Economiques
M. CAYATTE Jean Louis	Electronique
M. CHAPOTON Alain	Biochimie structurale
M. CHARET Pierre	Composants électroniques optiques
M. CHIVE Maurice	Informatique théorique
M. COMYN Gérard	Composants électroniques et optiques
Mme CONSTANT Monique	Psychophysiologie
M. COQUERY Jean Marie	Sciences Economiques
M. CORIAT Benjamin	Paléontologie
Mme CORSIN Paule	Physique nucléaire et corpusculaire
M. CORTOIS Jean	Chimie organique
M. COUTURIER Daniel	Tectonique géodynamique
M. CRAMPON Norbert	Biologie
M. CURGY Jean Jacques	Physique théorique
M. DANGOISSE Didier	Analyse
M. DE PARIS Jean Claude	Composants électroniques et optiques
M. DECOSTER Didier	Electrochimie et Cinétique
M. DEJAEGER Roger	Informatique
M. DELAHAYE Jean Paul	Physiologie animale
M. DELORME Pierre	Sciences Economiques
M. DELORME Robert	Sociologie
M. DEMUNTER Paul	Physique atomique, moléculaire et du rayonnement
Mme DEMUYNCK Claire	Informatique
M. DENEL Jacques	Physique du solide - cristallographie
M. DEPREZ Gilbert	

M. DERIEUX Jean Claude
 M. DERYCKE Alain
 M. DESCAMPS Marc
 M. DEVRAINNE Pierre
 M. DEWAILLY Jean Michel
 M. DHAMELINCOURT Paul
 M. DI PERSIO Jean
 M. DUBAR Claude
 M. DUBOIS Henri
 M. DUBOIS Jean Jacques
 M. DUBUS Jean Paul
 M. DUPONT Christophe
 M. DUTHOIT Bruno
 Mme DUVAL Anne
 Mme EVRARD Micheline
 M. FAKIR Sabah
 M. FARVACQUE Jean Louis
 M. FAUQUEMBERGUE Renaud
 M. FELIX Yves
 M. FERRIERE Jacky
 M. FISCHER Jean Claude
 M. FONTAINE Hubert
 M. FORSE Michel
 M. GADREY Jean
 M. GAMBLIN André
 M. GOBLOT Rémi
 M. GOURIEROUX Christian
 M. GREGORY Pierre
 M. GREMY Jean Paul
 M. GREVET Patrice
 M. GRIMBLOT Jean
 M. GUELTON Michel
 M. GUICHAOUA André
 M. HAIMAN Georges
 M. HOUDART René
 M. HUEBSCHMANN Johannes
 M. HUTTNER Marc
 M. ISAERT Noël
 M. JACOB Gérard
 M. JACOB Pierre
 M. JEAN Raymond
 M. JOFFRE Patrick
 M. JOURNAL Gérard
 M. KOENIG Gérard
 M. KOSTRUBIEC Benjamin
 M. KREMBEL Jean
 Mme KRIFA Hadjila
 M. LANGEVIN Michel
 M. LASSALLE Bernard
 M. LE MEHAUTE Alain
 M. LEBFEVRE Yannic
 M. LECLERCQ Lucien
 M. LEFEBVRE Jacques
 M. LEFEBVRE Marc
 M. LEFEBVRE Christian
 Mlle LEGRAND Denise
 M. LEGRAND Michel
 M. LEGRAND Pierre
 Mme LEGRAND Solange
 Mme LEHMANN Josiane
 M. LEMAIRE Jean

Microbiologie
 Informatique
 Physique de l'état condensé et cristallographie
 Chimie minérale
 Géographie humaine
 Chimie physique
 Physique de l'état condensé et cristallographie
 Sociologie démographique
 Spectroscopie hertzienne
 Géographie
 Spectrométrie des solides
 Vie de la firme
 Génie civil
 Algèbre
 Génie des procédés et réactions chimiques
 Algèbre
 Physique de l'état condensé et cristallographie
 Composants électroniques
 Mathématiques
 Tectonique - Géodynamique
 Chimie organique, minérale et analytique
 Dynamique des cristaux
 Sociologie
 Sciences économiques
 Géographie urbaine, industrielle et démographie
 Algèbre
 Probabilités et statistiques
 I. A. E.
 Sociologie
 Sciences Economiques
 Chimie organique
 Chimie physique
 Sociologie
 Modélisation, calcul scientifique, statistiques
 Physique atomique
 Mathématiques
 Algèbre
 Physique de l'état condensé et cristallographie
 Informatique
 Probabilités et statistiques
 Biologie des populations végétales
 Vie de la firme
 Spectroscopie hertzienne
 Sciences de gestion
 Géographie
 Biochimie
 Sciences Economiques
 Algèbre
 Embryologie et biologie de la différenciation
 Modélisation, calcul scientifique, statistiques
 Physique atomique, moléculaire et du rayonnement
 Chimie physique
 Physique
 Composants électroniques et optiques
 Pétrologie
 Algèbre
 Astronomie - Météorologie
 Chimie
 Algèbre
 Analyse
 Spectroscopie hertzienne

M. LE MAROIS Henri	Vie de la firme
M. LEMOINE Yves	Biologie et physiologie végétales
M. LESCURE François	Algèbre
M. LESENNE Jacques	Systèmes électroniques
M. LOCQUENEUX Robert	Physique théorique
Mme LOPES Maria	Mathématiques
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU François	Sciences économiques
M. MAHIEU Jean Marie	Optique - Physique atomique
M. MAIZIERES Christian	Automatique
M. MANSY Jean Louis	Géologie
M. MAURISSON Patrick	Sciences Economiques
M. MERIAUX Michel	EUDIL
M. MERLIN Jean Claude	Chimie
M. MESMACQUE Gérard	Génie mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MOCHE Raymond	Modélisation,calcul scientifique,statistiques
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie organique
M. MORE Marcel	Physique de l'état condensé et cristallographie
M. MORTREUX André	Chimie organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
M. NIAY Pierre	Physique atomique,moléculaire et du rayonnement
M. NICOLE Jacques	Spectrochimie
M. NOTELET Francis	Systèmes électroniques
M. PALAVIT Gérard	Génie chimique
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. PERTUZON Emile	Physiologie animale
M. PETIT Daniel	Biologie des populations et écosystèmes
M. PLIHON Dominique	Sciences Economiques
M. PONSOLLE Louis	Chimie physique
M. POSTAIRE Jack	Informatique industrielle
M. RAMBOUR Serge	Biologie
M. RENARD Jean Pierre	Géographie humaine
M. RENARD Philippe	Sciences de gestion
M. RICHARD Alain	Biologie animale
M. RIETSCH François	Physique des polymères
M. ROBINET Jean Claude	EUDIL
M. ROGALSKI Marc	Analyse
M. ROLLAND Paul	Composants électroniques et optiques
M. ROLLET Philippe	Sciences Economiques
Mme ROUSSEL Isabelle	Géographie physique
M. ROUSSIGNOL Michel	Modélisation,calcul scientifique,statistiques
M. ROY Jean Claude	Psychophysiologie
M. SALERNO Francis	Sciences de gestion
M. SANCHOLLE Michel	Biologie et physiologie végétales
Mme SANDIG Anna Margarete	
M. SAWERYSYN Jean Pierre	Chimie physique
M. STAROSWIECKI Marcel	Informatique
M. STEEN Jean Pierre	Informatique
Mme STELLMACHER Irène	Astronomie - Météorologie
M. STERBOUL François	Informatique
M. TAILLIEZ Roger	Génie alimentaire
M. TANRE Daniel	Géométrie - Topologie
M. THERY Pierre	Systèmes électroniques
Mme TJOTTA Jacqueline	Mathématiques
M. TOURSEL Bernard	Informatique
M. TREANTON Jean René	Sociologie du travail

M. TURREL Georges
M. VANDIJK Hendrik
Mme VAN ISEGHEM Jeanine
M. VANDORPE Bernard
M. VASSEUR Christian
M. VASSEUR Jacques
Mme VIANO Marie Claude
M. WACRENIER Jean Marie
M. WARTEL Michel
M. WATERLOT Michel
M. WEICHERT Dieter
M. WERNER Georges
M. WIGNACOURT Jean Pierre
M. WOZNIAK Michel
Mme ZINN JUSTIN Nicole

Spectrochimie infrarouge et raman

Modélisation, calcul scientifique, statistiques

Chimie minérale

Automatique

Biologie

Electronique

Chimie inorganique

géologie générale

Génie mécanique

Informatique théorique

Spectrochimie

Algèbre

Je remercie Jean-Claude Raoult qui me fait l'honneur de présider le jury de cette thèse.

J'exprime ma gratitude à Christian Choffrut qui s'est intéressé à ce travail et en a été un rapporteur attentif et enthousiaste. Je le remercie pour les remarques et conseils qu'il m'a prodigués.

Trotz sprachlicher Schwierigkeiten hat Helmut Seidl akzeptiert, die Ergebnisse dieser Doktorarbeit zu beurteilen, und darüber mit Interesse und Aufmerksamkeit Bericht zu erstatten. Ich bedanke mich sehr herzlich dafür ¹.

C'est sous la direction de Max Dauchet que ce travail a été entrepris et réalisé. Son exceptionnelle patience, sa gentillesse et sa compétence m'ont grandement aidé. Je lui exprime ici mon plus profond respect.

Ce travail doit beaucoup à Francis Bossut qui en a suivi la réalisation avec disponibilité et enthousiasme. Ses nombreux conseils et ses remarques toujours fécondes furent indispensables. Je le remercie sincèrement et j'espère que nous aurons encore souvent l'occasion de travailler ensemble.

Mes remerciements vont également à Sophie Tison qui me fait l'honneur de participer au jury.

Je tiens, enfin, à exprimer mes remerciements à ceux qui m'ont apporté aide et soutien au cours de l'élaboration de cette thèse: mes collègues du LIFL, avec une pensée particulière pour Marc Tommasi, et ceux du département Informatique de l'IUT "A" de Lille 1, et plus particulièrement Edouard Duszinski qui a accepté mon manque de disponibilité pendant la période de rédaction.

Henri Glanc a assuré la reproduction de cette thèse. Je lui en suis reconnaissant.

¹Malgré les problèmes linguistiques, Helmut Seidl a accepté de juger ce travail. Il en a été un rapporteur attentif. Je lui exprime ma profonde reconnaissance.

Table des matières

Introduction	5
L'objet de notre travail	5
Les outils manipulés	7
Notre travail	14
1 Préliminaires	17
1.1 Arbres	17
1.2 Systèmes de réécriture	18
1.3 Automates d'arbres	20
1.4 Transducteurs d'arbres	20
1.5 Problèmes de décision	23
2 Transducteurs linéaires et complets	25
2.1 Transducteurs sans-torsion	25
2.1.1 Décidabilité de l'équivalence	26
2.1.2 Algorithme de décision	28
2.2 Transducteurs linéaires et complets	29
2.2.1 Préliminaires	29
2.2.2 Transformations initiales réalisées à partir de deux ensembles d'états globalement équivalents d'un transducteur infinitaire	31
2.2.3 Forme Λ -normalisée d'un transducteur infinitaire	37
2.2.4 Décidabilité de l'équivalence dans LCT-LL	49
2.2.5 Cas des transducteurs ascendants	50
2.2.6 Cas de lettres de rang supérieur à 2	50
3 Transducteurs descendants linéaires	55
3.1 Transformations initiales	57
3.2 Forme Λ -normalisée d'un transducteur infinitaire	59
3.3 Décidabilité de l'équivalence	67

4	Automates avec contraintes d'équivalence	69
4.1	Automates avec contraintes d'équivalence	72
4.1.1	Définitions	72
4.1.2	Exemple	73
4.1.3	Propriétés	74
4.2	Classes REC_τ	77
4.2.1	Systèmes de réécriture lettre à lettre	78
4.2.2	Classes REC_τ	81
4.2.3	Automate des formes normales associé à un automate de REC_τ	82
4.2.4	Décidabilité de l'équivalence dans REC_τ	94
5	Transducteurs descendants complets	97
5.1	Transformations initiales	99
5.2	Formes Λ -normalisées	101
5.3	Décidabilité de l'équivalence	103
5.3.1	Codage des couples d'arbres	105
5.3.2	Automate de REC_τ associé à la forme Λ -normalisée d'un transducteur	105
5.4	Le cas général descendant	111
5.4.1	Cas général	111
5.4.2	Regard sur les techniques utilisées	113
6	Transducteurs ascendants	115
6.1	Extension des définitions	115
6.2	Transformations finales	116
6.3	Formes Λ -normalisées	119
6.3.1	Formes Λ -semi-normalisées	119
6.3.2	Formes Λ -normalisées	124
6.4	Décidabilité de l'équivalence	127
	Conclusion	129
	Index	131

Introduction

L'objet de notre travail

Cette thèse est une contribution à l'étude du problème de l'équivalence pour une classe de transducteurs non-déterministes d'arbres: les transducteurs lettre à lettre. Nos preuves s'appuyant sur le codage de transducteurs dans des automates, pour résoudre le problème dans le cas non-linéaire nous sommes amenés à introduire de nouvelles familles d'automates.

Pour mieux décrire notre travail, nous présentons d'abord et de façon informelle un exemple de transducteur.

Après avoir rappelé les notions de base, nous décrirons les différentes parties de notre travail. Enfin des extensions possibles de nos résultats seront proposées en conclusion.

Reprenons l'exemple classique du calcul de la factorielle d'un entier:

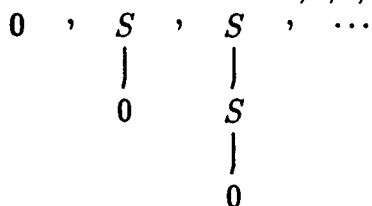
$$\begin{cases} f(0) &= 1 \\ f(n+1) &= (n+1) \times f(n) \end{cases}$$

Le calcul de $f(2)$ peut se présenter ainsi:

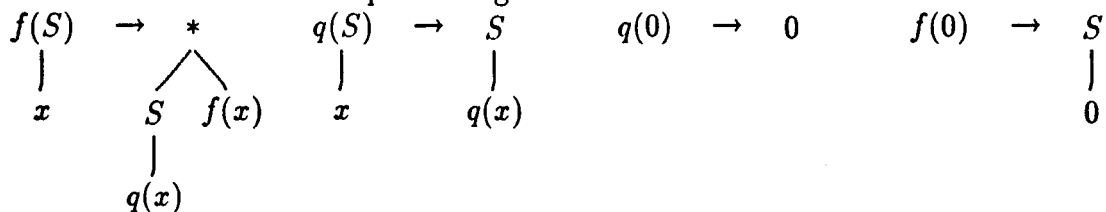
$$\begin{aligned} f(2) &= 2 \times f(1) \\ &= 2 \times [1 \times f(0)] \\ &= 2 \times [1 \times 1] \\ &= 2 \times 1 \\ &= 2 \end{aligned}$$

Les équations définissant $f(n)$ ont en fait été utilisées de la gauche vers la droite. On peut alors associer à cette définition un transducteur descendant d'arbres de la façon suivante:

Les entiers naturels 0, 1, 2, ... sont représentés par les arbres:



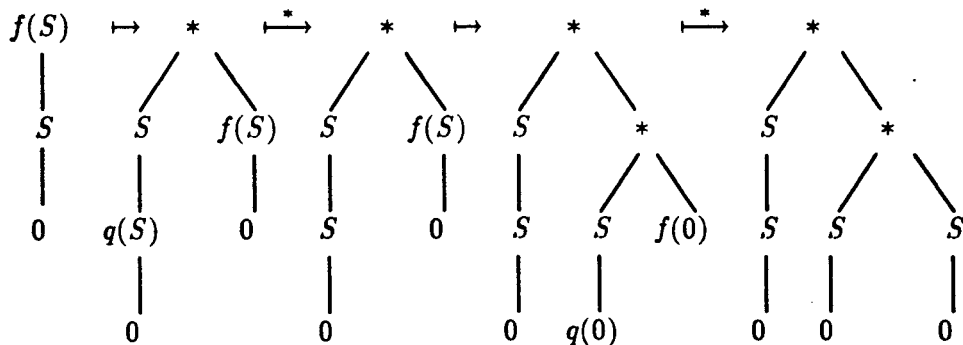
Le transducteur est défini par les règles suivantes



Ce transducteur, en consommant un arbre représentant un entier naturel, produit l'arbre correspondant à la factorielle de cet entier. Dans l'arbre obtenu en sortie, le symbole * doit être interprété comme le produit d'entiers.

On peut remarquer que ce transducteur comporte deux états: *q* à partir duquel un entier est recopié et *f* état à partir duquel est effectué le calcul de la factorielle.

Les transformations décrites ci-après correspondent au calcul de la factorielle de l'entier 2.



Ce transducteur n'est pas lettre à lettre à cause de la première règle qui produit un arbre de profondeur supérieure à 1.

Par contre, en introduisant le nouveau symbole "1", le transducteur défini par:

$$\begin{array}{ccccccc}
 f(S) & \rightarrow & * & q(S) & \rightarrow & S & q(0) & \rightarrow & 1 & f(0) & \rightarrow & 1 \\
 | & & \swarrow \searrow & | & & | & & & & & & \\
 x & & q(x) & f(x) & & q(x) & & & & & &
 \end{array}$$

est lettre à lettre. Les transformations associées correspondent toujours au calcul, pour tout entier n , de la factorielle de n .

Ce transducteur est non-linéaire, car x apparaît deux fois dans le membre droit de la première règle, mais il est déterministe puisque, pour tout entier, un seul calcul est possible.

Considérons maintenant les deux transducteurs définis par:

$$\begin{array}{ll}
 q(C(x)) \rightarrow *(q_0(x), q_1(x)) & q(C(x)) \rightarrow *(q_1(x), q_0(x)) \\
 q(C(x)) \rightarrow *(q'_0(x), q'_0(x)) & q(C(x)) \rightarrow *(q'_1(x), q'_1(x)) \\
 q_0(S(x)) \rightarrow S(q_1(x)) & q_1(S(x)) \rightarrow S(q_2(x)) \\
 q_2(S(x)) \rightarrow S(q_2(x)) & q_2(0) \rightarrow 0 \\
 q'_1(S(x)) \rightarrow S(q'_0(x)) & q'_0(0) \rightarrow 0
 \end{array}$$

et

$$\begin{array}{ll}
 k(C(x)) \rightarrow *(k_1(x), k_1(x)) & \\
 k_1(S(x)) \rightarrow S(k_1(x)) & k_1(0) \rightarrow 0.
 \end{array}$$

Ces deux transducteurs lettre à lettre sont équivalents, dans la mesure où ils calculent tous deux le carré d'un entier, mais le premier le fait de façon non déterministe. Dans cette thèse, nous nous intéressons à l'équivalence de tels schémas de calcul non-déterministes. On verra que l'association de la non-linéarité et du non-déterminisme complique les algorithmes de décision d'équivalence.

Rappelons que si l'équivalence est décidable dans le cas des transducteurs déterministes (Z. Esik [Esi80], Z. Zachar [Zac78]), elle est en général indécidable dans le cas non-déterministe. Comme résultat positif, citons la décidabilité de l'équivalence pour les transducteurs ascendants finiment-valués établie par H. Seidl [Sei90].

Les outils manipulés

La structure d'arbre est employée dans des domaines très différents. On la retrouve pour la description de hiérarchies ou de décompositions, pour l'énumération de choix,

en généalogie et même dans des livres de grammaire destinés aux enfants, pour ne citer que quelques exemples non-informatiques. Le vocabulaire utilisé est celui de la botanique. On parlera ainsi de racine, de branche, de noeud...

Ces objets sont aussi largement utilisés par les informaticiens comme structures de données et, plus généralement, comme termes (compositions formelles de fonctions). Nous considérons ici des arbres finis (ou termes) dont les noeuds sont étiquetés par des lettres d'arité fixe; l'arité étant le nombre de descendants possibles.

L'étude des langages d'arbres, encore appelés *forêts*, s'est beaucoup développée ces vingt dernières années. Les principaux concepts apparus sont en fait des généralisations naturelles de ceux existant dans les mots. Cependant la situation diffère de celles des mots puisque de nombreuses propriétés ne se transposent pas.

La classe de forêts la plus étudiée est celle des forêts reconnaissables, notée **REC**. Les arbres de dérivation de langages algébriques constituent un exemple de forêts reconnaissables.

Ces forêts sont reconnues par des automates d'arbres. Notons dès maintenant une différence fondamentale avec le cas des mots: il existe des automates, dits ascendants, pour lesquels la reconnaissance des arbres se fait des feuilles vers la racine, et des automates descendants pour lesquels le mouvement se fait dans l'autre sens. Et déjà, une difficulté liée au non-déterminisme dans le cas descendant apparaît: on ne peut pas, en général, réduire le non-déterminisme dans le cas des automates descendants alors qu'on peut le faire dans le cas ascendant.

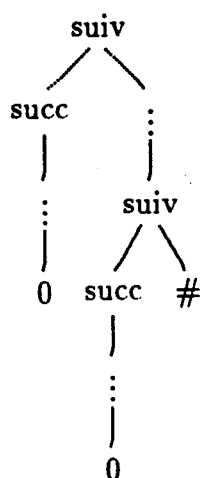
Les automates finis d'arbres sont fortement utilisés dans nos travaux. Rappelons que les automates finis d'arbres sont, exprimés dans le vocabulaire des sortes, des signatures d'algèbres (H. Comon [Com90]). A chaque sorte, correspond un état de l'automate.

L'exemple suivant illustre cette correspondance entre sortes et automates pour définir les listes d'entiers.

Soit la sorte *ListeEntiers* définie par:

$$\begin{aligned} 0 & : \text{Nat} \\ \text{succ} & : \text{Nat} \rightarrow \text{Nat} \\ \# & : \rightarrow \text{ListeEntiers} \\ \text{suiv} & : \text{Nat} \times \text{ListeEntiers} \rightarrow \text{ListeEntiers} \end{aligned}$$

Un terme t de la sorte *ListeEntiers* a pour forme:



A la sorte ListeEntier est associée la forêt reconnue par l'automate $A = \langle \Sigma, Q, F, R \rangle$ avec:

- $\Sigma = \{0, \#, succ, suiv\}$
- $Q = \{q_{Nat}, q_{LE}\}$ et $F = \{q_{LE}\}$
- les règles
 - $0 \rightarrow q_{Nat}(0)$
 - $succ(q_{Nat}(x)) \rightarrow q_{Nat}(succ(x))$
 - $\# \rightarrow q_{LE}(\#)$
 - $suiv(q_{Nat}(x), q_{LE}(y)) \rightarrow q_{LE}(suiv(x, y))$.

Un autre phénomène important, et qui n'existe pas dans les mots, est celui de la non-linéarité. La manipulation de termes non-linéaires se rencontre fréquemment en programmation logique par exemple. Dans le but de pouvoir manipuler des égalités de sous-arbres (c'est un symptôme de la non-linéarité) de nouveaux outils ont été définis. Des extensions des automates "classiques" d'arbres ont ainsi été proposées ces dernières années.

Une première famille d'automates, notée **RATEG**, permettant de manipuler des égalités entre sous-termes a été introduite par M. Dauchet et J. Mongy en 1981 [Mon81]. Ils ont montré que, sans restriction sur les contraintes d'égalité, la propriété du vide est indécidable.

Plus récemment, B. Bogaert et S. Tison ont défini la classe d'automates REC_{\neq} dans laquelle des tests d'égalité ou de non-égalité sont autorisés uniquement entre sous-termes directs d'un même noeud (frères). Cette classe est ainsi une algèbre de Boole close par morphisme alphabétique.

Dans sa thèse, A. C. Caron a défini une classe d'automates dont les contraintes sont de la forme " t est un facteur de x ". Cette classe d' "Automates de Filtrage dans les Arbres", est close par les opérations booléennes et le vide y est décidable [Car93].

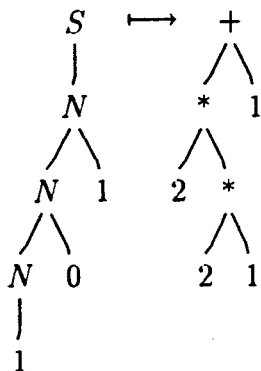
Historiquement les transformations d'arbres sont apparues dès qu'il s'est agi de traduire un langage, qu'il soit naturel ou informatique, en respectant la syntaxe. Des préoccupations linguistiques ont, par exemple, conduit aux grammaires transformationnelles. L'étude de certaines phases de la compilation a conduit à l'introduction des "syntax-directed translators" [AU69a], [AU69b].

Une transduction ascendante peut être vue comme une passe de calculs à attributs hérités; une transduction descendante synthétise des attributs comme l'illustre l'exemple suivant:

Considérons la grammaire suivante:

$$\begin{aligned} S &\rightarrow N & V(S) &= V(N) \\ N &\rightarrow N1 & V(N) &= +(* (2, V(N)), 1) \\ N &\rightarrow N0 & V(N) &= *(2, V(n)) \\ N &\rightarrow 1 \end{aligned}$$

Cette grammaire permet ainsi la transformation de l'arbre de dérivation de l'entier codé 101 en base 2 en l'arbre représentant le codage en base 10 de ce même entier:



Plus précisément, les transducteurs sont d'abord apparus, comme ce fut le cas pour les mots, comme des reconnaisseurs d'états finis avec sortie. Citons en particulier les travaux de W.C. Rounds et J.W. Thatcher ([Rou68], [Tha69], [Tha70]), les précurseurs, et ceux de J. Engelfriet qui a créé des outils permettant une étude systématique ([Eng75], [Eng77], [Eng80]...).

Ils ont été largement étudiés. Parmi les travaux les plus récents, citons ceux de S. Bozapalidis ([Boz92]), J. C. Raoult([Rao93]) et H. Seidl ([Sei 93]).

Les transducteurs modélisent aussi certaines sous-classes de programmes fonctionnels comme l'illustre l'exemple introductif. Le cas non-déterministe étudié ici correspond à certaines classes de programmes logiques.

D'un point de vue algébrique, les transducteurs d'arbres généralisent en fait les transductions rationnelles dans les mots. Les principaux résultats sur les transductions rationnelles peuvent être trouvés dans [Ber79] et [AuB88].

Comme c'est le cas pour les automates d'arbres, on distingue deux classes principales de transducteurs d'arbres:

- les transducteurs descendants ("top-down" ou "root-to-frontier" dans la littérature en langue anglaise) qui transforment les arbres en commençant par la racine pour finir aux feuilles
- et les transducteurs ascendants ("bottom-up" ou "frontier-to-root") pour lesquels, au contraire, le mouvement se fait des feuilles vers la racine.

D'autres distinctions, qui n'existent pas non plus dans les mots, entraînent une multiplication des classes de transducteurs.

On citera les transducteurs

- linéaires qui ne donnent pas en sortie plusieurs transformations d'un même sous-arbre
- et les transducteurs complets qui n'effacent aucune branche, c'est-à-dire qui, pour tout sous-arbre en entrée, donne au moins un sous-arbre en sortie.

Contrairement à ce qui se passe pour les automates, la classe des transducteurs ascendants est différente de celle des transducteurs descendants (J. Engelfriet [Eng75]). Informellement, on peut dire que cela est dû au fait que ces transducteurs réalisent et transforment des copies de sous-arbres dans un ordre différent. Nous illustrons les

différences sur un exemple.

Soit le transducteur ascendant défini par:

$$\begin{array}{ll} \alpha(q(x)) \rightarrow q(\sigma(x, x)) & \\ a(q(x)) \rightarrow q(a(x)) & a(q(x)) \rightarrow q(b(x)) \\ \bar{a} \rightarrow q(\bar{a}) & \bar{\alpha} \rightarrow q(\bar{\alpha}) \end{array}$$

Ainsi, par exemple

$$\begin{array}{ccc} \alpha & \xrightarrow{*} & \alpha \\ | & & | \\ a & & q(a) \\ | & & | \\ a & & b \\ | & & | \\ \bar{a} & & \bar{a} \end{array}$$

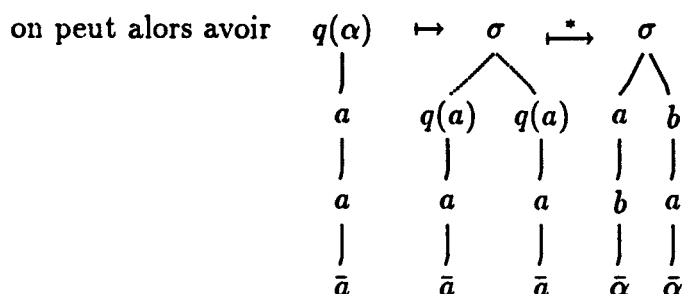
et ensuite la branche transformée est dupliquée:

$$\begin{array}{ccc} \alpha & \mapsto & \sigma \\ | & & / \quad \backslash \\ q(a) & & a \quad a \\ | & & | \quad | \\ b & & b \quad b \\ | & & | \quad | \\ \bar{a} & & \bar{\alpha} \quad \bar{\alpha} \end{array}$$

La situation diffère pour un transducteur descendant qui peut, lui, de par le non-déterminisme, réaliser des copies différentes.

Pour le transducteur défini par

$$\begin{array}{ll} q(\alpha(x)) \rightarrow \sigma(q(x), q(x)) & \\ q(a(x)) \rightarrow a(q(x)) & q(a(x)) \rightarrow b(q(x)) \\ q(\bar{a}) \rightarrow \bar{a} & q(\bar{\alpha}) \rightarrow \bar{\alpha} \end{array}$$



Cette différence disparaît dans le cas des transducteurs linéaires et complets. On remarquera que les transducteurs considérés dans cet exemple sont lettre à lettre.

Par transduction, on désignera l'ensemble des couples d'arbres produits par un transducteur.

Au niveau des propriétés, notons que seuls les transducteurs linéaires conservent la reconnaissabilité et l'algébricité des forêts.

De même, la plupart des classes de transductions ne sont pas stables ni par composition ni par inversion ([Dau75], [Dau77]).

De nouvelles familles de transducteurs ayant de meilleures propriétés ont été définies. Des transducteurs, dits avec "look-ahead", permettent par exemple un contrôle des sous-arbres d'un noeud avant que ceux-ci ne soient transformés. Lorsque ce contrôle est reconnaissable, c'est à dire calculable par un automate fini d'arbres, on obtient des familles closes par composition. De nombreuses classes de transducteurs avec look-ahead ont été étudiées. Citons, entre autres, les travaux de J. Engelfriet et de S. Vágvölgyi ([Eng77], [Fv89], [Fv90], [Vág92]).

Peu à peu, les auteurs ont été amenés à décomposer les transductions en transformations simples: morphisme, morphisme inverse et intersection avec une forêt reconnaissable. Le point de vue des bimorphismes est ainsi apparu.

Un bimorphisme est un triplet $B = (\phi, R, \psi)$ où ϕ et ψ sont des morphismes et R est une forêt reconnaissable.

La transformation associée à un bimorphisme est définie par $\hat{B} = \{(\phi(t), \psi(t)) \mid t \in R\}$.

En théorie des langages, M. Nivat a montré l'équivalence entre les transductions rationnelles, considérées comme transformations réalisées par des machines d'états finis, et les bimorphismes [Niv68].

Dans [ArD82], A. Arnold et M. Dauchet montrent que ce résultat ne se transpose pas si on reste dans le cas classique d'étude des arbres. Dans une structure algébrique où les termes manipulés sont des n -uplets d'arbres (les magmoïdes [ArD79a],[ArD79b]) des classes de transductions aux propriétés comparables à celles des transductions de langages sont obtenues.

Récemment, S. Bozapalidis a montré que la classe *Alph* des transductions alphabétiques, c'est-à-dire caractérisables par un bimorphisme (ϕ, R, ψ) où ϕ et ψ sont des homomorphismes alphabétiques linéaires, est stable par inversion et composition [Boz92].

Notre travail

Dans cette étude, nous nous intéressons au problème de l'équivalence pour la classe des transducteurs d'états finis lettre à lettre non-déterministes d'arbres. Les règles de tels transducteurs sont symétriques en ce sens que l'arbre figurant dans le membre droit des règles est réduit à une lettre, comme c'est le cas pour le membre gauche.

Cette symétrie dans la forme des règles nous permet de montrer facilement que si ces transducteurs sont de plus linéaires et complets alors ils sont inversibles. Le transducteur réalisant la relation inverse s'obtient par "renversement des flèches".

Rappelons d'abord que deux transducteurs sont équivalents si les transformations associées sont égales. Les résultats de décidabilité de l'équivalence sont peu nombreux. Rappelons que l'équivalence des transducteurs déterministes est décidable (Z. Esik [Esi80], et Z. Zachar [Zac78]). L'équivalence est en général indécidable pour les transducteurs non-déterministes. Citons les seuls résultats positifs pour ces transducteurs non-déterministes. Il est connu qu'on peut décider si un transducteur est fonctionnel (J. Engelfriet [Eng80]) et que ce problème est décidable en temps polynomial (H. Seidl [Sei92]). En corollaire, on obtient la décidabilité de l'équivalence des transducteurs fonctionnels. Plus récemment, H. Seidl a établi la décidabilité de l'équivalence pour les transducteurs ascendants finiment-valués [Sei90].

La première partie, correspondant à la publication [AnD93], présente le résultat de décidabilité de l'équivalence pour les transducteurs linéaires et complets, qu'ils soient descendants ou ascendants.

La preuve est faite d'abord pour les transducteurs descendants. L'idée qui nous a guidés est d'utiliser les automates d'arbres comme "outil de décision". L'idée n'est

pas nouvelle puisque, dans les années 60, les automates sur les mots et les arbres infinis avaient été introduits comme outil de décision pour des problèmes de logique (Büchi, Doner, Mac Naughton, Rabin ...). En particulier, Rabin développa la théorie des automates infinis pour résoudre le problème de la décidabilité de la théorie monadique du second ordre [Rab69].

Dans une première phase, nous codons facilement les transducteurs sans-torsion (encore appelés marqueurs), c'est-à-dire linéaires, complets et n'autorisant aucune permutation de sous-arbres, en des automates d'arbres. Ces transducteurs préservant le squelette des arbres, il nous suffit de les superposer. Le problème de l'équivalence pour les marqueurs peut ainsi se réduire à l'équivalence des automates d'arbres.

Dans la seconde phase, nous considérons des transducteurs où les permutations de sous-arbres (que nous appelons torsions) sont possibles. Nous montrons que deux transducteurs équivalents ne peuvent appliquer des torsions différentes que pour des arbres et sous-arbres dont la profondeur est bornée. En traitant globalement ces arbres et sous-arbres, nous rendons les torsions non-significatives. Ces transducteurs peuvent ainsi être codés en des marqueurs et nous utilisons alors le résultat précédent.

Ce résultat de décidabilité obtenu pour les transducteurs descendants se transpose sans difficulté au cas des transducteurs ascendants.

La seconde partie, correspondant à [AB93a], constitue une extension des résultats de la première partie au cas de transducteurs descendants qui restent linéaires mais ne sont plus complets.

On montre cette fois que deux transducteurs équivalents ne peuvent appliquer des torsions différentes que pour des arbres et sous-arbres dont l'image est de profondeur bornée. Par une démarche analogue à celle utilisée pour les transducteurs linéaires et complets, mais en ajoutant une phase de "complétion" de nos transducteurs, nous nous ramenons également au cas des marqueurs.

Le problème devient beaucoup plus complexe lorsque les transducteurs descendants sont non-linéaires, même s'ils restent complets. Les transducteurs non-linéaires ne préservant pas la reconnaissabilité, il semble naturel que le codage en des marqueurs, c'est-à-dire en fait en des forêts reconnaissables, soit impossible. Les transducteurs descendants peuvent dupliquer des sous-arbres et ensuite des transformations différentes peuvent leur être appliquées.

Ainsi, si on veut avoir un espoir d'étendre la technique de codage utilisée précédemment à nos transducteurs non-linéaires, outre le fait qu'il nous faudra obtenir des couples d'arbres de même squelette, il faudra prendre en compte le fait qu'un même

sous-arbre t peut être transformé en des sous-arbres différents u, u', u'', \dots

Pour ce faire, nous introduisons de nouvelles classes d'automates. Ces automates, décrits dans la troisième partie de ce mémoire et dans [AB93b], peuvent tester l'équivalence ou la non-équivalence de sous-termes directs. Ces automates constituent une extension des automates à tests d'égalité définis en 1990 par B. Bogaert et S. Tison ([Bog90], [BT92]). A toute relation d'équivalence \equiv correspond en fait une classe d'automates notée REC_{\neq} . Ces classes sont stables par les opérations booléennes.

Dans le cas particulier où l'équivalence est induite par un système de réécriture lettre à lettre, noethérien et confluent (deux arbres équivalents ont la même forme irréductible) la propriété du vide est décidable. On obtient ainsi de nouvelles classes d'automates pour lesquelles l'équivalence est décidable.

On montre ensuite, dans la quatrième partie (publication [AB93c]), que le problème de l'équivalence pour nos transducteurs descendants non-linéaires peut se ramener à l'équivalence d'automates à tests d'équivalence.

En répercutant dans les arbres en entrée les duplications qui sont effectuées lors des transformations nous obtenons des arbres de même squelette que nous pouvons alors superposer. Un système de réécriture bien choisi nous permet de considérer comme équivalentes des superpositions provenant d'un même sous-arbre en entrée. Nous obtenons ainsi la décidabilité de l'équivalence pour les transducteurs descendants complets.

Malheureusement, les techniques utilisées ne permettent pas de conclure dans le cas général des transducteurs descendants non-complets et non-linéaires. Un exemple en illustre la raison.

Dans la dernière partie, une extension au cas ascendant des techniques décrites dans les chapitres précédents nous permet de montrer que l'équivalence est décidable pour les transducteurs ascendants lettre à lettre.

Pour conclure ce mémoire, nous décrivons les développements que nous souhaitons donner à notre travail.

Chapitre 1

Préliminaires

Ce chapitre contient les définitions et principales propriétés des outils manipulés dans cette thèse.

La plupart des définitions et notations proviennent des articles de J. Engelfriet [Eng75] pour les arbres et transformations d'arbres et de G. Huet [Hue80] pour les systèmes de réécriture.

1.1 Arbres

Un *alphabet gradué* est un couple (Σ, ρ) où Σ est un alphabet fini et ρ est une application de Σ dans \mathbb{N} . Usuellement, on le notera simplement Σ .

Pour toute lettre σ de Σ , $\rho(\sigma)$ est appelé l'*arité* ou *rang* de σ .

Le sous-ensemble Σ_m de Σ est l'ensemble des lettres de rang m .

Pour tout $p \geq 1$, on désigne par X_p l'ensemble $\{x_1, \dots, x_p\}$ de variables. X_0 est l'ensemble vide.

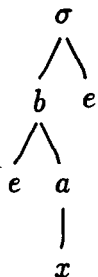
Etant donné un alphabet gradué Σ et un ensemble X_p de variables, l'ensemble de tous les *termes* construits sur l'alphabet Σ et indexé par X_p , noté $T_\Sigma(X_p)$, est défini récursivement par:

- $X_p \subseteq T_\Sigma(X_p)$ et
- si $\sigma \in \Sigma_n$ et $t_1, \dots, t_n \in T_\Sigma(X_p)$ alors $\sigma(t_1, \dots, t_n) \in T_\Sigma(X_p)$.

L'ensemble $T_\Sigma(X_0)$ des *termes clos* sera noté T_Σ .

Les termes ainsi construits seront vus comme des arbres.

Par exemple, le terme $t = \sigma(b(e, a(x)), e)$ peut être représenté par l'arbre



σ en est la racine et e , f et x en sont les feuilles.

Un sous-arbre d'un arbre t est dit *propre* s'il est différent de t .

On désigne par $\mathcal{V}(t)$ l'ensemble des variables apparaissant dans t .

Un terme t est *linéaire* si chaque variable apparaît au plus une fois dans t .

La *profondeur* d'un arbre $t \in T_{\Sigma}(X_p)$, notée $\pi(t)$, est définie par $\pi(t) = 0$ si $t \in \Sigma_0$ ou $t \in X_p$ et $\pi(t) = 1 + \max\{\pi(t_1), \dots, \pi(t_n)\}$ si $t = \sigma(t_1, \dots, t_n)$.

La profondeur de l'arbre $t = \sigma(b(e, a(x)), e)$, par exemple, est ainsi $\pi(t) = 3$.

Pour tout $p \in \mathbb{N}$, $[p]$ désigne l'ensemble $\{1, \dots, p\}$.

Une torsion θ de $[p]$ dans $[q]$ est une application de $[p]$ dans $[q]$. On la notera $\langle q; \theta(1), \dots, \theta(p) \rangle$. Par $id_{[n]}$ on désigne l'identité sur $[n]$.

1.2 Systèmes de réécriture

Une *règle de réécriture* est un couple (l, r) de termes tels que $\mathcal{V}(r) \subseteq \mathcal{V}(l)$. On la notera $l \rightarrow r$.

l est le *membre gauche* de la règle et r en est le *membre droit*.

Une règle $l \rightarrow r$ est *complète* si $\mathcal{V}(r) = \mathcal{V}(l)$.

