Document generated on 08/09/2025 10:44 p.m.

Revue québécoise de linguistique

Un exemple d'exploration linguistique du français à l'aide de Logo

Louisette Emirkanian and Lorne H. Bouchard

Volume 14, Number 2, 1985

Linguistique et informatique

URI: https://id.erudit.org/iderudit/602542ar
DOI: https://doi.org/10.7202/602542ar

See table of contents

Publisher(s)

Université du Québec à Montréal

ISSN

0710-0167 (print)
1705-4591 (digital)

Explore this journal

Cite this article

Emirkanian, L. & Bouchard, L. H. (1985). Un exemple d'exploration linguistique du français à l'aide de Logo. *Revue québécoise de linguistique*, *14*(2), 145–156. https://doi.org/10.7202/602542ar

Tous droits réservés © Université du Québec à Montréal, 1985

This document is protected by copyright law. Use of the services of Érudit (including reproduction) is subject to its terms and conditions, which can be viewed online.

https://apropos.erudit.org/en/users/policy-on-use/

This article is disseminated and preserved by Érudit.

Érudit is a non-profit inter-university consortium of the Université de Montréal, Université Laval, and the Université du Québec à Montréal. Its mission is to promote and disseminate research.

https://www.erudit.org/en/

UN EXEMPLE D'EXPLORATION LINGUISTIQUE DU FRANÇAIS À L'AIDE DE LOGO*

Louisette Émirkanian
Lorne H. Bouchard

Dans cet article, nous présentons un programme écrit en Logo qui permet de rendre compte d'une partie d'un phénomène syntaxique du français, la coordination. Après avoir examiné les phénomènes de coordination que nous allons traiter en Logo, nous parlerons du programme constitué de deux procédures principales : GÉNÈRE[1] et COORDONNE. Le programme génère d'abord de façon aléatoire des phrases coordonnées susceptibles d'avoir une forme réduite correspondante (GÉNÈRE); puis, il analyse ces phrases, et enfin, propose leur contrepartie réduite (COORDONNE).

Avant d'aborder l'étude des phénomènes de coordination, nous voudrions justifier notre choix du langage Logo.

1. Le langage Logo

Logo est un langage de programmation conçu pour servir de support concret à la résolution de problèmes (Papert 1980). Le langage Logo est rapidement devenu un langage de premier choix pour initier les enfants à l'ordinateur, principalement grâce à la simplicité de la géométrie de la tortue; comme le signalent Abelson et di Sessa (1980), cet environnement est cependant suffisamment riche pour maintenir l'intérêt des étudiants en sciences du premier cycle universitaire. Mais Logo, c'est beaucoup plus que la géométrie de la tortue. Logo est basé sur Lisp, langage de programmation de l'intelligence artificielle (Winston 1977); en effet, Logo est une version

* Une version préliminaire de cette étude a fait l'objet d'une communication au 52e Congrès de l'ACFAS.

1. Les titres des procédures et certaines instructions sont notées avec accent dans ce texte afin d'en faciliter la lecture.

«sucrée» de Lisp. Par rapport à Lisp, Logo réduit au minimum l'usage des parenthèses; l'interface de Logo avec l'utilisateur est plus amical. De plus, l'environnement interactif de Logo sert de support à la programmation exploratoire telle que définie par Sheil (1984).

2. Les phénomènes de coordination

Nous allons traiter ici d'un phénomène particulier de la coordination en français. On distingue traditionnellement des coordinations de constituants et des coordinations de non-constituants. (1) et (2) sont deux exemples de coordination de constituants :

(1) Pierre donne une maison à Marie et vend une voiture à François.

(2) Les enfants et les étudiants programment en Logo.

En (1), deux syntagmes verbaux sont directement coordonnés, en (2), il s'agit de deux syntagmes nominaux. En (3), (4) et (5), en revanche, nous sommes en présence de coordinations de non-constituants :

(3) Pierre mange une pomme et Jacques, une poire.

(4) Pierre fait du ski l'hiver et de la plongée l'été.

(5) Michèle économise et François dépense, tout l'argent de la semaine.

Dans ces trois exemples, deux phrases sont coordonnées avec dans chacune d'elles des éléments ayant leur contrepartie dans l'autre. En (5) par exemple, *Michèle économise* et *François dépense* ne constituent pas deux constituants directement coordonnés. Ce sont ces coordinations de non-constituants qui feront l'objet de cette étude. Nous nous intéresserons ici aux phrases coordonnées complètes et réduites en considérant non pas que les formes réduites sont dérivées des formes complètes mais qu'elles sont équivalentes[2].

3. La procédure : GÉNÈRE

Il s'agit dans un premier temps de générer de façon aléatoire des phrases coordonnées avec contraintes. Nous posons deux contraintes : d'une part, il faut que les deux phrases coordonnées générées aient au moins un élément identique pour qu'elles puissent avoir une forme réduite

2. Nous considérerons pas ici l'aspect linguistique proprement dit; nous nous appuyons sur les résultats d'une étude antérieure (Émirkanian 1979).