Une règle $l \rightarrow r$ est *linéaire à gauche* (respectivement *linéaire à droite*) si l (respectivement r) est un terme linéaire.

Un ensemble \mathcal{S} de règles de réécriture est appelé un *système de réécriture*.

On écrit $t \rightarrow_{\mathcal{S}} t'$ (ou $t \rightarrow t'$ s'il n'y a pas d'ambiguïté quant au système de réécriture considéré) si t est réécrit en t' en utilisant une règle de \mathcal{S} .

$\overset{*}{\rightarrow}$ désigne la clôture réflexive et transitive \rightarrow .

Un système de réécriture est *lettre à lettre* si et seulement si pour chaque règle l'arbre apparaissant dans le membre droit est, comme dans le membre gauche, réduit à une lettre.

Un système de réécriture est *noethérien* si et seulement si il n'y a pas de séquence infinie de réécritures: $t_0 \rightarrow_{\mathcal{S}} t_1 \rightarrow_{\mathcal{S}} \dots$

Un système de réécriture est *confluent* si et seulement si

$$\forall x, \forall y, \forall z \in T_{\Sigma}(X), (z \overset{*}{\rightarrow} x \text{ et } z \overset{*}{\rightarrow} y) \Rightarrow \exists t \in T_{\Sigma}(X) (x \overset{*}{\rightarrow} t \text{ et } y \overset{*}{\rightarrow} t).$$

Soit \mathcal{S} un système de réécriture. Un terme clos t est *irréductible* si et seulement si il n'existe aucun terme t' tel que t puisse être réécrit en t' en utilisant une règle de \mathcal{S} .

Plus particulièrement, un terme non clos $\sigma(x)$ de profondeur 1 (respectivement $\sigma(x_{\theta(1)}, x_{\theta(2)})$) est *irréductible* si et seulement si il n'existe aucune règle de partie gauche $\sigma(x)$ (respectivement $\sigma(x_{\theta(1)}, x_{\theta(2)})$).

Soit \mathcal{S} un système de réécriture. Pour tout terme t , par \hat{t} on désigne une quelconque forme irréductible de t .

Si \mathcal{S} est confluent, alors la forme irréductible (ou forme normale) \hat{t} de tout terme t , si elle existe, est unique. De plus, si \mathcal{S} est noethérien, on peut obtenir \hat{t} à partir de t quelle que soit la séquence de réécritures utilisant des règles de \mathcal{S} ([Hue80]).

On désigne par $IRR(E)$ l'ensemble des formes irréductibles des termes de E .

Un système de réécriture est *non-dégénéré* si et seulement si il existe au moins un couple d'arbres (t_1, t_2) tels que $\hat{t}_1 \neq \hat{t}_2$. Si \mathcal{S} est un système lettre à lettre dégénéré alors il existe une lettre unique α de rang 0 telle que toute lettre σ est réécrite en α .

1.3 Automates d'arbres

Les principaux résultats concernant les automates d'arbres peuvent être trouvés dans le livre de F. Gecseg et M. Steinby [GS84]. On ne rappelle que ceux qui sont utilisés dans ce mémoire.

Un *automate ascendant* d'arbres est un quadruplet $A = \langle \Sigma, Q, F, R \rangle$ où

- Σ est un alphabet gradué
- Q est un ensemble fini d'états
- F est un sous-ensemble de Q composé des états finaux
- R est un ensemble fini de règles de la forme

$$\sigma(q_1(x_1), \dots, q_n(x_n)) \rightarrow q(\sigma(x_1, \dots, x_n)), \text{ avec } q, q_1, \dots, q_n \in Q, \sigma \in \Sigma_n.$$

Le langage reconnu (ou *forêt*) par un automate ascendant d'arbres A est défini par $\mathcal{F}(A) = \{t \in T(\Sigma) \mid t \xrightarrow{*}_A q(t), q \in F\}$.

Un *automate descendant* d'arbres se définit de façon analogue mais F est l'ensemble des états initiaux et les règles de R sont de la forme $q(\sigma(x_1, \dots, x_n)) \rightarrow \sigma(q_1(x_1), \dots, q_n(x_n))$.

Les forêts reconnues par les automates ascendants d'arbres sont exactement celles reconnues par les automates descendants. On les appelle *forêts reconnaissables* et on note **REC** la classe ainsi définie.

Théorème 1.3.1 *La classe **REC** est close par union, intersection et complémentaire.*

Théorème 1.3.2 *Soit \mathcal{F} une forêt reconnaissable. La propriété " $\mathcal{F} = \emptyset$ " est décidable (en temps polynômial).*

1.4 Transducteurs d'arbres

Il existe de nombreux types de transducteurs mais on peut distinguer deux classes principales: les transducteurs descendants pour lesquels le mouvement se fait de la racine vers les feuilles et les transducteurs ascendants pour lesquels, au contraire, le mouvement commence aux feuilles pour se terminer à la racine.

Un *transducteur d'états finis descendant d'arbres* est un quintuplet $T = \langle \Sigma, \Delta, Q, I, R \rangle$ où

- Σ et Δ sont les alphabets gradués des symboles d'entrée et de sortie
- Q est un ensemble fini d'états
- I est un sous-ensemble de Q composé des états initiaux
- R est un ensemble fini de règles
 $R \subseteq Q(T_\Sigma[X]) \times T_\Delta[Q(X)]$ où $Q(T_\Sigma[X]) = \{q(t) \mid q \in Q, t \in T_\Sigma[X]\}$ et
 $Q(X) = \{q(x) \mid q \in Q, x \in X\}$.

Les règles des transducteurs descendants que nous considérons sont de la forme:

$$q(\sigma(x_1, \dots, x_n)) \rightarrow \tau(q_1(x_{\theta(1)}), \dots, q_p(x_{\theta(p)}))$$

avec $\sigma \in \Sigma_n$, $\tau \in T_\Delta(X_p)$,
 q, q_1, \dots, q_p états de Q ,
et θ application de $[p]$ dans $[n]$.

Si $n = 0$ on a alors une règle de la forme $q(\sigma) \rightarrow \tau$.

Un transducteur est *lettre à lettre* si pour toute règle l'arbre τ figurant dans le membre droit est réduit à une lettre de Δ .

Un transducteur est *linéaire* si pour toute règle la torsion θ figurant dans le membre droit est injective.

Un transducteur est *complet* si pour toute règle la torsion θ figurant dans le membre droit est surjective.

Un transducteur est *sans-torsion* si pour toute règle la torsion θ figurant dans le membre droit est l'identité.

Les règles de R sont en fait des règles de réécriture.

$t \mapsto t'$ si et seulement si

il existe $t_0 \in T_\Sigma(X_1)$, $\sigma \in \Sigma_n$, $t_1, \dots, t_n \in T_\Sigma$, $\delta \in \Delta_m$,

il existe $q, q_1, \dots, q_m \in Q$,

il existe une règle $q(\sigma(x_1, \dots, x_n)) \rightarrow \delta(q_1(x_{\theta(1)}), \dots, q_m(x_{\theta(m)}))$ dans R

et $t = t_0(q(\sigma(t_1, \dots, t_n)))$, $t' = t_0(\delta(q_1(t_{\theta(1)}), \dots, q_m(t_{\theta(m)})))$.

\mapsto^* désigne la clôture réflexive et transitive de \mapsto .

Pour tout état q de Q , \hat{T}_q désigne la transformation réalisée à partir de l'état q . Formellement, $\hat{T}_q = \{(t, u) \in T_\Sigma \times T_\Delta / q(t) \mapsto^* u\}$.

Cette notation s'étend à tout ensemble d'états E : $\hat{T}_E = \cup_{q \in E} \hat{T}_q$.

\hat{T} désigne la transformation associée à T (ou *transduction*): $\hat{T} = \cup_{q \in I} \hat{T}_q$.

Le *domaine* d'une transduction \hat{T} , noté $dom(\hat{T})$, est l'ensemble $\{t \in T_\Sigma / \exists u \in T_\Delta, (t, u) \in \hat{T}\}$.

Le *codomaine* d'une transduction \hat{T} , noté $im(\hat{T})$, est l'ensemble $\{u \in T_\Delta / \exists t \in T_\Sigma, (t, u) \in \hat{T}\}$.

Une transformation \hat{T} est d'*image finie* si et seulement si $im(\hat{T})$ est fini.

Cette définition est étendue à tout sous-ensemble de \hat{T} :

Une partie E de \hat{T} est d'*image finie* si et seulement si $\{u \mid \exists t \text{ tel que } (t, u) \in E\}$ est fini.

Un état d'un transducteur descendant est dit *infinitaire* (respectivement *finitaire*) si et seulement si une infinité (respectivement un nombre fini) d'arbres sont transformés à partir de cet état.

Un transducteur dont tous les états sont infinitaires (respectivement finitaires) est dit *infinitaire* (respectivement *finitaire*).

Un transducteur descendant est *déterministe* si et seulement si l'ensemble des états initiaux est un singleton et il n'existe pas deux règles de même partie gauche.

Un *transducteur ascendant* se définit de façon analogue mais cette fois I est l'ensemble des états finaux et $R \subseteq T_\Sigma[Q(X)] \times Q(T_\Delta[X])$.

Rappelons que les transducteurs ascendants et descendants sont incomparables en général mais que cette différence disparaît lorsqu'ils sont linéaires et complets.

Théorème 1.4.1 [Eng75] *La classe des transducteurs ascendants linéaires et complets est égale à celle des transducteurs descendants linéaires et complets.*

Nous avons, pour les transducteurs lettre à lettre, la propriété fondamentale suivante:

Propriété 1.4.1 Soit T un transducteur lettre à lettre. Pour tout couple d'arbres (t, u) de la transformation associée \hat{T} , $\pi(t) = \pi(u)$ si T est complet et $\pi(t) \geq \pi(u)$ dans le cas contraire.

PREUVE:

Elle se fait par récurrence sur la profondeur des arbres. \square

A ces notions classiques, nous ajoutons les définitions et notation suivantes:

Définition 1.4.1 Deux ensembles d'états $\{q_1, \dots, q_n\}$ et $\{k_1, \dots, k_m\}$ d'un transducteur T sont globalement équivalents si et seulement si $\bigcup_{i \in [n]} (\hat{T}_{q_i}) = \bigcup_{j \in [m]} (\hat{T}_{k_j})$.

Définition 1.4.2 Soit \mathcal{T} un mouvement de transformation de $t = \sigma(t_1, \dots, t_n)$ à partir de l'état q :

$$\mathcal{T} : q(\sigma(t_1, \dots, t_n)) \mapsto \delta(q_1(t_{\theta(1)}), \dots, q_m(t_{\theta(m)})) \xrightarrow{*} \delta(u_1, \dots, u_m).$$

On appelle transformation initiale sur t à partir de l'état q le triplet (σ, δ, θ) .

Notation

$\hat{T}_{q(\sigma, \delta, \theta)}$ désigne la transformation réalisée à partir de l'état q en appliquant la transformation initiale (σ, δ, θ) .

1.5 Problèmes de décision

Un problème P , fonction des données d_1, \dots, d_n , est dit *décidable* s'il existe un programme déterminant si, pour tout jeu de données d_1, \dots, d_n , $P(d_1, \dots, d_n)$ est vraie ou fausse.

Dans ce mémoire, nous contribuons à préciser la frontière entre le décidable et le non-décidable pour des classes de transducteurs non-déterministes d'arbres.

Définition 1.5.1 Deux transducteurs T et T' sont équivalents si et seulement si les transformations associées \hat{T} et \hat{T}' sont égales.

Pour montrer qu'on peut décider de l'équivalence pour une classe donnée de transducteurs, nous réduisons le problème considéré à un problème pour lequel la décidabilité de l'équivalence a déjà été établie. En fait dans cette thèse, nous nous

ramenons à l'égalité d'automates d'arbres, qui est un problème décidable.

Les résultats de décidabilité de l'équivalence pour les transducteurs sont relativement peu nombreux:

1. L'équivalence est en général indécidable dans le cas non déterministe (conséquence du résultat obtenu dans les mots par T. Griffiths [Gri68])
2. En 1979, Z. Esik a montré que le caractère fonctionnel d'un transducteur, ascendant ou descendant, est décidable. L'algorithme proposé peut être utilisé pour décider de l'équivalence de deux transducteurs. Les transducteurs déterministes étant fonctionnels, il établit ainsi la décidabilité de l'équivalence dans le cas déterministe ascendant et descendant [Esi80].
3. Par d'autres méthodes, le même résultat a été obtenu pour les transducteurs déterministes ascendants par Z. Zachar en 1978 [Zac78] et par B. Courcelle et P. Franchi-Zannettachi en 1982, pour les transducteurs descendants [CF82].
4. Plus récemment, en 1990, H. Seidl a établi la décidabilité de l'équivalence pour les transducteurs ascendants finiment-valués [Sei90], [Sei92].

Remarques

- Dans cette thèse, tous les transducteurs considérés seront lettre à lettre. Le terme "lettre à lettre" sera donc généralement omis pour les transducteurs.
- Les résultats et preuves sont généralement énoncés pour des lettres d'arité 2. Ils se généralisent naturellement au cas de lettres d'arité quelconque mais au prix, entre autres, d'une plus grande complexité des notations et d'un accroissement du nombre de cas à étudier. De plus l'énoncé de certains lemmes peut être sensiblement différent comme nous l'illustrons dans le paragraphe 2.2.6 pour la généralisation du lemme 2.2.2 qui est fondamental dans notre démarche.

Chapitre 2

Transducteurs linéaires et complets

Nous considérons dans ce chapitre la classe des transducteurs lettre à lettre linéaires et complets. Ces transducteurs changent seulement le label des noeuds des arbres qu'ils transforment et, pour chaque noeud, peuvent effectuer une permutation des sous-arbres.

Nous établissons d'abord (paragraphe 1) la décidabilité de l'équivalence pour la sous-classe des transducteurs, appelés sans-torsion, pour lesquels aucune permutation de sous-arbres n'est effectuée. Nous montrons ensuite (paragraphe 2) que le problème de l'équivalence pour les transducteurs linéaires et complets peut se réduire au problème de l'équivalence pour les transducteurs sans-torsion.

2.1 Transducteurs sans-torsion

Les transducteurs descendants lettre à lettre sans-torsion sont étudiés dans ce paragraphe. Ces transducteurs, encore appelés *marqueurs* dans [Dau75] (*relabelings* dans [Eng75]), ne peuvent en fait modifier que le label des noeuds des arbres reconnus.

Les règles sont de la forme:

$$q(\sigma(x_1, \dots, x_n)) \rightarrow \delta(q_1(x_1), \dots, q_n(x_n))$$

ou de la forme:

$$q(\sigma) \rightarrow \delta.$$

Classe reconnue

On note **T-LAB** la classe des transformations réalisées par les marqueurs descendants.

Les transformations associés à cette classe correspondent en fait aux relations définies par Takahashi. En utilisant le point de vue des bimorphismes, S. Bozapalidis montre que l'égalité de telles relations est décidable [Boz92]. Le résultat obtenu dans ce premier paragraphe n'est donc pas nouveau.

2.1.1 Décidabilité de l'équivalence

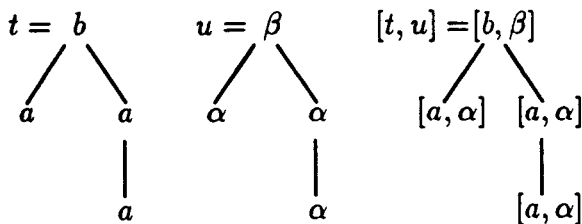
Pour établir la décidabilité de l'équivalence dans cette classe de transducteurs, nous utilisons les bonnes propriétés des forêts reconnaissables en codant nos transducteurs dans des automates. L'idée de considérer les automates comme "outil de décision" n'est pas nouvelle. Dès la fin des années 60, cette technique avait été introduite par Doner [Don70] et utilisée dans le théorème de Rabin dans le cas des arbres infinis [Rab69]. Cette construction a été choisie par C. Frougny et J. Sakarovitch [FS90] dans leur étude des relations rationnelles à délai borné et, dans le cas des arbres, elle fut utilisée par M. Dauchet et S. Tison pour établir la décidabilité de la théorie de la réécriture close [DT90], [Tis90].

L'idée est d'associer à tout n -uple d'arbres un arbre dont les noeuds sont des n -uples. Ce codage repose ici sur la propriété suivante:

Propriété 2.1.1 *Pour tout marqueur T , pour tout couple d'arbres (t, u) de la transformation associée \hat{T} , les arbres t et u ont le même squelette.*

Par conséquent, une simple superposition nous permet de coder facilement tout couple d'arbres (t, u) en un arbre qu'on notera $[t, u]$.

Exemple



A tout marqueur $T = \langle \Sigma, \Delta, Q, I, R \rangle$

nous associons l'automate $A_T = \langle \Gamma, Q, I, R' \rangle$ défini par:

- $\Gamma = \Sigma \times \Delta$

- R' tel que $q([\sigma, \delta](x_1, \dots, x_n)) \rightarrow [\sigma, \delta](q_1(x_1), \dots, q_n(x_n)) \in R'$ si et seulement si $q(\sigma(x_1, \dots, x_n)) \rightarrow \delta(q_1(x_1), \dots, q_n(x_n)) \in R$.

On obtient de façon évidente:

Propriété 2.1.2 $[t, u] \in \mathcal{F}(A_T)$ si et seulement si $(t, u) \in \hat{T}$.

Il nous est ainsi possible d'associer une forêt reconnaissable à la transformation d'arbres considérée et donc d'hériter des bonnes propriétés de clôture et de décidabilité de la classe **REC**.

Propriétés de clôture de **T-LAB**

Clôture de **T-LAB** par union.

Soient T_1 et T_2 deux marqueurs. Nous associons à T_1 et T_2 les automates A_1 et A_2 (comme ils ont été définis précédemment). Ainsi

$$\begin{aligned} \hat{T}_1 \cup \hat{T}_2 &= \{(t, u) / (t, u) \in \hat{T}_1 \text{ ou } (t, u) \in \hat{T}_2\} \\ &= \{(t, u) / [t, u] \in F(A_1) \text{ ou } [t, u] \in F(A_2)\} \\ &= \{(t, u) / [t, u] \in F(A_1) \cup F(A_2)\}. \end{aligned}$$

La classe **REC** étant close par union, il existe donc un automate A tel que $\mathcal{F}(A) = \mathcal{F}(A_1) \cup \mathcal{F}(A_2)$.

Donc, $\hat{T}_1 \cup \hat{T}_2 = \{(t, u) / [t, u] \in \mathcal{F}(A)\}$. Soit T le marqueur associé à A . On obtient alors $\hat{T}_1 \cup \hat{T}_2 = \{(t, u) \in \hat{T}\} = \hat{T}$ et par conséquent **T-LAB** est stable par union. \square

De la même façon, on montre que **T-LAB** est close par intersection et différence, et que le vide est décidable.

On obtient ainsi

Théorème 2.1.1 *L'équivalence des transducteurs descendants lettre à lettre sans-torsion est décidable.*

PREUVE:

On utilise le fait que T_1 et T_2 sont équivalents si et seulement si $(\hat{T}_1 - \hat{T}_2) \cup (\hat{T}_2 - \hat{T}_1) = \emptyset$. \square

Notons que l'équivalence est ici décidable en temps exponentiel puisque l'équivalence des automates d'arbres l'est avec cette complexité.

Remarque: propriétés de non clôture

- **T-LAB** n'est pas close par complémentaire.

Exemple

Considérons $\hat{T} = \{(b(a^n(\bar{a}), \bar{a}), \beta(\alpha^n(\bar{\alpha}), \bar{\alpha})), n \in \mathbb{N}\}$.

$(t, u) = (b(a(\bar{a}), \bar{a}), \beta(\bar{\alpha}, \alpha(\bar{\alpha}))) \in T_\Sigma \times T_\Delta - \hat{T}$ mais n'est pas réalisable par un transducteur sans-torsion.

- De la même façon, on perd, dans le cas de transducteurs complets mais non linéaires, la clôture par intersection.

Exemple

Soient $\hat{T}_1 = \{(b(a^n(\bar{a}), a^m(\bar{a})), c(a_1^n(\bar{a}), a_2^n(\bar{a}), a_3^n(\bar{a}))), n \in \mathbb{N}, m \in \mathbb{N}\}$ et

$\hat{T}_2 = \{(b(a^n(\bar{a}), a^m(\bar{a})), c(a_1^n(\bar{a}), a_2^m(\bar{a}), a_3^m(\bar{a}))), n \in \mathbb{N}, m \in \mathbb{N}\}$.

On a alors $\hat{T}_1 \cap \hat{T}_2 = \{(b(a^n(\bar{a}), a^n(\bar{a})), c(a_1^n(\bar{a}), a_2^n(\bar{a}), a_3^n(\bar{a}))), n \in \mathbb{N}\}$ qui n'est pas réalisable par un transducteur parce que son domaine n'est pas reconnaissable [GS84].

- Comme dans le cas non linéaire, nous perdons, pour les transducteurs linéaires complets avec torsions, la clôture par intersection.

Exemple

Soient $\hat{T}_1 = \{(b(a^n(\bar{a}), a^m(\bar{a})), b(a_1^n(\bar{a}), a_2^m(\bar{a}))), n \in \mathbb{N}, m \in \mathbb{N}\}$ et

$\hat{T}_2 = \{(b(a^n(\bar{a}), a^m(\bar{a})), b(a_1^m(\bar{a}), a_2^n(\bar{a}))), n \in \mathbb{N}, m \in \mathbb{N}\}$.

On obtient $\hat{T}_1 \cap \hat{T}_2 = \{(b(a^n(\bar{a}), a^n(\bar{a})), b(a_1^n(\bar{a}), a_2^n(\bar{a}))), n \in \mathbb{N}\}$ qui n'est pas réalisable par un transducteur.

2.1.2 Algorithme de décision

Nous proposons maintenant un algorithme de décision de l'équivalence, pour les marqueurs descendants, basé sur une preuve de la décidabilité de l'équivalence obtenue pour les transducteurs ascendants déterministes.

La propriété de factorisation (T_1) de J. Engelfriet [Eng80] permet de conclure que pour résoudre le problème de l'équivalence pour deux transducteurs ascendants déterministes B_1 et B_2 , il suffit de tester l'ensemble des arbres de profondeur au plus $4 \times N_1 N_2$, où N_i désigne le nombre d'états de B_i .

Soient T_1 et T_2 deux marqueurs descendants ayant respectivement N_1 et N_2 états. Notre algorithme comprend les étapes suivantes:

1. Construction des automates descendants A_1^t et A_2^t , comme cela a été décrit au début de ce paragraphe.

Ces automates ont également respectivement N_1 et N_2 états.

2. Construction des automates ascendants A_1^b et A_2^b équivalents respectivement aux automates A_1^t et A_2^t . Ces automates étant obtenus par simple "renversement des flèches", les nombres d'états sont inchangés.
3. Construction des automates ascendants déterministes $A_1^{b,d}$ et $A_2^{b,d}$ équivalents à A_1^b et A_2^b . Ces automates ont au plus respectivement 2^{N_1} et 2^{N_2} états.
4. On associe, à ces automates déterministes, les transducteurs ascendants "identité" B_1 et B_2 : un couple $(t, t) \in \hat{B}_i$ si et seulement si $t \in \mathcal{F}(A_i^{b,d})$. Ces transducteurs sont évidemment déterministes. Les nombres d'états sont inchangés, à savoir respectivement 2^{N_1} et 2^{N_2} .

Il est évident que T_1 et T_2 sont équivalents si et seulement si B_1 et B_2 le sont.

Il nous suffira donc de tester l'ensemble de tous les arbres de profondeur au plus $4 \times 2^{N_1} 2^{N_2} = 2^{N_1+N_2+2}$.

2.2 Transducteurs linéaires et complets

2.2.1 Préliminaires

Nous considérons d'abord le problème de la décidabilité de l'équivalence pour les transducteurs linéaires et complets dans le cas descendant (classe **LCT-LL**). Nous montrons que ce problème peut se réduire au problème de l'équivalence pour les marqueurs.

Le principal problème rencontré, et illustré dans l'exemple suivant, est que, même si T et T' sont des transducteurs équivalents, pour certains couples d'arbres les transformations ne peuvent pas être réalisées avec les mêmes torsions dans T et dans T' .

Exemple

Considérons les transducteurs T et T' définis par:

$$\mathbf{T} : \begin{array}{ll} q(\sigma(x, y)) \rightarrow \delta(q_1(x), q_2(y)) & \\ q_1(a(x)) \rightarrow a(q_1(x)) & q_1(\bar{a}) \rightarrow \bar{a} \\ q_2(\alpha(x)) \rightarrow \alpha(q_2(x)) & q_2(\bar{\alpha}) \rightarrow \bar{\alpha} \end{array}$$

$$\mathbf{T}' : \begin{array}{ll} k(\sigma(x, y)) \rightarrow \delta(k'_1(x), k_2(y)) & k(\sigma(x, y)) \rightarrow \delta(k_1(x), k'_2(y)) \\ k(\sigma(x, y)) \rightarrow \delta(k_3(y), k_4(x)) & \\ k_1(a(x)) \rightarrow a(k_1(x)) & k_1(\bar{a}) \rightarrow \bar{a} \\ k_2(\alpha(x)) \rightarrow \alpha(k_2(x)) & k_2(\bar{\alpha}) \rightarrow \bar{\alpha} \\ k'_1(a(x)) \rightarrow a(k_{11}(x)) & k'_2(\alpha(x)) \rightarrow \alpha(k_{22}(x)) \\ k_{11}(a(x)) \rightarrow a(k_{11}(x)) & k_{11}(\bar{a}) \rightarrow \bar{a} \\ k_{22}(\alpha(x)) \rightarrow \alpha(k_{22}(x)) & k_{22}(\bar{\alpha}) \rightarrow \bar{\alpha} \\ k_3(\bar{\alpha}) \rightarrow \bar{a} & k_4(\bar{a}) \rightarrow \bar{\alpha} \end{array}$$

T et T' sont des transducteurs équivalents puisqu'ils réalisent les mêmes transformations:

$$\hat{T} = \hat{T}' = \{(\sigma(a^n(\bar{a}), \alpha^m(\bar{\alpha})), \delta(a^n(\bar{a}), \alpha^m(\bar{\alpha}))), n, m \in \mathbb{N}\}.$$

Mais, pour le couple d'arbres $(\sigma(\bar{a}, \bar{\alpha}), \delta(\bar{a}, \bar{\alpha}))$ des torsions différentes sont nécessairement appliquées dans le premier pas de transformation:

pour T , la règle utilisée initialement est

$$q(\sigma(x, y)) \rightarrow \delta(q_1(x), q_2(y))$$

alors que pour T' c'est

$$k(\sigma(x, y)) \rightarrow \delta(k_3(y), k_4(x))$$

qui est appliquée.

Le point essentiel de notre preuve consiste à montrer que ce phénomène est de profondeur bornée, c'est à dire en fait que pour des transducteurs équivalents les mêmes torsions peuvent être appliquées, excepté éventuellement pour un nombre fini d'arbres (lemmes 2.2.1, 2.2.2).

Ensuite, pour tout entier Λ nous construisons la forme Λ -normalisée T_{Θ}^{Λ} d'un transducteur T de telle façon que:

1. l'équivalence des formes Λ -normalisées est facile à décider: ces formes normalisées sont en fait des marqueurs et nous utiliserons donc les résultats du paragraphe précédent.
2. si T et T' sont des transducteurs équivalents alors il existe un entier Λ tel que leurs formes normalisées T_{Θ}^{Λ} et T'_{Θ}^{Λ} sont équivalentes (on se base sur le fait que si T et T' sont équivalents alors les mêmes torsions sont appliquées sauf pour un nombre fini d'arbres).

L'équivalence de T_{Θ}^{Λ} et T'_{Θ}^{Λ} étant décidable, l'équivalence de T et T' est semi-décidable.

La non-équivalence étant évidemment semi-décidable, on en conclut que l'équivalence est décidable (théorème 2.2.1).

En corollaire, on étend ce résultat au cas des transducteurs ascendants (théorème 2.2.2).

Remarque

Pour des raisons techniques, dans les paragraphes 2.2.2 et 2.2.3, nous ne considérons que la sous-classe de **LCT-LL**, notée **LCT-LL_i**, et composée des transducteurs infinitaires. Les résultats obtenus sont valables dans le cas général (paragraphe 2.2.4).

2.2.2 Transformations initiales réalisées à partir de deux ensembles d'états globalement équivalents d'un transducteur infinitaire

Cas des arbres de la forme $\sigma(t_1, t_2)$ avec $\pi(t_1) \neq \pi(t_2)$

Lemme 2.2.1 *A partir de deux ensembles d'états globalement équivalents, les mêmes transformations initiales sont réalisées sur les arbres de la forme $\sigma(t_1, t_2)$ pour lesquels $\pi(t_1) \neq \pi(t_2)$.*

PREUVE:

Considérons deux ensembles d'états globalement équivalents E et F d'un transducteur T de **LCT-LL₁**.

Soit q un état de E et soit $(\sigma(t_1, t_2), \delta(u_1, u_2))$ un couple d'arbres de la transformation $\hat{T}_{q(\sigma, \delta, \theta)}$ tel que $\pi(t_1) \neq \pi(t_2)$.

Pour simplifier les notations, on prendra $\theta = id_{[2]}$ (l'autre cas, $\theta = \langle 2; 2, 1 \rangle$, se traite de la même façon).

On a donc (propriété 1.4.1) $\pi(u_1) = \pi(t_1)$, $\pi(u_2) = \pi(t_2)$ et par conséquent $\pi(u_1) \neq \pi(u_2)$.

Supposons qu'il existe un état k de F et une torsion μ , $\mu \neq \theta$ (ici $\mu = \langle 2; 2, 1 \rangle$), telle que $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{k(\sigma, \delta, \mu)}$.

On aurait donc $k(\sigma(t_1, t_2)) \xrightarrow{*} \delta(k_1(t_2), k_2(t_1)) \xrightarrow{*} \delta(u_1, u_2)$, ce qui impliquerait l'égalité de $\pi(t_2)$ et de $\pi(u_1)$. C'est en contradiction avec les hypothèses.

Par conséquent, parce que E et F sont globalement équivalents, il existe au moins un état $k \in F$ tel que $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{k(\sigma, \delta, \theta)}$. \square

Cas des arbres de la forme $\sigma(t_1, t_2)$ pour lesquels $\pi(t_1) = \pi(t_2)$

L'exemple donné au début de ce paragraphe illustre le fait que, à partir d'états équivalents, des transformations initiales avec des torsions différentes peuvent être réalisées sur les arbres de cette forme. Nous montrons par le lemme ci-dessous que ce phénomène est "de profondeur bornée".

Lemme 2.2.2 *A partir de deux ensembles d'états globalement équivalents les mêmes transformations initiales peuvent être réalisées sur les arbres de la forme $\sigma(t_1, t_2)$ pour lesquels $\pi(t_1) = \pi(t_2)$, excepté éventuellement pour un nombre fini d'arbres.*

PREUVE:

Soient E et F deux ensembles d'états globalement équivalents de **LCT-LL₁**. On considère la différence $\hat{T}_{E(\sigma, \delta, \theta)} - \hat{T}_{F(\sigma, \delta, \theta)}$.

Dans le but de simplifier les notations nous prenons $\theta = id_{[2]}$.

A $\hat{T}_{E(\sigma, \delta, \theta)}$ nous associons l'ensemble \mathcal{Q} défini par:

$$\mathcal{Q} = \{((t, u), (t', u')) \text{ tel que } (\sigma(t, t'), \delta(u, u')) \in \hat{T}_{E(\sigma, \delta, \theta)}\},$$

et à $\hat{T}_{F(\sigma, \delta, \theta)}$ nous associons l'ensemble \mathcal{K} défini par:

$$\mathcal{K} = \{((t, u), (t', u')) \text{ tel que } (\sigma(t, t'), \delta(u, u')) \in \hat{T}_{F(\sigma, \delta, \theta)}\}.$$

On a donc $(\sigma(t, t'), \delta(u, u')) \in \hat{T}_{E(\sigma, \delta, \theta)} - \hat{T}_{F(\sigma, \delta, \theta)} \Leftrightarrow ((t, u), (t', u')) \in \mathcal{Q} - \mathcal{K}$.

Pour tout état q_i de E , aux α_i règles de la forme $q_i(\sigma(x_1, x_2)) \rightarrow \delta(q_{i_\alpha}(x_1), q'_{i_\alpha}(x_2))$ ($\alpha \in [\alpha_i]$) nous associons les ensembles $Q_{i_1}, Q'_{i_1}, \dots, Q_{i_{\alpha_i}}, Q'_{i_{\alpha_i}}$ qui sont les transformations réalisées à partir des états $q_{i_1}, q'_{i_1}, \dots, q_{i_{\alpha_i}}, q'_{i_{\alpha_i}}$.

De la même façon, pour tout état k_j de F , on associe aux α_j règles de la forme $k_j(\sigma(x_1, x_2)) \rightarrow \delta(k_{j_\beta}(x_1), k'_{j_\beta}(x_2))$ les ensembles $K_{j_1}, K'_{j_1}, \dots, K_{j_{\alpha_j}}, K'_{j_{\alpha_j}}$.

On désignera par \overline{Q}_i le complémentaire de Q_i dans $T_\Sigma \times T_\Delta$.

Des propriétés élémentaires de la théorie des ensembles nous permettent d'exprimer $\mathcal{Q} - \mathcal{K}$ en fonction de $Q_{i_\alpha}, Q'_{i_\alpha}, \overline{K}_{j_\beta}, \overline{K}'_{j_\beta}$ (avec $i \in [n], \alpha \in [\alpha_i], j \in [m], \beta \in [\alpha_j]$).

$$\begin{aligned} \mathcal{Q} - \mathcal{K} &= \left(\bigcup_{i \in [n], \alpha \in [\alpha_i]} Q_{i_\alpha} \times Q'_{i_\alpha} \right) - \left(\bigcup_{j \in [m], \beta \in [\alpha_j]} K_{j_\beta} \times K'_{j_\beta} \right) \\ &= \bigcup_{i \in [n], \alpha \in [\alpha_i]} \left(\bigcup_{P \subseteq A} (Q_{i_\alpha} \cap \left(\bigcap_{e \in P} \overline{K}_e \right)) \times (Q'_{i_\alpha} \cap \left(\bigcap_{f \in A-P} \overline{K}'_f \right)) \right) \\ &\quad \text{avec } A = \{j_\beta / j \in [m], \beta \in [\alpha_j]\}. \end{aligned}$$

Dans le cas particulier $P = \emptyset$, on pose $\bigcap_{e \in P} (\overline{K}_e) = T_\Sigma$.

Supposons que $\mathcal{Q} - \mathcal{K}$ soit infini.

Il existerait alors au moins $i \in [n], \alpha \in [\alpha_i]$ et $P \subseteq A$ tel que le produit $(Q_{i_\alpha} \cap \left(\bigcap_{e \in P} \overline{K}_e \right)) \times (Q'_{i_\alpha} \cap \left(\bigcap_{f \in A-P} \overline{K}'_f \right))$ soit infini.

On pourrait donc trouver dans ce produit au moins un couple $((t, u), (t', u'))$ pour lequel $\pi(t) \neq \pi(t')$,

c'est à dire en fait

au moins un couple $(\sigma(t, t'), \delta(u, u'))$ de la différence $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$ avec $\pi(t) \neq \pi(t')$.

Ceci est impossible puisque contraire au lemme 2.2.1.

La différence $\mathcal{Q} - \mathcal{K}$, et par conséquent $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$, est donc finie. \square

Nous proposons maintenant une autre preuve de ce lemme.

AUTRE PREUVE:

Signalons que cette preuve nous a été suggérée par l'un des rapporteurs de notre article [AnD93] soumis à la revue Rairo.

Comme précédemment E et F sont deux ensembles d'états globalement équivalents d'un transducteur T de LCT-LL_i et on considère la différence $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$ en prenant $\theta = id_{[2]}$.

A chaque couple $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$ nous associons l'ensemble $C_1 = \{(k_1, k_2) \text{ tel que } \exists k \in F, k(\sigma(x_1, x_2)) \rightarrow \delta(k_1(x_1), k_2(x_2)) \text{ est une règle de } T \text{ et } (t_1, u_1) \notin \widehat{T}_{k_1}\}$

et l'ensemble

$C_2 = \{(k_1, k_2) \text{ tel que } \exists k \in F, k(\sigma(x_1, x_2)) \rightarrow \delta(k_1(x_1), k_2(x_2)) \text{ est une règle de } T \text{ et } (t_2, u_2) \notin \widehat{T}_{k_2}\}$.

Si $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$ était infini alors, parce que T est un transducteur d'états finis, il existerait au moins deux couples d'arbres

$(\sigma(t_1, t_2), \delta(u_1, u_2))$ et $(\sigma(t'_1, t'_2), \delta(u'_1, u'_2))$,

- éléments de $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$,
- obtenus à partir d'un même état q de E par une règle de la forme $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_1), q_j(x_2))$,
- associés aux mêmes ensembles C_1 et C_2
- et tels que $\pi(t_1) \neq \pi(t'_2)$.

Considérons maintenant le couple $(\sigma(t_1, t'_2), \delta(u_1, u'_2))$.

Il peut s'obtenir en appliquant initialement la règle $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_1), q_j(x_2))$ et il est donc élément de $\widehat{T}_{E(\sigma, \delta, \theta)}$. Par contre il ne peut être obtenu à partir de F puisque, quelle que soit la règle $k(\sigma(x_1, x_2)) \rightarrow \delta(k_i(x_1), k_j(x_2))$ utilisée (avec $k \in F$), on a soit $(t_1, u_1) \notin \widehat{T}_{k_i}$, soit $(t'_2, u'_2) \notin \widehat{T}_{k_j}$. Ce couple serait donc élément de $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$ ce qui contredit le lemme précédent (lemme 2.2.1).

La différence $\widehat{T}_{E(\sigma, \delta, \theta)} - \widehat{T}_{F(\sigma, \delta, \theta)}$ est donc finie. \square

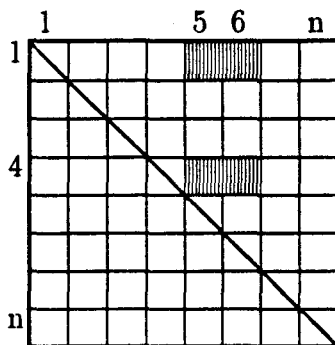
Ce résultat peut également être obtenu par une interprétation géométrique du problème considéré.

Soit n un nombre entier.

On appelle *produit* tout sous-ensemble $E \times F$ de $[n] \times [n]$ où E et F sont deux parties disjointes de $[n]$.

On désigne par T_n l'ensemble $\{(i, j) \mid i \in [n], j \in [n] \text{ et } i < j\}$.

Géométriquement, le produit $[n] \times [n]$ sera interprété comme une surface carrée de côté n . L'ensemble T_n représentera alors le triangle supérieur du carré. La figure suivante donne un exemple de produit dans le cas $n = 8$.



La surface hachurée est associée au produit $\{1, 4\} \times \{5, 6\}$.

1) Montrons d'abord que le recouvrement de T_n par des produits nécessite au moins $\log_2 n$ produits.

La preuve se fait par récurrence sur n .

La propriété est évidemment vraie pour $n = 2$.

Supposons la propriété vraie jusque n , c'est-à-dire qu'il faut au moins $\log_2 n$ produits pour recouvrir T_n , et montrons qu'elle l'est encore pour T_{2n} .

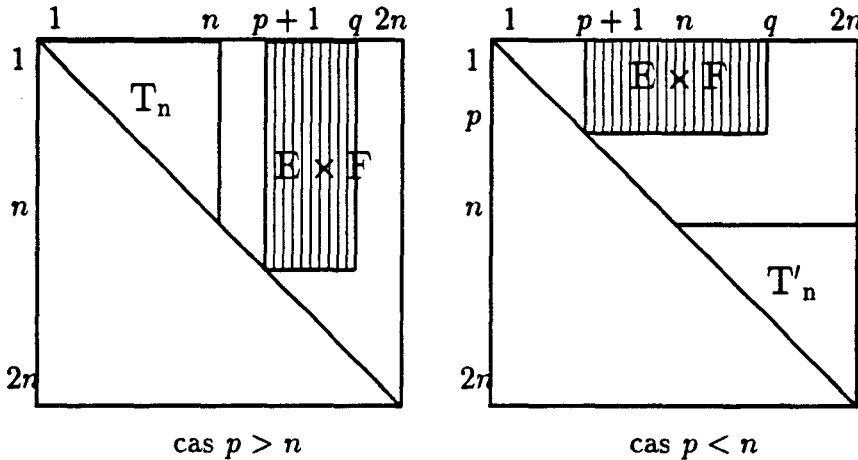
Pour cela supposons T_{2n} recouvert par $\log_2 n$ produits. Soit $E \times F$ un de ces produits. Par "renumérotation", on peut toujours obtenir $E = \{1, \dots, p\}$ et $F = \{p+1, \dots, q\}$. On distinguera deux cas selon la valeur de p (voir figure page suivante).

Premier cas: $p > n$

Dans ce cas $E \times F$ n'intercepte pas T_n , et T_n pourrait donc être recouvert par $\log_2 n - 1$ produits restants. Ce qui contredit l'hypothèse.

Second cas: $p < n$

Dans ce cas, le triangle $T'_n = \{(i, j) \mid i < j, i \in [n + 1, 2n]\}$ n'est pas intercepté par $E \times F$ et par conséquent T'_n pourrait être recouvert par les $\log_2 n - 1$ produits restants. Ce qui est impossible également.



Par conséquent, il faut au moins $\log_2 n + 1 = \log_2 2n$ produits pour recouvrir T_{2n} .

Le recouvrement de T_n , pour $n = 8$ par exemple, nécessite au moins 3 produits:

- $P_1 = \{1, 2, 3, 4\} \times \{5, 6, 7, 8\}$,
- $P_2 = \{1, 2, 5, 6\} \times \{3, 4, 7, 8\}$
- et $P_3 = \{1, 3, 5, 7\} \times \{2, 4, 6, 8\}$.

2) Interprétation du résultat précédent.

Considérons deux états équivalents q et k , et supposons qu'il existe n arbres de la forme $\sigma(\tau_i, \tau'_i)$, avec $\pi(\tau_i) = \pi(\tau'_i)$, pour lesquels la transformation initiale ne peut se faire à partir de l'état k avec la même torsion que celle qui est appliquée à partir de l'état q .

Pour les arbres $\sigma(\tau_i, \tau'_j)$ avec $i \in [n], j \in [n]$ et $i \neq j$ les mêmes torsions initiales doivent pouvoir être appliquées.

Cet ensemble d'arbres correspond en fait au carré de côté n privé de sa diagonale. Les n arbres $\sigma(\tau_i, \tau'_i)$ ($i \in [n]$) étant associés à cette diagonale.

Ainsi, si on considère seulement l'ensemble E des arbres $\sigma(\tau_i, \tau'_j)$ avec $i \in [n], j \in [n]$ et $i < j$ on retrouve le triangle supérieur T_n .

A toute règle, à partir de l'état k , assurant la transformation d'une partie de cet ensemble E , on associe un produit.

Par exemple le produit $P = \{1, 2\} \times \{5, 8\}$ correspond à une règle assurant la transformation des arbres $\sigma(\tau_1, \tau'_5)$, $\sigma(\tau_1, \tau'_8)$, $\sigma(\tau_2, \tau'_5)$ et $\sigma(\tau_2, \tau'_8)$.

En corollaire du résultat combinatoire obtenu en 2) on obtient alors:

Pour que les arbres $\sigma(\tau_i, \tau'_j)$ avec $i \in [n], j \in [n]$ et $i < j$ soient transduqués à partir de l'état k avec la même transformation initiale que pour l'état q il faut au moins $\log_2 n$ règles.

Il faut donc au moins $2 \times \log_2 n$ règles pour assurer la transformation de tous les arbres $\sigma(\tau_i, \tau'_j)$ avec $i \in [n], j \in [n]$ et $i \neq j$.

Le nombre de règles d'un transducteur étant fini, il n'existera qu'un nombre fini d'arbres pour lesquels des transformations initiales différentes sont nécessairement réalisées à partir de deux états équivalents. \square

A l'issue des lemmes 2.2.1 et 2.2.2, on est donc assuré de l'existence, pour tout couple (E, F) d'ensembles d'états globalement équivalents d'un transducteur T et pour toute transformation initiale (σ, δ, θ) , d'un entier, qu'on peut noter $\lambda_{(E,F),(\sigma,\delta,\theta)}$, tel que dès que $\pi(t) > \lambda_{(E,F),(\sigma,\delta,\theta)}$, on a $(t, u) \in \hat{T}_{E(\sigma,\delta,\theta)} \Leftrightarrow (t, u) \in \hat{T}_{F(\sigma,\delta,\theta)}$.

C'est parce que nous ne calculons pas de majorant pour ces entiers $\lambda_{(E,F),(\sigma,\delta,\theta)}$ que la décision de l'équivalence repose sur deux algorithmes de semi-décision.

2.2.3 Forme Λ -normalisée d'un transducteur infinitaire

Pour tout entier Λ , nous associons maintenant à tout transducteur T de LCT-LL_i une forme normalisée appelée forme Λ -normalisée et construite en deux étapes.

1. Dans une première phase, nous construisons une forme intermédiaire, appelée forme Λ -semi-normalisée, pour laquelle tous les arbres et sous-arbres de profondeur inférieure ou égale à Λ sont transformés globalement. Si l'entier Λ est suffisamment grand, on obtient ainsi un transducteur pour lequel à partir d'états équivalents les transformations peuvent se faire en appliquant les mêmes torsions pour tous les arbres.
2. La forme Λ -normalisée, qui est un marqueur, s'obtient ensuite en modifiant les règles non-closes de telle façon que les torsions ne soient plus appliquées.

La figure 2.3 illustre la démarche.

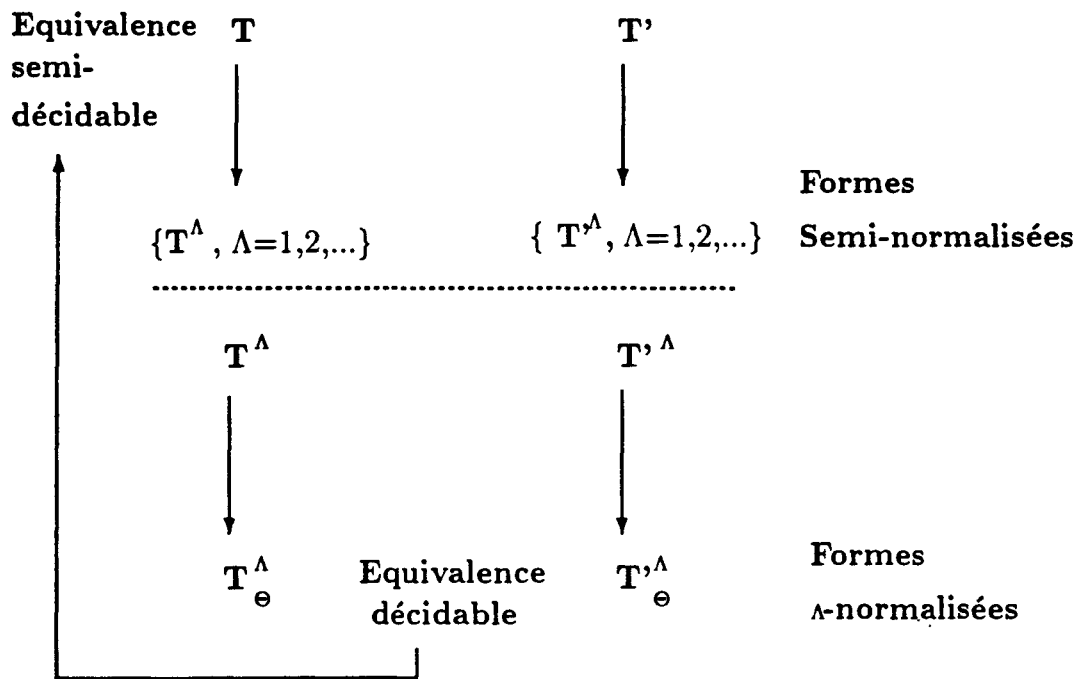


FIG. 2.1 - : Démarche dans le cas linéaire et complet

Les deux paragraphes qui suivent contiennent les constructions de ces formes et leurs propriétés.

Forme Λ -semi-normalisée

Nous avons montré dans le paragraphe précédent que, pour tout couple de transducteurs équivalents, il existait un entier N , tel que dès que la profondeur des arbres (et sous-arbres) est supérieure à N , les transformations peuvent se faire avec les mêmes torsions dans les deux transducteurs.

L'idée est donc de construire, pour tout entier Λ , un transducteur traitant globalement les arbres et sous-arbres dont la profondeur est inférieure ou égale à Λ , c'est à dire les transformant en un seul pas de réécriture. La forme ainsi obtenue, appelée Λ -semi-normalisée, est notée T^Λ .

Tout d'abord, pour tout état q d'un transducteur T , pour tout couple d'arbres $(t, u) \in \hat{T}_q$ et tel que $\pi(t) \leq \Lambda$, on ajoute une règle de la forme

- $q^{<\Lambda}(t) \rightarrow u$ si $\pi(t) < \Lambda$
- ou de la forme $q^\Lambda(t) \rightarrow u$ si $\pi(t) = \Lambda$.

Les arbres t et u sont de plus identifiés à de nouvelles lettres.

Nous devons également modifier les règles "non-closes" de T de telle façon que pour ces couples (t, u) , avec $\pi(t) \leq \Lambda$, la transformation ne puisse pas se faire autrement. Dans ce but nous introduisons, par l'intermédiaire des états, un contrôle des profondeurs des arbres qui sont transduits.

Ainsi à tout état q du transducteur de départ T , vont être associés deux états:

- $q^{<\Lambda}$, à partir duquel seules des transformations en un pas peuvent être réalisées
- et q^Λ permettant la transformation des arbres de profondeur égale ou supérieure à Λ .

Ainsi par exemple, on ne trouvera dans T^Λ aucune règle

- de la forme $q^\Lambda(\sigma(x)) \rightarrow \delta(q_i^{<\Lambda}(x))$, pour les lettres de rang 1,
- ou de la forme $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^{<\Lambda}(x_{\theta(1)}), q_j^{<\Lambda}(x_{\theta(2)}))$, pour les lettres de rang 2.

Pour les lettres de rang 2, l'un au moins des états de sortie sera donc indicé par Λ , c'est-à-dire sera de la forme q^Λ .

Nous montrons alors que (lemme 2.2.4) si Λ est suffisamment grand, à partir de deux ensembles d'états globalement équivalents de la forme Λ -semi-normalisée, les transformations peuvent être réalisées en appliquant les mêmes torsions pour tous les arbres.

Construction

Soit $T = \langle \Sigma, \Delta, Q, I, R \rangle$ un transducteur de LCT-LL_i.

Pour tout entier Λ , nous associons à T

le transducteur $T^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R^\Lambda \rangle$ pour lequel:

Σ^Λ et Δ^Λ sont de nouveaux alphabets dont les lettres peuvent être interprétées comme des arbres de $\text{dom}(\hat{T})$ et $\text{im}(\hat{T})$ de profondeur inférieure ou égale à Λ ,

Q^Λ , I^Λ et R^Λ sont définis par:

- $q^{<\Lambda}$ et q^Λ sont des états de Q^Λ si et seulement si q appartient à Q , et ils sont éléments de I^Λ si et seulement si q est dans I .
- $q^{<\Lambda}(t) \rightarrow u$ (resp. $q^\Lambda(t) \rightarrow u$) est une règle de R^Λ , t est une lettre de Σ^Λ et u est une lettre de Δ^Λ si et seulement si $(t, u) \in \hat{T}_q$ avec $t \in T_\Sigma$ et $\pi(t) < \Lambda$ (resp. $\pi(t) = \Lambda$).
- $q^\Lambda(\sigma(x)) \rightarrow \delta(q_i^\Lambda(x))$ est une règle de R^Λ si et seulement si $q(\sigma(x)) \rightarrow \delta(q_i(x))$ est une règle de R .
- $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_{\theta(1)}), q_j^\Lambda(x_{\theta(2)}))$,
 $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^{<\Lambda}(x_{\theta(1)}), q_j^\Lambda(x_{\theta(2)}))$ et
 $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_{\theta(1)}), q_j^{<\Lambda}(x_{\theta(2)}))$ sont des règles de R^Λ si et seulement si
 $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_{\theta(1)}), q_j(x_{\theta(2)}))$ est une règle de R .

Exemple

Reprenons les transducteurs T et T' définis dans le paragraphe 2.2.1.

Pour $\Lambda = 1$, par exemple, leurs formes Λ -semi-normalisées T^Λ et T'^Λ sont définies par:

Règles closes de T^Λ

$$\begin{aligned} q_1^{<\Lambda}(\bar{a}) &\rightarrow \bar{a} \\ q_2^{<\Lambda}(\bar{\alpha}) &\rightarrow \bar{\alpha} \\ q_1^\Lambda(a(\bar{a})) &\rightarrow a(\bar{a}) \\ q_2^\Lambda(\alpha(\bar{\alpha})) &\rightarrow \alpha(\bar{\alpha}) \\ q^\Lambda(\sigma(\bar{a}, \bar{\alpha})) &\rightarrow \delta(\bar{a}, \bar{\alpha}) \end{aligned}$$

Règles non-closes de T^Λ

$$\begin{aligned} q^\Lambda(\sigma(x, y)) &\rightarrow \delta(q_1^\Lambda(x), q_2^\Lambda(y)) \\ q^\Lambda(\sigma(x, y)) &\rightarrow \delta(q_1^{<\Lambda}(x), q_2^\Lambda(y)) \\ q^\Lambda(\sigma(x, y)) &\rightarrow \delta(q_1^\Lambda(x), q_2^{<\Lambda}(y)) \\ q_1^\Lambda(a(x)) &\rightarrow a(q_1^\Lambda(x)) \\ q_2^\Lambda(\alpha(x)) &\rightarrow \alpha(q_2^\Lambda(x)) \end{aligned}$$

Règles closes de T'^Λ

$$\begin{aligned} k_1^{<\Lambda}(\bar{a}) &\rightarrow \bar{a} & k_2^{<\Lambda}(\bar{\alpha}) &\rightarrow \bar{\alpha} \\ k_{11}^{<\Lambda}(\bar{a}) &\rightarrow \bar{a} & k_{22}^{<\Lambda}(\bar{\alpha}) &\rightarrow \bar{\alpha} \\ k_3^{<\Lambda}(\bar{\alpha}) &\rightarrow \bar{\alpha} & k_4^{<\Lambda}(\bar{a}) &\rightarrow \bar{a} \\ k_1^\Lambda(a(\bar{a})) &\rightarrow a(\bar{a}) & k_2^\Lambda(\alpha(\bar{\alpha})) &\rightarrow \alpha(\bar{\alpha}) \\ k_{11}^\Lambda(a(\bar{a})) &\rightarrow a(\bar{a}) & k_{22}^\Lambda(\alpha(\bar{\alpha})) &\rightarrow \alpha(\bar{\alpha}) \\ k_1'^\Lambda(a(\bar{a})) &\rightarrow a(\bar{a}) & k_2'^\Lambda(\alpha(\bar{\alpha})) &\rightarrow \alpha(\bar{\alpha}) \\ k^\Lambda(\sigma(\bar{a}, \bar{\alpha})) &\rightarrow \delta(\bar{a}, \bar{\alpha}) \end{aligned}$$

Règles non-closes de T'^Λ

$$\begin{aligned} k^\Lambda(\sigma(x, y)) &\rightarrow \delta(k_1'^\Lambda(x), k_2^\Lambda(y)) \\ k^\Lambda(\sigma(x, y)) &\rightarrow \delta(k_1^{<\Lambda}(x), k_2^\Lambda(y)) \\ k^\Lambda(\sigma(x, y)) &\rightarrow \delta(k_1'^\Lambda(x), k_2^{<\Lambda}(y)) \\ \\ k^\Lambda(\sigma(x, y)) &\rightarrow \delta(k_1^\Lambda(x), k_2'^\Lambda(y)) \\ k^\Lambda(\sigma(x, y)) &\rightarrow \delta(k_1^{<\Lambda}(x), k_2'^\Lambda(y)) \\ k^\Lambda(\sigma(x, y)) &\rightarrow \delta(k_1^\Lambda(x), k_2'^\Lambda(y)) \end{aligned}$$

$$k^\Lambda(\sigma(x, y)) \rightarrow \delta(k_3^\Lambda(y), k_4^\Lambda(x))$$

$$k^\Lambda(\sigma(x, y)) \rightarrow \delta(k_3^{<\Lambda}(y), k_4^\Lambda(x))$$

$$k^\Lambda(\sigma(x, y)) \rightarrow \delta(k_3^\Lambda(y), k_4^{<\Lambda}(x))$$

$$k_1^\Lambda(a(x)) \rightarrow a(k_1^\Lambda(x))$$

$$k_1^{\prime\Lambda}(a(x)) \rightarrow a(k_{11}^\Lambda(x))$$

$$k_{11}^\Lambda(a(x)) \rightarrow a(k_{11}^\Lambda(x))$$

$$k_2^\Lambda(\alpha(x)) \rightarrow \alpha(k_2^\Lambda(x))$$

$$k_2^{\prime\Lambda}(\alpha(x)) \rightarrow \alpha(k_{22}^\Lambda(x))$$

$$k_{22}^\Lambda(\alpha(x)) \rightarrow \alpha(k_{22}^\Lambda(x))$$

Le couple d'arbres $(\sigma(\bar{a}, \bar{\alpha}), \delta(\bar{a}, \bar{\alpha}))$, pour lequel des transformations initiales différentes étaient réalisées dans T et T' , est maintenant traité globalement.

Les règles utilisées sont:

$$q^\Lambda(\sigma(\bar{a}, \bar{\alpha})) \rightarrow \delta(\bar{a}, \bar{\alpha}) \text{ dans } T^\Lambda$$

$$\text{et } k^\Lambda(\sigma(\bar{a}, \bar{\alpha})) \rightarrow \delta(\bar{a}, \bar{\alpha}) \text{ dans } T'^\Lambda.$$

Pour ce couple, aucun autre calcul n'est possible dans T^Λ et T'^Λ .

La similitude entre les transformations réalisées dans le transducteur d'origine et celles obtenues dans les formes Λ -semi-normalisées nous conduit naturellement à identifier ces deux formes pour toute valeur de Λ .

Identification de T et T^Λ

• Pour tout mouvement $q(t) \xrightarrow{*} u$ dans T avec $\pi(t) \leq \Lambda$, il existe dans T^Λ

- une règle de la forme $q^{<\Lambda}(t) \rightarrow u$ si $\pi(t) < \Lambda$
- ou de la forme $q^\Lambda(t) \rightarrow u$ si $\pi(t) = \Lambda$.

Ces termes clos t de $\text{dom}(\hat{T})$ et u de $\text{im}(\hat{T})$, dont la profondeur est inférieure ou égale à Λ , sont identifiés à de nouvelles lettres dans T^Λ . C'est donc la seule dérivation possible pour (t, u) dans T^Λ .

• Pour tout couple d'arbres (t, u) de \hat{T} avec $\pi(t) > \Lambda$, il existe une décomposition unique de t et u en $t = t_0(t_1, \dots, t_n)$ et $u = u_0(u_1, \dots, u_n)$ telle que:

- pour tout i de $[n]$, $\pi(t_i) \leq \Lambda$ et il n'existe aucun sous-arbre de t , de profondeur inférieure ou égale à Λ , dont t_i est un sous-arbre propre

- les dérivations

$$q(t_0(t_1, \dots, t_n)) \xrightarrow{*} u_0(q_1(t_{\theta(1)}), \dots, q_n(t_{\theta(n)})) \text{ dans } T$$

et

$$q^\Lambda(t_0(t_1, \dots, t_n)) \xrightarrow{*} u_0(q'_1(t_{\theta(1)}), \dots, q'_n(t_{\theta(n)})) \text{ dans } T^\Lambda$$

(avec $\forall i \in [n]$, q'_i est soit $q_i^{<\Lambda}$ soit q_i^Λ) sont analogues, c'est à dire qu'elles ne diffèrent l'une de l'autre que par les labels des états: $q_i^{<\Lambda}$ (ou q_i^Λ) est utilisé dans T^Λ lorsque q_i l'est dans T .

Pour ces raisons, pour tout Λ nous identifions T et T^Λ et, dans la suite, pour tout couple d'arbres t de $T_{\Sigma \cup \Sigma^\Lambda}$ (resp. $T_{\Delta \cup \Delta^\Lambda}$), $\pi(t)$ désignera la profondeur de l'arbre correspondant de T_Σ (resp. T_Δ).

Il nous faut maintenant montrer que, lorsque Λ est suffisamment grand, pour les formes Λ -semi-normalisées les transformations peuvent être réalisées avec les mêmes torsions pour tous les arbres. Par le premier lemme nous montrons que c'est le cas pour les transformations initiales, c'est-à-dire pour la transformation de la racine. Le second lemme généralise cette propriété à la transformation complète.

Lemme 2.2.3 *Lorsque Λ est suffisamment grand, à partir de deux ensembles d'états globalement équivalents de T^Λ , les mêmes transformations initiales peuvent être réalisées pour tous les arbres.*

PREUVE:

Soient E et F deux ensembles d'états globalement équivalents de T^Λ et soit (t, u) un couple d'arbres de $\hat{T}_E^\Lambda (= \hat{T}_F^\Lambda)$.

Prenons comme valeur pour Λ , un majorant de l'ensemble des entiers $\lambda_{(E,F),(\sigma,\delta,\theta)}$ obtenus pour tous les couples (E, F) d'ensembles d'états globalement équivalents et pour toutes les transformations initiales (σ, δ, θ) .

Le lemme 2.2.2 nous permet d'affirmer que, pour tout couple (t, u) tel que $\pi(t) > \Lambda$, les mêmes torsions initiales sont utilisées.

On obtient le même résultat dans le cas $\pi(t) \leq \Lambda$, puisqu' alors t et u sont traités globalement (assimilés à des lettres de rang 0); la construction nous assurant de plus, grâce à un contrôle des profondeurs d'arbres effectués par les états, que c'est le seul calcul possible. \square

Lemme 2.2.4 *Lorsque Λ est suffisamment grand, si E et F sont deux ensembles d'états globalement équivalents de T^Λ alors pour tout calcul $q(t) \xrightarrow{*} u$, avec $q \in E$, il existe $k \in F$ tel que les torsions appliquées lors du calcul $k(t) \xrightarrow{*} u$ sont les mêmes que celles appliquées à partir de q .*

PREUVE:

La preuve se fait par récurrence sur la profondeur des dérivations.

Un mouvement de dérivations $q(t_0(t_1, \dots, t_n)) \xrightarrow{*} u_0(q_1(t_{\theta(1)}), \dots, q_n(t_{\theta(n)}))$ est dit de *profondeur p* si et seulement si chaque état q_i (pour $i \in [n]$) est obtenu après exactement $p - 1$ pas de réécriture.

Dans le but de simplifier l'écriture de la preuve, nous considérons ici deux états équivalents q et k . Le résultat obtenu peut se généraliser, sans difficultés majeures mais au prix d'une surcharge des notations, au cas d'ensembles d'états globalement équivalents.

Prenons comme valeur de Λ un majorant de l'ensemble des entiers $\lambda_{(E,F),(\sigma,\delta,\theta)}$ obtenus pour tous les couples (E, F) d'ensembles d'états globalement équivalents et pour toutes les transformations initiales (σ, δ, θ) .

On ne considère que des couples (t, u) de $\hat{T}_q^\Lambda (= \hat{T}_k^\Lambda)$, tels que $\pi(t) > \Lambda$ (dans le cas $\pi(t) \leq \Lambda$, t est en fait assimilé à une lettre).

Supposons la propriété vraie jusqu'à la profondeur p , et montrons qu'elle l'est encore

à la profondeur $p + 1$.

Premier cas: $p < \pi(t) - \Lambda$.

Soit $(t, u) \in \hat{T}_q^\Lambda (= \hat{T}_k^\Lambda)$. Considérons le mouvement

$$\begin{aligned} q(t) = q(t_0(t_1, \dots, t_n)) &\xrightarrow{*} u_0(q_1(t_{\theta(1)}), \dots, q_n(t_{\theta(n)})) \\ &\xrightarrow{*} u_0(u_1, \dots, u_n) = u \end{aligned}$$

avec $\pi(t_0) = p$.

La propriété étant supposée vraie jusqu'à la profondeur p , il existe à partir de k au moins un calcul appliquant les mêmes torsions (que celles utilisées dans le calcul à partir de q) jusqu'à la profondeur p :

$$\begin{aligned} k(t) = k(t_0(t_1, \dots, t_n)) &\xrightarrow{*} u_0(k_1(t_{\theta(1)}), \dots, k_n(t_{\theta(n)})) \\ &\xrightarrow{*} u_0(u_1, \dots, u_n) = u \end{aligned}$$

Soit $i \in [n]$; considérons les ensembles

$C_i = \{q_i/q(t_0(x_1, \dots, x_n)) \xrightarrow{*} u_0(q_{j_1}(x_{\theta(1)}), \dots, q_i(x_{\theta(i)}), \dots, q_{j_n}(x_{\theta(n)}))\}$ et

$D_i = \{k_i/k(t_0(x_1, \dots, x_n)) \xrightarrow{*} u_0(k_{j_1}(x_{\theta(1)}), \dots, k_i(x_{\theta(i)}), \dots, k_{j_n}(x_{\theta(n)}))\}$.

Supposons maintenant que, à partir de ces ensembles d'états C_i et D_i , la transformation $(t_{\theta(i)}, u_i)$ ne puisse pas se faire avec la même torsion initiale.

Les ensembles d'états C_i et D_i ne seraient donc pas globalement équivalents (lemme 2.2.3 et Λ choisi suffisamment grand) et il existerait alors au moins un couple d'arbres (\bar{t}, \bar{u}) dans $\hat{T}_{C_i} - \hat{T}_{D_i}$.

Par conséquent, pour le couple $(t_0(\bar{t}_1, \dots, \bar{t}, \dots, \bar{t}_n), u_0(\bar{u}_1, \dots, \bar{u}, \dots, \bar{u}_n))$ on aurait

$$\begin{aligned} q(t_0(\bar{t}_1, \dots, \bar{t}, \dots, \bar{t}_n)) &\xrightarrow{*} u_0(q_1(\bar{t}_{\theta(1)}), \dots, q_i(\bar{t}), \dots, q_n(\bar{t}_{\theta(n)})) \\ &\xrightarrow{*} u_0(\bar{u}_1, \dots, \bar{u}, \dots, \bar{u}_n) \end{aligned}$$

alors que quel que soit le mouvement de dérivation

$$k(t_0(\bar{t}_1, \dots, \bar{t}, \dots, \bar{t}_n)) \xrightarrow{*} u_0(k_{j_1}(\bar{t}_{\theta(1)}), \dots, k_i(\bar{t}), \dots, k_{j_n}(\bar{t}_{\theta(n)})),$$

où $k_i \in D_i$, (\bar{t}, \bar{u}) n'est pas transformé à partir de k_i .

Les états q et k étant équivalents, $(t_0(\bar{t}_1, \dots, \bar{t}, \dots, \bar{t}_n), u_0(\bar{u}_1, \dots, \bar{u}, \dots, \bar{u}_n)) \in \hat{T}_k$.

On aurait donc nécessairement, à partir de l'état k , un mouvement de dérivation tel

que les torsions appliquées sont différentes de celles appliquées à partir de q avant la profondeur p . Ceci contredit les hypothèses. Les mêmes torsions peuvent donc être appliquées à la profondeur $p + 1$.

Second cas: $p = \pi(t) - \Lambda$.

Les mêmes torsions sont appliquées à la profondeur $p + 1$ puisque dans ce cas les arbres transformés sont identifiés à des lettres de Σ^Λ . \square

Forme Λ -normalisée d'un transducteur infinitaire

Pour tout transducteur, nous avons obtenu à l'issue de la première phase une forme Λ -semi-normalisée telle que si l'entier Λ est assez grand alors à partir d'ensembles d'états globalement équivalents les transformations peuvent se faire avec les mêmes torsions pour tous les arbres.

Il n'est donc plus nécessaire d'appliquer effectivement ces torsions. Mais, comme le montre l'exemple ci-dessous, il n'est cependant pas possible de simplement les supprimer.

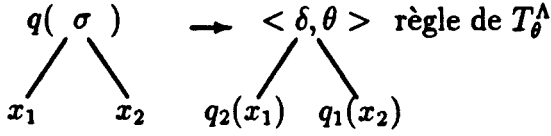
Exemple:

$$\begin{aligned} q(\sigma(x_1, x_2)) &\rightarrow \delta(q_1(x_1), q_2(x_2)) \\ q_1(\bar{a}) &\rightarrow \bar{a} \\ q_2(\bar{\alpha}) &\rightarrow \bar{\alpha} \end{aligned}$$

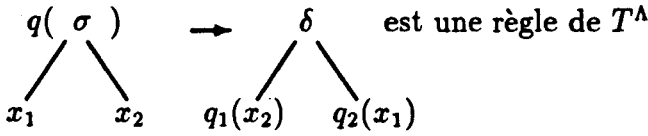
$$\begin{aligned} k(\sigma(x_1, x_2)) &\rightarrow \delta(k_1(x_2), k_2(x_1)) \\ k_1(\bar{\alpha}) &\rightarrow \bar{\alpha} \\ k_2(\bar{a}) &\rightarrow \bar{a} \end{aligned}$$

Les états q et k ne sont pas équivalents parce que $(\sigma(\bar{a}, \bar{\alpha}), \delta(\bar{a}, \bar{\alpha})) \in \hat{T}_q - \hat{T}_k$. Par simple suppression des torsions, ce même couple deviendrait élément de \hat{T}_k . On pourrait ainsi obtenir deux états équivalents dans la forme normalisée alors qu'ils ne le sont pas dans les transducteurs de départ. Pour cette raison les torsions, qui ne sont plus appliquées, sont codées dans les noeuds de l'arbre image.

On aura,



si et seulement si



$\langle \delta, \theta \rangle$ est considéré comme une nouvelle lettre (ici, $\theta = \langle 2; 2, 1 \rangle$).

Ainsi pour l'exemple précédent, le couple $(\sigma(\bar{a}, \bar{\alpha}), \delta(\bar{a}, \bar{\alpha}))$ sera codé:

- $(\sigma(\bar{a}, \bar{\alpha}), \langle \delta, id \rangle (\langle \bar{a}, id \rangle, \langle \bar{\alpha}, id \rangle))$ à partir de l'état q
- et $(\sigma(\bar{a}, \bar{\alpha}), \langle \delta, \theta \rangle (\langle \bar{a}, id \rangle, \langle \bar{\alpha}, id \rangle))$ à partir de k .

La non-équivalence des états q et k est ainsi préservée.

Construction

Soient $T = \langle \Sigma, \Delta, Q, I, R \rangle$ un transducteur de **LCT-LL_i**

et $T^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R^\Lambda \rangle$ sa forme Λ -semi-normalisée.

On associe à T le transducteur $T_\theta^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta_\theta, Q^\Lambda, I^\Lambda, R_\theta \rangle$ où Δ_θ et R_θ sont définis par:

- $q(\sigma) \rightarrow \langle \delta, id \rangle$ est une règle de R_θ et $\langle \delta, id \rangle$ est une lettre de Δ_θ si et seulement si $q(\sigma) \rightarrow \delta$ est une règle de R^Λ .
- $q(\sigma(x)) \rightarrow \langle \delta, id \rangle (q_i(x))$ est une règle de R_θ et $\langle \delta, id \rangle$ est une lettre de Δ_θ si et seulement si $q(\sigma(x)) \rightarrow \delta(q_i(x))$ est une règle de R^Λ .
- $q(\sigma(x_1, x_2)) \rightarrow \langle \delta, \theta \rangle (q_{\theta^{-1}(1)}(x_1), q_{\theta^{-1}(2)}(x_2))$ est une règle de R_θ et $\langle \delta, \theta \rangle$ est une lettre de Δ_θ si et seulement si $q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$ est une règle de R^Λ .

Remarque

Pour tout entier Λ , la forme Λ -normalisée de tout transducteur linéaire et complet infiniitaire est un marqueur.

Exemple

Reprenons les transducteurs T et T' décrits dans le paragraphe 2.2.1 de ce chapitre et dont la forme Λ -semi-normalisée a été construite pour $\Lambda = 1$ (paragraphe 2.2.3). Dans la forme 1-normalisée, pour l'indication de la torsion dans la lettre image, on désignera par id l'identité et par μ la torsion $\langle 2; 2, 1 \rangle$.

Règles closes de T_{Θ}^{Λ} .

$$\begin{aligned} q_1^{\leq \Lambda}(\bar{a}) &\rightarrow \langle \bar{a}, id \rangle & q_2^{\leq \Lambda}(\bar{\alpha}) &\rightarrow \langle \bar{\alpha}, id \rangle \\ q_1^{\Lambda}(a(\bar{a})) &\rightarrow \langle a(\bar{a}), id \rangle & q_2^{\Lambda}(\alpha(\bar{\alpha})) &\rightarrow \langle \alpha(\bar{\alpha}), id \rangle \\ q^{\Lambda}(\sigma(\bar{a}, \bar{\alpha})) &\rightarrow \langle \delta(\bar{a}, \bar{\alpha}), id \rangle \end{aligned}$$

Règles non-closes de T_{Θ}^{Λ} .

$$\begin{aligned} q^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (q_1^{\Lambda}(x), q_2^{\Lambda}(y)) \\ q^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (q_1^{\leq \Lambda}(x), q_2^{\Lambda}(y)) \\ q^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (q_1^{\Lambda}(x), q_2^{\leq \Lambda}(y)) \\ q_1^{\Lambda}(a(x)) &\rightarrow \langle a, id \rangle (q_1^{\Lambda}(x)) \\ q_2^{\Lambda}(\alpha(x)) &\rightarrow \langle \alpha, id \rangle (q_2^{\Lambda}(x)) \end{aligned}$$

Règles closes de T'_{Θ}^{Λ} .

$$\begin{aligned} k_1^{\leq \Lambda}(\bar{a}) &\rightarrow \langle \bar{a}, id \rangle & k_2^{\leq \Lambda}(\bar{\alpha}) &\rightarrow \langle \bar{\alpha}, id \rangle \\ k_{11}^{\leq \Lambda}(\bar{a}) &\rightarrow \langle \bar{a}, id \rangle & k_{22}^{\leq \Lambda}(\bar{\alpha}) &\rightarrow \langle \bar{\alpha}, id \rangle \\ k_3^{\leq \Lambda}(\bar{\alpha}) &\rightarrow \langle \bar{\alpha}, id \rangle & k_4^{\leq \Lambda}(\bar{a}) &\rightarrow \langle \bar{a}, id \rangle \\ k_1^{\Lambda}(a(\bar{a})) &\rightarrow \langle a(\bar{a}), id \rangle & k_2^{\Lambda}(\alpha(\bar{\alpha})) &\rightarrow \langle \alpha(\bar{\alpha}), id \rangle \\ k_{11}^{\Lambda}(a(\bar{a})) &\rightarrow \langle a(\bar{a}), id \rangle & k_{22}^{\Lambda}(\alpha(\bar{\alpha})) &\rightarrow \langle \alpha(\bar{\alpha}), id \rangle \\ k_1^{\prime \Lambda}(a(\bar{a})) &\rightarrow \langle a(\bar{a}), id \rangle & k_2^{\prime \Lambda}(\alpha(\bar{\alpha})) &\rightarrow \langle \alpha(\bar{\alpha}), id \rangle \\ k^{\Lambda}(\sigma(\bar{a}, \bar{\alpha})) &\rightarrow \langle \delta(\bar{a}, \bar{\alpha}), id \rangle \end{aligned}$$

Règles non-closes de T'_{Θ}^{Λ} .

$$\begin{aligned} k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (k_1^{\Lambda}(x), k_2^{\Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (k_1^{\prime \Lambda}(x), k_2^{\Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (k_1^{\Lambda}(x), k_2^{\leq \Lambda}(y)) & k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, \mu \rangle (k_4^{\Lambda}(x), k_3^{\Lambda}(y)) \\ & & & k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, \mu \rangle (k_4^{\Lambda}(x), k_3^{\leq \Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (k_1^{\Lambda}(x), k_2^{\prime \Lambda}(y)) & k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, \mu \rangle (k_4^{\leq \Lambda}(x), k_3^{\Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (k_1^{\leq \Lambda}(x), k_2^{\prime \Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) &\rightarrow \langle \delta, id \rangle (k_1^{\Lambda}(x), k_2^{\prime \leq \Lambda}(y)) \end{aligned}$$

$$\begin{aligned} k_1^{\Lambda}(a(x)) &\rightarrow \langle a, id \rangle (k_1^{\Lambda}(x)) & k_2^{\Lambda}(\alpha(x)) &\rightarrow \langle \alpha, id \rangle (k_2^{\Lambda}(x)) \\ k_1^{\prime \Lambda}(a(x)) &\rightarrow \langle a, id \rangle (k_1^{\Lambda}(x)) & k_2^{\prime \Lambda}(\alpha(x)) &\rightarrow \langle \alpha, id \rangle (k_2^{\Lambda}(x)) \\ k_{11}^{\Lambda}(a(x)) &\rightarrow \langle a, id \rangle (k_{11}^{\Lambda}(x)) & k_{22}^{\Lambda}(\alpha(x)) &\rightarrow \langle \alpha, id \rangle (k_{22}^{\Lambda}(x)) \end{aligned}$$

2.2.4 Décidabilité de l'équivalence dans LCT-LL

Les résultats obtenus dans les paragraphes précédents s'étendent facilement au cas général des transducteurs non nécessairement infinitaires.

Il est clair que, pour tout Λ , la forme Λ -normalisée de tout transducteur de LCT-LL se construit de la même façon.

De plus, si N_F est le nombre d'états finitaires d'un transducteur T , alors la profondeur de tout arbre de $dom(\hat{T})$ transformé à partir d'un état finitaire est au plus N_F . Par conséquent dès que Λ est supérieur à N_F les seuls états intervenant sont infinitaires. Les lemmes 2.2.3 et 2.2.4 sont donc valables dans le cas général.

Lemme 2.2.5 *Soit T et T' deux transducteurs de LCT-LL. Lorsque Λ est suffisamment grand, T et T' sont équivalents si et seulement si les marqueurs T_{\emptyset}^{Λ} et T'_{\emptyset}^{Λ} sont équivalents.*

ELEMENTS DE PREUVE:

Par le lemme 2.2.4 nous savons que, lorsque l'entier Λ est suffisamment grand, si E et F sont deux ensembles d'états globalement équivalents alors les transformations à partir de E et F peuvent se faire avec les mêmes torsions pour tous les couples d'arbres. Il est maintenant évident que l'équivalence des états est préservée lors de la construction de la forme Λ -normalisée puisque les torsions qui ne sont plus appliquées sont codées dans les noeuds des arbres produits en sortie. Ainsi, pour une valeur suffisamment grande de Λ , $\hat{T}_{\emptyset_E}^{\Lambda} = \hat{T}_{\emptyset_F}^{\Lambda} \Leftrightarrow \hat{T}_E = \hat{T}_F$.

Pour transposer ce résultat au cas de deux transducteurs $T = \langle \Sigma, \Delta, Q, I, R \rangle$ et $T' = \langle \Sigma, \Delta, Q', I', R' \rangle$, avec $Q \cap Q' = \emptyset$ (par renommage des états d'un des deux transducteurs, on peut toujours se ramener à cette situation), il nous suffit de considérer le transducteur $\tau = \langle \Sigma, \Delta, Q \cup Q', I \cup I', R \cup R' \rangle$. On applique alors les résultats précédents aux ensembles d'états I et I' . \square

On peut maintenant établir le résultat suivant:

Théorème 2.2.1 *L'équivalence des transducteurs descendants lettre à lettre linéaires et complets est décidable.*

PREUVE:

Parce que l'équivalence des marqueurs est décidable, le lemme 2.2.5 établit la semi-décidabilité de l'équivalence.

La non-équivalence étant évidemment semi-décidable, nous concluons. \square

2.2.5 Cas des transducteurs ascendants

Dans ce paragraphe, nous étendons le résultat précédent au cas des transducteurs lettre à lettre, linéaires et complets mais maintenant ascendants .

On désignera par **LCB-LL** la classe ainsi définie.

Dans [Eng75], J.Engelfriet a montré que la classe des transducteurs ascendants linéaires et complets est égale à celle des transducteurs descendants linéaires et complets (théorème 1.4.1 dans ce mémoire).

Soient B_1 et B_2 deux transducteurs ascendants lettre à lettre linéaires et complets, et T_1 et T_2 les transducteurs descendants linéaires et complets réalisant respectivement les mêmes transformations.

Les transducteurs T_1 et T_2 s'obtiennent à partir de B_1 et B_2 par "renversement des flèches":

si $\sigma(q_1(x_1), \dots, q_n(x_n)) \rightarrow q(\sigma(x_{\theta(1)}, \dots, x_{\theta(1)}))$ est une règle de B_1 (resp. B_2)
alors $q(\sigma(x_1, \dots, x_n)) \rightarrow \sigma(q_1(x_{\theta(1)}), \dots, q_n(x_{\theta(1)}))$ est une règle de T_1 (resp. T_2).

T_1 et T_2 sont donc lettre à lettre.

L'équivalence est décidable dans **LCT-LL**, elle l'est donc dans **LCB-LL**.

Théorème 2.2.2 *L'équivalence des transducteurs ascendants lettre à lettre linéaires et complets est décidable.*

2.2.6 Cas de lettres de rang supérieur à 2

Dans les paragraphes précédents, les preuves ont été réalisées pour des lettres de rang au plus 2. Des résultats de même nature sont obtenus pour des lettres de rang supérieur à 2 mais quelques différences existent cependant dans l'énoncé de certains lemmes. C'est le cas, par exemple, pour la généralisation du lemme 2.2.2. En effet comme l'illustre l'exemple suivant, la différence $\hat{T}_{q(\sigma, \delta, \theta)} - \hat{T}_{k(\sigma, \delta, \theta)}$ peut être infinie (q et k étant équivalents).

Considérons le transducteur T défini par:

$$\begin{array}{ll}
q(\sigma(x, y, z)) \rightarrow \delta(q_1(x), q_2(y), q_3(z)) & \\
q_1(\alpha(x)) \rightarrow \alpha(q_1(x)) & q_1(\bar{\alpha}) \rightarrow \bar{\alpha} \\
q_2(\beta(x)) \rightarrow \beta(q_2(x)) & q_2(\bar{\beta}) \rightarrow \bar{\beta} \\
q_3(\gamma(x)) \rightarrow \gamma(q_3(x)) & q_3(\bar{\gamma}) \rightarrow \bar{\gamma}
\end{array}$$

$$\begin{array}{ll}
k(\sigma(x, y, z)) \rightarrow \delta(q_1(x), k_2(y), q_3(z)) & k(\sigma(x, y, z)) \rightarrow \delta(q_1(x), q_2(y), k_3(z)) \\
k(\sigma(x, y, z)) \rightarrow \delta(q_1(x), k_4(z), k_5(y)) & \\
k_2(\beta(x)) \rightarrow \beta(q_2(x)) & \\
k_3(\gamma(x)) \rightarrow \gamma(q_3(x)) & \\
k_4(\bar{\gamma}) \rightarrow \bar{\beta} & \\
k_5(\bar{\beta}) \rightarrow \bar{\gamma} &
\end{array}$$

Les états q et k sont équivalents et pourtant, quel que soit $(\tau, \tau') \in \hat{T}_{q_1}$, pour tous les couples de la forme $(\sigma(\tau, \bar{\beta}, \bar{\gamma}), \delta(\tau', \bar{\beta}, \bar{\gamma}))$, c'est à dire donc pour une infinité de couples, des transformations initiales différentes seront appliquées à partir de q et de k . A partir de q la torsion initiale est $\theta = \langle 3; 1, 2, 3 \rangle$ et à partir de k on applique la torsion $\theta' = \langle 3; 1, 3, 2 \rangle$. On remarquera cependant que cette infinité est uniquement due aux couples (τ, τ') et qu'en considérant la "projection" de cet ensemble de couples sur les deux dernières composantes (celles sur lesquelles θ et θ' diffèrent), on se ramène à des ensembles finis.

Nous donnons maintenant la généralisation du lemme 2.2.2 en considérant, pour simplifier les notations, des lettres de rang 3.

Pour toute transformation initiale (σ, δ, θ) , on définit

$$\mathcal{Q}_{(\sigma, \delta, \theta)} = \{((t, u), (t', u'), (t'', u'')) \mid (\sigma(t, t', t''), \delta(u, u', u'')) \in \hat{T}_{q_{(\sigma, \delta, \theta)}}\}$$

$$\text{et } \mathcal{K}_{(\sigma, \delta, \theta)} = \{((t, u), (t', u'), (t'', u'')) \mid (\sigma(t, t', t''), \delta(u, u', u'')) \in \hat{T}_{k_{(\sigma, \delta, \theta)}}\}.$$

Considérons $\mathcal{D} = (\mathcal{Q}_{(\sigma, \delta, \theta)} - \mathcal{K}_{(\sigma, \delta, \theta)}) \cap \mathcal{K}_{(\sigma, \delta, \theta')}$ où θ' est une torsion différente de θ . On montre alors que la projection de \mathcal{D} sur l'ensemble $\{i \mid \theta(i) \neq \theta'(i)\}$, notée $\mathcal{D}_{\{i \mid \theta(i) \neq \theta'(i)\}}$, est finie. Le lemme s'énonce alors ainsi:

Lemme 2.2.6 *Soient q et k deux états d'un transducteur T linéaire et complet. Pour toute transformation initiale (σ, δ, θ) , pour toute torsion $\theta' \neq \theta$, la différence $((\mathcal{Q}_{(\sigma, \delta, \theta)} - \mathcal{K}_{(\sigma, \delta, \theta)}) \cap \mathcal{K}_{(\sigma, \delta, \theta')})_{\{i \mid \theta(i) \neq \theta'(i)\}}$ est finie.*

PREUVE:

Prenons $\theta = id_{[3]}$.

A partir de q les règles sont donc de la forme $q(\sigma(x_1, x_2, x_3)) \rightarrow \delta(q_i(x_1), q'_i(x_2), q''_i(x_3))$ avec $i \in [n]$

et à partir de k les règles sont de la forme $k(\sigma(x_1, x_2, x_3)) \rightarrow \delta(k_j(x_1), k'_j(x_2), k''_j(x_3))$ ($j \in [m]$).

Supposons qu'il existe p règles $k(\sigma(x_1, x_2, x_3)) \rightarrow \delta(l_r(x_{\theta'(1)}), l'_r(x_{\theta'(2)}), l''_r(x_{\theta'(3)}))$ ($r \in [p]$) pour la transformation initiale $(\sigma, \delta, \theta')$. Pour tout i on notera respectivement Q_i , K_i et L_i les transformations \hat{T}_{q_i} , \hat{T}_{k_i} et \hat{T}_{l_i} .

Comme dans la preuve du lemme 2.2.2, on exprime \mathcal{D} sous forme d'une union de produits cartésiens.

$$\begin{aligned} \mathcal{D} &= \left(\left(\bigcup_{i \in [n]} Q_i \times Q'_i \times Q''_i \right) - \left(\bigcup_{j \in [m]} K_j \times K'_j \times K''_j \right) \right) \cap \left(\bigcup_{r \in [p]} L_r \times L'_r \times L''_r \right) \\ &= \bigcup_{i \in [n]} \left(\bigcup_{\{J_1, J_2, J_3\}} (Q_i \cap \left(\bigcap_{\alpha \in J_1} \overline{K}_\alpha \right)) \times Q'_i \cap \left(\bigcap_{\beta \in J_2} \overline{K}'_\beta \right) \times Q''_i \cap \left(\bigcap_{\gamma \in J_3} \overline{K}''_\gamma \right) \right) \\ &\quad \cap \left(\bigcup_{r \in [p]} L_r \times L'_r \times L''_r \right). \end{aligned}$$

avec J_1, J_2 et J_3 parties de $[m]$ et $J_1 \cup J_2 \cup J_3 = [m]$.

En utilisant des propriétés élémentaires de la théorie des ensembles, on obtient:

$$\mathcal{D} = \bigcup_{i \in [n]} \left(\bigcup_{\{J_1, J_2, J_3\}} \left(\bigcup_{r \in [p]} (Q_i \cap \left(\bigcap_{\alpha \in J_1} \overline{K}_\alpha \right)) \cap L_r \times Q'_i \cap \left(\bigcap_{\beta \in J_2} \overline{K}'_\beta \right) \cap L'_r \times Q''_i \cap \left(\bigcap_{\gamma \in J_3} \overline{K}''_\gamma \right) \cap L''_r \right) \right).$$

Montrons maintenant que $\mathcal{D}_{\{i | \theta(i) \neq \theta'(i)\}}$ est finie.

Premier cas: $\text{card}(\{i | \theta(i) \neq \theta'(i)\}) = 3$

Supposons $\mathcal{D}_{\{i | \theta(i) \neq \theta'(i)\}}$ infinie.

Il existerait alors $i \in [n]$, J_1, J_2, J_3 parties de $[m]$ et $r \in [p]$ tels que

$$(Q_i \cap \left(\bigcap_{\alpha \in J_1} \overline{K}_\alpha \right)) \cap L_r \times Q'_i \cap \left(\bigcap_{\beta \in J_2} \overline{K}'_\beta \right) \cap L'_r \times Q''_i \cap \left(\bigcap_{\gamma \in J_3} \overline{K}''_\gamma \right) \cap L''_r \text{ est infini.}$$

L'un des facteurs, au moins, est donc infini. Supposons que ce soit le premier. Il existerait alors au moins un triplet $((t, u), (t', u'), (t'', u''))$ élément de \mathcal{D} et tel que $\pi(t) > \pi(t')$ et $\pi(t) > \pi(t'')$. Parce que le transducteur considéré est complet, il y a conservation des hauteurs. On a donc $\pi(t) = \pi(u)$, $\pi(t') = \pi(u')$, $\pi(t'') = \pi(u'')$ et par conséquent $\pi(u) > \pi(t')$ et $\pi(u) > \pi(t'')$. Dans le cas considéré ici, les seules valeurs possibles pour θ' sont $\langle 3; 2, 3, 1 \rangle$ et $\langle 3; 3, 1, 2 \rangle$. Dans le premier cas on devrait donc produire u à partir de t' ce qui est impossible puisque $\pi(u) > \pi(t')$ et pour la seconde torsion on devrait produire u à partir de t'' , ce qui est impossible également puisque $\pi(u) > \pi(t'')$. Cette différence est donc finie.

Second cas: $\text{card}(\{i | \theta(i) \neq \theta'(i)\}) = 2$

C'est le cas des torsions $\theta' = \langle 3; 1, 3, 2 \rangle$, $\theta' = \langle 3; 3, 2, 1 \rangle$ et $\theta' = \langle 3; 2, 1, 3 \rangle$.

On retrouve alors ici la situation du cas des lettres de rang 2. \square

Cette extension du lemme 2.2.2 nous permet ensuite, par la construction de formes semi-normalisées qui généralisent celles construites dans les paragraphes précédents, de rendre vides ces différences $((\mathcal{Q}_{(\sigma,\delta,\theta)} - \mathcal{K}_{(\sigma,\delta,\theta)}) \cap \mathcal{K}_{(\sigma,\delta,\theta')})_{\{i|\theta(i) \neq \theta'(i)\}}$ et donc d'obtenir des transformations avec les mêmes torsions pour tous les couples d'arbres.

Chapitre 3

Transducteurs descendants linéaires

Nous proposons dans ce chapitre une première extension du résultat de décidabilité de l'équivalence, obtenu dans le chapitre précédent, au cas des transducteurs descendants lettre à lettre linéaires mais qui maintenant ne sont plus complets. Ces transducteurs se caractérisent par le fait que lors des transformations certains sous-arbres d'un noeud (voire tous les sous-arbres) peuvent être supprimés avant même d'avoir été reconnus puisque le mouvement est ici descendant.

On note **LT-LL** la classe des transducteurs descendants lettre à lettre linéaires.

Nous adopterons la même démarche que dans le chapitre précédent, à savoir:

1. étudier, du point de vue des torsions, les transformations réalisées à partir d'états équivalents
2. construire des formes normalisées qui nous permettent d'utiliser des résultats existants de décidabilité. Ici encore, ces formes normalisées seront des marqueurs.

La figure 3.1 illustre la démarche. Celle-ci est analogue à celle utilisée dans le chapitre 2 en lui ajoutant cependant une phase de "complétion" des transducteurs.

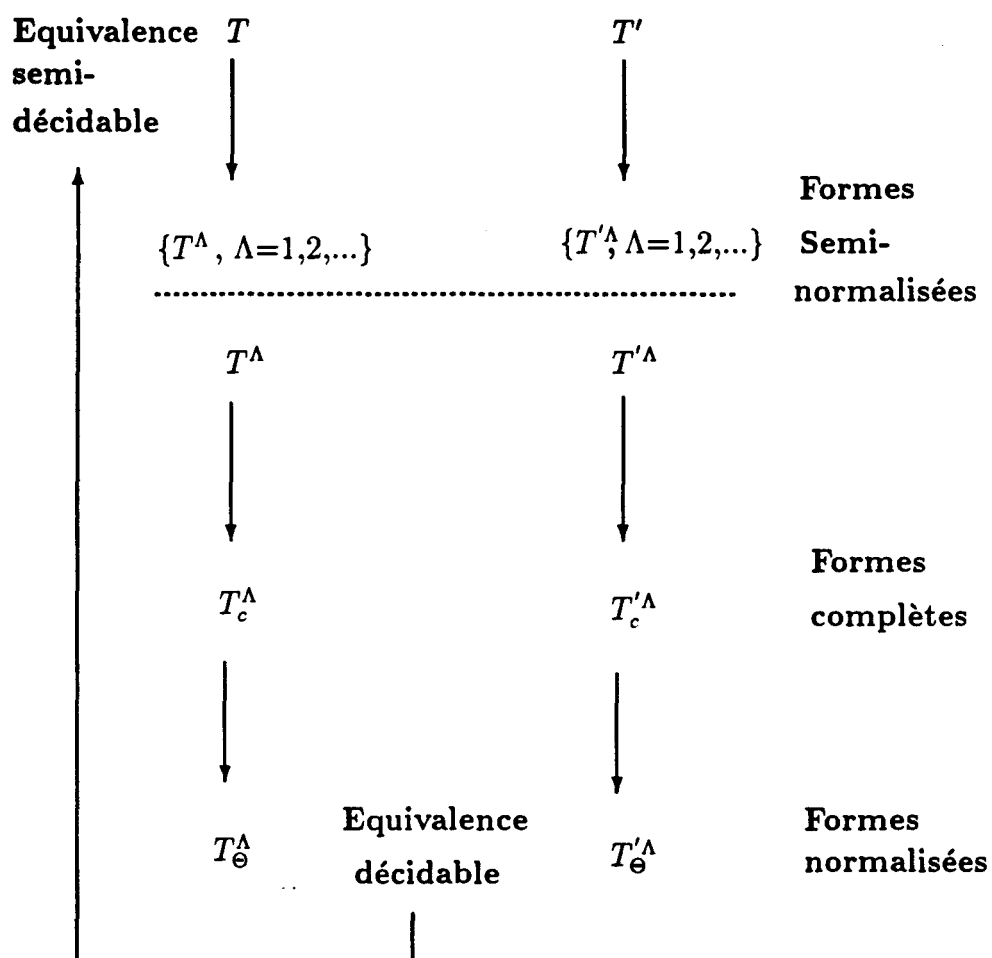


FIG. 3.1 - : Démarche dans le cas linéaire

Nous ne donnerons donc, dans ce chapitre, que les constructions et propriétés qui sont spécifiques au cas des transducteurs non complets en ne considérant, dans un premier temps, que des transducteurs infinitaires (sous-classe **LT-LL₁**).

3.1 Transformations initiales

Considérons le transducteur non-complet $T = \langle \Sigma, \Delta, Q, I, R \rangle$ avec $\Sigma_0 = \{\bar{a}\}$, $\Sigma_1 = \{a, \alpha\}$ et $\Sigma_2 = \{\sigma\}$, $\Delta_0 = \{\bar{a}, e, f\}$, $\Delta_2 = \{\delta\}$ et les règles:

$$\begin{array}{ll} q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_2), q_2(x_1)) & k(\sigma(x_1, x_2)) \rightarrow \delta(k_1(x_1), k_2(x_2)) \\ q_1(\alpha(x)) \rightarrow e & k_1(a(x)) \rightarrow e \\ q_2(a(x)) \rightarrow f & k_2(\alpha(x)) \rightarrow f \end{array}$$

Les transformations réalisées à partir des états q et k sont égales:

$$\hat{T}_q = \hat{T}_k = \{(\sigma(a(\tau), \alpha(\tau')), \delta(e, f)) \mid \tau, \tau' \in T(\Sigma)\}.$$

Et pourtant des torsions différentes sont appliquées initialement à tous les arbres: à partir de l'état q une permutation des sous-arbres de la racine est effectuée alors qu'à partir de k la transformation est sans-torsion.

A la différence de ce qui a été montré dans le cas des transducteurs linéaires et complets, nous avons ici une infinité de couples d'arbres pour lesquels les transformations initiales se font nécessairement avec des torsions différentes. Nous montrons par le lemme ci-dessous que cet ensemble de couples est d'image finie.

Lemme 3.1.1 *Soient q et k deux états équivalents d'un transducteur T de **LT-LL₁**. Pour tout couple $(\sigma, \delta) \in \Sigma_2 \times \Delta_p$ avec $p \leq 2$, pour toute torsion θ de $[p]$ dans $[2]$, la différence $\hat{T}_{q_\theta} - \hat{T}_{k_\theta}$ est d'image finie.*

PREUVE:

Nous distinguons trois cas.

Premier cas: σ et δ sont de même rang.

Les règles sont alors de la forme $q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$.

Dans le but de simplifier les notations, nous prendrons $\theta = id_{[2]}$ et la seule autre

torsion possible sera $\mu = \langle 2; 2, 1 \rangle$.

Considérons la différence $\hat{T}_{q_\theta} - \hat{T}_{k_\theta}$.

Supposons qu'il existe,

à partir de l'état q , n règles de la forme $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_1), q'_i(x_2))$ avec $i \in [n]$ et

à partir de l'état k , m règles de la forme $k(\sigma(x_1, x_2)) \rightarrow \delta(k_j(x_1), k'_j(x_2))$ avec $j \in [m]$.

On désigne par \mathcal{Q} l'ensemble $\{((u, v), (u', v')) \in \hat{T}_{q_i} \times \hat{T}_{q'_i}, i \in [n]\}$ et par \mathcal{K} l'ensemble $\{((u, v), (u', v')) \in \hat{T}_{k_j} \times \hat{T}_{k'_j}, j \in [m]\}$.

Ainsi, $((u, v), (u', v')) \in \mathcal{Q} - \mathcal{K} \Leftrightarrow (\sigma(u, u'), \delta(v, v')) \in \hat{T}_{q_\theta} - \hat{T}_{k_\theta}$.

\bar{T}_q désigne le complémentaire de \hat{T}_q dans $T_\Sigma \times T_\Delta$.

Des propriétés élémentaires de la théorie des ensembles nous permettent d'exprimer $\mathcal{Q} - \mathcal{K}$ en fonction de \hat{T}_{q_i} , $\hat{T}_{q'_i}$, \hat{T}_{k_j} , $\hat{T}_{k'_j}$.

$$\begin{aligned} \mathcal{Q} - \mathcal{K} &= \left(\bigcup_{i \in [n]} \hat{T}_{q_i} \times \hat{T}_{q'_i} \right) - \left(\bigcup_{j \in [m]} \hat{T}_{k_j} \times \hat{T}_{k'_j} \right) \\ &= \bigcup_{i \in [n]} \left(\bigcup_{J \subseteq [m]} (\hat{T}_{q_i} \cap (\bigcap_{j \in J} (\bar{T}_{k_j}))) \times (\hat{T}_{q'_i} \cap (\bigcap_{l \in \bar{J}} (\bar{T}_{k'_l}))) \right). \end{aligned}$$

Supposons que $\mathcal{Q} - \mathcal{K}$ ne soit pas d'image finie (ce qui est possible dans le cas de transducteurs infinitaires).

Il existerait alors au moins $i \in [n]$ et $J \subseteq [m]$ tels que $\hat{T}_{q_i} \cap (\bigcap_{j \in J} (\bar{T}_{k_j}))$ ou $\hat{T}_{q'_i} \cap (\bigcap_{l \in \bar{J}} (\bar{T}_{k'_l}))$ ne soit pas d'image finie.

Prenons, par exemple, le cas où $\hat{T}_{q_i} \cap (\bigcap_{j \in J} (\bar{T}_{k_j}))$ n'est pas d'image finie. Pour tout couple (u', v') de $\hat{T}_{q'_i} \cap (\bigcap_{l \in \bar{J}} (\bar{T}_{k'_l}))$, il existerait un couple (u, v) de $\hat{T}_{q_i} \cap (\bigcap_{j \in J} (\bar{T}_{k_j}))$ tel que $\pi(v) > \pi(u')$.

C'est-à-dire $(\sigma(u, u'), \delta(v, v')) \in \hat{T}_{q_\theta} - \hat{T}_{k_\theta}$ avec $\pi(v) > \pi(u')$. Ce couple ne pouvant pas être élément de \hat{T}_{k_μ} (propriété 1.4.1), on arrive à une contradiction avec l'hypothèse d'équivalence de q et k .

Par conséquent, $\mathcal{Q} - \mathcal{K}$ est d'image finie.

Second cas: δ est de rang 1.

Considérons, par exemple, le cas où $\theta = \langle 2; 1 \rangle$.

Soit $(\sigma(t, t'), \delta(u)) \in \hat{T}_{q_\theta} - \hat{T}_{k_\theta}$.

Alors, pour tout arbre $\tau \in T_\Sigma$, $(\sigma(t, \tau), \delta(u)) \in \hat{T}_{q_\theta} - \hat{T}_{k_\theta}$.

Les états q et k étant équivalents, il existerait au moins une autre torsion μ (ici $\mu = \langle 2; 2 \rangle$) telle que $(\sigma(t, \tau), \delta(u)) \in \hat{T}_{k_\mu}$ pour tout τ , et en particulier dans le cas $\tau \in \Sigma_0$. Donc $u \in \Delta_0$.

Par conséquent, pour tout $(\sigma(t, t'), \delta(u)) \in \hat{T}_{q_\theta} - \hat{T}_{k_\theta}$, $\pi(u) = 0$.

Troisième cas: les règles sont de la forme $q(\sigma(x_1, x_2)) \rightarrow f$ où f est une lettre de rang 0. Dans ce cas, la différence $\hat{T}_{q_\theta} - \hat{T}_{k_\theta}$ est évidemment vide. \square

Remarque

Ces résultats, obtenus pour des états équivalents, s'étendent naturellement au cas d'ensembles d'états globalement équivalents.

3.2 Forme Λ -normalisée d'un transducteur infinitaire

Comme dans le chapitre 2, pour tout Λ on associe à tout transducteur T de **LT-LL**_i une forme Λ -normalisée construite en deux étapes.

La forme Λ -semi-normalisée de T , obtenue par la première construction, vise le même objectif à savoir avoir un traitement global de tous les couples d'arbres ou de sous-arbres dont la profondeur est au plus Λ .

Mais cette fois, à cause de la non-complétude, c'est la profondeur de l'arbre obtenu en sortie qui sera prise en compte.

Ainsi, pour tout couple (t, u) de \hat{T} , avec $\pi(u) \leq \Lambda$, t sera transformé en u en un seul pas dans T^Λ ; les arbres t et u étant identifiés à de nouvelles lettres, éléments de nouveaux alphabets notés Σ^Λ et Δ^Λ , et t pouvant être un terme non clos.

De même, nous montrons que si Λ est suffisamment grand alors, dans la forme Λ -semi-normalisée, à partir d'ensembles d'états globalement équivalents, les transformations peuvent être réalisées avec les mêmes torsions pour tous les arbres (lemme 3.2.2).

Dans la seconde phase, après avoir rendu complète la forme semi-normalisée, nous construisons la forme Λ -normalisée dans laquelle les torsions sont exprimées dans l'image des arbres mais non appliquées effectivement. Cette forme Λ -normalisée est un marqueur.

Forme Λ -semi-normalisée

Au transducteur $T = \langle \Sigma, \Delta, Q, I, R \rangle$, de **LT-LL_i**, nous associons le transducteur $T^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R^\Lambda \rangle$, pour lequel Σ^Λ et Δ^Λ sont de nouveaux alphabets dont les lettres peuvent être interprétées respectivement comme des arbres ou des préfixes d'arbres de $\text{dom}(\hat{T})$ et des arbres de $\text{im}(\hat{T})$ de profondeur inférieure ou égale à Λ , et construit de la façon suivante:

- $q^{<\Lambda}$ et q^Λ sont des états de Q^Λ si et seulement si q est un état de Q et ils sont éléments de I^Λ si et seulement si q est dans I .
- $q^{<\Lambda}(t) \rightarrow u$ (resp. $q^\Lambda(t) \rightarrow u$) est une règle de R^Λ , t est une lettre de Σ^Λ et u est une lettre de Δ^Λ si et seulement si $(t, u) \in \hat{T}_q$ avec $t \in T_\Sigma$ et $\pi(t) = \pi(u) < \Lambda$ (resp. $\pi(t) = \pi(u) = \Lambda$).
- $q^{<\Lambda}(t_0(x_1, \dots, x_n)) \rightarrow u$ (resp. $q^\Lambda(t_0(x_1, \dots, x_n)) \rightarrow u$) est une règle de R^Λ , t_0 est une lettre de rang n de Σ^Λ et u est une lettre de Δ^Λ si et seulement si $q(t_0(x_1, \dots, x_n)) \xrightarrow{*} u$ dans T avec $\pi(t_0) = \pi(u) < \Lambda$ (resp. $\pi(t_0) = \pi(u) = \Lambda$).
- $q^\Lambda(\sigma(x)) \rightarrow \delta(q_i^\Lambda(x))$ est une règle de R^Λ si et seulement si $q(\sigma(x)) \rightarrow \delta(q_i(x))$ est une règle de R .
- $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_{\theta(1)}), q_j^\Lambda(x_{\theta(2)}))$,
 $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^{<\Lambda}(x_{\theta(1)}), q_j^\Lambda(x_{\theta(2)}))$ et
 $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_{\theta(1)}), q_j^{<\Lambda}(x_{\theta(2)}))$ sont des règles de R^Λ si et seulement si
 $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_{\theta(1)}), q_j(x_{\theta(2)}))$ est une règle de R .
- $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_{\theta(1)}))$ est une règle de R^Λ si et seulement si $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_{\theta(1)}))$ est une règle de R .

La similitude des transformations réalisées dans le transducteur d'origine et dans toute forme Λ -semi-normalisée, nous conduit naturellement à les identifier.

Identification de T et T^Λ

• Soit $q(t) \xrightarrow{*} u$ un mouvement de dérivation dans T , avec $\pi(u) \leq \Lambda$.

Premier cas: $\pi(t) = \pi(u)$.

Nous avons alors dans T^Λ une règle de la forme $q^{<\Lambda}(t) \rightarrow u$ si $\pi(t) < \Lambda$, ou de la forme $q^\Lambda(t) \rightarrow u$ si $\pi(t) = \Lambda$.

Second cas: $\pi(t) > \pi(u)$

Il existe une décomposition unique de t en $t_0(t_1, \dots, t_n)$ telle que $\pi(t_0) = \pi(u)$ et, pour tout n-uple d'arbres τ_1, \dots, τ_n , $q(t_0(\tau_1, \dots, \tau_n)) \xrightarrow{*} u$.
On a alors, dans T^Λ

- une règle de la forme $q^{<\Lambda}(t_0(x_1, \dots, x_n)) \rightarrow u$ si $\pi(u) < \Lambda$,
- ou de la forme $q^\Lambda(t_0(x_1, \dots, x_n)) \rightarrow u$ si $\pi(u) = \Lambda$

et ainsi $q^{<\Lambda}(t) \mapsto u$ ou $q^\Lambda(t) \mapsto u$.

En fait, ici nous confondons les arbres de profondeur inférieure ou égale à Λ , tels que t_0 ou u , avec de nouvelles lettres. Le contrôle des profondeurs d'arbres assuré par les états ($q^{<\Lambda}$ ou q^Λ) assurent que c'est le seul mouvement de dérivation possible pour (t, u) dans T^Λ .

• Pour tout couple d'arbres (t, u) de \hat{T} avec $\pi(u) > \Lambda$, il existe une décomposition unique $u = u_0(u_1, \dots, u_n)$ et $t = t_0(t_1, \dots, t_m)$ ($m \geq n$) telle que

- pour tout $i \in [n]$, $\pi(u_i) \leq \Lambda$
- il n'existe aucun sous-arbre de u , de profondeur inférieure ou égale à Λ , dont u_i est un sous-arbre propre
- et $q(t_0(x_1, \dots, x_m)) \xrightarrow{*} u_0(q_1(x_{\theta(1)}), \dots, q_n(x_{\theta(n)}))$ dans T .

Les mouvements de dérivation

- $q(t_0(t_1, \dots, t_m)) \xrightarrow{*} u_0(q_1(t_{\theta(1)}), \dots, q_n(t_{\theta(n)}))$ dans T
- et $q^\Lambda(t_0(t_1, \dots, t_m)) \xrightarrow{*} u_0(q'_1(t_{\theta(1)}), \dots, q'_n(t_{\theta(n)}))$ dans T^Λ (pour tout i de $[n]$, q'_i est soit $q_i^{<\Lambda}$ si $\pi(t_{\theta(i)}) < \Lambda$, soit q_i^Λ si $\pi(t_{\theta(i)}) = \Lambda$)

sont analogues, en ce sens qu'ils ne diffèrent que par les états rencontrés.

Par conséquent, pour tout Λ nous identifions T et T^Λ .

Dans la suite, pour tout arbre t de $T_{\Sigma \cup \Sigma^\Lambda}$, $\pi(t)$ désignera la profondeur de "l'arbre correspondant" de T_Σ .

De la même façon que dans le cas de transducteurs linéaires et complets (chapitre 2), on prouve successivement les résultats suivants:

Lemme 3.2.1 *Lorsque Λ est suffisamment grand, à partir de deux ensembles d'états globalement équivalents de T^Λ les transformations initiales peuvent être réalisées avec les mêmes torsions pour tous les arbres.*

PREUVE:

La preuve est analogue à celle du lemme 2.2.3.

Soient E et F deux ensembles d'états globalement équivalents de T^Λ et soit (t, u) un couple d'arbres de $\widehat{T}_E^\Lambda (= \widehat{T}_F^\Lambda)$.

On prend comme valeur pour Λ un majorant de l'ensemble des entiers $\lambda_{(E,F),(\sigma,\delta,\theta)}$ obtenus pour tous les couples (E, F) d'ensembles d'états globalement équivalents et pour toutes les transformations initiales (σ, δ, θ) .

Dans le cas où $\pi(u) > \Lambda$, le lemme 3.1.1 et la construction de T^Λ nous permettent d'affirmer que les mêmes torsions initiales peuvent être appliquées.

Dans le cas $\pi(u) \leq \Lambda$, u est en fait une lettre de Δ^Λ et t est alors soit une lettre de Σ^Λ , soit un arbre de la forme $t_0(t_1, \dots, t_n)$ avec $t_0 \in \Sigma^\Lambda$. Dans ces deux cas, il existe au moins un état $q_i \in E$ et un état $k_j \in F$ à partir desquels t est réécrit en u en un seul pas. \square

Lemme 3.2.2 *Lorsque Λ est suffisamment grand, à partir de deux ensembles d'états globalement équivalents de T^Λ les transformations peuvent être réalisées avec les mêmes torsions pour tous les arbres.*

PREUVE:

La preuve est analogue à celle du lemme 2.2.4. \square

Forme Λ -normalisée

A partir de T^Λ nous construisons le marqueur T_Θ^Λ en deux étapes.

Puisque, pour les formes Λ -semi-normalisées (avec Λ suffisamment grand), les mêmes torsions peuvent être appliquées à tous les arbres à partir d'états équivalents, les mêmes sous-arbres seront supprimés. De plus, lors d'une transformation descendante, les sous-arbres supprimés ne sont pas préalablement reconnus. On peut par conséquent, tout en préservant l'équivalence des états, répercuter dans les arbres d'origine ces suppressions. On obtient ainsi un transducteur complet. Son alphabet d'entrée est noté Σ_c^Λ . Chaque lettre σ^- de cet alphabet est obtenue à partir d'une lettre σ de $\Sigma \cup \Sigma^\Lambda$ et son arité (notée $\rho(\sigma^-)$) est inférieure ou égale à celle de σ . On aura $\rho(\sigma^-) < \rho(\sigma)$ dans le cas où σ est transformée par une règle non-complète (la différence $\rho(\sigma) - \rho(\sigma^-)$ étant égale au nombre de branches supprimées) et $\sigma^- = \sigma$ dans le cas contraire. La seconde phase, de construction d'un marqueur, est identique à celle décrite dans le chapitre 2.

• **Première phase: complétion de T^Λ**

A partir de $T^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R^\Lambda \rangle$

construction de $T_c^\Lambda = \langle \Sigma_c^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R_c^\Lambda \rangle$ de la façon suivante:

- $q^{<\Lambda}(t_{1,\dots,n}^-) \rightarrow u$ (resp. $q^\Lambda(t_{1,\dots,n}^-) \rightarrow u$) est une règle de R_c^Λ et $t_{1,\dots,n}^-$ est une lettre de rang 0 de Σ_c^Λ si et seulement si $q^{<\Lambda}(t(x_1, \dots, x_n)) \rightarrow u$ (resp. $q^\Lambda(t(x_1, \dots, x_n)) \rightarrow u$) est une règle de R^Λ .
- $q^\Lambda(\sigma_2^-(x)) \rightarrow \delta(q_i^\Lambda(x))$ (resp. $q^\Lambda(\sigma_1^-(x)) \rightarrow \delta(q_i^\Lambda(x))$) est une règle de R_c^Λ et σ_2^- (resp. σ_1^-) est une lettre de rang 1 de Σ_c^Λ si et seulement si $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_1))$ (resp. $q^\Lambda(\sigma(x_1, x_2)) \rightarrow \delta(q_i^\Lambda(x_2))$) est une règle de R^Λ .
- Chaque règle complète (c'est à dire une règle pour laquelle la torsion en partie droite est une bijection) de R^Λ est une règle de R_c^Λ et la lettre transformée par cette règle est une lettre de Σ_c^Λ .

• **Seconde phase: construction du marqueur**

A partir de $T_c^\Lambda = \langle \Sigma_c^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R_c^\Lambda \rangle$,

construction du marqueur $T_\Theta^\Lambda = \langle \Sigma_c^\Lambda, \Delta_\Theta, Q^\Lambda, I^\Lambda, R_\Theta \rangle$.

La construction, identique à celle réalisée précédemment pour les transducteurs linéaires et complets, peut être trouvée dans le chapitre 2.

Exemple

Reprenons les transducteurs T et T' définis dans le paragraphe 3.1 et construisons les formes Λ -normalisées pour $\Lambda = 2$.

Formes semi-normalisées

$$\begin{aligned} \mathbf{T}^2 : q^\Lambda(\sigma(x_1, x_2)) &\rightarrow \delta(q_1^\Lambda(x_2), q_2^\Lambda(x_1)) \\ q^\Lambda(\sigma(x_1, x_2)) &\rightarrow \delta(q_1^{<\Lambda}(x_2), q_2^\Lambda(x_1)) \\ q^\Lambda(\sigma(x_1, x_2)) &\rightarrow \delta(q_1^\Lambda(x_2), q_2^{<\Lambda}(x_1)) \\ q_1^{<\Lambda}(\alpha(x)) &\rightarrow e \\ q_2^{<\Lambda}(a(x)) &\rightarrow f \end{aligned}$$

$$q^{<\Lambda}(\sigma(a(x_1), \alpha(x_2))) \rightarrow \delta(e, f)$$

$$\begin{aligned} \mathbf{T}'^2 : k^\Lambda(\sigma(x_1, x_2)) &\rightarrow \delta(k_1^\Lambda(x_1), k_2^\Lambda(x_2)) \\ k^\Lambda(\sigma(x_1, x_2)) &\rightarrow \delta(k_1^{<\Lambda}(x_1), k_2^\Lambda(x_2)) \\ k^\Lambda(\sigma(x_1, x_2)) &\rightarrow \delta(k_1^\Lambda(x_1), k_2^{<\Lambda}(x_2)) \\ k_1^{<\Lambda}(a(x)) &\rightarrow e \\ k_2^{<\Lambda}(\alpha(x)) &\rightarrow f \end{aligned}$$

$$k^{<\Lambda}(\sigma(a(x_1), \alpha(x_2))) \rightarrow \delta(e, f)$$

Formes normalisées

$$\begin{aligned} \mathbf{T}_\Theta^2 : q^\Lambda(\sigma(x_1, x_2)) &\rightarrow \langle \delta, \mu \rangle (q_2^\Lambda(x_1), q_1^\Lambda(x_2)) \\ q^\Lambda(\sigma(x_1, x_2)) &\rightarrow \langle \delta, \mu \rangle (q_2^\Lambda(x_1), q_1^{<\Lambda}(x_2)) \\ q^\Lambda(\sigma(x_1, x_2)) &\rightarrow \langle \delta, \mu \rangle (q_2^{<\Lambda}(x_1), q_1^\Lambda(x_2)) \\ q_1^{<\Lambda}(\alpha_1^-) &\rightarrow \langle e, id \rangle \\ q_2^{<\Lambda}(a_1^-) &\rightarrow \langle f, id \rangle \end{aligned}$$

$$q^{<\Lambda}(\sigma(a_1^-, \alpha_1^-)) \rightarrow \langle \delta(e, f), id \rangle$$

$$\begin{aligned} \mathbf{T}'_\Theta^2 : k^\Lambda(\sigma(x_1, x_2)) &\rightarrow \langle \delta, id \rangle (k_1^\Lambda(x_1), k_2^\Lambda(x_2)) \\ k^\Lambda(\sigma(x_1, x_2)) &\rightarrow \langle \delta, id \rangle (k_1^{<\Lambda}(x_1), k_2^\Lambda(x_2)) \\ k^\Lambda(\sigma(x_1, x_2)) &\rightarrow \langle \delta, id \rangle (k_1^\Lambda(x_1), k_2^{<\Lambda}(x_2)) \end{aligned}$$

$$k_1^{<\Lambda}(a_1^-) \rightarrow \langle e, id \rangle$$

$$k_2^{<\Lambda}(\alpha_1^-) \rightarrow \langle f, id \rangle$$

$$k^{<\Lambda}(\sigma(a_1^-, \alpha_1^-)) \rightarrow \langle \delta(e, f), id \rangle$$

Dans ces formes normalisées id et μ désignent respectivement l'identité et la torsion $\langle 2; 2, 1 \rangle$.

On remarquera que les 3 premières règles non-closes de T_{Θ}^2 et $T_{\Theta}'^2$ génèrent initialement des couples de lettres différentes mais que tout mouvement de dérivation s'arrête là puisqu'il n'existe aucune règle à partir des états q_1^{Λ} , q_2^{Λ} , k_1^{Λ} et k_2^{Λ} .

Les transformations \hat{T}_{Θ}^2 et $\hat{T}_{\Theta}'^2$ sont en fait complètement réalisées par les règles

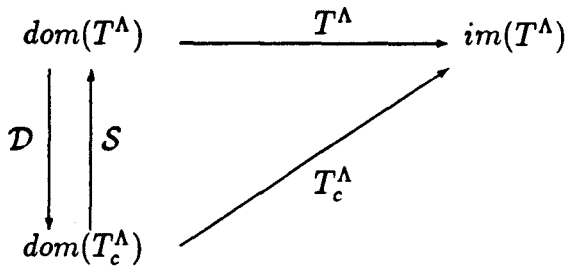
$$q^{<\Lambda}(\sigma(a_1^-, \alpha_1^-)) \rightarrow \langle \delta(e, f), id \rangle$$

$$\text{et } k^{<\Lambda}(\sigma(a_1^-, \alpha_1^-)) \rightarrow \langle \delta(e, f), id \rangle.$$

Lemme 3.2.3 Soient E et F deux ensembles d'états globalement équivalents de la forme Λ -semi-normalisée T^Λ d'un transducteur T et soit T_{Θ}^Λ sa forme Λ -normalisée. Lorsque Λ est suffisamment grand, $\hat{T}_E^\Lambda = \hat{T}_F^\Lambda$ si et seulement si $\hat{T}_{\Theta_E}^\Lambda = \hat{T}_{\Theta_F}^\Lambda$.

ELEMENTS DE PREUVE:

Entre T^Λ et T_c^Λ existent les relations décrites dans le diagramme ci-dessous.



où \mathcal{D} est le transducteur lettre à lettre, construit à partir de T^Λ , et modifiant $\text{dom}(\hat{T}^\Lambda)$ de telle façon que T_c^Λ soit complet:

Regles de T^Λ	Regles de \mathcal{D}
$q(\sigma) \rightarrow \delta$	$q(\sigma) \rightarrow \sigma$
$q(\sigma(x_1)) \rightarrow \delta(q_1(x_1))$	$q(\sigma(x_1)) \rightarrow \sigma(q_1(x_1))$
$q(\sigma(x_1)) \rightarrow \delta$	$q(\sigma(x_1)) \rightarrow \sigma_1^-$
$q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_1), q_2(x_2))$	$q(\sigma(x_1, x_2)) \rightarrow \sigma(q_1(x_1), q_2(x_2))$
$q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_2), q_2(x_1))$	$q(\sigma(x_1, x_2)) \rightarrow \sigma(q_2(x_1), q_1(x_2))$
$q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_1))$	$q(\sigma(x_1, x_2)) \rightarrow \sigma_2^-(q_1(x_1))$
$q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_2))$	$q(\sigma(x_1, x_2)) \rightarrow \sigma_1^-(q_1(x_2))$
$q(\sigma(x_1, x_2)) \rightarrow \delta$	$q(\sigma(x_1, x_2)) \rightarrow \sigma_{1,2}^-$

Pour tout couple (t, t') de $\hat{\mathcal{D}}$, on a $(t, u) \in \hat{T}^\Lambda$ si et seulement si $(t', u) \in \hat{T}_c^\Lambda$.

\mathcal{S} est la substitution réalisant la transformation inverse.

Ainsi, si Λ est suffisamment grand, on obtient $\hat{T}_E^\Lambda = \hat{T}_F^\Lambda$ si et seulement si $\hat{T}_{c_E}^\Lambda = \hat{T}_{c_F}^\Lambda$.

Dès que Λ est suffisamment grand, à partir d'ensembles d'états globalement équivalents du transducteur linéaire et complet T_c^Λ les mêmes torsions peuvent être appliquées pour tous les arbres. Il est par conséquent évident que l'équivalence des ensembles d'états est préservée lorsque les torsions ne sont plus appliquées mais sont codées dans les noeuds des arbres produits en sortie. Ainsi, pour une valeur suffisamment grande de Λ , on obtient $\hat{T}_{c_E}^\Lambda = \hat{T}_{c_F}^\Lambda$ si et seulement si $\hat{T}_{\Theta_E}^\Lambda = \hat{T}_{\Theta_F}^\Lambda$. \square

3.3 Décidabilité de l'équivalence

Les résultats obtenus dans le paragraphe précédent s'étendent au cas général des transducteurs non nécessairement infinitaires. Il suffit pour cela de choisir Λ de telle façon qu'il soit de plus supérieur au nombre d'états finitaires du transducteur.

Lemme 3.3.1 *Soient T et T' deux transducteurs de LT-LL et soient T_{Θ}^{Λ} et T'_{Θ}^{Λ} leurs formes Λ -normalisées. Lorsque Λ est suffisamment grand, T et T' sont équivalents si et seulement si T_{Θ}^{Λ} et T'_{Θ}^{Λ} sont équivalents.*

PREUVE:

La preuve se fait en appliquant le lemme 3.2.3 aux ensembles d'états initiaux des transducteurs considérés. \square

Comme dans le chapitre précédent, on obtient:

Théorème 3.3.1 *L'équivalence des transducteurs descendants lettre à lettre linéaires est décidable.*

PREUVE:

Parce que l'équivalence des marqueurs est décidable, le lemme 3.3.1 établit la semi-décidabilité de l'équivalence dans LT-LL. La non équivalence étant semi décidable, nous concluons. \square

Chapitre 4

Automates avec contraintes d'équivalence

La non-linéarité est un phénomène important qui apparaît souvent dans les arbres (par exemple en programmation logique, dans les systèmes de réécriture ...) et l'ensemble des instances closes d'un terme non-linéaire n'est pas une forêt reconnaissable. Pour les reconnaisseurs "classiques" d'arbres, les conditions d'application d'une règle sont constituées uniquement par la lettre et les états apparaissant en dessous de cette lettre (pour un automate ascendant). Cette notion a été étendue par l'ajout de conditions portant sur les sous-arbres du noeud considéré. Ces conditions peuvent prendre différentes formes.

Une première classe de tels automates a été introduite par J. Mongy [Mon81] au début des années 80. Pour ces reconnaisseurs ascendants à tests, appelés **RATEG** (Reconnaisseurs Ascendants à Tests d'EGalité), chaque règle est dotée d'une condition d'application sous la forme d'un arbre fini, non clos en général. La règle ne peut s'appliquer que si l'arbre considéré a la forme de "l'arbre-condition".

Reprenons l'exemple d'automate RATEG donné dans [Mon81] et défini par:

$$\Sigma_0 = \{\bar{a}\}, \Sigma_1 = \{a\} \text{ et } \Sigma_2 = \{b\}$$

$$Q = \{q_0, q_1\} \text{ et } F = \{q_1\}$$

R composé des règles suivantes:

$$(r1) \quad \bar{a} \rightarrow q_0$$

$$(r2) \quad a(q_0) \rightarrow q_0$$

$$(r3) \quad b(q_0, q_0) \rightarrow q_1$$

$$(r4) \quad b(q_0, q_1) \rightarrow q_1 \text{ avec la condition } b(a(x_1), b(x_1, x_2))$$

L'arbre-condition de la dernière règle est non-linéaire et cette règle ne pourra s'appliquer qu'en cas d'égalité du sous-arbre situé sous le noeud a et du sous-arbre gauche du noeud b .

Par exemple l'arbre $b(a(a(a^p(\bar{a}))), b(a(a^p(\bar{a})), b(a^p(\bar{a})), a^q(\bar{a})))$ est reconnu par cet automate.

On remarque immédiatement que, même si les comparaisons ne se font qu'à des profondeurs bornées (puisque les arbres conditions sont finis), il est impossible de borner la distance entre des sous-arbres liés par la condition d'égalité. Le nombre de " a " du premier sous-arbre de gauche est ici lié au nombre de " a " apparaissant en bas de l'arbre.

Cette propagation des contraintes est à l'origine de la non-décidabilité du vide.

Une autre voie a consisté à ne comparer que les sous-termes directs de la lettre courante. B. Bogaert et S. Tison ont ainsi défini une nouvelle classe d'automates, notée REC_{\neq} , et permettant de tester l'égalité ou la non-égalité de sous-termes directs ([Bog90],[BT92]).

Considérons l'exemple suivant (de [Bog90]):

Soit $A = \langle \Sigma, Q, F, R \rangle$ l'automate de REC_{\neq} défini par:

$$\Sigma_0 = \{\bar{a}\}, \Sigma_1 = \{a\} \text{ et } \Sigma_2 = \{b\}$$

$$Q = \{q, q_f\} \text{ et } F = \{q_f\}$$

R composé des règles suivantes:

$$(r1) \quad \bar{a} \rightarrow q$$

$$(r2) \quad a(q) \rightarrow q$$

$$(r3) \quad a(q_f) \rightarrow q_f$$

$$(r4) \quad b(q, q)[x_1 \neq x_2] \rightarrow q_f$$

$$(r5) \quad b(q, q)[x_1 = x_2] \rightarrow q$$

$$(r6) \quad b(q, q_f)[x_1 \neq x_2] \rightarrow q_f$$

$$(r7) \quad b(q, q_f)[x_1 = x_2] \rightarrow q_f$$

$$(r8) \quad b(q_f, q)[x_1 \neq x_2] \rightarrow q_f$$

$$(r9) \quad b(q_f, q)[x_1 = x_2] \rightarrow q_f$$

$$(r10) \quad b(q_f, q_f)[x_1 \neq x_2] \rightarrow q_f$$

$$(r11) \quad b(q_f, q_f)[x_1 = x_2] \rightarrow q_f$$

La règle (r4) ne s'appliquera qu'en cas de sous-arbres distincts, (r5) étant appliquée pour des sous-arbres identiques.

La forêt reconnue par cet automate est composée des arbres déséquilibrés sur l'alphabet Σ .

Cette nouvelle classe possède les propriétés suivantes:

Propriété 4.0.1 *La classe REC_{\neq} est close par union, intersection et complémentaire.*

Propriété 4.0.2 *Soit A un automate de REC_{\neq} . La propriété " $\mathcal{F}(A) = \emptyset$ " est décidable.*

Dans le cas où on autorise des tests à profondeur 1 (entre fils) et 2 (entre cousins germains) la propriété du vide est indécidable [Tom91].

Plus récemment D. Lugiez et J.L. Moysset ont étendu la classe précédente à une classe d'automates avec égalités entre fils modulo l'associativité et la commutativité [LM93].

Dans ce chapitre, nous étendons les travaux de B. Bogaert et S. Tison ([Bog90], [BT92]) au cas de classes d'automates où les contraintes entre sous-termes directs d'un noeud sont des contraintes d'équivalence ou de non-équivalence.

Les règles ont une forme analogue à celle des règles des automates de REC_{\neq} ; on substitue des contraintes d'équivalence (ou de non-équivalence) aux contraintes d'égalité (ou de non-égalité).

Par exemple, si $\sigma(q_1(x_1), q_2(x_2))[x_1 \equiv x_2] \rightarrow q$ est une règle d'un tel automate alors $\sigma(t_1, t_2) \xrightarrow{*} q$ si et seulement si:

- $t_1 \xrightarrow{*} q_1, t_2 \xrightarrow{*} q_2$
- et t_1 et t_2 sont des arbres équivalents.

Dans ce chapitre, nous montrons que la classe ainsi définie est close par union, intersection et complémentaire. De plus, lorsque la relation d'équivalence est induite par un système de réécriture lettre à lettre confluent et noethérien, la propriété du vide est décidable.

On obtient ainsi une classe d'automates pour laquelle l'équivalence est décidable. Ce résultat est utilisé pour établir la décidabilité de l'équivalence pour les transducteurs descendants lettre à lettre et complets (chapitre 5).

4.1 Automates avec contraintes d'équivalence

Les définitions et propriétés, que nous énonçons dans ce paragraphe pour les automates avec contraintes d'équivalence entre sous-termes directs, sont inspirées de celles données dans [Bog90] et [BT92] pour les automates de la classe \mathbf{REC}_{\neq} .

4.1.1 Définitions

Soit \equiv une relation d'équivalence sur T_{Σ} .

Définition 4.1.1 Une description d'équivalence sur n éléments est une partition de l'ensemble $[n]$. EqD_n désignera l'ensemble de toutes les descriptions d'équivalence sur $[n]$.

Définition 4.1.2 Soit $d \in EqD_n$. Un n -uplet de termes $(t_i)_{i \in [n]}$ satisfait la description d'équivalence d si et seulement si:

- pour tout $X \in d$, pour tout i et $j \in X$, $t_i \equiv t_j$
- et pour tout X et Y de d avec $X \neq Y$, $i \in X$, $j \in Y$ implique $t_i \not\equiv t_j$.

Définition 4.1.3 Un automate ascendant avec tests d'équivalence entre sous-termes directs (ou fils) est un 4-uplet $\langle \Sigma, Q, F, R \rangle$ où

- Σ est un alphabet gradué
- Q est un ensemble fini d'états
- F est le sous-ensemble de Q des états terminaux
- R est un ensemble fini de règles; $R \subseteq \bigcup_i (\Sigma_i \times EqD_i \times Q^{i+1})$.

Notation

Les règles sont généralement notées $\sigma(q_1, \dots, q_n)[d] \rightarrow q$ où d est une description d'équivalence sur $[n]$.

Dans le cas des lettres d'arité 0 ou 1, la description d est omise.

De la même façon, lorsque pour une lettre σ d'arité n ($n \geq 2$) et pour un n -uplet d'états q_1, \dots, q_n toutes les descriptions d'équivalence conduisent au même état de sortie, on omettra ces descriptions et on n'écrira ainsi qu'une seule règle.

Fonctionnement des automates

Un arbre t est reconnu par un automate A , à tests d'équivalence entre fils, si et seulement si il existe un état final q tel que $t \xrightarrow{*}_A q$.

$\xrightarrow{*}_A$ désigne la clôture réflexive et transitive de \mapsto_A définie par:

$t \mapsto t'$ si et seulement si
 $\exists u \in T_\Sigma(X_1), \exists \sigma \in \Sigma_n$ et $\exists t_1, \dots, t_n \in T_\Sigma$,
 $\exists q$ et q_1, \dots, q_n éléments de Q ,
 $\exists \sigma(q_1, \dots, q_n)[d] \rightarrow q \in R$ tels que
 $t = u(\sigma(q_1(t_1), \dots, q_n(t_n)))$, $t' = u(q(\sigma(t_1, \dots, t_n)))$
 et $(t_i)_{i \in [n]}$ satisfait d .

Définition 4.1.4 Pour tout état q , $\mathcal{F}_q(A) = \{t \in T(\Sigma) \mid t \xrightarrow{*}_A q(t)\}$.

Ainsi la forêt reconnue par A est $\mathcal{F}(A) = \bigcup_{q \in F} \mathcal{F}_q(A)$.

Classes définies

Pour toute relation d'équivalence \equiv , \mathbf{REC}_\equiv désigne la classe des automates à tests d'équivalence entre fils (pour cette équivalence).

Remarque

La classe \mathbf{REC}_\neq est en fait un cas particulier de classe \mathbf{REC}_\equiv pour laquelle deux termes sont équivalents s'ils sont égaux.

4.1.2 Exemple

Σ est l'alphabet défini par $\Sigma_0 = \{\bar{a}\}$, $\Sigma_1 = \{a, \alpha\}$, et $\Sigma_2 = \{\sigma, \delta\}$.

On considère la relation d'équivalence sur T_Σ , notée \equiv , et définie par $t \equiv t'$ si et seulement si t et t' ont le "même squelette".

$A = \langle \Sigma, Q, F, R \rangle$ est l'automate de \mathbf{REC}_\neq définie par:

$$Q = \{q, \text{puits}\}, F = \{q\}$$

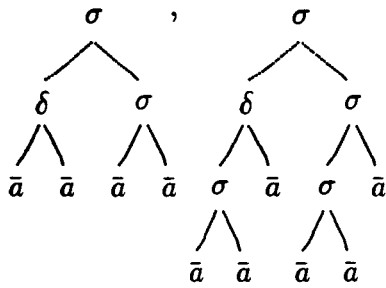
et R constitué des règles suivantes:

- (r1) $\sigma(q, q)[1, 2] \rightarrow q$
- (r2) $\sigma(q, q)[1], [2] \rightarrow \text{puits}$
- (r3) $\delta(q, q)[1, 2] \rightarrow q$
- (r4) $\delta(q, q)[1], [2] \rightarrow \text{puits}$
- (r5) $a(q) \rightarrow q$
- (r6) $\alpha(q) \rightarrow q$
- (r7) $\bar{a} \rightarrow q$

Les règles (r1) et (r3) assurent à chaque noeud des arbres reconnus l'équivalence des sous-termes.

La forêt reconnue est ainsi constituée des arbres équilibrés de T_Σ .

Considérons, par exemple, les deux arbres:



Le premier est reconnu par l'automate A alors que le second ne l'est pas car les sous-arbres σ et \bar{a} n'ont pas le même squelette.



4.1.3 Propriétés

Définition 4.1.5 Un automate A est déterministe si et seulement si pour toute lettre σ de Σ_n , pour tout n -uple d'états q_1, \dots, q_n et pour toute description d'équivalence d de EqD_n , il existe au plus une règle de la forme $\sigma(q_1, \dots, q_n)[d] \rightarrow q$.

Définition 4.1.6 Un automate A est complètement spécifié si et seulement si pour toute lettre σ de Σ_n , pour tout n -uple d'états q_1, \dots, q_n et pour toute description d'équivalence d de EqD_n , il existe au moins une règle de la forme $\sigma(q_1, \dots, q_n)[d] \rightarrow q$.

En utilisant la méthode décrite par B. Bogaert et S. Tison dans [BT92], nous pouvons associer à tout automate de REC_{\neq} un automate déterministe et complètement spécifié reconnaissant le même langage.

Spécification complète

Lemme 4.1.1 *Soit $A = \langle \Sigma, Q, F, R \rangle$ un automate de REC_{\neq} . On peut construire un automate $A' = \langle \Sigma, Q', F, R' \rangle$ complètement spécifié tel que $\mathcal{F}(A') = \mathcal{F}(A)$.*

PREUVE:

L'algorithme est similaire à celui utilisé pour les automates classiques d'arbres.

On ajoute un nouvel état noté "puits": $Q' = Q \cup \{\text{puits}\}$.

De plus, pour toute lettre σ de Σ_n , pour tout n-uple d'états q_1, \dots, q_n , pour toute description d'équivalence d de EqD_n s'il n'existe aucune règle de R de membre gauche $\sigma(q_1, \dots, q_n)[d]$ alors on ajoute la règle $\sigma(q_1, \dots, q_n)[d] \rightarrow \text{puits}$. On obtient ainsi R' .

On a alors, pour tout état $q \neq \text{puits}$, $t \xrightarrow{*}_{A'} q \Leftrightarrow t \xrightarrow{*}_A q$. \square

Algorithme de déterminisation

Lemme 4.1.2 Soit $A = \langle \Sigma, Q, F, R \rangle$ un automate complètement spécifié de REC_{\neq} . On peut construire un automate $A' = \langle \Sigma, Q', F, R' \rangle$ déterministe tel que $\mathcal{F}(A') = \mathcal{F}(A)$.

PREUVE:

Algorithme

Debut

$Q_0 \leftarrow \{X \mid X = \{q \mid \exists \bar{a} \in \Sigma_0, (\bar{a} \rightarrow q) \in R\}$

$R_0 \leftarrow \{\bar{a} \rightarrow X \mid X = \{q \mid (a \rightarrow q) \in R\}$

Repeter

$\bar{Q}_i \leftarrow \bar{Q}_{i-1} \cup$

$\{X \mid \exists \sigma \in \Sigma_n, \exists (X_i)_{i \in [n]} \text{ dans } \bar{Q}_{i-1}, \exists d \in EqD_n,$

$X = \{q \mid \exists (q_i)_{i \in [n]} \text{ dans } Q, q_k \in X_k, (\sigma(q_1, \dots, q_n)[d] \rightarrow q) \in R\}$

$\bar{R}_i \leftarrow \bar{R}_{i-1} \cup$

$\{(\sigma(X_1, \dots, X_n)[d] \rightarrow X) \mid$

$\sigma \in \Sigma_n, (X_i)_{i \in [n]} \text{ dans } \bar{Q}_{i-1}, d \in EqD_n$

$X = \{q \mid \exists (q_i)_{i \in [n]} \text{ dans } Q, q_k \in X_k, (\sigma(q_1, \dots, q_n)[d] \rightarrow q) \in R\}$

Jusqu'a ce que $\bar{Q}_i = \bar{Q}_{i-1}$

$\bar{Q} \leftarrow \bar{Q}_i$

$\bar{R} \leftarrow \bar{R}_i$

$\bar{F} \leftarrow \{X \mid X \cap F \neq \emptyset\}$.

Fin

Par récurrence sur la profondeur des termes on vérifie que:

$t \xrightarrow{*}_{A'} X \Leftrightarrow X = \{q \mid t \xrightarrow{*}_A q\}$. \square

Propriétés de clôture

Proposition 4.1.1 La classe REC_{\neq} est close par union et intersection.

PREUVE:

La preuve se fait par un produit d'automates.

Soient A et B deux automates de REC_{\neq} et A' et B' les automates déterministes et complètement spécifiés reconnaissant respectivement $\mathcal{F}(A)$ et $\mathcal{F}(B)$.

Soit $A' = \langle \Sigma, Q_A, F_A, R_A \rangle$ et $B' = \langle \Sigma, Q_B, F_B, R_B \rangle$.

Les automates "produits" $A' \cap B'$ et $A' \cup B'$ sont définis par:

$A' \cap B' = \langle \Sigma, Q_{\cap}, F_{\cap}, R_{\cap} \rangle$ et $A' \cup B' = \langle \Sigma, Q_{\cup}, F_{\cup}, R_{\cup} \rangle$ avec

$$- Q_{\cap} = Q_{\cup} = Q \times Q'$$

$$- F_\cap = F \times F', F_\cup = (Q \times F') \cup (F \times Q')$$

$$- R_\cap = R_\cup = \{\sigma((q_1, q'_1), \dots, (q_n, q'_n))[d] \rightarrow (q, q') \text{ tel que } \sigma(q_1, \dots, q_n)[d] \rightarrow q \in R \text{ et } \sigma(q'_1, \dots, q'_n)[d] \rightarrow q' \in R'\}.$$

Pour tout arbre t , $t \mapsto (q, q')$ dans l'automate "produit" si et seulement si $t \mapsto_{A'} q$ et $t \mapsto_{B'} q'$.

Le même résultat s'obtient pour la clôture transitive et réflexive de \mapsto .

Les automates $A' \cap B'$ et $A' \cup B'$ ainsi construits sont donc des automates de la classe REC_{\neq} reconnaissant respectivement $\mathcal{F}(A) \cap \mathcal{F}(B)$ et $\mathcal{F}(A) \cup \mathcal{F}(B)$. \square

Proposition 4.1.2 *La classe REC_{\neq} est close par complémentaire.*

PREUVE:

Soit A un automate de REC_{\neq} et A' l'automate déterministe et complètement spécifié reconnaissant la même forêt.

Si $A' = \langle \Sigma, Q_A, F_A, R_A \rangle$ alors l'automate $C = \langle \Sigma, Q_A, Q_A - F_A, R_A \rangle$ reconnaît le complémentaire. \square

4.2 Classes REC_τ

Nous nous intéressons maintenant aux classes d'automates à tests d'équivalence entre sous-termes directs pour lesquelles la relation d'équivalence est induite par un système de réécriture lettre à lettre, complet, noethérien et confluent.

Si τ est un tel système de réécriture alors deux arbres sont équivalents si et seulement si leurs formes irréductibles, obtenues par des réécritures utilisant les règles de τ , sont égales. Pour tout système de réécriture τ (ayant les propriétés ci-dessus), la classe de tous les automates à contraintes d'équivalence ainsi définie est notée REC_τ .

Nous montrons d'abord que, pour de tels systèmes de réécriture, une forme normalisée, appelée forme *1-réduite*, peut être obtenue (paragraphe 4.2.1). Cette forme nous permet par une stratégie de réécriture ascendante d'obtenir la forme irréductible de tout terme en ne réécrivant qu'une seule fois chaque noeud.

En utilisant la forme 1-réduite d'un système de réécriture donné τ , nous associons à tout automate A de REC_τ l'automate A_τ de REC_{\neq} (paragraphe 4.2.3) qui reconnaît l'ensemble des formes irréductibles des arbres de $\mathcal{F}(A)$. Cela nous permet de décider si la forêt reconnue par A est vide.

Les classes REC_τ ayant, de plus, de bonnes propriétés de clôture par les opérations

booléennes (propriétés 4.1.1 et 4.1.2), on en déduit que l'équivalence dans REC_τ est décidable.

4.2.1 Systèmes de réécriture lettre à lettre

Propriétés des systèmes de réécriture lettre à lettre noethériens et confluents

Lemme 4.2.1 *Soit \mathcal{S} un système de réécriture non-dégénéré, lettre à lettre, noethérien et confluent.*

Pour tout couple (σ, θ) , où σ est une lettre et θ est une application de $[n]$ dans $[n]$ (n étant le rang de σ , $n \leq 2$), il existe un couple unique $(\tilde{\sigma}, \mu)$ tel que:

- μ est une application de $[m]$ dans $\theta([n])$ (m est le rang de $\tilde{\sigma}$, $m \leq 2$),
- $\sigma(x_{\theta(1)}, \dots, x_{\theta(n)}) \xrightarrow{*}_{\mathcal{S}} \tilde{\sigma}(x_{\mu(1)}, \dots, x_{\mu(m)})$
- et $\tilde{\sigma}(x_{\mu(1)}, \dots, x_{\mu(m)})$ est un terme irréductible.

PREUVE:

Soit σ , $\tilde{\sigma}'$ et $\tilde{\sigma}''$ des lettres de rang 2 et soit $\theta = id_{[2]}$ (les autres cas se traitent de façon similaire).

Supposons que, pour le couple (σ, θ) , il existe deux couples $(\tilde{\sigma}', \mu')$ et $(\tilde{\sigma}'', \mu'')$ tels que:

- $\tilde{\sigma}' \neq \tilde{\sigma}''$ ou $\mu' \neq \mu''$
- $\sigma(x_1, x_2) \xrightarrow{*}_{\mathcal{S}} \tilde{\sigma}'(x_{\mu'(1)}, x_{\mu'(2)})$ et $\sigma(x_1, x_2) \xrightarrow{*}_{\mathcal{S}} \tilde{\sigma}''(x_{\mu''(1)}, x_{\mu''(2)})$,
- les termes $\tilde{\sigma}'(x_{\mu'(1)}, x_{\mu'(2)})$ et $\tilde{\sigma}''(x_{\mu''(1)}, x_{\mu''(2)})$ étant irréductibles.

Considérons un couple de termes (t_1, t_2) tel que $\hat{t}_1 \neq \hat{t}_2$ (dans ce cas, nous avons évidemment $t_1 \neq t_2$ parce que \mathcal{S} est confluent).

On aurait alors

$$\begin{aligned} t &= \sigma(t_1, t_2) \xrightarrow{*}_{\mathcal{S}} \tilde{\sigma}'(\hat{t}_{\mu'(1)}, \hat{t}_{\mu'(2)}) = \hat{t}' \text{ et} \\ t &= \sigma(t_1, t_2) \xrightarrow{*}_{\mathcal{S}} \tilde{\sigma}''(\hat{t}_{\mu''(1)}, \hat{t}_{\mu''(2)}) = \hat{t}'' \text{ avec } \hat{t}' \text{ et } \hat{t}'' \text{ termes irréductibles.} \end{aligned}$$

Or, $\hat{t}' \neq \hat{t}''$ parce que nous avons supposé $\tilde{\sigma}' \neq \tilde{\sigma}''$ ou $\hat{t}_{\mu'(i)} \neq \hat{t}_{\mu''(i)}$ ($i \in [2]$), ce qui est en contradiction avec l'hypothèse de confluence de \mathcal{S} et la propriété est donc établie. \square

Lemme 4.2.2 *Soit \mathcal{S} un système de réécriture non-dégénéré, lettre à lettre, noethérien et confluent.*

Pour toute lettre σ de rang 2, si $\sigma(x_1, x_2) \xrightarrow{}_{\mathcal{S}} \tilde{\sigma}'(x_{\mu(1)}, x_{\mu(2)})$ et $\sigma(x, x) \xrightarrow{*}_{\mathcal{S}} \tilde{\sigma}''(x, x)$ (resp. $\tilde{\sigma}''(x)$) avec:*

- $\tilde{\sigma}' \neq \tilde{\sigma}''$ et $\mu(1) \neq \mu(2)$,

- $\tilde{\sigma}'(x_{\mu(1)}, x_{\mu(2)})$ et $\tilde{\sigma}''(x, x)$ (resp. $\tilde{\sigma}''(x)$) termes irréductibles
alors il existe un mouvement $\tilde{\sigma}'(x, x) \xrightarrow{*}_S \tilde{\sigma}''(x, x)$ (resp. $\tilde{\sigma}''(x)$).

PREUVE:

Le contraire contredirait le fait que \mathcal{S} est un système de réécriture confluent. \square

Système de réécriture 1-réduit

Définition 4.2.1 Un système de réécriture lettre à lettre noethérien et confluent est 1-réduit si et seulement si pour tout terme $t = \sigma(t_1, \dots, t_n)$ sa forme irréductible $\hat{t} = \tilde{\sigma}(\hat{t}_{\theta(1)}, \dots, \hat{t}_{\theta(m)})$ peut être obtenue de la façon suivante:

1. pour tout $i \in [n]$, $t_i \xrightarrow{*} \hat{t}_i$
2. $\sigma(\hat{t}_1, \dots, \hat{t}_n) \rightarrow \tilde{\sigma}(\hat{t}_{\theta(1)}, \dots, \hat{t}_{\theta(m)})$ avec θ application de $[m]$ dans $[n]$
ou alors $\hat{t} = \sigma(\hat{t}_1, \dots, \hat{t}_n)^1$.

Construction d'un système 1-réduit.

Nous associons à tout système de réécriture lettre à lettre noethérien et confluent \mathcal{S} sa forme 1-réduite, notée \mathcal{S}_{1r} , et construite de la façon suivante:

- Pour toute lettre σ de rang 0,
 $\sigma \rightarrow \tilde{\sigma}$ est une règle de \mathcal{S}_{1r} si et seulement si $\sigma \xrightarrow{*}_S \tilde{\sigma}$ et $\tilde{\sigma}$ est un terme irréductible.
- Pour toute lettre σ de rang 1,
 $\sigma(x) \rightarrow \tilde{\sigma}(x)$ (resp. $\tilde{\sigma}(x, x)$) est une règle de \mathcal{S}_{1r}
si et seulement si $\sigma(x) \xrightarrow{*}_S \tilde{\sigma}(x)$ (resp. $\tilde{\sigma}(x, x)$) et $\tilde{\sigma}(x)$ (resp. $\tilde{\sigma}(x, x)$) est un terme irréductible.
- Pour toute lettre σ de rang 2,
 $\sigma(x_{\theta(1)}, x_{\theta(2)}) \rightarrow \tilde{\sigma}(x_{\mu(1)}, x_{\mu(2)})$ (resp. $\tilde{\sigma}(x_{\mu(1)})$) est une règle \mathcal{S}_{1r} si et seulement si $\sigma(x_{\theta(1)}, x_{\theta(2)}) \xrightarrow{*}_S \tilde{\sigma}(x_{\mu(1)}, x_{\mu(2)})$ (resp. $\tilde{\sigma}(x_{\mu(1)})$) et $\tilde{\sigma}(x_{\mu(1)}, x_{\mu(2)})$ (resp. $\tilde{\sigma}(x_{\mu(1)})$) est un terme irréductible.
- De plus, $\tilde{\sigma}'(x, x) \rightarrow \tilde{\sigma}''(x, x)$ (resp. $\tilde{\sigma}''(x)$) est une règle de \mathcal{S}_{1r} si et seulement si $\sigma(x_1, x_2) \rightarrow \tilde{\sigma}'(x_{\theta(1)}, x_{\theta(2)})$ (avec $\theta \neq Id_{[2]}$) et $\sigma(x, x) \rightarrow \tilde{\sigma}''(x, x)$ (resp. $\tilde{\sigma}''(x)$) sont des règles de \mathcal{S}_{1r} .

¹On ne peut avoir de règle de la forme $\sigma(x, y) \rightarrow \sigma(y, x)$ puisque le système considéré est noethérien

Il est clair que \mathcal{S}_{1r} est un système de réécriture lettre à lettre noethérien et confluent. Nous vérifions maintenant que \mathcal{S}_{1r} réalise bien la même transformation que \mathcal{S} .

Lemme 4.2.3 *Soit \mathcal{S} un système de réécriture lettre à lettre noethérien et confluent et \mathcal{S}_{1r} sa forme 1-réduite. Pour tout terme t , $t \xrightarrow{\ast_{\mathcal{S}_{1r}}} \hat{t}$ si et seulement si $t \xrightarrow{\ast_{\mathcal{S}}} \hat{t}$.*

PREUVE:

La preuve se fait par récurrence sur la profondeur des termes.

La preuve étant immédiate dans le cas d'un système dégénéré puisqu'il existe alors une lettre α unique telle que toute lettre de Σ est réécrite en α , on ne considèrera que le cas d'un système de réécriture non-dégénéré.

La propriété est évidemment vraie pour tout terme de profondeur 0.

Supposons-la vraie pour des termes jusqu'à la profondeur p .

Soit $t = \sigma(t_1, t_2)$ un terme de profondeur $p + 1$ et soit $\hat{t} = \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$ sa forme irréductible (on procéderait de la même façon dans les autres cas, c'est à dire $\hat{t} = \tilde{\sigma}(\hat{t}_2, \hat{t}_1)$ ou alors σ et $\tilde{\sigma}$ de rangs différents).

La propriété étant supposée vraie jusqu'à la profondeur p ,

$(t_1 \xrightarrow{\ast_{\mathcal{S}}} \hat{t}_1$ et $t_2 \xrightarrow{\ast_{\mathcal{S}}} \hat{t}_2)$ si et seulement si $(t_1 \xrightarrow{\ast_{\mathcal{S}_{1r}}} \hat{t}_1$ et $t_2 \xrightarrow{\ast_{\mathcal{S}_{1r}}} \hat{t}_2)$.

Premier cas: Il n'existe aucune règle de partie gauche $\sigma(x, x)$ dans \mathcal{S} ou $\hat{t}_1 \neq \hat{t}_2$.

On a alors $t = \sigma(t_1, t_2) \xrightarrow{\ast_{\mathcal{S}}} \hat{t} = \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$ si et seulement si il existe un mouvement $\sigma(x, y) \xrightarrow{\ast_{\mathcal{S}}} \tilde{\sigma}(x, y)$ et $t_1 \xrightarrow{\ast_{\mathcal{S}}} \hat{t}_1$, $t_2 \xrightarrow{\ast_{\mathcal{S}}} \hat{t}_2$.

Par construction, $\sigma(x, y) \rightarrow \tilde{\sigma}(x, y)$ est alors une règle de \mathcal{S}_{1r} , et donc $\sigma(t_1, t_2) \xrightarrow{\ast_{\mathcal{S}_{1r}}} \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$.

Second cas: Il existe dans \mathcal{S} une règle de partie gauche $\sigma(x, x)$ et $\hat{t}_1 = \hat{t}_2$.

Considérons les dérivations $\sigma(x, y) \xrightarrow{\ast_{\mathcal{S}}} \tilde{\sigma}'(x, y)$ et $\sigma(x, x) \xrightarrow{\ast_{\mathcal{S}}} \tilde{\sigma}(x, x)$ dans \mathcal{S} , avec $\tilde{\sigma}'(x, y)$ et $\tilde{\sigma}(x, x)$ termes irréductibles.

Si $\tilde{\sigma}' \neq \tilde{\sigma}$ alors, parce que \mathcal{S} est un système de réécriture confluent, il existe un mouvement $\tilde{\sigma}'(x, x) \xrightarrow{\ast_{\mathcal{S}}} \tilde{\sigma}(x, x)$ (lemme 4.2.2).

Par construction, $\sigma(x, y) \rightarrow \tilde{\sigma}'(x, y)$, $\sigma(x, x) \rightarrow \tilde{\sigma}(x, x)$ et $\tilde{\sigma}'(x, x) \rightarrow \tilde{\sigma}(x, x)$ sont des règles de \mathcal{S}_{1r} (si $\tilde{\sigma}' \neq \tilde{\sigma}$).

Par conséquent, quelle que soit la séquence de réécritures, on obtient $\sigma(t_1, t_2) \xrightarrow{\ast_{\mathcal{S}_{1r}}} \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$.

Réciproquement, en utilisant la même méthode, on montrerait que, pour ces deux cas, $t \xrightarrow{\ast_{\mathcal{S}_{1r}}} \hat{t}$ implique $t \xrightarrow{\ast_{\mathcal{S}}} \hat{t}$. \square

Remarque

Le système de réécriture $S_{1,\tau}$ est bien 1-réduit.

En effet,

- pour tout terme, chaque noeud, de rang 0 ou de rang 1, est réécrit en sa forme irréductible en 1 pas.
- quelle que soit la séquence de réécritures, nous avons au plus 2 pas pour la transformation des noeuds de rang 2.

Considérons, par exemple $t = \sigma(t_1, t_2)$ et $\hat{t} = \bar{\sigma}(\hat{t}_1, \hat{t}_2)$.

Lorsque $\hat{t}_1 = \hat{t}_2$, si on réécrit d'abord la racine de t , on peut avoir la séquence de réécritures suivante:

$$\sigma(t_1, t_2) \rightarrow_{S_{1,\tau}} \bar{\sigma}'(t_1, t_2) \xrightarrow{*}_{S_{1,\tau}} \bar{\sigma}'(\hat{t}_1, \hat{t}_2) \rightarrow_{S_{1,\tau}} \bar{\sigma}(\hat{t}_1, \hat{t}_2).$$

Donc, en choisissant une stratégie de réécriture ascendante, chaque noeud peut être réécrit en sa forme irréductible en un seul pas.

4.2.2 Classes REC_τ

Soit τ un système de réécriture lettre à lettre, noethérien, confluent et complet.

On désigne par REC_τ la classe de tous les automates à tests d'équivalence entre fils où la relation d'équivalence \equiv est définie par:

$$t \equiv t' \text{ si et seulement si } \tau(t) = \tau(t'),$$

$\tau(t)$ désignant la forme irréductible de t obtenue par des réécritures utilisant les règles de τ .

De plus, pour tout système de réécriture lettre à lettre noethérien et confluent τ , il existe une forme 1-réduite τ' telle que $t \xrightarrow{*}_\tau \hat{t} \Leftrightarrow t \xrightarrow{*}_{\tau'} \hat{t}$ (lemme 4.2.3). On a alors $REC_\tau = REC_{\tau'}$.

Par conséquent, dans la suite, on ne considèrera que des systèmes 1-réduits.

Comme cela a été montré dans le cas général (classes REC_{\neq}), quel que soit τ la classe REC_τ est close par les opérations d'union, d'intersection et de complémentaire.

Il nous reste donc à montrer que la "propriété du vide" y est décidable pour obtenir une classe d'automates dans laquelle l'équivalence est décidable.

La méthode choisie consiste à associer à tout automate A de REC_τ un automate, noté A_τ , reconnaissant les formes normales de tous les arbres reconnus par A . L'automate A_τ doit donc vérifier $IRR(\mathcal{F}(A)) = \mathcal{F}(A_\tau)$.

Le système de réécriture τ nous permet de réduire les équivalences d'arbres à des égalités. L'automate des formes normales sera ainsi un automate à contrainte d'égalités entre fils. Ainsi en utilisant les bonnes propriétés de décidabilité de la classe \mathbf{REC}_\neq (propriété 4.0.2) nous montrons que la "propriété du vide" est décidable dans \mathbf{REC}_τ .

Le paragraphe suivant contient l'algorithme de construction de l'automate des formes normales associé à un automate de \mathbf{REC}_τ ainsi que quelques exemples.

4.2.3 Automate des formes normales associé à un automate de \mathbf{REC}_τ

Construction.

Soit τ un système de réécriture 1-réduit défini sur un alphabet gradué Σ_τ .

A tout automate $A = \langle \Sigma, Q, F, R \rangle$ de \mathbf{REC}_τ , où $\Sigma \subseteq \Sigma_\tau$, on associe l'automate $A_\tau = \langle \Sigma_\tau, Q_\tau, F, R_\tau \rangle$ de \mathbf{REC}_\neq construit en deux étapes.

Première phase: Construction de $Q_0 \subseteq Q_\tau$ et $R_0 \subseteq R_\tau$.

Les tableaux suivants décrivent pour chaque type de lettre, la règle de R_0 obtenue à partir d'une règle de τ (ligne) et d'une règle de R (colonne).

$\text{arite}(\sigma) = 0$	$\sigma \rightarrow q$	$\text{arite}(\sigma) = 1$	$\sigma(q_i) \rightarrow q_j$
$\sigma \rightarrow \tilde{\sigma}$	$\tilde{\sigma} \rightarrow q$	$\sigma(x) \rightarrow \tilde{\sigma}(x)$	$\tilde{\sigma}(q_i) \rightarrow q_j$
<i>pas de règle</i>	$\sigma \rightarrow q$	$\sigma(x) \rightarrow \tilde{\sigma}(x, x)$	$\tilde{\sigma}(q_i, q_i)[1, 2] \rightarrow q_j$
		<i>pas de règle</i>	$\sigma(q_i) \rightarrow q_j$

$\text{arite}(\sigma) = 2$	$\sigma(q_i, q_j)[1, 2] \rightarrow q$	$\sigma(q_i, q_j)[1, 2] \rightarrow q$
$\sigma(x, y) \rightarrow \tilde{\sigma}(x, y)$	$\tilde{\sigma}(q_i, q_j)[1, 2] \rightarrow q$	$\tilde{\sigma}(q_i, q_j)[1, 2] \rightarrow q$
$\sigma(x, y) \rightarrow \tilde{\sigma}(y, x)$	$\tilde{\sigma}(q_j, q_i)[1, 2] \rightarrow q$	$\tilde{\sigma}(q_j, q_i)[1, 2] \rightarrow q$
$\sigma(x, x) \rightarrow \tilde{\sigma}(x, x)$	\emptyset	$\tilde{\sigma}(q_i, q_j)[1, 2] \rightarrow q$
$\sigma(x, x) \rightarrow \tilde{\sigma}(x)$	\emptyset	$\tilde{\sigma}(\{q_i, q_j\}) \rightarrow q, \{q_i, q_j\} \in Q_0$
<i>pas de règle</i>	$\sigma(q_i, q_j)[1, 2] \rightarrow q$	$\sigma(q_i, q_j)[1, 2] \rightarrow q$

Dans le tableau ci-dessus, $\{q_i, q_j\}$ désigne une paire d'états et Q_0 est en fait l'ensemble des paires d'états obtenues dans le cas de règles linéaires à droite mais non-linéaires à gauche.

Remarque 1

On peut observer qu'un grand nombre de règles de A et de A_τ ont la même forme, mais la contrainte d qui est interprétée dans A comme une contrainte d'équivalence sera interprétée dans A_τ comme une contrainte d'égalité.

Remarque 2

Nous avons obtenu, en partie gauche de certaines règles, de nouveaux états qui sont des ensembles d'états de l'automate A mais, parce qu'il n'existe aucune règle avec de tels états en partie droite, ces nouveaux états ne sont pas accessibles.

La seconde phase a donc pour but de construire le plus petit ensemble de règles R_τ qui contient R_0 , avec ces nouveaux états en partie droite de règles et tel que la propriété suivante soit vérifiée:

Propriété 4.2.1 *Pour tout état $\{q_1, \dots, q_p\}$ de Q_τ , pour toute lettre σ de Σ_τ :*

- si $\sigma \rightarrow \{q_1, \dots, q_p\}$ est une règle de R_τ alors $\forall i \in [p]$, $\sigma \rightarrow q_i$ est une règle de R_0 .
- si $\sigma(E) \rightarrow F$ est une règle de R_τ alors $\forall q_i \in F$, $\exists E_i \subseteq E$ tel que $\sigma(E_i) \rightarrow q_i$ est une règle de R_0 .
- si $\sigma(X, Y)[d] \rightarrow F$ est une règle de R_τ alors $\forall q_i \in F$, $\exists k_i \in X$, $\exists k'_i \in Y$ tel que $\sigma(k_i, k'_i)[d] \rightarrow q_i$ est une règle de R_0 .

Seconde phase: Algorithme de complétion de Q_0 et R_0 .

Debut

$R_\tau \leftarrow R_0$

$I \leftarrow 0$;

Repeter

$I \leftarrow I + 1$; $Q_I \leftarrow Q_{I-1}$;

Pour tout p-uple d'états $\{q_1, \dots, q_p\}$ de Q_{I-1} Faire

Pour toute lettre σ de Σ_τ Faire

Cas σ de rang 0:

Si $\forall i \in [p], \sigma \rightarrow q_i \in R_\tau$ Alors $R_\tau \leftarrow R_\tau \cup \{\sigma \rightarrow \{q_1, \dots, q_p\}\}$ Fin si.

Cas σ de rang 1:

Soit $E = \bigcup_{i \in [p]} E_i$ avec $E_i \in Q_{i-1}$ et tel que

pour tout $i \in [p], \sigma(E_i) \rightarrow q_i$ est une règle de R_τ alors

$R_\tau \leftarrow R_\tau \cup \{\sigma(E) \rightarrow \{q_1, \dots, q_p\}\}$,

$Q_I \leftarrow Q_I \cup \{E\}$.

Cas σ de rang 2:

Soit $E' = \bigcup_{i \in [p]} \{k'_i\}$ avec $k'_i \in Q$, $E'' = \bigcup_{i \in [p]} \{k''_i\}$ ou $k''_i \in Q$ et tel que

pour tout $i \in [p], \sigma(k'_i, k''_i)[d] \rightarrow q_i$ est une règle de R_τ alors

$R_\tau \leftarrow R_\tau \cup \{\sigma(E', E'')[d] \rightarrow \{q_1, \dots, q_p\}\}$,

$Q_I \leftarrow Q_I \cup \{E', E''\}$.

Fin pour

Fin pour

Jusqu'à $Q_I = Q_{I-1}$

$Q_\tau \leftarrow Q \cup Q_I$

Fin

Si N est le nombre d'états de l'automate A alors le nombre d'états de l'automate A_τ est au plus $2^N - 1$ (les nouveaux états générés étant des parties de l'ensemble des états de A). Ce maximum peut être atteint. C'est le cas par exemple si on considère le système de réécriture réduit à la règle $\sigma(x, x) \rightarrow \tilde{\sigma}(x)$ et l'automate A défini par $Q = \{q_1, q_2, q_3\}$, les règles $\sigma(q_i, q_j)[1, 2] \rightarrow q_k$ pour tout $i, j, k \in [3]$ avec $i \neq j, i \neq k, j \neq k$ et les règles $a \rightarrow q_i$ pour tout $i \in [3]$.

Exemple 1

Reprenons l'automate A défini dans le paragraphe 4.1.2.

La relation d'équivalence \equiv , "avoir le même squelette", peut être associée au système de réécriture lettre à lettre, noethérien, confluent et complet τ suivant:

Forme 1-réduite de τ

$$\sigma(x, y) \rightarrow \tilde{\sigma}(x, y)$$

$$\delta(x, y) \rightarrow \tilde{\delta}(x, y)$$

$$a(x) \rightarrow \tilde{a}(x)$$

$$\alpha(x) \rightarrow \tilde{\alpha}(x)$$

Ce système de réécriture, en identifiant d'une part les noeuds d'arité 2 que sont σ et δ , et d'autre part les noeuds d'arité 1 que sont a et α , réduit les équivalences à des égalités.

Automate des formes normales associé à A

$$\tilde{\sigma}(q, q)[1, 2] \rightarrow q$$

$$\tilde{\sigma}(q, q)[1], [2] \rightarrow \text{puits}$$

$$\tilde{\alpha}(q) \rightarrow q$$

$$\bar{a} \rightarrow q.$$

Par exemple, on a

$$t_1 = \begin{array}{c} \sigma \\ \swarrow \quad \searrow \\ \delta \quad \alpha \\ \swarrow \quad \downarrow \quad \swarrow \\ \bar{a} \quad a \quad \bar{a} \\ \downarrow \\ \bar{a} \end{array} \xrightarrow{*}_\tau \hat{t}_1 = \begin{array}{c} \tilde{\sigma} \\ \swarrow \quad \searrow \\ \tilde{\sigma} \quad \tilde{\alpha} \\ \swarrow \quad \downarrow \quad \swarrow \\ \bar{a} \quad \tilde{\alpha} \quad \bar{a} \\ \downarrow \\ \bar{a} \end{array} \text{ et } t_2 = \begin{array}{c} \delta \\ \swarrow \quad \searrow \\ \sigma \quad a \\ \swarrow \quad \downarrow \quad \swarrow \\ \bar{a} \quad \alpha \quad \bar{a} \\ \downarrow \\ \bar{a} \end{array} \xrightarrow{*}_\tau \hat{t}_2 = \begin{array}{c} \tilde{\delta} \\ \swarrow \quad \searrow \\ \tilde{\sigma} \quad \tilde{\alpha} \\ \swarrow \quad \downarrow \quad \swarrow \\ \bar{a} \quad \tilde{\alpha} \quad \bar{a} \\ \downarrow \\ \bar{a} \end{array}$$

$$\hat{t}_1 = \hat{t}_2 \text{ et donc } \hat{t} = \begin{array}{c} \tilde{\sigma} \\ \swarrow \quad \searrow \\ \hat{t}_1 \quad \hat{t}_2 \end{array} \xrightarrow{*}_{A_\tau} q$$

Exemple 2

Soit $\Sigma_0 = \{\bar{a}\}$, $\Sigma_1 = \{\alpha\}$ et $\Sigma_2 = \{\sigma\}$.

On considère la relation d'équivalence \equiv sur T_Σ induite par le système de réécriture suivant:

Forme 1-réduite de τ

$$\sigma(x, x) \rightarrow \tilde{\sigma}(x)$$

$$\alpha(x) \rightarrow \tilde{\sigma}(x)$$

$$\bar{a} \rightarrow \tilde{a}$$

Automate de REC_τ

Soit $A = \langle \Sigma, Q, F, R \rangle$ l'automate défini par:

$$Q = \{q, \text{puits}\}, F = \{q\}$$

et R composé des règles suivantes:

$$\sigma(q, q)[1, 2] \rightarrow q$$

$$\sigma(q, q)[1], [2] \rightarrow \text{puits}$$

$$\alpha(q) \rightarrow q$$

$$\bar{a} \rightarrow q$$

La forêt reconnue par l'automate A est l'ensemble des arbres sur T_Σ dont toutes les branches ont la même longueur.

C'est la première règle de l'automate qui assure à chaque niveau de l'arbre que cette propriété est vérifiée.

Le système de réécriture, en substituant l'arbre en "forme de fil" $\tilde{\sigma}$ à tout

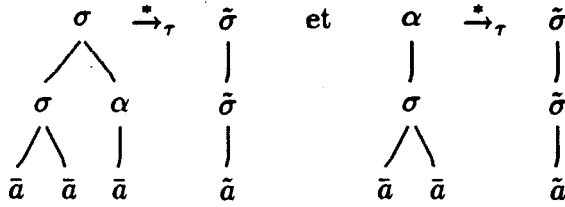
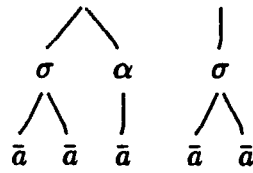
$$\begin{array}{c} | \\ t \end{array}$$

sous-arbre de la forme σ , permet de tester si deux sous-arbres sont

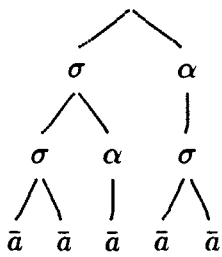
$$\begin{array}{c} \wedge \\ t \quad t \end{array}$$

équivalents.

Par exemple, les arbres σ et α sont équivalents car



et ainsi $\sigma \xrightarrow{*}_A q$.



Automate des formes normales associé à A

$$\begin{aligned} \tilde{\sigma}(q) &\rightarrow q \\ \tilde{a} &\rightarrow q. \end{aligned}$$

Cet automate reconnaît les arbres de la forme



Exemple 3

On considère le même alphabet Σ que dans l'exemple 2 et soit \equiv la relation d'équivalence sur T_Σ induite par le système de réécriture suivant:

Forme 1-réduite de τ

$$\sigma(x, x) \rightarrow \alpha(x)$$

Automate de REC_τ

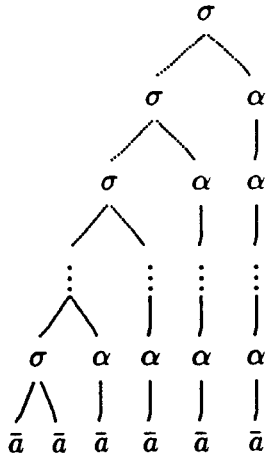
$A = \langle \Sigma, Q, F, R \rangle$ est défini par

$$Q = \{q, q_f, \text{puits}\}, F = \{q\}$$

et R composé des règles suivantes:

$$\begin{aligned} \sigma(q, q_f)[1, 2] &\rightarrow q \\ \sigma(q, q_f)[1], [2] &\rightarrow \text{puits} \\ \alpha(q_f) &\rightarrow q_f \\ \bar{a} &\rightarrow q \\ \bar{a} &\rightarrow q_f. \end{aligned}$$

Cet automate reconnaît les arbres de la forme:



Exemple 4

Il ne s'agit que d'une variante de l'exemple 3.

Prenons $\Sigma_0 = \{1\}$, $\Sigma_1 = \{S\}$ et $\Sigma_2 = \{*\}$.

La relation d'équivalence sur T_Σ est induite par le système de réécriture suivant:

Forme 1-réduite de τ

$$*(x, x) \rightarrow S(x)$$

Automate de REC_τ

$A = \langle \Sigma, Q, F, R \rangle$ est défini par

$$Q = \{q, q_f, \text{puits}\}, F = \{q\}$$

et R composé des règles suivantes:

$$*(q_f, q)[1, 2] \rightarrow q$$

$$*(q_f, q)[1], [2] \rightarrow \text{puits}$$

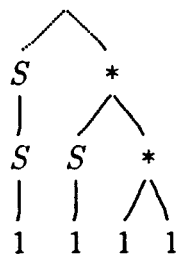
$$S(q_f) \rightarrow q_f$$

$$1 \rightarrow q$$

$$1 \rightarrow q_f.$$

La forêt reconnue par cet automate est l'ensemble des arbres pouvant être interprétés comme la factorielle d'un entier (voir exemple donné en introduction).

Ainsi, l'arbre $*$ correspondant à $f(3)$ est reconnu par l'automate A .



Le lemme suivant établit une propriété fondamentale de l'automate des formes normales.

Cette propriété est ensuite utilisée pour montrer que $IRR(\mathcal{F}(A)) = \mathcal{F}(A_\tau)$.

Lemme 4.2.4 *Soit τ un système de réécriture 1-réduit, soient A un automate de REC_τ et A_τ l'automate des formes normales associé à A . Soit $\{q_1, \dots, q_p\}$ un état de A_τ . On aura $t \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ si et seulement si pour tout i de $[p]$, $t \xrightarrow{*}_{A_\tau} q_i$.*

PREUVE:

La preuve se fait par récurrence sur la profondeur des arbres.

La propriété est évidemment vraie pour les arbres de profondeur 0. Supposons-la vraie pour les arbres de profondeur inférieure ou égale à n et montrons qu'elle l'est encore pour les arbres de profondeur $n + 1$.

Premier cas: $t = \sigma(t')$

$t \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ si et seulement si:

- il existe une règle de la forme $\sigma(\{k_1, \dots, k_m\}) \rightarrow \{q_1, \dots, q_p\}$ dans R_τ
- et $t' \xrightarrow{*}_{A_\tau} \{k_1, \dots, k_m\}$.

Par construction, $\forall i \in [p] \exists E_i \subseteq \{k_1, \dots, k_m\}$ tel que $\sigma(E_i) \rightarrow q_i$ est une règle de R_τ et, parce que la propriété est supposée vraie jusqu'à la profondeur n , $\forall j \in [m] t' \xrightarrow{*}_{A_\tau} k_j$.

Par conséquent, $\forall E_i \subseteq \{k_1, \dots, k_m\}$, $t' \xrightarrow{*}_{A_\tau} E_i$ et donc $\forall i \in [p]$, $t \xrightarrow{*}_{A_\tau} q_i$.

Second cas: $t = \sigma(t', t'')$

$t \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ si et seulement si:

- il existe une règle $\sigma(E', E'')[d] \rightarrow \{q_1, \dots, q_p\}$ dans R_τ
- et $t' \xrightarrow{*}_{A_\tau} E'$, $t'' \xrightarrow{*}_{A_\tau} E''$.

C'est à dire par construction,

$\forall i \in [p]$, $\exists k'_i \in E'$, $\exists k''_i \in E''$ tel que $\sigma(k'_i, k''_i)[d] \rightarrow q_i$ est une règle de R_τ et, parce que la propriété est supposée vraie jusqu'à la profondeur n , $\forall k'_j \in E'$, $t' \xrightarrow{*}_{A_\tau} k'_j$ et $\forall k''_j \in E''$, $t'' \xrightarrow{*}_{A_\tau} k''_j$.

Par conséquent, $\forall i \in [p]$, $\sigma(t', t'') \xrightarrow{*}_{A_\tau} q_i$. \square

Il nous faut maintenant montrer que l'automate A_τ reconnaît effectivement les formes irréductibles de tous les termes reconnus par l'automate A (et elles seules).

Lemme 4.2.5 *Soit τ un système de réécriture 1-réduit, soient A un automate de REC_τ et A_τ l'automate des formes normales associé à A . Pour tout terme t , si $t \xrightarrow{*}_A q$ alors $\hat{t} \xrightarrow{*}_{A_\tau} q$.*

PREUVE:

La preuve se fait par récurrence sur la profondeur des arbres reconnus.

• Soit $t = \sigma$, avec σ lettre de rang 0.

$t \xrightarrow{*}_A q$ implique l'existence d'une règle $\sigma \rightarrow q$ dans A .

Si $\sigma \rightarrow \tilde{\sigma}$ est une règle de τ alors $\hat{t} = \tilde{\sigma}$ et, par construction, $\tilde{\sigma} \rightarrow q$ est une règle de A_τ .

Dans le cas contraire, c'est à dire s'il n'existe aucune règle de τ de partie gauche σ , $\hat{t} = \sigma$ et $\sigma \rightarrow q$ est une règle de A_τ .

Par conséquent, dans ces deux cas, $\hat{t} \xrightarrow{*}_{A_\tau} q$.

• On suppose la propriété vraie pour les arbres jusqu'à la profondeur n et on considère $t = \sigma(t_1, t_2)$ arbre de profondeur $n + 1$ (on ferait le même type de raisonnement dans le cas $t = \sigma(t')$).

La propriété étant supposée vraie jusqu'à la profondeur n , $t_1 \xrightarrow{*}_A q_1$ et $t_2 \xrightarrow{*}_A q_2$ implique que $\hat{t}_1 \xrightarrow{*}_{A_\tau} q_1$ et $\hat{t}_2 \xrightarrow{*}_{A_\tau} q_2$.

Premier cas: $t_1 \neq t_2$

$t \xrightarrow{*}_A q$ implique qu'il existe dans A une règle de la forme $\sigma(q_1, q_2)[1], [2] \rightarrow q$ et $t_1 \xrightarrow{*}_A q_1, t_2 \xrightarrow{*}_A q_2$.

Si $\sigma(x, y) \rightarrow \tilde{\sigma}(x, y)$ est une règle de τ (on ferait le même raisonnement pour la règle $\sigma(x, y) \rightarrow \tilde{\sigma}(y, x)$) alors $\hat{t} = \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$ (parce que τ est 1-réduit) et, par construction, $\tilde{\sigma}(q_1, q_2)[1], [2] \rightarrow q$ est une règle de A_τ .

Dans le cas contraire, s'il n'existe aucune règle de τ de partie gauche $\sigma(x, y)$, $\hat{t} = \sigma(\hat{t}_1, \hat{t}_2)$ et $\sigma(q_1, q_2)[1], [2] \rightarrow q$ est une règle de A_τ .

On a donc $\hat{t} \xrightarrow{*}_{A_\tau} q$.

Second cas: $t_1 \equiv t_2$

$t \xrightarrow{*}_A q$ implique

- l'existence dans A d'une règle de la forme $\sigma(q_1, q_2)[1, 2] \rightarrow q$

- et $t_1 \xrightarrow{*}_A q_1, t_2 \xrightarrow{*}_A q_2$.

En fonction des règles de τ , on peut distinguer trois cas:

2.1: $\sigma(x, x) \rightarrow \tilde{\sigma}(x, x)$ ou $\sigma(x, y) \rightarrow \tilde{\sigma}(x, y)$ est une règle de τ .

Dans ce cas, $\hat{t} = \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$ et $\tilde{\sigma}(q_1, q_2)[1, 2] \rightarrow q$ est une règle de A_τ .

Or $\hat{t}_1 \xrightarrow{*}_{A_\tau} q_1$ et $\hat{t}_2 \xrightarrow{*}_{A_\tau} q_2$, par conséquent $\hat{t} \xrightarrow{*}_{A_\tau} q$.

2.2: Il n'existe dans τ , aucune règle de partie gauche $\sigma(x, x)$ ou $\sigma(x, y)$.

On a alors $\hat{t} = \sigma(\hat{t}_1, \hat{t}_2)$ et $\sigma(q_1, q_2)[1, 2] \rightarrow q$ est une règle de A_τ .

Par conséquent, comme en 2.1, $\hat{t} \xrightarrow{*}_{A_\tau} q$.

2.3: $\sigma(x, x) \rightarrow \tilde{\sigma}(x)$ est une règle de τ .

Dans ce cas, $\hat{t} = \tilde{\sigma}(\hat{t}')$ avec $\hat{t}_1 = \hat{t}_2 = \hat{t}'$ et $\tilde{\sigma}(\{q_1, q_2\}(x)) \rightarrow q$ est une règle de A_τ .

Or $\hat{t}_1 \xrightarrow{*}_{A_\tau} q_1$, $\hat{t}_2 \xrightarrow{*}_{A_\tau} q_2$ et $\hat{t}_1 = \hat{t}_2 = \hat{t}'$ donc $\hat{t}' \xrightarrow{*}_{A_\tau} \{q_1, q_2\}$ (lemme 4.2.4) et finalement $\hat{t} \xrightarrow{*}_{A_\tau} q$. \square

Lemme 4.2.6 Soit τ un système de réécriture 1-réduit, soient A un automate de REC_τ et A_τ l'automate des formes normales associé à A . Pour tout terme \hat{t} , si $\hat{t} \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ alors pour tout q_i de $\{q_1, \dots, q_p\}$, il existe au moins un terme t_i tel que $t_i \xrightarrow{*}_\tau \hat{t}$ et $t_i \xrightarrow{*}_A q_i$.

PREUVE:

La preuve se fait par récurrence sur la profondeur des arbres.

• Soit $\hat{t} = \tilde{\sigma}$ un terme de profondeur 0.

$\hat{t} \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ implique (lemme 4.2.4) que pour tout état q_i de $\{q_1, \dots, q_p\}$ on a $\hat{t} \xrightarrow{*}_{A_\tau} q_i$, c'est à dire qu'il existe une règle $\tilde{\sigma} \rightarrow q_i$ dans A_τ .

Par construction de A_τ (section 4.2.3), pour tout q_i de $\{q_1, \dots, q_p\}$ il existe dans A :

- soit une règle $\sigma_i \rightarrow q_i$ et $\sigma_i \rightarrow \tilde{\sigma}$ est une règle de τ ,
- soit une règle $\tilde{\sigma} \rightarrow q_i$ s'il n'existe aucune règle de partie gauche $\tilde{\sigma}$ dans τ .

Par conséquent, dans ces deux cas, pour tout q_i de $\{q_1, \dots, q_p\}$ il existe au moins un terme t_i tel que $t_i \xrightarrow{*}_\tau \hat{t}$ et $t_i \xrightarrow{*}_A q_i$.

• Supposons la propriété vraie pour les arbres de profondeur inférieure à n et considérons $\hat{t} = \tilde{\sigma}(\hat{t}_1, \hat{t}_2)$ avec $\pi(\hat{t}) = n$ (la preuve est analogue si $\hat{t} = \tilde{\sigma}(\hat{t}')$).

Par l'hypothèse de récurrence, si $\hat{t} \xrightarrow{*}_{A_\tau} E$ (avec $\pi(\hat{t}) < n$) alors pour tout k de E il existe au moins un arbre t_k tel que $t_k \xrightarrow{*}_\tau \hat{t}$ et $t_k \xrightarrow{*}_A k$.

Premier cas: $\hat{t}_1 \neq \hat{t}_2$

$\hat{t} \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ implique

- l'existence dans A_τ d'une règle $\tilde{\sigma}(E_1, E_2)[1], [2] \rightarrow \{q_1, \dots, q_p\}$
- et $\hat{t}_1 \xrightarrow{*}_{A_\tau} E_1$, $\hat{t}_2 \xrightarrow{*}_{A_\tau} E_2$.

De plus, par construction de A_τ (section 4.2.3), pour tout état q_i de $\{q_1, \dots, q_p\}$ il existe $k_i \in E_1$ et $l_i \in E_2$ tel que $\tilde{\sigma}(k_i, l_i)[1], [2] \rightarrow q_i$ est une règle A_τ .

Par conséquent,

soit $\sigma(k_i, l_i)[1], [2] \rightarrow q_i$ est une règle de A et $\sigma(x, y) \rightarrow_\tau \tilde{\sigma}(x, y)$

soit $\tilde{\sigma}(k_i, l_i)[1], [2] \rightarrow q_i$ est une règle de A s'il n'existe aucune règle de τ de partie gauche $\tilde{\sigma}(x, y)$.

Or, $\hat{t}_1 \xrightarrow{*}_{A_\tau} E_1$ implique que pour tout $k \in E_1$, il existe au moins t_{1k} tel que $t_{1k} \xrightarrow{*}_\tau \hat{t}_1$ et $t_{1k} \xrightarrow{*}_A k$, et

$\hat{t}_2 \xrightarrow{*}_{A_\tau} E_2$ implique que pour tout $l \in E_2$, il existe au moins t_{2i} tel que $t_{2i} \xrightarrow{*}_\tau \hat{t}_2$ et $t_{2i} \xrightarrow{*}_A l$.

Donc, dans ces deux cas pour tout q_i de $\{q_1, \dots, q_p\}$ il existe au moins un arbre t_i tel que $t_i = \sigma(t_{1k}, t_{2i})$ ou $t_i = \tilde{\sigma}(t_{1k}, t_{2i})$ (τ est 1-réduit), $t_i \xrightarrow{*}_\tau \hat{t}$ et $t_i \xrightarrow{*}_A q_i$.

Second cas: $\hat{t}_1 = \hat{t}_2$

$\hat{t} \xrightarrow{*}_{A_\tau} \{q_1, \dots, q_p\}$ implique l'existence dans A_τ d'une règle $\tilde{\sigma}(E_1, E_2)[1, 2] \rightarrow \{q_1, \dots, q_p\}$ et donc, comme dans le cas précédent, pour tout q_i de $\{q_1, \dots, q_p\}$ il existe $k_i \in E_1$ et $l_i \in E_2$ tel que $\tilde{\sigma}(k_i, l_i)[1, 2] \rightarrow q_i$ est une règle de A_τ .

En fonction des règles de τ , pour tout q_i de $\{q_1, \dots, q_p\}$, on peut distinguer trois situations.

2.1: $\sigma(x, x) \rightarrow_\tau \tilde{\sigma}(x, x)$ ou $\sigma(x, y) \rightarrow_\tau \tilde{\sigma}(x, y)$ et $\sigma(k_i, l_i)[1, 2] \rightarrow q_i$ est une règle de A .

2.2: Il n'existe aucune règle de τ ayant $\tilde{\sigma}(x, x)$ ou $\tilde{\sigma}(x, y)$ comme partie gauche et $\tilde{\sigma}(k_i, l_i)[1, 2] \rightarrow q_i$ est une règle de A .

2.3: $\sigma(x) \rightarrow_\tau \tilde{\sigma}(x, x)$, $k_i = l_i$ et $\sigma(k_i) \rightarrow q_i$ est une règle de A .

Par conséquent, pour ces trois cas, pour tout q_i de $\{q_1, \dots, q_p\}$ il existe t_i tel que $t_i \xrightarrow{*}_\tau \hat{t}$ et $t_i \xrightarrow{*}_A q_i$. \square

Théorème 4.2.1 *Soit A un automate de REC_τ . L'ensemble $\text{IRR}(\mathcal{F}(A))$, des formes irréductibles des arbres de $\mathcal{F}(A)$, est reconnu par un automate de REC_{\neq} .*

PREUVE:

C'est une conséquence directe des lemmes 4.2.5 et 4.2.6. \square

4.2.4 Décidabilité de l'équivalence dans REC_τ

Théorème 4.2.2 *Pour tout automate A de REC_τ , la propriété " $\mathcal{F}(A) = \emptyset$ " est décidable.*

PREUVE:

Soit A un automate de REC_τ et soit A_τ l'automate des formes normales associé à l'automate A .

- i) Supposons $\mathcal{F}(A) \neq \emptyset$. Il existerait alors au moins un état final q et un arbre t tel que $t \xrightarrow{*}_A q$. En utilisant le lemme 4.2.5, on obtient $\hat{t} \xrightarrow{*}_{A_\tau} q$ et donc $\mathcal{F}(A_\tau) \neq \emptyset$.
- ii) Supposons $\mathcal{F}(A_\tau) \neq \emptyset$. Il existerait alors au moins un état final q et un arbre \hat{t} tel que $\hat{t} \xrightarrow{*}_{A_\tau} q$. En utilisant le lemme 4.2.6, on conclut qu'il existe au moins un terme t tel que $\tau(t) = \hat{t}$ et $t \xrightarrow{*}_A q$ et par conséquent que $\mathcal{F}(A) \neq \emptyset$.

Ainsi $\mathcal{F}(A) = \emptyset$ si et seulement si $\mathcal{F}(A_\tau) = \emptyset$. Le vide est décidable dans REC_\neq , il l'est donc dans REC_τ . \square

Théorème 4.2.3 *L'équivalence est décidable dans REC_τ .*

PREUVE:

La classe REC_τ est stable par union, intersection et complémentaire (partie 4.1) et la propriété du vide est décidable (théorème 4.2.2). \square

Chapitre 5

Transducteurs descendants complets

Dans ce chapitre, nous étendons le résultat de décidabilité de l'équivalence obtenu dans le chapitre 2, pour des transducteurs linéaires et complets, au cas des transducteurs complets mais qui ne sont plus linéaires.

On désigne par **CT-LL** la classe des transducteurs descendants lettre à lettre complets.

La démarche est la même que dans les chapitres précédents, à savoir nous ramener à des automates d'arbres par superposition des arbres d'origine avec leurs transformés. Mais ce codage est ici moins facile à cause de la non-linéarité. En effet à tout niveau d'un arbre en entrée des duplications de sous-arbres peuvent être réalisées, les copies étant éventuellement ensuite transformées de façons différentes. Nous proposons, pour cette raison, une technique de superposition des arbres d'origine avec les arbres produits en sortie nous permettant de coder nos transducteurs dans des automates avec contraintes d'équivalence.

Ce chapitre comprendra trois parties:

1. étude des transformations initiales réalisées à partir d'états équivalents
2. construction de formes normalisées
3. codage des formes normalisées dans des automates à contraintes d'équivalence.

La figure 5.1 illustre la démarche.

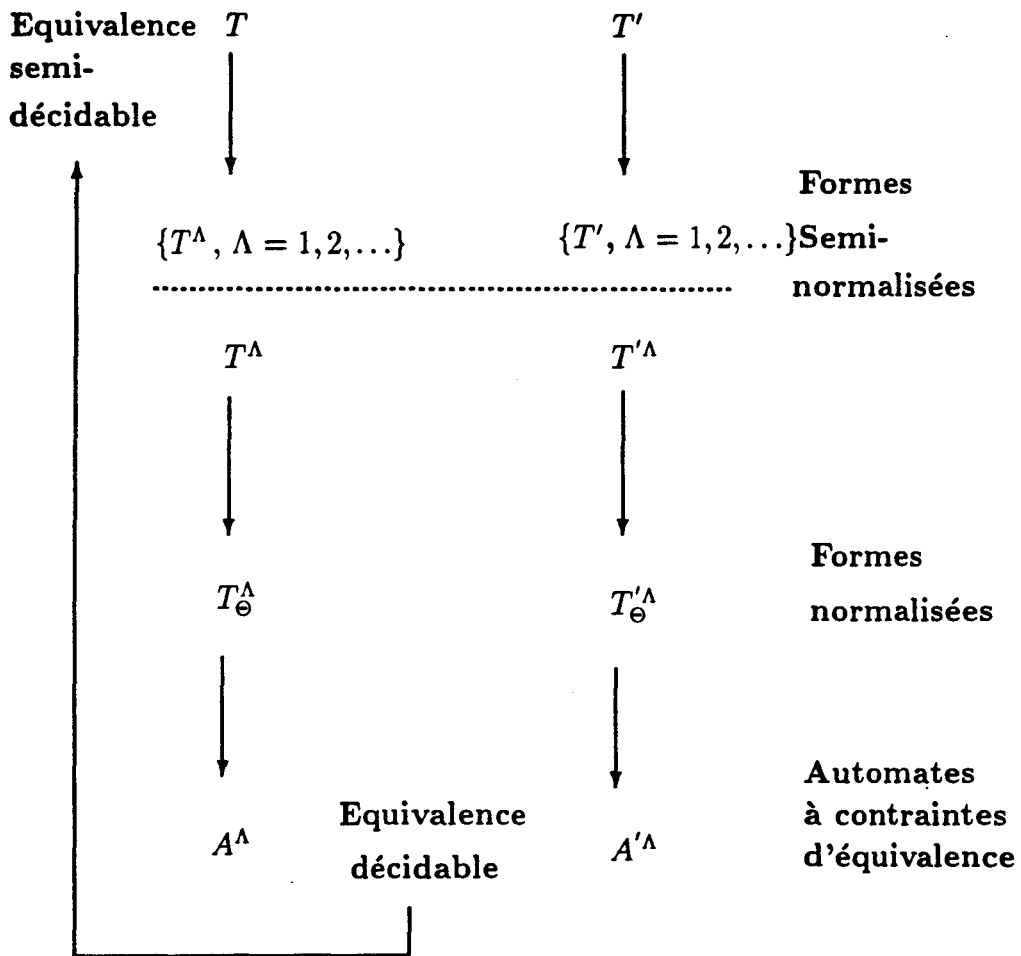


FIG. 5.1 - : Démarche dans le cas complet

5.1 Transformations initiales

L'exemple suivant illustre le problème principal rencontré dans la comparaison des calculs réalisés par deux transducteurs équivalents.

Exemple

Soient T et T' deux transducteurs de CT-LL définis par:

$$\mathbf{T} : q(\sigma(x, y)) \rightarrow \delta(q_1(x), q_2(y), q_1(x))$$

$$\begin{aligned} q_1(a(x)) &\rightarrow a(q_1(x)) \\ q_2(\alpha(x)) &\rightarrow \alpha(q_2(x)) \end{aligned}$$

$$\begin{aligned} q_1(\bar{a}) &\rightarrow \bar{a} \\ q_2(\bar{\alpha}) &\rightarrow \bar{\alpha} \end{aligned}$$

$$\mathbf{T}' : \begin{aligned} k(\sigma(x, y)) &\rightarrow \delta(k_1(x), k_2(y), k'_1(x)) & k(\sigma(x, y)) &\rightarrow \delta(k_1(x), k'_2(y), k_1(x)) \\ k(\sigma(x, y)) &\rightarrow \delta(k_1(x), k_3(y), k_4(y)) \end{aligned}$$

$$k_1(a(x)) \rightarrow a(k_1(x))$$

$$k_1(\bar{a}) \rightarrow \bar{a}$$

$$k'_1(a(x)) \rightarrow a(k'_{11}(x))$$

$$k'_{11}(a(x)) \rightarrow a(k'_{11}(x))$$

$$k'_{11}(\bar{a}) \rightarrow \bar{a}$$

$$k_3(\bar{\alpha}) \rightarrow \bar{\alpha}$$

$$k_2(\alpha(x)) \rightarrow \alpha(k_2(x))$$

$$k_2(\bar{\alpha}) \rightarrow \bar{\alpha}$$

$$k'_2(\alpha(x)) \rightarrow \alpha(k'_{22}(x))$$

$$k'_{22}(\alpha(x)) \rightarrow \alpha(k'_{22}(x))$$

$$k'_{22}(\bar{\alpha}) \rightarrow \bar{\alpha}$$

$$k_4(\bar{\alpha}) \rightarrow \bar{a}$$



T et T' sont équivalents parce qu'ils réalisent la même transformation:

$$\hat{T} = \hat{T}' = \{(\sigma(a^n(\bar{a}), \alpha^m(\bar{\alpha})), \delta(a^n(\bar{a}), \alpha^m(\bar{\alpha}), a^n(\bar{a}))), n \text{ et } m \in \mathbb{N}\}$$

mais pour $(\sigma(\bar{a}, \bar{\alpha}), \delta(\bar{a}, \bar{\alpha}, \bar{a}))$ des torsions différentes sont appliquées dans le premier pas de dérivation.

Nous montrons d'abord que ce phénomène est de profondeur bornée.

Lemme 5.1.1 *A partir de deux états équivalents q et k d'un transducteur T de CT-LL les mêmes transformations initiales sont réalisées sur les arbres de la forme $\sigma(t_1, \dots, t_n)$ pour lesquels pour tout i et j de $[n]$, avec $i \neq j$, $\pi(t_i) \neq \pi(t_j)$.*

PREUVE:

Pour simplifier les notations nous considérons deux lettres σ et δ de rang 2 et nous prenons $\theta = id_{[2]}$ (les autres cas se traitent de façon similaire).

Considérons $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{q(\sigma, \delta, \theta)}$ tel que $\pi(t_1) \neq \pi(t_2)$.

Dans ce cas, parce que T est complet on a $\pi(u_1) = \pi(t_1)$, $\pi(u_2) = \pi(t_2)$ (propriété 1.4.1) et donc $\pi(u_1) \neq \pi(u_2)$.

Supposons que $(\sigma(t_1, t_2), \delta(u_1, u_2)) \notin \hat{T}_{k(\sigma, \delta, \theta)}$.

Il existerait alors au moins une torsion θ' , $\theta' \neq \theta$ telle que $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{k(\sigma, \delta, \theta')}$ (ici $\theta' = \langle 2; 2, 1 \rangle$).

On devrait donc avoir $k(\sigma(t_1, t_2)) \mapsto \delta(k_i(t_2), k_j(t_1)) \xrightarrow{*} \delta(u_1, u_2)$ ce qui est impossible puisque $\pi(t_1) \neq \pi(t_2)$. \square

Lemme 5.1.2 *A partir de deux états équivalents q et k d'un transducteur T de CT-LL, les mêmes transformations initiales peuvent être réalisées sur les arbres de la forme $\sigma(t_1, \dots, t_n)$, pour lesquels il existe $A \subseteq [n]$ tel que quels que soient i et j éléments de A on a $\pi(t_i) = \pi(t_j)$, sauf éventuellement pour un nombre fini d'arbres.*

PREUVE:

Premier cas: σ et δ ont des rangs différents.

Considérons par exemple, σ lettre de rang 2 et δ lettre de rang 3 (ce qui nous autorise un choix quant au sous-arbre qui va être dupliqué) et prenons $\theta = \langle 2; 1, 1, 2 \rangle$.

Nous considérons la différence $\hat{T}_{q(\sigma, \delta, \theta)} - \hat{T}_{k(\sigma, \delta, \theta)}$.

Soient q_i et q'_i deux états.

On désigne par $Q_i \otimes Q'_i$ l'ensemble $\{(t, (u, u')) / (t, u) \in \hat{T}_{q_i} \text{ et } (t, u') \in \hat{T}_{q'_i}\}$. Ainsi Q_i est en fait \hat{T}_{q_i} .

Aux n règles de la forme $q(\sigma(x_1, x_2)) \rightarrow \delta(q_i(x_1), q'_i(x_1), q''_i(x_2))$ ($i \in [n]$) nous associons l'ensemble $\bigcup_{i \in [n]} ((Q_i \otimes Q'_i) \times Q''_i)$.

De la même façon, aux m règles de la forme $k(\sigma(x_1, x_2)) \rightarrow \delta(k_j(x_1), k'_j(x_1), k''_j(x_2))$

nous associons l'ensemble $\bigcup_{j \in [m]} ((K_j \otimes K'_j) \times K''_j)$.

Soient $\mathcal{Q} = \{((t, (u, u')), (t', u'')) / (\sigma(t, t'), \delta(u, u', u'')) \in \hat{T}_{q(\sigma, \delta, \theta)}\}$ et

$\mathcal{K} = \{((t, (u, u')), (t', u'')) / (\sigma(t, t'), \delta(u, u', u'')) \in \hat{T}_{k(\sigma, \delta, \theta)}\}$.

On a donc $\hat{T}_{q(\sigma, \delta, \theta)} - \hat{T}_{k(\sigma, \delta, \theta)}$ infini si et seulement si $\mathcal{Q} - \mathcal{K}$ est infini.

Des propriétés élémentaires de la théorie des ensembles nous permettent d'écrire:

$$\begin{aligned} \mathcal{Q} - \mathcal{K} &= \bigcup_{i \in [n]} ((Q_i \otimes Q'_i) \times Q''_i) - \bigcup_{j \in [m]} ((K_j \otimes K'_j) \times K''_j) \\ &= \bigcup_{i \in [n]} \left(\bigcup_{J \subseteq [m]} (Q_i \otimes Q'_i) \cap \left(\bigcap_{j \in J} \overline{K_j \otimes K'_j} \right) \times Q''_i \cap \left(\bigcap_{k \in [m] - J} \overline{K''_k} \right) \right). \end{aligned}$$

Supposons maintenant $\mathcal{Q} - \mathcal{K}$ infini.

Il existerait alors $i \in [n]$, $J \subseteq [m]$ tel que

$(Q_i \otimes Q'_i) \cap \left(\bigcap_{j \in J} \overline{K_j \otimes K'_j} \right) \times Q''_i \cap \left(\bigcap_{k \in [m] - J} \overline{K''_k} \right)$ est infini.

On aurait ainsi au moins un couple $((t, (u, u')), (t', u'')) \in \mathcal{Q} - \mathcal{K}$ avec $\pi(t) \neq \pi(t')$ ce qui est contradiction avec le lemme 5.1.1.

Second cas: σ et δ ont le même rang.

T étant un transducteur complet, la transformation initiale est dans ce cas linéaire.

En utilisant la même démarche que dans le premier cas (avec seulement des produits cartésiens cette fois) on montre que la différence $\hat{T}_{q(\sigma, \delta, \theta)} - \hat{T}_{k(\sigma, \delta, \theta)}$ est finie (la preuve est analogue à celle du chapitre 2). \square

On obtient ainsi:

Proposition 5.1.1 *Si q et k sont deux états équivalents d'un transducteur T de CT-LL alors pour toute transformation initiale (σ, δ, θ) la différence $\hat{T}_{q(\sigma, \delta, \theta)} - \hat{T}_{k(\sigma, \delta, \theta)}$ est finie.*

Un résultat analogue est obtenu pour des ensembles d'états globalement équivalents. Nous construisons maintenant des formes normalisées de nos transducteurs.

5.2 Formes Λ -normalisées

La construction se fait en deux temps:

1. pour tout entier Λ , une forme Λ -semi-normalisée est d'abord proposée. La construction est identique à celle décrite dans le chapitre 2 et, de la même

façon, la forme Λ -semi-normalisée est identifiée au transducteur d'origine.

2. la seconde phase est consacrée à la construction de formes Λ -normalisées pour lesquelles les torsions sont codées dans les labels des noeuds des arbres mais restent ici appliquées. En effet, les transducteurs descendants peuvent dupliquer des sous-arbres d'un arbre en entrée et ensuite des transformations différentes peuvent être réalisées sur ces copies. En construisant des formes normalisées qui soient des marqueurs, on ne pourrait pas prendre en compte cet aspect des transformations.

La première phase étant analogue à celle décrite dans le second chapitre, nous ne décrivons ici que la seconde phase.

De plus, dans le paragraphe précédent, afin d'avoir un choix quant au sous-arbre de l'arbre en entrée qui allait être dupliqué, nous avons considéré des lettres d'arité 3 pour l'alphabet de sortie. Parce qu'il n'y a pas de différences majeures avec le cas de lettres d'arité 2, excepté la multiplication des cas, nous ne considérons de nouveau que des lettres d'arité au plus 2.

Formes Λ -normalisées

Soit $T = \langle \Sigma, \Delta, Q, I, R \rangle$ un transducteur de CT-LL_i

et soit $T^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R^\Lambda \rangle$ sa forme Λ -semi-normalisée.

Pour tout entier Λ , on associe à T sa forme Λ -normalisée

$T_\Theta^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta_\Theta^\Lambda, Q^\Lambda, I^\Lambda, R_\Theta^\Lambda \rangle$ où Δ_Θ^Λ et R_Θ^Λ sont définis par:

- $q(\sigma) \rightarrow \langle \delta, id \rangle$ est une règle de R_Θ^Λ et $\langle \delta, id \rangle$ est une lettre de Δ_Θ si et seulement si $q(\sigma) \rightarrow \delta$ est une règle de R^Λ .
- $q(\sigma(x)) \rightarrow \langle \delta, id \rangle (q_i(x))$ est une règle de R_Θ^Λ et $\langle \delta, id \rangle$ est une lettre de Δ_Θ si et seulement si $q(\sigma(x)) \rightarrow \delta(q_i(x))$ est une règle de R^Λ .
- $q(\sigma(x)) \rightarrow \langle \delta, \theta \rangle (q_i(x), q_j(x))$ (avec $\theta = \langle 1; 1, 1 \rangle$) est une règle de R_Θ^Λ et $\langle \delta, \theta \rangle$ est une lettre de Δ_Θ si et seulement si $q(\sigma(x)) \rightarrow \delta(q_i(x), q_j(x))$ est une règle de R^Λ .
- $q(\sigma(x_1, x_2)) \rightarrow \langle \delta, \theta \rangle (q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$ (où θ est une bijection sur [2]) est une règle de R_Θ^Λ et $\langle \delta, \theta \rangle$ est une lettre de Δ_Θ si et seulement si $q(\sigma(x_1, x_2)) \rightarrow \delta(q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$ est une règle de R^Λ .

On établit ensuite la propriété suivante

Lemme 5.2.1 *Soient T et T' deux transducteurs de CT-LL et T_{Θ}^{Λ} et T'_{Θ}^{Λ} leurs formes Λ -normalisées. Si Λ est suffisamment grand, T et T' sont équivalents si et seulement si T_{Θ}^{Λ} et T'_{Θ}^{Λ} sont équivalents.*

ELEMENTS DE PREUVE:

Comme dans le chapitre 2, dès que l'entier Λ est de plus supérieur au nombre d'états finitaires du transducteur considéré, les résultats obtenus sont valables pour des transducteurs non nécessairement infinitaires.

On montre de même que si Λ est suffisamment grand alors, à partir d'ensembles d'états globalement équivalents, les transformations peuvent être réalisées avec les mêmes torsions pour tous les arbres dans la forme Λ -semi-normalisée. Ainsi, en codant ces torsions dans les noeuds des arbres produits en sortie, on ne modifie pas l'équivalence des états et donc, si Λ est suffisamment grand, $\hat{T}_E = \hat{T}_F \Leftrightarrow \hat{T}_{\Theta_E}^{\Lambda} = \hat{T}_{\Theta_F}^{\Lambda}$. Pour transposer ce résultat au cas de deux transducteurs T et T' , nous considérons (comme cela est décrit dans la preuve du lemme 2.2.5) les ensembles d'états initiaux de T et T' . \square

5.3 Décidabilité de l'équivalence

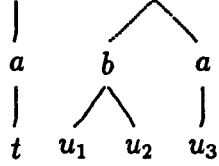
Dans les chapitres 2 et 3, les formes normalisées construites étaient des marqueurs et on utilisait alors directement les résultats de décidabilité de la classe **T-LAB**.

La situation diffère ici puisque les formes normalisées obtenues ne sont pas sans-torsion. Rappelons seulement que, parce que les transducteurs considérés sont descendants, les sous-arbres dupliqués peuvent ensuite être transformés de façons différentes et c'est pour cette raison que les torsions restent appliquées.

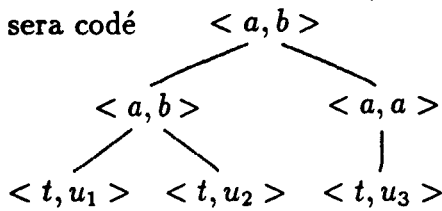
Parce que nous souhaitons réduire le problème de l'équivalence de nos transducteurs à celui de l'équivalence pour des automates d'arbres, il nous faut superposer les arbres d'origine et les arbres produits en sortie. On est ainsi amené à répercuter dans l'arbre d'origine les duplications de sous-arbres réalisées pour produire l'arbre de sortie.

Par exemple,

Le couple (a , b) de la transduction considérée



sera codé

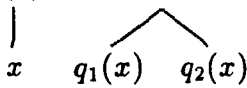


Les arbres $< t, u_1 >$, $< t, u_2 >$ et $< t, u_3 >$ qui correspondent à des transformations différentes du même sous-arbre t_1 seront considérés comme des arbres équivalents. Un système de réécriture, défini de telle façon qu'il associe l'arbre t à toute superposition $< t, u >$, nous permet de tester si deux arbres $< t, u >$ et $< t', u' >$ sont équivalents.

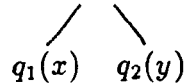
Nos transducteurs descendants sont alors codés dans des automates ascendants à tests d'équivalence entre sous-termes directs; la relation d'équivalence étant induite par le système de réécriture évoqué ci-dessus.

Ainsi,

à partir de la règle $q(a) \rightarrow b$ de T



on obtient la règle $< a, b > [1, 2] \rightarrow q$



L'équivalence étant décidable dans REC_τ , nous en déduisons qu'il en est de même dans $CT-LL$.

5.3.1 Codage des couples d'arbres

Soient Σ et Δ deux alphabets gradués. Les lettres de Δ_θ étant des couples de la forme $\langle \delta, \theta \rangle$, où θ est une application ayant $\text{rang}(\delta)$ comme ensemble de départ. Dans le but de coder tout couple d'arbres (t, u) de $T(\Sigma) \times T(\Delta_\theta)$ en un arbre de $T(\Sigma \times \Delta_\theta)$, noté $[t, u]$, nous définissons la fonction partielle suivante.

Fonction $\text{code}(t, u)$

debut

si $\pi(t) = 0$ et $u = \langle \delta, id \rangle$

alors retourner $\langle t, u \rangle$

sinon

cas: $t = \alpha(t_1)$ et $u = \langle \delta, id \rangle (u_1)$

retourner $\langle \alpha, \langle \delta, id \rangle \rangle (\text{code}(t_1, u_1))$

cas: $t = \alpha(t_1)$ et $u = \langle \delta, \theta \rangle (u_1, u_2)$, $\theta = \langle 1; 1, 1 \rangle$

retourner $\langle \alpha, \langle \delta, \theta \rangle \rangle (\text{code}(t_{\theta(1)}, u_1), \text{code}(t_{\theta(2)}, u_2))$

cas: $t = \alpha(t_1, t_2)$ et $u = \langle \delta, \theta \rangle (u_1, u_2)$ ou θ est une bijection sur [2]

retourner $\langle \alpha, \langle \delta, \theta \rangle \rangle (\text{code}(t_{\theta(1)}, u_1), \text{code}(t_{\theta(2)}, u_2))$

fin si

fin

Remarque

Pour tout couple d'arbres (t, u) de $T(\Sigma) \times T(\Delta_\theta)$, l'indication d'une torsion pour tout noeud nous permet d'obtenir au plus un arbre $[t, u]$.

5.3.2 Automate de REC_τ associé à la forme Λ -normalisée d'un transducteur

Soit $T_\theta^\Lambda = \langle \Sigma^\Lambda, \Delta_\theta^\Lambda, Q^\Lambda, I^\Lambda, R_\theta^\Lambda \rangle$ la forme Λ -normalisée, pour un entier donné Λ , d'un transducteur T de CT-LL. Nous associons à T_θ^Λ l'automate à tests d'équivalence entre fils $A^\Lambda = \langle \Sigma^\Lambda \times \Delta_\theta^\Lambda, Q^\Lambda, F^\Lambda, \mathcal{R}^\Lambda \rangle$ où la relation d'équivalence est définie par un système de réécriture lettre à lettre noethérien, confluent τ sur l'alphabet $(\Sigma^\Lambda \times \Delta_\theta^\Lambda) \cup \Sigma^\Lambda$.

Définition du système de réécriture τ .

- pour toute lettre $\langle \sigma, \langle \delta, id \rangle \rangle$ de $\Sigma^\Lambda \times \Delta_\theta^\Lambda$, pour laquelle $\text{rang}(\delta) = \text{rang}(\sigma) = 0$, $\langle \sigma, \langle \delta, id \rangle \rangle \rightarrow \sigma$ est une règle de τ .

- pour toute lettre $\langle \sigma, \langle \delta, id \rangle \rangle$ de $\Sigma^\Lambda \times \Delta_\Theta^\Lambda$, pour laquelle $rang(\delta) = rang(\sigma) = 1$, $\langle \sigma, \langle \delta, id \rangle \rangle (x) \rightarrow \sigma(x)$ est une règle de τ .
- pour toute lettre $\langle \sigma, \langle \delta, \theta \rangle \rangle$ de $\Sigma^\Lambda \times \Delta_\Theta^\Lambda$, pour laquelle $rang(\delta) = rang(\sigma) = 2$ et θ est une bijection sur $[2]$, $\langle \sigma, \langle \delta, \theta \rangle \rangle (x, y) \rightarrow \sigma(x, y)$ est une règle de τ .
- pour toute lettre $\langle \sigma, \langle \delta, \theta \rangle \rangle$ de $\Sigma^\Lambda \times \Delta_\Theta^\Lambda$, pour laquelle $rang(\delta) = 2 > rang(\sigma) = 1$ et $\theta = \langle 1; 1, 1 \rangle$, $\langle \sigma, \langle \delta, \theta \rangle \rangle (x, x) \rightarrow \sigma(x)$ est une règle de τ .

Il est clair que τ est un système de réécriture noethérien et confluent. De plus τ est sous forme 1-réduite (paragraphe 4.2.1).

Lemme 5.3.1 Pour tout couple d'arbres (t, u) de \hat{T}_Θ^Λ , $[t, u] \xrightarrow{*}_\tau t$.

PREUVE:

La preuve se fait par récurrence sur la profondeur des arbres.

• Soit $(t, u) = (\sigma, \langle \delta, id \rangle) \in T_\Theta^\Lambda$ avec $rang(\sigma) = rang(\delta) = 0$.

$[t, u] = \langle \sigma, \langle \delta, id \rangle \rangle$ et de plus $\langle \sigma, \langle \delta, id \rangle \rangle \rightarrow \sigma$ est une règle de τ . On a donc bien $[t, u] \xrightarrow{*}_\tau t$.

• Supposons la propriété vraie pour les arbres de profondeur inférieure ou égale à p et soit (t, u) un couple de \hat{T}_Θ^Λ avec $\pi(t) = p + 1$.

Nous ne considérerons que le cas $t = \sigma(t_1)$, $u = \langle \delta, \theta \rangle (u_1, u_2)$ (les autres cas se traitent de façon analogue).

$(t, u) \in \hat{T}_\Theta^\Lambda$ et donc $\theta = \langle 1; 1, 1 \rangle$.

Par conséquent $[t, u] = \langle \sigma, \langle \delta, \theta \rangle \rangle ([t_1, u_1], [t_1, u_2])$.

La propriété étant supposée vraie jusqu'à la profondeur p , on a:

$[t_1, u_1] \xrightarrow{*}_\tau t_1$ et $[t_1, u_2] \xrightarrow{*}_\tau t_1$.

Ainsi, $\langle \sigma, \langle \delta, \theta \rangle \rangle ([t_1, u_1], [t_1, u_2]) \xrightarrow{*}_\tau \langle \sigma, \langle \delta, \theta \rangle \rangle (t_1, t_1)$.

De plus, $\langle \sigma, \langle \delta, \theta \rangle \rangle (x, x) \rightarrow \sigma(x)$ est une règle de τ et donc $[t, u] \xrightarrow{*}_\tau \sigma(t_1) = t$.

□

Construction de l'automate A^Λ .

Pour tout entier Λ , $A^\Lambda = \langle \Sigma^\Lambda \times \Delta_\Theta^\Lambda, Q^\Lambda, F^\Lambda, \mathcal{R}^\Lambda \rangle$ est défini par $F^\Lambda = I^\Lambda$ et \mathcal{R}^Λ construit de la façon suivante:

- $\langle \sigma, \langle \sigma', id \rangle \rangle \rightarrow q$ est une règle de \mathcal{R}^Λ si et seulement si $q(\sigma) \rightarrow \langle \sigma', id \rangle$ est une règle de R_Θ^Λ .

- $\langle \sigma, \langle \sigma', id \rangle \rangle (q_j) \rightarrow q_i$ est une règle de \mathcal{R}^Λ si et seulement si $q_i(\sigma(x)) \rightarrow \langle \sigma', id \rangle (q_j(x))$ est une règle de R_Θ^Λ .
- $\langle \sigma, \langle \delta, \theta \rangle \rangle (q_1, q_2) \rightarrow q$ (avec θ bijection sur $[2]$) est une règle de \mathcal{R}^Λ si et seulement si $q(\sigma(x_1, x_2)) \rightarrow \langle \delta, \theta \rangle (q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$ est une règle de R_Θ^Λ .
- $\langle \sigma, \langle \delta, \theta \rangle \rangle (q_1, q_2)[1, 2] \rightarrow q$ (avec $\theta = \langle 1; 1, 1 \rangle$) est une règle de \mathcal{R}^Λ si et seulement si $q(\sigma(x)) \rightarrow \langle \delta, \theta \rangle (q_1(x), q_2(x))$ est une règle de R_Θ^Λ .

Exemple

Soit $T_{\Theta}^{\Lambda} = \langle \Sigma^{\Lambda}, \Delta^{\Lambda}, Q^{\Lambda}, I^{\Lambda}, R^{\Lambda} \rangle$ la forme Λ -normalisée, pour un entier donné Λ , d'un transducteur T de CT-LL avec

$$\begin{aligned} \Sigma_0^{\Lambda} &= \{\bar{a}\}, \Sigma_1^{\Lambda} = \{a, \alpha\}, \Delta_0^{\Lambda} = \{\langle \bar{a}, id \rangle\}, \\ \Delta_1^{\Lambda} &= \{\langle a, id \rangle, \langle a', id \rangle, \langle a'', id \rangle, \langle \delta, id \rangle\}, \\ \Delta_2^{\Lambda} &= \{\langle b, \theta \rangle, \langle \sigma, \theta \rangle\} \text{ avec } \theta = \langle 1; 1, 1 \rangle. \\ Q^{\Lambda} &= \{q_0, q_1, q_2, q_3, q_4\}, Q^{\Lambda} = \{q_0\} \end{aligned}$$

et les règles suivantes:

$$\begin{aligned} q_0(a(x)) &\rightarrow \langle b, \theta \rangle (q_1(x), q_2(x)) \\ q_1(\alpha(x)) &\rightarrow \langle \sigma, \theta \rangle (q_3(x), q_4(x)) \\ q_1(\alpha(x)) &\rightarrow \langle \delta, id \rangle (q_3(x)) \\ q_2(\alpha(x)) &\rightarrow \langle \delta, id \rangle (q_3(x)) \\ q_3(a(x)) &\rightarrow \langle a, id \rangle (q_3(x)) \\ q_3(a(x)) &\rightarrow \langle a', id \rangle (q_3(x)) \\ q_3(\bar{a}) &\rightarrow \langle \bar{a}, id \rangle \\ q_4(a(x)) &\rightarrow \langle a'', id \rangle (q_4(x)) \\ q_4(\bar{a}) &\rightarrow \langle \bar{a}, id \rangle \end{aligned}$$

Système de réécriture τ

$$\begin{aligned} \langle a, \langle b, \theta \rangle \rangle (x, x) &\rightarrow a(x) \\ \langle \alpha, \langle \sigma, \theta \rangle \rangle (x, x) &\rightarrow \alpha(x) \\ \langle \alpha, \langle \delta, id \rangle \rangle (x) &\rightarrow \alpha(x) \\ \langle a, \langle a, id \rangle \rangle (x) &\rightarrow a(x) \\ \langle a, \langle a', id \rangle \rangle (x) &\rightarrow a(x) \\ \langle a, \langle a'', id \rangle \rangle (x) &\rightarrow a(x) \\ \langle \bar{a}, \langle \bar{a}, id \rangle \rangle &\rightarrow \bar{a} \end{aligned}$$

Automate de REC_{τ} associé à T_{Θ}^{Λ}

$A^{\Lambda} = \langle \Sigma^{\Lambda} \times \Delta^{\Lambda}, Q^{\Lambda}, F^{\Lambda} = I^{\Lambda}, \mathcal{R}^{\Lambda} \rangle$ est défini par:

$$\begin{aligned} \langle a, \langle b, \theta \rangle \rangle (q_1, q_2)[1, 2] &\rightarrow q_0 \\ \langle \alpha, \langle \sigma, \theta \rangle \rangle (q_3, q_4)[1, 2] &\rightarrow q_1 \\ \langle \alpha, \langle \delta, id \rangle \rangle (q_3) &\rightarrow q_1 \\ \langle \alpha, \langle \delta, id \rangle \rangle (q_3) &\rightarrow q_2 \\ \langle a, \langle a, id \rangle \rangle (q_3) &\rightarrow q_3 \\ \langle a, \langle a', id \rangle \rangle (q_3) &\rightarrow q_3 \\ \langle a, \langle a'', id \rangle \rangle (q_4) &\rightarrow q_4 \\ \langle \bar{a}, \langle \bar{a}, id \rangle \rangle &\rightarrow q_3 \\ \langle \bar{a}, \langle \bar{a}, id \rangle \rangle &\rightarrow q_4 \end{aligned}$$

Considérons, par exemple le couple $(t, u) = (a(t_1), \langle b, \theta \rangle (u_1, u_2)) \in T_{\Theta}^{\Lambda}$ avec

$$t_1 = \alpha(\bar{a}),$$

$$u_1 = \langle \sigma, \theta \rangle (\langle \bar{a}, id \rangle, \langle \bar{a}, id \rangle)$$

$$\text{et } u_2 = \langle \delta, id \rangle (\langle \bar{a}, id \rangle).$$

$[t, u] = \langle a, \langle b, \theta \rangle \rangle ([t_1, u_1], [t_1, u_2])$, résultat de la fonction définie en 5.3.1, est un arbre de $\mathcal{F}(A^{\Lambda})$.

En effet, $[t_1, u_1] \xrightarrow{*}_{A^{\Lambda}} q_1$ et $[t_1, u_2] \xrightarrow{*}_{A^{\Lambda}} q_2$.

De plus, $[t_1, u_1]$ et $[t_1, u_2]$ sont des arbres équivalents parce que

$$[t_1, u_1] \xrightarrow{*}_{\tau} t_1$$

$$\text{et } [t_1, u_2] \xrightarrow{*}_{\tau} t_1.$$

Par conséquent $[t, u] = \langle a, \langle b, \theta \rangle \rangle ([t_1, u_1], [t_1, u_2]) \xrightarrow{*}_{A^{\tau}} q_0$.

Lemme 5.3.2 Soient T_{Θ}^{Λ} la forme Λ -normalisée d'un transducteur T et A^{Λ} l'automate de REC_{τ} qui lui est associé. Pour tout état q , $q(t) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u$ si et seulement si $[t, u] \xrightarrow{*}_{A^{\Lambda}} q$.

PREUVE:

La preuve se fait par récurrence sur la profondeur des arbres.

• La propriété est vraie pour les arbres de profondeur 0.

Soit (t, u) un couple d'arbres de profondeur 0. t et u sont, dans ce cas, des lettres d'arité 0.

$q(t) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u$ si et seulement si il existe dans T_{Θ}^{Λ} une règle $q(t) \rightarrow u$.

Il existe donc par construction une règle $\langle t, u \rangle \rightarrow q$ dans A^{Λ} . Et ainsi, $[t, u] \xrightarrow{*}_{A^{\Lambda}} q$.

• Supposons la propriété vraie jusqu'à la profondeur n et soit $t = \sigma(t_1)$ un arbre de profondeur $n + 1$ (on procéderait de façon analogue dans le cas $t = \sigma(t_1, t_2)$).

Premier cas: $u = \langle \delta, id \rangle (u_1)$.

Nous avons alors $[t, u] = \langle \sigma, \langle \delta, id \rangle \rangle ([t_1, u_1])$. $q(t) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u$ si et seulement si il existe une règle $q(\sigma(x)) \rightarrow \langle \delta, id \rangle (q'(x))$ dans T_{Θ}^{Λ} et $q'(t_1) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u_1$.

Il existe donc, par construction, une règle $\langle \sigma, \langle \delta, id \rangle \rangle (q') \rightarrow q$ dans A^{Λ}

et, parce que la propriété est supposée vraie jusqu'à la profondeur n , $[t_1, u_1] \xrightarrow{*}_{A^{\Lambda}} q'$.

Finalement, $[t, u] \xrightarrow{*}_{A^{\Lambda}} q$.

Second cas: $t = \sigma(t_1)$ et $u = \langle \delta, \theta \rangle (u_1, u_2)$, avec $\theta = \langle 1; 1, 1 \rangle$.

Dans ce cas, nous avons $[t, u] = \langle \sigma, \langle \delta, \theta \rangle \rangle ([t_1, u_1], [t_1, u_2])$.

$q(t) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u$ si et seulement si il existe une règle $q(\sigma(x)) \rightarrow \langle \delta, \theta \rangle (q_1(x), q_2(x))$ dans T_{Θ}^{Λ} et $q_1(t_1) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u_1$, $q_2(t_1) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u_2$.

Il existe donc une règle $\langle \sigma, \langle \delta, \theta \rangle \rangle (q_1, q_2)[1, 2] \rightarrow q$ dans A^{Λ} et, parce que la propriété est supposée vraie jusqu'à la profondeur n , $[t_1, u_1] \xrightarrow{*}_{A^{\Lambda}} q_1$, $[t_1, u_2] \xrightarrow{*}_{A^{\Lambda}} q_2$.

De plus, $[t_1, u_1] \xrightarrow{*}_{\tau} t_1$ et $[t_1, u_2] \xrightarrow{*}_{\tau} t_1$ (lemme 5.3.1) et ainsi $[t_1, u_1] \equiv [t_1, u_2]$.

Finalement, $[t, u] \xrightarrow{*}_{A^{\Lambda}} q$.

Inversement, en utilisant la même méthode, on montre que dans ces deux cas $[t, u] \xrightarrow{*}_{A^{\Lambda}} q$ implique $q(t) \xrightarrow{*}_{T_{\Theta}^{\Lambda}} u$. \square

Décidabilité de l'équivalence

Lemme 5.3.3 *Soient T et T' deux transducteurs de CT-LL, T_Θ^Λ , $T'_{\Theta'}^\Lambda$ leurs formes Λ -normalisées et A^Λ , A'^Λ les automates de REC_τ associés à T_Θ^Λ et $T'_{\Theta'}^\Lambda$. Lorsque Λ est suffisamment grand, T et T' sont équivalents si et seulement si A^Λ et A'^Λ sont équivalents.*

PREUVE:

Le lemme 5.2.1 établit que, lorsque Λ est suffisamment grand, T et T' sont équivalents si et seulement si leurs formes Λ -normalisées T_Θ^Λ et $T'_{\Theta'}^\Lambda$ le sont. Le lemme 5.3.2 nous permet alors de conclure. \square

On obtient ainsi

Théorème 5.3.1 *L'équivalence des transducteurs descendants lettre à lettre complets est décidable.*

PREUVE:

Le lemme 5.3.3 établit en fait la semi-décidabilité de l'équivalence dans CT-LL. La non-équivalence étant semi-décidable, nous concluons. \square

5.4 Le cas général descendant

Nous venons d'établir la décidabilité de l'équivalence pour les transducteurs descendants lettre à lettre linéaires (chapitre 3) ou complets (chapitre 5).

La question qui se pose alors naturellement est de savoir si nos résultats peuvent s'étendre au cas général de transducteurs descendants lettre à lettre non-linéaires et non-complets?

Il n'en est rien malheureusement. Nous en illustrons la raison dans la première partie de ce paragraphe.

Pour conclure sur le cas des transducteurs descendants lettre à lettre, nous résumons dans la seconde partie l'ensemble de la démarche en utilisant, à chaque étape, la présentation la plus générale possible.

5.4.1 Cas général

L'exemple suivant illustre le fait que les lemmes fondamentaux (lemmes 3.1.1 ou 5.1.2 selon que les transducteurs sont linéaires ou complets) sur lesquels reposent

nos techniques, ne peuvent être étendus au cas général.

Exemple

Considérons les transducteurs T et T' définis par:

$$\mathbf{T} : \begin{array}{l} q(\sigma(x, y)) \rightarrow \delta(q_1(x), q_2(y)) \\ q(\sigma(x, y)) \rightarrow \delta(q_3(x), q_4(x)) \quad q(\sigma(x, y)) \rightarrow \delta(q_5(y), q_6(y)) \end{array}$$

$$\begin{array}{ll} q_1(a(x)) \rightarrow a(q_1(x)) & q_1(\bar{a}) \rightarrow \bar{a} \\ q_2(\alpha(x)) \rightarrow \alpha(q_2(x)) & q_2(\bar{\alpha}) \rightarrow \bar{\alpha} \end{array}$$

$$\begin{array}{ll} q_3(a(x)) \rightarrow a(q_3(x)) & q_3(\bar{a}) \rightarrow \bar{a} \\ q_4(a(x)) \rightarrow \alpha(q_4(x)) & q_4(\bar{a}) \rightarrow \bar{\alpha} \\ q_4(a(x)) \rightarrow \bar{\alpha} & \end{array}$$

$$\begin{array}{ll} q_5(\alpha(x)) \rightarrow a(q_5(x)) & q_5(\bar{\alpha}) \rightarrow \bar{a} \\ q_5(\alpha(x)) \rightarrow \bar{a} & \\ q_6(\alpha(x)) \rightarrow \alpha(q_6(x)) & q_6(\bar{\alpha}) \rightarrow \bar{\alpha} \end{array}$$

$$\mathbf{T}' : \begin{array}{l} k(\sigma(x, y)) \rightarrow \delta(k_3(x), k_4(x)) \quad k(\sigma(x, y)) \rightarrow \delta(k_5(y), k_6(y)) \end{array}$$

$$\begin{array}{ll} k_3(a(x)) \rightarrow a(k_3(x)) & k_3(\bar{a}) \rightarrow \bar{a} \\ k_4(a(x)) \rightarrow \alpha(k_4(x)) & k_4(\bar{a}) \rightarrow \bar{\alpha} \\ k_4(a(x)) \rightarrow \bar{\alpha} & \end{array}$$

$$\begin{array}{ll} k_5(\alpha(x)) \rightarrow a(k_5(x)) & k_5(\bar{\alpha}) \rightarrow \bar{a} \\ k_5(\alpha(x)) \rightarrow \bar{a} & \\ k_6(\alpha(x)) \rightarrow \alpha(k_6(x)) & k_6(\bar{\alpha}) \rightarrow \bar{\alpha} \end{array}$$

Les transducteurs T et T' sont équivalents:

$$\hat{T} = \hat{T}' = \{(\sigma(a^m(\bar{a}), \alpha^n(\bar{\alpha})), \delta(a^{m'}(\bar{a}), \alpha^{n'}(\bar{\alpha}))) \mid m' \text{ et } n' \leq \text{Sup}\{m, n\}\}.$$

On remarque cependant que la première règle du transducteur T , qui est sans-torsion, n'a pas d'équivalent dans T' .

Tous les couples $(\sigma(a^m(\bar{a}), \alpha^n(\bar{\alpha})), \delta(a^m(\bar{a}), \alpha^n(\bar{\alpha})))$ obtenus en appliquant initialement la règle sans-torsion de T seront produits dans T' en appliquant initialement

une torsion différente de l'identité. On aura:

$$\begin{aligned} k(\sigma(a^m(\bar{a}), \alpha^n(\bar{\alpha}))) &\xrightarrow{*} \delta(k_3(a^m(\bar{a})), k_4(a^m(\bar{a}))) \\ &\xrightarrow{*} \delta(a^m(\bar{a}), \alpha^n(\bar{\alpha})) \end{aligned}$$

dans le cas $n \leq m$

et

$$\begin{aligned} k(\sigma(a^m(\bar{a}), \alpha^n(\bar{\alpha}))) &\xrightarrow{*} \delta(k_5(\alpha^n(\bar{\alpha})), k_6(\alpha^n(\bar{\alpha}))) \\ &\xrightarrow{*} \delta(a^m(\bar{a}), \alpha^n(\bar{\alpha})) \end{aligned}$$

dans le cas $m \leq n$.

Ainsi, à partir d'états équivalents des transformations initiales différentes sont nécessairement réalisées pour une infinité de couples d'arbres. Les techniques précédentes ne peuvent donc s'appliquer ici.

5.4.2 Regard sur les techniques utilisées

Nous résumons, maintenant, l'ensemble de la démarche en adoptant, à chaque étape, la plus générale des constructions décrites précédemment.

Transformations initiales

Les lemmes 3.1.1 et 5.1.2, obtenus respectivement pour les cas linéaires et complets, et le contre-exemple du paragraphe précédent nous permettent d'énoncer le résultat suivant:

Lemme fondamental 1 *Si q et k sont deux états équivalents d'un transducteur descendant lettre à lettre T alors pour toute transformation initiale (σ, δ, θ) la différence $\hat{T}_{q(\sigma, \delta, \theta)} - \hat{T}_{k(\sigma, \delta, \theta)}$ est:*

- d'image finie si T est linéaire
- finie si T est complet
- éventuellement infinie sinon.

Construction des formes Λ -semi-normalisées

Pour tout entier Λ , on associe à tout transducteur T descendant lettre à lettre sa forme Λ -semi-normalisée T^Λ en utilisant la construction décrite dans le paragraphe 3.2 (cas linéaire).

Des résultats obtenus dans les chapitres 2 et 5, on déduit:

Lemme 1 *Si T est linéaire ou complet alors, lorsque Λ est suffisamment grand, à partir d'états équivalents, les calculs dans la forme Λ -semi-normalisée peuvent être réalisées avec les mêmes torsions pour tous les arbres.*

Construction des formes Λ -normalisées

Pour tout entier Λ , à partir de la forme Λ -semi-normalisée T^Λ d'un transducteur on construit sa forme Λ -normalisée en deux phases:

- complétion du transducteur (algorithme donné en 3.2). Aucune modification du transducteur n'est évidemment réalisée si le transducteur est complet d'origine.
- construction de la forme Λ -normalisée. On utilise ici la construction définie dans le paragraphe 5.2 pour des transducteurs complets. La forme normalisée obtenue n'est pas un marqueur puisque les torsions qui sont codées au niveau des noeuds des arbres restent appliquées.

On obtient:

Lemme 2 *Si T et T' sont deux transducteurs linéaires ou s'ils sont tous deux complets alors, lorsque Λ est suffisamment grand, T et T' sont équivalents si et seulement si leurs formes Λ -normalisées sont équivalentes.*

Décision de l'équivalence

Nous ramenons enfin le problème de l'équivalence de nos transducteurs linéaires ou complets à l'équivalence d'automates de REC_τ . La construction des automates à tests d'équivalence, à partir des formes normalisées, est décrite dans le paragraphe 5.3.2. Notons que, pour un transducteur qui serait linéaire d'origine, les contraintes d'équivalence sont indifférentes (l'automate obtenu dans ce cas est en fait un automate "classique").

Chapitre 6

Transducteurs ascendants

Dans ce chapitre, en transposant au cas ascendant la démarche utilisée pour les transducteurs descendants, nous établissons la décidabilité de l'équivalence pour les transducteurs ascendants lettre à lettre.

Notation

La classe des transducteurs ascendants lettre à lettre est notée **B-LL**.

6.1 Extension des définitions

Nous proposons dans ce paragraphe une extension au cas ascendant des notions introduites dans le chapitre 1, pour les transducteurs descendants.

Définition 6.1.1 Soit \mathcal{D} un mouvement de dérivation pour le terme $t = \sigma(t_1, \dots, t_m)$ et aboutissant à l'état q .

$\mathcal{D} : \sigma(t_1, \dots, t_m) \xrightarrow{*} \sigma(q_1(u_1), \dots, q_m(u_m)) \mapsto q(\delta(u_{\theta(1)}, \dots, u_{\theta(n)}))$.

On appelle transformation finale le triplet (σ, δ, θ) .

Notation

$\hat{B}_{q(\sigma, \delta, \theta)}$ désignera l'ensemble des couples (t, u) tels que $t \xrightarrow{*}_B q(u)$ en appliquant la transformation finale (σ, δ, θ) . On écrira, plus simplement, \hat{B}_q s'il n'y a aucune ambiguïté sur le couple de lettres transformées.

Définition 6.1.2 Un état q est dit infinitaire s'il existe une infinité de couples (t, u) tels que $t \xrightarrow{*} q(u)$.

Définition 6.1.3 Deux états q et k , d'un transducteur ascendant B , sont équivalents si $t \xrightarrow{*}_B q(u) \Leftrightarrow t \xrightarrow{*}_B k(u)$.

Cette définition s'étend à des ensembles d'états:

Définition 6.1.4 Deux ensembles d'états $\{q_1, \dots, q_n\}$ et $\{k_1, \dots, k_m\}$ d'un transducteur B sont globalement équivalents si et seulement si $\bigcup_{i \in [n]} (\widehat{B}_{q_i}) = \bigcup_{j \in [m]} (\widehat{B}_{k_j})$.

Comme dans les chapitres précédents, pour des raisons techniques nous étudions d'abord les transformations finales aboutissant à des états infinitaires équivalents.

6.2 Transformations finales

Considérons le transducteur $B = \langle \Sigma, \Delta, Q, I, R \rangle$ avec

$\Sigma_0 = \{\bar{a}\}$, $\Sigma_1 = \{a, \alpha\}$ et $\Sigma_2 = \{\sigma\}$,

$\Delta_0 = \{e, f\}$, $\Delta_2 = \{\delta\}$

et les règles:

$$\begin{array}{ll} \sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_1, x_2)) & \sigma(k_2(x_1), k_1(x_2)) \rightarrow k(\delta(x_2, x_1)) \\ a(q'_1(x)) \rightarrow q_1(e) & \alpha(k'_1(x)) \rightarrow k_1(e) \\ \alpha(q'_2(x)) \rightarrow q_2(f) & a(k'_2(x)) \rightarrow k_2(f) \end{array}$$

avec de plus, $\forall \tau \in T_\Sigma$,

$$\tau \xrightarrow{*} q'_1(\tau) \text{ et } \tau \xrightarrow{*} q'_2(\tau) \quad \tau \xrightarrow{*} k'_1(\tau) \text{ et } \tau \xrightarrow{*} k'_2(\tau)$$

Les transformations aboutissant aux états q et k sont égales:

$$\widehat{B}_q = \widehat{B}_k = \{(\sigma(a(\tau), \alpha(\tau')), \delta(e, f)) \mid \tau, \tau' \in T_\Sigma\}.$$

Et pourtant, des torsions finales différentes sont appliquées à tous les arbres: pour l'état k une permutation des sous-arbres de la racine est effectuée alors que pour q la transformation finale est sans-torsion.

Nous montrons d'abord que ce phénomène est d'image finie.

Lemme 6.2.1 Soient q et k deux états infinitaires équivalents d'un transducteur B . Pour toute transformation finale (σ, δ, θ) , la différence $\widehat{B}_{q(\sigma, \delta, \theta)} - \widehat{B}_{k(\sigma, \delta, \theta)}$ est d'image finie.

PREUVE:

Nous distinguerons deux cas selon les arités de σ et δ .

Premier cas: σ et δ sont de rang 2.

Les règles considérées sont donc de la forme $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_{\theta(1)}, x_{\theta(2)}))$.

Considérons d'abord le cas $\theta = id$.

Supposons qu'il existe n règles de la forme $\sigma(q_i(x_1), q'_i(x_2)) \rightarrow q(\delta(x_1, x_2))$, c'est-à-dire aboutissant à l'état q ,

et m règles de la forme $\sigma(k_i(x_1), k'_i(x_2)) \rightarrow k(\delta(x_1, x_2))$, avec k comme état de sortie.

Nous considérons la différence $\hat{B}_{q\theta} - \hat{B}_{k\theta}$.

Nous notons \mathcal{Q} l'ensemble $\{((u, v), (u', v')) \in \hat{B}_{q_i} \times \hat{B}_{q'_i}, i \in [n]\}$

et \mathcal{K} l'ensemble $\{((u, v), (u', v')) \in \hat{B}_{k_j} \times \hat{B}_{k'_j}, j \in [m]\}$.

On aura alors $((u, v), (u', v')) \in \mathcal{Q} - \mathcal{K} \Leftrightarrow (\sigma(u, u'), \delta(v, v')) \in \hat{B}_{q\theta} - \hat{B}_{k\theta}$.

Par \bar{B}_q on désigne le complémentaire de \hat{B}_q dans $T_\Sigma \times T_\Delta$.

Des propriétés de base de la théorie des ensembles nous permettent d'exprimer la différence $\mathcal{Q} - \mathcal{K}$ en fonction de $\hat{B}_{q_i}, \hat{B}_{q'_i}, \hat{B}_{k_j}, \hat{B}_{k'_j}$.

$$\begin{aligned} \mathcal{Q} - \mathcal{K} &= \left(\bigcup_{i \in [n]} \hat{B}_{q_i} \times \hat{B}_{q'_i} \right) - \left(\bigcup_{j \in [m]} \hat{B}_{k_j} \times \hat{B}_{k'_j} \right) \\ &= \bigcup_{i \in [n]} \left(\bigcup_{J \subseteq [m]} \left(\hat{B}_{q_i} \cap \left(\bigcap_{j \in J} \bar{B}_{k_j} \right) \right) \times \left(\hat{B}_{q'_i} \cap \left(\bigcap_{l \in [m] - J} \bar{B}_{k'_l} \right) \right) \right). \end{aligned}$$

Supposons que $\mathcal{Q} - \mathcal{K}$ ne soit pas d'image finie.

Il existerait alors au moins $i \in [n]$ et $J \subseteq [m]$ tels que $\hat{B}_{q_i} \cap \left(\bigcap_{j \in J} \bar{B}_{k_j} \right)$

ou $\hat{B}_{q'_i} \cap \left(\bigcap_{l \in [m] - J} \bar{B}_{k'_l} \right)$ n'est pas d'image finie.

Choisissons, par exemple, le cas où $\hat{B}_{q_i} \cap \left(\bigcap_{j \in J} \bar{B}_{k_j} \right)$ n'est pas d'image finie.

Pour tout couple (u', v') de $\hat{B}_{q_i} \cap \left(\bigcap_{l \in [m] - J} \bar{B}_{k'_l} \right)$, on pourrait donc trouver au moins

un couple (u, v) de $\hat{B}_{q_i} \cap \left(\bigcap_{j \in J} \bar{B}_{k_j} \right)$ tel que $\pi(v) > \pi(u')$.

C'est-à-dire $(\sigma(u, u'), \delta(v, v')) \in \hat{B}_{q\theta} - \hat{B}_{k\theta}$ avec $\pi(v) > \pi(u')$.

Les états q et k étant équivalents, il existerait donc une torsion μ , différente de θ ,

et telle que $(\sigma(u, u'), \delta(v, v')) \in \hat{B}_{k_\mu}$. Trois cas sont possibles.

cas 1.1: $\mu = \langle 2; 2, 1 \rangle$.

On devrait donc avoir

$\sigma(u, u') \xrightarrow{*} \sigma(k_j(w), k'_j(w')) \mapsto k(\delta(w', w))$ avec $\delta(w', w) = \delta(v, v')$,

c'est-à-dire en fait $u' \xrightarrow{*} k'_j(w')$ avec $w' = v$.

Cette transformation est impossible puisque d'une part $\pi(w') \leq \pi(u')$ (lemme 1.4.1) et d'autre part $\pi(v) > \pi(u')$.

cas 1.2: $\mu = \langle 2; 1, 1 \rangle$.

On aurait ici

$\sigma(u, u') \xrightarrow{*} \sigma(k_j(w), k'_j(w')) \mapsto k(\delta(w, w))$ avec $\delta(w, w) = \delta(v, v')$.

Ce mouvement est impossible également puisque $\pi(v) \neq \pi(v')$ dans la mesure où on ne peut avoir simultanément $\pi(v) > \pi(u')$ et $\pi(v') \leq \pi(u')$.

cas 1.3: $\mu = \langle 2; 2, 2 \rangle$.

Le mouvement de transformation à considérer ici est

$\sigma(u, u') \xrightarrow{*} \sigma(k_j(w), k'_j(w')) \mapsto k(\delta(w', w'))$ avec $\delta(w', w') = \delta(v, v')$.

On arrive également à une impossibilité puisque $\pi(w') \leq \pi(u')$ et $\pi(v) > \pi(u')$.

On fait le même type de raisonnement pour les autres valeurs possibles pour la torsion θ : $\theta = \langle 2; 2, 1 \rangle$, $\theta = \langle 2; 1, 1 \rangle$ ou $\theta = \langle 2; 2, 2 \rangle$.

On en conclut que la différence $\hat{B}_{q(\sigma, \delta, \theta)} - \hat{B}_{k(\sigma, \delta, \theta)}$ est d'image finie lorsque σ et δ sont de rang 2.

Second cas: σ est de rang 2 et δ est de rang 1.

Les règles sont donc de la forme $\sigma(q_i(x_1), q'_i(x_2)) \rightarrow q(\delta(x_{\theta(1)}))$. Supposons que la différence $\hat{B}_{q_\theta} - \hat{B}_{k_\theta}$ ne soit pas d'image finie.

On est donc amené à distinguer deux cas selon que $\theta = \langle 2; 1 \rangle$ ou que $\theta = \langle 2; 2 \rangle$.

2.1: $\theta = \langle 2; 1 \rangle$.

On aurait donc $\sigma(u, u') \xrightarrow{*} \sigma(q_i(v), q'_i(v')) \mapsto q(\delta(v))$.

Supposons que la différence $\hat{B}_{q_\theta} - \hat{B}_{k_\theta}$ ne soit pas d'image finie, il existerait donc une infinité d'arbres v tels que $(\sigma(u, u'), \delta(v)) \notin \hat{B}_{k_\theta}$. Ce serait donc en particulier le cas pour v tel que $\pi(v) > \pi(u')$.

Les états q et k étant supposés équivalents, il existerait donc une torsion μ , différente de θ , telle que $(\sigma(u, u'), \delta(v)) \in \hat{B}_{k_\mu}$. Ici, la seule autre torsion possible est

$\mu = \langle 2; 2 \rangle$.

On aurait donc:

$\sigma(u, u') \xrightarrow{*} \sigma(k_i(w), k'_i(w')) \mapsto k(\delta(w'))$ avec $w' = v$.

Cette transformation est impossible puisque $\pi(v) > \pi(u')$ et $\pi(w') \leq \pi(u')$.

La différence considérée est donc d'image finie.

2.2: $\theta = \langle 2; 2 \rangle$.

Ce cas se traite de façon analogue au cas précédent.

Ainsi dans tous les cas possibles, pour toute transformation finale (σ, δ, θ) la différence $\hat{B}_{q(\sigma, \delta, \theta)} - \hat{B}_{k(\sigma, \delta, \theta)}$ est d'image finie. \square

Un résultat analogue est obtenu lorsqu'on considère des ensembles d'états globalement équivalents. Ainsi, pour tout couple (E, F) d'ensembles d'états globalement équivalents, pour toute transformation finale (σ, δ, θ) il existe un entier, qu'on notera $\lambda_{(E, F), (\sigma, \delta, \theta)}$, tel que dès que $\pi(u) > \lambda_{(E, F), (\sigma, \delta, \theta)}$, on a $(t, u) \in \hat{B}_{E(\sigma, \delta, \theta)} \Leftrightarrow (t, u) \in \hat{B}_{F(\sigma, \delta, \theta)}$. Ce résultat est valable dans le cas d'ensembles d'états non nécessairement infinitaires.

Comme pour le cas des transducteurs descendants, la démarche se poursuit par la construction, en deux étapes, de formes Λ -normalisées.

6.3 Formes Λ -normalisées

Pour tout entier Λ , nous construisons d'abord la forme Λ -semi-normalisée d'un transducteur B pour laquelle tous les couples (t, u) (d'arbres ou de sous-arbres) tels que $\pi(u) \leq \Lambda$ sont traités globalement, c'est-à-dire en fait identifiés à de nouvelles lettres.

6.3.1 Formes Λ -semi-normalisées

Au transducteur ascendant $B = \langle \Sigma, \Delta, Q, F, R \rangle$ nous associons le transducteur $B^\Lambda = \langle \Sigma^\Lambda, \Delta^\Lambda, Q^\Lambda, F^\Lambda, R^\Lambda \rangle$ construit de la façon suivante.

- $q^{<\Lambda}$ et q^Λ sont des états de Q^Λ si et seulement si q est un état de Q et ils sont éléments de F^Λ si et seulement si q est dans F .
- $t \rightarrow q^{<\Lambda}(u)$ (respectivement $t \rightarrow q^\Lambda(u)$) est une règle de R^Λ ,
 t est une lettre de Σ^Λ et u est une lettre de Δ^Λ
 si et seulement si
 $(t, u) \in \hat{B}_q$ avec $t \in T_\Sigma$ et $\pi(t) = \pi(u) < \Lambda$ (respectivement $\pi(t) = \pi(u) = \Lambda$).

- $t_0(q'_1(x_1), \dots, q'_n(x_n)) \rightarrow q^{<\Lambda}(u)$ est une règle de R^Λ (q'_i est soit $q_i^{<\Lambda}$ soit q_i^Λ),
 t_0 est une lettre de rang n de Σ^Λ et u est une lettre de Δ^Λ
 si et seulement si
 $t_0(q_1(x_1), \dots, q_n(x_n)) \xrightarrow{*}_B q(u)$ dans B avec $\pi(t_0) = \pi(u) < \Lambda$.
 On aura $t_0(q'_1(x_1), \dots, q'_n(x_n)) \rightarrow q^\Lambda(u)$ dans le cas où $\pi(t_0) = \pi(u) = \Lambda$.
- $\sigma(q_i^\Lambda(x)) \rightarrow q^\Lambda(\delta(x))$ est une règle de R^Λ
 si et seulement si $\sigma(q_i(x)) \rightarrow q(\delta(x))$ est une règle de R .
- $\sigma(q_i^\Lambda(x)) \rightarrow q^\Lambda(\delta(x, x))$ est une règle de R^Λ
 si et seulement si $\sigma(q_i(x)) \rightarrow q(\delta(x, x))$ est une règle de R .
- $\sigma(q_i^\Lambda(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_{\theta(1)}, x_{\theta(2)})),$
 $\sigma(q_i^{<\Lambda}(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_{\theta(1)}, x_{\theta(2)}))$ et
 $\sigma(q_i^\Lambda(x_1), q_j^{<\Lambda}(x_2)) \rightarrow q^\Lambda(\delta(x_{\theta(1)}, x_{\theta(2)}))$ sont des règles de R^Λ
 si et seulement si
 $\sigma(q_i(x_1), q_j(x_2)) \rightarrow q(\delta(x_{\theta(1)}, x_{\theta(2)}))$, avec θ bijection, est une règle de R .
- $\sigma(q_i^\Lambda(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_1, x_1))$ et
 $\sigma(q_i^\Lambda(x_1), q_j^{<\Lambda}(x_2)) \rightarrow q^\Lambda(\delta(x_1, x_1))$ sont des règles de R^Λ
 si et seulement si
 $\sigma(q_i(x_1), q_j(x_2)) \rightarrow q(\delta(x_1, x_1))$ est une règle de R .
- $\sigma(q_i^\Lambda(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_2, x_2))$ et
 $\sigma(q_i^{<\Lambda}(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_2, x_2))$ sont des règles de R^Λ
 si et seulement si
 $\sigma(q_i(x_1), q_j(x_2)) \rightarrow q(\delta(x_2, x_2))$ est une règle de R .
- $\sigma(q_i^\Lambda(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_1))$ et
 $\sigma(q_i^\Lambda(x_1), q_j^{<\Lambda}(x_2)) \rightarrow q^\Lambda(\delta(x_1))$ sont des règles de R^Λ
 si et seulement si
 $\sigma(q_i(x_1), q_j(x_2)) \rightarrow q(\delta(x_1))$ est une règle de R .
- $\sigma(q_i^\Lambda(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_2))$ et
 $\sigma(q_i^{<\Lambda}(x_1), q_j^\Lambda(x_2)) \rightarrow q^\Lambda(\delta(x_2))$ sont des règles de R^Λ
 si et seulement si
 $\sigma(q_i(x_1), q_j(x_2)) \rightarrow q(\delta(x_2))$ est une règle de R .

Comme pour le cas des transducteurs descendants, la similitude des transformations réalisées dans le transducteur d'origine et dans toute forme Λ -semi-normalisée, nous

conduit à les identifier.

Nous montrons maintenant que, lorsque Λ est suffisamment grand, dans la forme Λ -semi-normalisée, les calculs aboutissant à des ensembles d'états globalement équivalents peuvent être réalisés en appliquant les mêmes torsions pour tous les couples d'arbres, sauf éventuellement pour les sous-arbres qui sont ensuite supprimés.

Nous illustrons, sur l'exemple suivant, le problème posé par la non-complétude.

Considérons le transducteur défini par:

$$\begin{aligned} (r1) : & \sigma(q_1(x), q_2(y)) \rightarrow q(e) \\ (r2) : & b(q_1(x), q_1(y)) \rightarrow q_1(b(x, y)) & (r3) : & \bar{a} \rightarrow q_1(\bar{a}) \\ (r4) : & a(q_2(x)) \rightarrow q_2(a(x)) & (r5) : & \bar{a} \rightarrow q_2(\bar{a}) \end{aligned}$$

$$\begin{aligned} (s1) : & \sigma(k_1(x), k_2(y)) \rightarrow k(e) \\ (s2) : & b(k_1(x), k_1(y)) \rightarrow k_1(b(y, x)) & (s3) : & \bar{a} \rightarrow k_1(\bar{a}) \\ (s4) : & a(k_2(x)) \rightarrow k_2(b(x, x)) & (s5) : & \bar{a} \rightarrow k_2(\bar{a}) \end{aligned}$$

Les états q et k sont équivalents. Cependant on remarquera que d'une part, pour les arbres de la forme $\sigma(b(t_1, t_2), a(t_3))$, des torsions différentes (règles (r2) et (s2)) sont appliquées lors de la transformation de $b(t_1, t_2)$ et d'autre part que les arbres produits à partir de $a(t_3)$ sont différents (règles (r4) et (s4)).

C'est en réalité sans conséquence sur les transformations aboutissant aux états q et k puisque ces sous-arbres sont ensuite supprimés. Seul importe, ici, le fait que $\text{dom}(\hat{B}_{q_1}) = \text{dom}(\hat{B}_{k_1})$ et $\text{dom}(\hat{B}_{q_2}) = \text{dom}(\hat{B}_{k_2})$.

Ainsi pour tout couple d'arbres (t, u) , les seules torsions significatives, et donc à prendre en compte, sont celles qui sont appliquées aux sous-arbres apparaissant effectivement dans l'arbre u produit par le transducteur.

Pour formaliser cette situation, nous associons au transducteur B^Λ considéré un transducteur, noté $B^{\Lambda, d}$, pour lequel les torsions appliquées lors des calculs sont exprimées dans les noeuds des arbres produits en sortie.

Construction de $B^{\Lambda, d}$

A partir de $B^\Lambda = \langle \Sigma^\Lambda, \Delta^\Lambda, Q^\Lambda, F^\Lambda, R^\Lambda \rangle$,

construction de $B^{\Lambda, d} = \langle \Sigma^\Lambda, \Delta^{\Lambda, d}, Q^\Lambda, F^\Lambda, R^{\Lambda, d} \rangle$ de la façon suivante:

- $t \rightarrow q(\langle u, id \rangle)$ est une règle de $R^{\Lambda, d}$ et $\langle u, id \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t \rightarrow q(u)$ est une règle de R^Λ , avec $t \in \Sigma^\Lambda$ et $u \in \Delta^\Lambda$

- $t(x) \rightarrow q(\langle u, id \rangle(x))$ est une règle de $R^{\Lambda, d}$ et $\langle u, id \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t(x) \rightarrow q(u(x))$ est une règle de R^Λ
- $t(x) \rightarrow q(\langle u, \theta \rangle(x, x))$ est une règle de $R^{\Lambda, d}$, $\theta = \langle 1; 1, 1 \rangle$ et $\langle u, \theta \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t(x) \rightarrow q(u(x, x))$ est une règle de R^Λ
- $t(x) \rightarrow q(\langle u, id \rangle)$ est une règle de $R^{\Lambda, d}$ et $\langle u, id \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t(x) \rightarrow q(u)$ est une règle de R^Λ avec $u \in \Delta^\Lambda$
- $t(x_1, x_2) \rightarrow q(\langle u, \theta \rangle(x_{\theta(1)}, x_{\theta(2)}))$ est une règle de $R^{\Lambda, d}$ et $\langle u, \theta \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t(x_1, x_2) \rightarrow q(u(x_{\theta(1)}, x_{\theta(2)}))$ est une règle de R_Λ
- $t(x_1, x_2) \rightarrow q(\langle u, \theta \rangle(x_{\theta(1)}))$ est une règle de $R^{\Lambda, d}$ et $\langle u, \theta \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t(x_1, x_2) \rightarrow q(u(x_{\theta(1)}))$ est une règle de R^Λ
- $t(x_1, x_2) \rightarrow q(\langle u, id \rangle)$ est une règle de $R^{\Lambda, d}$ et $\langle u, id \rangle \in \Delta^{\Lambda, d}$ si et seulement si $t(x_1, x_2) \rightarrow q(u)$ est une règle de R^Λ , avec $u \in \Delta^\Lambda$

Pour tout couple (t, u) de \hat{B}^Λ , notons (t, u^d) le couple correspondant de $\hat{B}^{\Lambda, d}$, c'est-à-dire pour lequel les torsions appliquées lors du calcul sont codées dans les noeuds de l'arbre u produit en sortie. Nous nous proposons maintenant de montrer que si Λ est suffisamment grand, alors $\hat{B}_E = \hat{B}_F \Leftrightarrow \hat{B}_E^{\Lambda, d} = \hat{B}_F^{\Lambda, d}$. En considérant de plus Λ supérieur au nombre d'états finitaires du transducteur, les résultats peuvent être établis dans le cas général des transducteurs non nécessairement infinitaires. Nous utilisons le lemme technique suivant:

Lemme 6.3.1 *Soient E et F deux ensembles d'états globalement équivalents de transducteur B^Λ . Lorsque Λ est suffisamment grand, pour tout couple (t, u) de $\hat{B}_E^\Lambda (= \hat{B}_F^\Lambda)$, pour tout préfixe u_0 de u , pour tout état q de Q , pour tout calcul*

$$t_0(t_1, \dots, t_n) \xrightarrow{*}_{B^\Lambda} t_0(q_1(u_1), \dots, q_n(u_n)) \xrightarrow{*}_{B^{\Lambda, d}} q(u_0^d(u_{\theta(1)}, \dots, u_{\theta(m)}))$$

il existe $k \in F$ tel que

$$t_0(t_1, \dots, t_n) \xrightarrow{*}_{B^\Lambda} t_0(k_1(u_1), \dots, k_n(u_n)) \xrightarrow{*}_{B^{\Lambda, d}} k(u_0^d(u_{\theta(1)}, \dots, u_{\theta(m)})).$$

PREUVE

Elle se fait par induction sur le préfixe u_0 de u .

Soit λ un majorant de l'ensemble des entiers $\lambda_{(E, F), (\sigma, \delta, \theta)}$ obtenus pour E et F ensembles d'états globalement équivalents et (σ, δ, θ) transformation finale, et soit N_F le nombre d'états finitaires du transducteur B . On prend $\Lambda = \sup(\lambda, N_F)$.

- La construction de B^Λ avec Λ suffisamment grand et le lemme 6.2.1 assurent

que la propriété est vraie pour $u_0 \in \Delta^\Lambda$.

• Supposons maintenant que la propriété est vraie pour tous les couples de la forme $(t, u_0(u_1, \dots, u_m))$ de \hat{B}_E^Λ et montrons qu'elle l'est encore pour ce préfixe u_0 augmenté d'une lettre.

Pour tout $j \in [m]$, considérons les ensembles

$$Q_j = \{q_{\theta(j)} \mid t_0(q_1(x_1), \dots, q_{\theta(j)}(x_{\theta(j)}), \dots, q_n(x_n)) \xrightarrow{*}_{B^{\Lambda,d}} q(u_0^d(x_{\theta(1)}, \dots, x_{\theta(j)}, \dots, x_{\theta(m)})), \\ \text{avec } q \in E \text{ et } \forall i \in [n], \hat{B}_{q_i} \neq \emptyset\}$$

et

$$K_j = \{k_{\theta(j)} \mid t_0(k_1(x_1), \dots, k_{\theta(j)}(x_{\theta(j)}), \dots, k_n(x_n)) \xrightarrow{*}_{B^{\Lambda,d}} k(u_0^d(x_{\theta(1)}, \dots, x_{\theta(j)}, \dots, x_{\theta(m)})), \\ \text{avec } k \in F \text{ et } \forall i \in [n], \hat{B}_{k_i} \neq \emptyset\}$$

Montrons d'abord que, pour tout $j \in [m]$, les ensembles d'états Q_j et K_j sont globalement équivalents.

Supposons, pour cela, qu'il existe $j \in [m]$ tel que Q_j et K_j ne soient pas globalement équivalents. Il existerait donc alors au moins un couple $(\bar{\tau}, \bar{\mu})$ élément de $\hat{B}_{Q_j} - \hat{B}_{K_j}$, c'est-à-dire tel que

$$t_0(\bar{u}_1, \dots, \bar{\tau}, \dots, \bar{u}_n) \xrightarrow{*}_{B^{\Lambda,d}} q(u_0^d(\bar{u}_{\theta(1)}, \dots, \bar{\mu}, \dots, \bar{u}_{\theta(m)})), \text{ avec } \forall i \in [n], \bar{t}_i \xrightarrow{*} q_i(\bar{u}_i)$$

alors que $(\bar{\tau}, \bar{\mu}) \notin \hat{B}_{K_j}$, et donc que ce calcul ne peut aboutir dans aucun état k de l'ensemble F .

Ceci contredit le fait que la propriété est supposée vraie pour le préfixe u_0 et donc pour tout $j \in [m]$ les ensembles d'états Q_j et K_j sont globalement équivalents.

Par conséquent, pour tout $j \in [m]$, pour tout état $q' \in Q_j$, il existe au moins un état $k' \in K_j$ tel que les transformations $t_{\theta(j)} \xrightarrow{*} q'(u_{\theta(j)})$ et $t_{\theta(j)} \xrightarrow{*} k'(u_{\theta(j)})$ sont réalisées en appliquant la même torsion finale (premier point de cette preuve). La propriété est donc vraie pour un préfixe de u contenant une lettre de plus que u_0 (la racine de $u_{\theta(j)}$). \square

Proposition 6.3.1 *Soit B un transducteur ascendant lettre à lettre. Il existe un entier Λ tel que, pour tout couple (E, F) d'ensembles d'états, $\hat{B}_E^{\Lambda,d} = \hat{B}_F^{\Lambda,d} \Leftrightarrow \hat{B}_E = \hat{B}_F$.*

ELEMENTS DE PREUVE:

On a en effet, de façon évidente, $\hat{B}_E^{\Lambda,d} = \hat{B}_F^{\Lambda,d} \Rightarrow \hat{B}_E = \hat{B}_F$. Réciproquement, on utilise le lemme précédent, en prenant comme préfixe de tout arbre t , l'arbre t lui-même. \square

6.3.2 Formes Λ -normalisées

La proposition 6.3.1 nous permet entre autres d'affirmer que, pour Λ suffisamment grand, si lors d'un calcul $t \xrightarrow{*} q(u)$, avec $q \in E$, certains sous-arbres de t sont supprimés (respectivement dupliqués) alors il existe au moins un état $k \in F$ (E et F sont des ensembles d'états globalement équivalents) tel que lors du calcul $t \xrightarrow{*} k(u)$ les mêmes sous-arbres sont supprimés (respectivement dupliqués). Cela nous permet, tout en préservant l'équivalence des états, de construire une forme Λ -normalisée dans laquelle les torsions sont codées dans les noeuds des arbres produits en sortie mais ne sont plus appliquées effectivement. Cette forme est ainsi un marqueur, c'est-à-dire entre autres un transducteur complet et linéaire. La construction de toute forme Λ -normalisée intègre donc une "complétion" et une "suppression de la non-linéarité".

La "complétion" du transducteur ascendant $B^{\Lambda,d}$ ne peut cependant pas se faire aussi simplement que dans le cas des transducteurs descendants. En effet, un transducteur ascendant ne supprime un sous-arbre qu'après l'avoir reconnu. Pour obtenir une forme complète du transducteur on est, pour cette raison, amené à conserver les sous-arbres qui devaient être supprimés et à les faire apparaître sans transformation dans l'arbre produit en sortie.

Ainsi, par exemple

à partir de la règle $\sigma \rightarrow q(\delta)$ de $B^{\Lambda,d}$

$$\begin{array}{ccc} & \sigma & \\ & \swarrow \quad \searrow & \\ q_1(x) & & q_2(y) \end{array} \quad \rightarrow \quad \begin{array}{c} q(\delta) \\ | \\ x \end{array}$$

on construit la règle $\sigma \rightarrow q(\delta')$

$$\begin{array}{ccc} & \sigma & \\ & \swarrow \quad \searrow & \\ q_1(x) & & \bar{q}_2(y) \end{array} \quad \rightarrow \quad \begin{array}{cc} q(\delta') & \\ / \quad \backslash & \\ x & y \end{array}$$

où la transformation associée à \bar{q}_2 est l'ensemble $\{(t, t) \mid t \in \text{dom}(\hat{B}_{q_2}^{\Lambda,d})\}$.

D'autre part, un transducteur ascendant ne peut réaliser, par application d'une règle non-linéaire, que des copies identiques puisque les duplications sont faites après transformations. Les torsions étant déjà codées dans les noeuds des arbres, il est donc possible de ne pas réaliser effectivement ces duplications, tout en préservant l'équivalence des états.

Phase de complétion

A toute forme Λ -semi-normalisée $B^{\Lambda,d} = \langle \Sigma, \Delta, Q, F, R \rangle$, on associe le transducteur complet $B'^{\Lambda,d} = \langle \Sigma, \Delta', Q', F', R' \rangle$ défini par:

- pour tout état q de Q , q et \bar{q} sont des états de Q' . Ils sont de plus dans F' si q est dans F .
- toute règle complète de R est une règle de R'
- $\sigma(q_1(x), \bar{q}_2(y)) \rightarrow q(\delta'(x, y))$ et $\sigma(\bar{q}_1(x), \bar{q}_2(y)) \rightarrow \bar{q}(\sigma(x, y))$ sont des règles de R' , δ' et σ sont des lettres de Δ'
si et seulement si $\sigma(q_1(x), q_2(y)) \rightarrow q(\delta(x))$ est une règle de R
- $\sigma(\bar{q}_1(x), q_2(y)) \rightarrow q(\delta'(x, y))$ et $\sigma(\bar{q}_1(x), \bar{q}_2(y)) \rightarrow \bar{q}(\sigma(x, y))$ sont des règles de R' , δ' et σ sont des lettres de Δ'
si et seulement si $\sigma(q_1(x), q_2(y)) \rightarrow q(\delta(y))$ est une règle de R
- $\sigma(\bar{q}_1(x), \bar{q}_2(y)) \rightarrow q(\delta'(x, y))$ et $\sigma(\bar{q}_1(x), \bar{q}_2(y)) \rightarrow \bar{q}(\sigma(x, y))$ sont des règles de R' , δ' et σ sont des lettres de Δ'
si et seulement si $\sigma(q_1(x), q_2(y)) \rightarrow q(\delta)$ est une règle de R

- $\sigma(\bar{q}_1(x)) \rightarrow q(\delta'(x))$ et $\sigma(\bar{q}_1(x)) \rightarrow \bar{q}(\sigma(x))$ sont des règles de R' , δ' et σ sont des lettres de Δ'
si et seulement si $\sigma(q_1(x)) \rightarrow q(\delta)$ est une règle de R
- de plus
- pour toute règle $\sigma \rightarrow q(\delta)$ de R , $\sigma \rightarrow \bar{q}(\sigma)$ est une règle de R' et σ est une lettre de Δ'
- pour toute règle $\sigma(q_1(x)) \rightarrow q(\delta(x))$ de R , $\sigma(\bar{q}_1(x)) \rightarrow \bar{q}(\sigma(x))$ est une règle de R' et σ est une lettre de Δ'
- pour toute règle $\sigma(q_1(x)) \rightarrow q(\sigma(x, x))$ de R , $\sigma(\bar{q}_1(x)) \rightarrow \bar{q}(\sigma(x))$ est une règle de R^d
- pour toute règle $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_{\theta(1)}, x_{\theta(2)}))$ de R , $\sigma(\bar{q}_1(x_1), \bar{q}_2(x_2)) \rightarrow \bar{q}(\sigma(x_{\theta(1)}, x_{\theta(2)}))$ est une règle de R' et σ est une lettre de Δ'

Suppression de la non-linéarité et construction du marqueur

Au transducteur complet $B'^{\Lambda, d}$ construit précédemment, on associe le marqueur B_{Θ}^{Λ} défini par

- $\sigma \rightarrow q(\langle \delta, id \rangle)$ est une règle de R_{Θ}^{Λ} et $\langle \delta, id \rangle \in \Delta_{\Theta}$ si et seulement si $\sigma \rightarrow q(\delta)$ est une règle de R'^{Λ}
- $\sigma(q_1(x)) \rightarrow q(\langle \delta, id \rangle(x))$ est une règle de R_{Θ}^{Λ} et $\langle \delta, id \rangle \in \Delta_{\Theta}$ si et seulement si $\sigma(q_1(x)) \rightarrow q(\delta(x))$ est une règle de R'^{Λ}
- $\sigma(q_1(x)) \rightarrow q(\langle \delta', \theta \rangle(x))$, avec $\theta = \langle 1; 1, 1 \rangle$ est une règle de R_{Θ}^{Λ} et $\langle \delta, \theta \rangle \in \Delta_{\Theta}$ si et seulement si $\sigma(q_1(x)) \rightarrow q(\delta(x, x))$ est une règle de R'^{Λ}
- $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\langle \delta, \theta \rangle(x_1, x_2))$ est une règle de R_{Θ}^{Λ} , $\langle \delta, \theta \rangle \in \Delta_{\Theta}$ si et seulement si $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_{\theta(1)}, x_{\theta(2)}))$ est une règle de R

Proposition 6.3.2 *Soit B un transducteur ascendant lettre à lettre. Il existe un entier Λ tel que pour tout couple (E, F) d'ensembles d'états, $\hat{B}_E = \hat{B}_F \Leftrightarrow \hat{B}_{\Theta_E}^{\Lambda} = \hat{B}_{\Theta_F}^{\Lambda}$.*

ELEMENTS DE PREUVE:

D'une part, la proposition 6.3.1 nous assure de l'existence d'un entier Λ tel que pour tout couple d'ensembles d'états (E, F) $\hat{B}_E^{\Lambda, d} = \hat{B}_F^{\Lambda, d} \Leftrightarrow \hat{B}_E = \hat{B}_F$.

D'autre part, par la méthode de "complétion" choisie, pour toute règle de la forme $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_1))$, nous substituons à l'état q_2 un état \bar{q}_2 tel que $t \in \text{dom}(\hat{B}_{q_2}^{\Lambda, d}) \Leftrightarrow t \in \text{dom}(\hat{B}_{\bar{q}_2}^{\Lambda})$ (on procède de façon analogue pour les règles de la forme $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_2))$ ou $\sigma(q_1(x_1), q_2(x_2)) \rightarrow q(\delta(x_1))$). L'équivalence des états est ainsi préservée. De la même façon, parce qu'un transducteur ascendant ne peut réaliser que des copies identiques, la suppression de la non-linéarité qui consiste ici à ne conserver qu'un seul exemplaire de ces copies ne modifie pas l'équivalence des états. On a par conséquent $\hat{B}_E = \hat{B}_F \Leftrightarrow \hat{B}_{\bar{E}}^{\Lambda} = \hat{B}_E^{\Lambda}$. \square

6.4 Décidabilité de l'équivalence

Les résultats obtenus précédemment pour des ensembles d'états globalement équivalents d'un transducteur nous permettent d'établir le résultat suivant

Théorème 6.4.1 *L'équivalence pour les transducteurs ascendants lettre à lettre est décidable.*

PREUVE:

Soient $B = \langle \Sigma, \Delta, Q, F, R \rangle$ et $B' = \langle \Sigma, \Delta, Q', F', R' \rangle$ avec $Q \cap Q' \neq \emptyset$ (en renommant les états d'un des deux transducteurs, on peut toujours se ramener à cette situation). La proposition 6.3.2 appliquée aux ensembles d'états finaux F et F' et la décidabilité de l'équivalence pour les marqueurs ascendants 2.2.2 nous permettent d'obtenir la semi-décidabilité de l'équivalence. La non-équivalence étant semi-décidable, nous concluons. \square

Conclusion

Nous avons établi, dans cette thèse, des résultats de décidabilité pour les transducteurs lettre à lettre d'arbres (à l'exception des transducteurs non-linéaires et non-complets). Citons quelques applications immédiates de ces résultats:

- Tester la "pertinence" d'une règle.

Soit $T = \langle \Sigma, \Delta, Q, I, R \rangle$ un transducteur de **LT-LL**, **CT-LL** ou **B-LL** et soit r une règle de R . Considérons maintenant le transducteur $T' = \langle \Sigma, \Delta, Q, I, R - \{r\} \rangle$. Si T et T' sont équivalents, alors la règle r n'est pas "pertinente". L'équivalence étant décidable pour ces classes de transducteurs, on peut donc décider de la "pertinence" d'une règle.

- Décider de l'équivalence de transducteurs de **LT-LL**, **CT-LL** ou **B-LL** modulo la commutativité de certains opérateurs de l'alphabet de sortie.

Considérons par exemple le transducteur lettre à lettre, décrit dans l'introduction de ce mémoire, et dont la transformation associée est composée des couples (t, u) où u peut être interprété comme la factorielle de l'entier correspondant à l'arbre d'entrée t . La prise en compte de la commutativité de l'opérateur associé au symbole $*$ permettrait de considérer comme équivalents ce transducteur et le même transducteur dans lequel on aurait substitué la règle $f(S(x)) \rightarrow *(f(x), q(x))$ à la règle $f(S(x)) \rightarrow *(q(x), f(x))$.

Pour réduire le problème de l'équivalence de transducteurs modulo la commutativité de certains opérateurs à l'équivalence dans **LT-LL**, **CT-LL** ou **B-LL**, il nous suffit, par exemple pour un transducteur descendant, d'ajouter pour toute règle de la forme $q(\sigma(x, y)) \rightarrow \delta(q_1(x), q_2(y))$ (où δ est interprété comme un opérateur commutatif) la règle $q(\sigma(x, y)) \rightarrow \delta(q_2(y), q_1(x))$. On remarquera que, de la même façon, on peut prendre en compte la commutativité de certains opérateurs de l'alphabet d'entrée.

Pour clore ce mémoire, citons maintenant les développements que nous souhaitons

donner à notre travail:

- Obtenir un résultat pour le cas général (c'est-à-dire associant la non-complétude et la non-linéarité) descendant lettre à lettre.
- Etudier les liens avec la programmation (programmation logique par exemple). C'étaient d'ailleurs les motivations de notre travail mais nous n'en avons étudié que l'aspect théorique.
- Déterminer la complexité des algorithmes de décision. La décision de l'équivalence est obtenue ici à partir de deux algorithmes complémentaires de semi-décision. Ils ne donnent malheureusement pas une borne. Dans ce genre de problèmes, on a souvent une borne dans le pire des cas impraticable. Il serait intéressant d'obtenir un algorithme de nettoyage empiriquement efficace.
- Considérer des transducteurs "quasi-plats", c'est à dire où les membres droits des règles ne sont pas nécessairement réduits à une lettre, les variables apparaissant toujours à la profondeur 1.

Index

- A_τ , 78
- EqD_n , 68
- $IRR(E)$, 19
- T_Σ , 17
- $T_\Sigma(X_p)$, 17
- $[p]$, 18
- $[t, u]$, 26
- Σ_m , 17
- \bar{q} , 121
- \rightarrow_S , 19
- \rightarrow , 19
- $\xrightarrow{*}_S$, 19
- $\pi(t)$, 18
- $\xrightarrow{*}$, 22
- \mapsto , 21
- \hat{T} , 22
- $\hat{T}_{q(\sigma, \delta, \theta)}$, 23
- \hat{T}_q , 22
- \hat{t} , 19
- u^d , 118
- $\mathcal{F}(A)$, 20
- \mathcal{S}_{1r} , 75
- $\mathcal{V}(t)$, 18
- B-LL**, 111
- CT-LL**, 93
- LCB-LL**, 50
- LCT-LL_i**, 31
- LCT-LL**, 29
- LT-LL_i**, 53
- LT-LL**, 51
- RATEG**, 65
- REC_≠**, 66
- REC_≠**, 69
- REC_τ**, 73
- REC**, 20
- T-LAB**, 25
- alphabet gradué, 17
- arité, 17
- automate
 - ascendant, 20
 - descendant, 20
- automate des formes normales, 78
- codomaine, 22
- complétion, 59
- confluent, 19
- description d'équivalence, 68
- domaine, 22
- etats globalement équivalents, 23
- finitaire, 22
- forêt, 20
 - reconnaissable, 20
- forme
 - Λ -normalisée, 37
 - Λ -semi-normalisée, 37
- forme irréductible, 19
- forme normale, 19
- image finie, 22

- infinitaire, 22
- marqueur, 25
- membre gauche, droit, 18
- noethérien, 19
- non-dégénéré, 19
- profondeur, 18
- règle de réécriture, 18
 - complète, 18
 - linéaire à droite, 19
 - linéaire à gauche, 19
- rang, 17
- relabeling, 25
- sous-terme propre, 18
- système de réécriture, 19
 - 1-réduit, 75
 - lettre à lettre, 19
- terme, 17
 - clos, 17
 - irréductible, 19
 - linéaire, 18
- torsion, 18
- transducteur
 - ascendant, 22
 - déterministe, 22
 - descendant, 21
 - lettre à lettre, 21
- transducteurs équivalents, 23
- transduction, 22
- transformation
 - finale, 111
 - initiale, 23
- transformation associée, 22

Bibliographie

- [AU69a] A.V. Aho et J.D. Ullman. Syntax directed translations and the pushdown assembler. *Journal Computer System Science* 3. 1969.
- [AU69b] A.V. Aho et J.D. Ullman. Properties of syntax directed translations. *Journal Computer System Science* 3. 1969.
- [AnD93] Y. André et M. Dauchet. Decidability of equivalence for a class of non deterministic tree transducers. *Rapport technique IT-243 LIFL. 1993*. Soumis à la revue Rairo - Informatique Théorique et Applications. 1993.
- [AB93a] Y. André et F. Bossut. Decidability of equivalence for linear letter to letter top-down tree transducers. *Rapport technique IT-248 LIFL. Université de Lille 1. 1993. Proceedings of FCT'93. LNCS 710. pp 142-151*.
- [AB93b] Y. André et F. Bossut. Tree automata with equivalence constraints on direct subterms. *Rapport technique IT-250 LIFL. Université de Lille 1. 1993*.
- [AB93c] Y. André et F. Bossut. Decidability of equivalence for variable-preserving letter to letter top-down tree transducers. *Rapport technique IT-251. LIFL. Université de Lille 1. 1993*.
- [ArD79a] A. Arnold et M. Dauchet. Théorie des magmoïdes (1). *RAIRO Informatique théorique* 12. 1979. pp 235-257.
- [ArD79b] A. Arnold et M. Dauchet. Théorie des magmoïdes (2). *RAIRO Informatique théorique* 13. 1979. pp 135-154.
- [ArD82] A. Arnold et M. Dauchet. Morphismes et bimorphismes d'arbres. *Theoretical Computer Science* 20. 1982. pp 33-93

- [AuB88] J.M. Autebert et L. Boasson. Transductions rationnelles. Applications aux langages algébriques. *Collection ERI, Masson*. 1988.
- [Ber79] J. Berstel. Transductions and context-free languages. *Teubner Verlag Stuttgart*, 1979.
- [Bog90] B. Bogaert. Automates d'arbres avec tests d'égalité. *Thèse université de Lille 1*. 1990.
- [BT92] B. Bogaert et S. Tison. Equality and disequality constraints on direct subterms in tree automata. *Proceedings of STACS'1992. LNCS 577*. pp 161-171.
- [Boz92] S. Bozapalidis. Alphabetic tree relations. *TCS 99*. pp 177-211. 1992.
- [Car93] A.C. Caron. Structures et décision en réécriture. *Thèse université de Lille 1*. 1993.
- [CCD93] A.C. Caron, J.L. Coquidé et M. Dauchet. Encompassment Properties and Automata with Constraints. *Proceedings of RTA '93. LNCS 690*, pp 328-341.
- [Com90] H. Comon. Equational formulas on order-sorted algebras. *Proceedings of ICALP'90*. pp 674-688. 1990.
- [CF82] B. Courcelle et P. Franchi-Zannettachi. Attribute grammars and recursive program schemes, part I. *TCS 17*. pp 163-191. 1982.
- [Dau75] M. Dauchet. Transductions inversibles de forêts. *Thèse spécialité. Université de Lille 1*. 1975.
- [Dau77] M. Dauchet. Transductions de forêts. Bimorphismes de magmoïdes. *Thèse d'état. Université de Lille 1*. 1977.
- [DT90] M. Dauchet et S. Tison. The theory of ground rewrite systems is decidable. *Proceedings of 5th IEEE Symposium on Logic in Computer Sciences*, Philadelphia, June 4-7, pp 242-248, 1990.
- [Don70] J. Doner. Tree acceptors and some of their applications. *Journal Computer System Science 4*, pp 406-451. 1970.
- [Eng75] J. Engelfriet. Bottom-up and top-down tree transformations: a comparison. *Mathematical system theory*. Vol 9. pp 198-231. 1975.

- [Eng77] J. Engelfriet. Top-down tree transducers with regular look-ahead. *Mathematical System Theory*. Vol 10. pp 289-303. 1977.
- [Eng78] J. Engelfriet. A hierarchy of tree transducers. *Proc. 3ème colloque sur les Arbres en Algèbre et en Programmation*. Université de Lille 1. 1978.
- [Eng80] J. Engelfriet. Some open questions and recent results on tree transducers and tree languages. *Formal language theory*, Academic press, pp 241-286. 1980.
- [Esi80] Z. Esik. Decidability results concerning tree transducers I. *Acta Cybernetica*, Tom.5, Fasc.1, pp 1-20, Szeged, 1980.
- [FS90] C. Frougny et J. Sakarovitch. Relations rationnelles à délai borné. Actes des Journées Montoises, pp 49-52. Université de Mons-Hainaut, Belgique. 1990. *Proceedings of International Colloquium on Words, Languages and Combinatorics*, Kyoto 1990.
- [Fv89] Z. Fülöp et S. Vágvölgyi. Variants of top-down tree transducers with look-ahead. *Math. Systems Theory* 21. 1989. pp 125-145.
- [Fv90] Z. Fülöp et S. Vágvölgyi. Top-down tree transducers with deterministic top-down look-ahead. *Inform. Process. Letter* 33. 1989/90. pp 3-5.
- [GS84] F. Gecseg et M. Steinby. Tree automata. Akademiai Kiado, Budapest. 1984.
- [Gri68] T. Griffiths. The unsolvability of the equivalence for λ -free non-deterministic generalized machines. *J. Assoc. Comput. Mach.* v15. 1968. pp 409-413.
- [Hue80] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting system. *J.A.C.M.* 27. pp 797-821. 1980.
- [LM93] D. Lugiez et J.L. Moysset. Complement problems and tree automata in AC-like theories. *Proceedings of STACS 93*. pp 515-524.
- [Mon81] J. Mongy. Transformations de noyaux reconnaissables d'arbres. Forêts RA-TEG. *Thèse Université de Lille 1*. 1981.
- [Niv68] M. Nivat. Transduction des langages de Chomsky. *Annales Institut Fourier* 18. Grenoble. 1968. pp 339-456.

- [Rab69] M.O. Rabin. Decidability of second-order theories and automata on infinite trees. 1969. *Trans. Amer. Math. Soc.* 141, pp1-35.
- [Rao93] J. C. Raoult. Recursively defined tree transductions. *Proceedings of RTA'93. LNCS 690. pp 343-357.*
- [Rou68] W.C. Rounds. Trees, transducers and transformations. *Ph.D. Thesis.* Standford University. 1968.
- [Sei90] H. Seidl. Equivalence of finite-valued bottom-up finite state tree transducers is decidable. Universität des Saarlandes. Saarbrücken. Germany. *Proceeding of CAAP 90. LNCS 431, pp 269-284.*
- [Sei92] H. Seidl. Single-valuedness of tree transducers is decidable in polynomial time. *Theoretical Computer Science 106.* 1992. pp 135-181.
- [Sei 93] H. Seidl. When is a functional tree transduction deterministic? *Proceedings of CAAP'93. LNCS 668.* pp 251-265.
- [Tha69] J.W. Thatcher. Transformations and translations from the point of view of generalized finite automata theory. *1st A.C.M. proc on theory of computing,* pp 129-142. 1969.
- [Tha70] J.W. Thatcher. *Generalized² sequential machines.* *Journal Computer System Science* 3. 1969.
- [Tis90] S. Tison. Automates comme outil de décision dans les arbres. Dossier d'habilitation à diriger des recherches. Lille 1990.
- [Tom91] M. Tommasi. Tree automata with equality tests between first cousins. *Mémoire de D.E.A.* Université de Lille 1. 1991.
- [Vág92] S. Vágvölgyi. Top-down tree transducers with two-way tree walking lookahead. *Theoretical Computer Science 93.* 1992. pp 43-74.
- [Zac78] Z. Zachar. The solvability of the equivalence problem for deterministic frontier-to-root tree transducers. *Acta Cybernetica,* v.4, 1978, pp 167-177.