équivalente; d'autre part, il faut que les deux phrases coordonnées générées aient au moins deux éléments différents pour que la forme réduite leur correspondant ne soit pas une coordination de constituants. La première procédure, GÉNÈRE, doit générer des phrases coordonnées en tenant compte de ces deux contraintes. Nous partons d'une grammaire très simple où S se réécrit P1 et P2, où P se réécrit SN1 SV ADV et où SV se réécrit V1 SN2, ce qui nous donne les constituants suivants : ADV1, ADV2, SN11, SN12, SN21, SN22, V11 et V12. Nous dressons alors la liste RELATIONS de tous les couples de constituants :

(6) MAKE "RELATIONS [[ADV1 ADV2] [SN11 SN12] [SN21 SN22] [SV11 SV12] [V11 V12]]

En Logo, l'instruction MAKE permet d'affecter une valeur à une variable. La variable RELATIONS prend ici comme valeur la liste des couples de constituants susceptibles d'être identiques dans une phrase.

Illustrons comment le programme détermine les contraintes. Dans un premier temps, il choisit au hasard un de ces couples pour déterminer les éléments identiques puis l'enlève pour qu'il ne soit pas choisi une seconde fois. En (7), l'instruction conditionnelle IF permet d'exécuter une liste d'instructions si sa condition est vérifiée. Ainsi, si le couple [SV11 SV12] est dans la relation IDENTIQUES, alors le programme enlève d'une part le couple [V11 V12] et d'autre part le couple [SN21 SN22] de la variable RELATIONS.

(7) IF MEMBERP [SV11 SV12] : IDENTIQUES [MAKE "IDENTIQUES [[V11 V12] [SN21 SN22]] MAKE "RELATIONS ENLÈVE [V11 V12] : RELATIONS MAKE "RELATIONS ENLÈVE [SN21 SN22] : RELATIONS]

Il s'agit dans un second temps de choisir les deux couples qui déterminent les éléments différents. Le premier couple est choisi en puisant dans RELATIONS; il est enlevé; puis un second couple est choisi et également enlevé. Il se pourrait alors que les couples déterminant la différence soient [V11 V12] et [SN21 SN22]. Dans ce cas, [V1 SN2] formant un constituant, le syntagme verbal, le programme prévoit alors d'aller chercher un troisième couple pour déterminer les éléments différents. Voici un exemple de ce que nous pouvons obtenir à cette étape (8) tel qu'imprimé par le programme :

(8) IDENTIQUE IS [[V11 V12]]
DIFFERENTS IS [[ADV1 ADV2] [SN21 SN22]]
RELATIONS IS [[SN11 SN12]]

Ici nous obtenons un couple pour déterminer les éléments identiques (les verbes), deux pour déterminer les éléments différents (les adverbes et les syntagmes nominaux objets). Le couple restant pourra être constitué d'éléments identiques ou différents. Une fois les contraintes déterminées, le programme procède à la génération proprement dite (GÉN). La fonction GÉN : X : Y est une fonction utilitaire que nous avons définie; elle permet d'affecter une valeur à un constituant : X différente de celle du constituant : Y. Cette fonction GÉN est au cœur de la génération sous contraintes. Par exemple GÉN «V11 [ ] affectera à V11 une valeur au hasard retournée par le générateur G.V1; en revanche, GÉN «SN22 «SN21 affectera à SN22 la première valeur au hasard retournée par le générateur G.SN2 différente de la valeur du constituant SN21. Nous générons pour chacun des membres gauches de tous les couples une valeur en allant piocher au hasard dans la liste des V1, des ADV, des SN2, etc. selon le cas. Pour le couple dans la relation d'identité, la valeur du membre droit devra être identique à celle du membre gauche; pour les deux couples dans la relation de différence, la valeur des membres droits devra être différente de celle des membres gauches. Poursuivant avec l'exemple (8), nous obtenons à cette étape :

(9) V12 IS [achète]
V11 IS [achète]
SV12 IS [ ]
SV11 IS [ ]
SN22 IS [du foie gras du Gers]
SN21 IS [des bonbons à la réglisse]
SN12 IS [ ]
SN11 IS [ ]
ADV2 IS [au restaurant du coin]
ADV1 IS [au sous-sol de l'église]

Nous affectons finalement à tous les constituants restants une valeur générée au hasard, sans contrainte. Une fois tous les constituants déterminés, nous appliquons la fonction CONSTRUIT, qui nous permet d'élaborer une structure arborescente à partir du patron ou du schéma de l'arborescence. En fait, le patron est une arborescence incluant des éléments (précédés du signe «$») qui désignent des variables. L'opération CONSTRUIT consiste à copier le patron en substituant aux variables leur valeur. À partir des valeurs de (9), l'instruction (10) produit l'arborescence (11) :

(10) OP CONSTRUIT [S [P [SN1 $SN11] [SV1 [V1 $V11] [SN2 $SN21]] [ADV $ADV1]] et [P [SN1 $SN12] [SV1 [V1 $V12] [SN2 $SN22]] [ADV $ADV2]]]

(11) S [P [SN1 Gertrude] [SV1 [V1 achète] [SN2 des bonbons à la réglisse]] [ADV au sous-sol de l'église]] et [P [SN1 la voisine] [SV1 [V1 achète] [SN2 du foie gras du Gers]] [ADV au restaurant du coin]]

L'instruction OP en Logo sert à identifier le résultat qui est retourné par une fonction.

Nous remarquons, dans l'exemple (11), que les SN1 sont différents dans les deux coordonnées. Voilà définie la première procédure GÉNÈRE, celle qui génère avec contraintes des phrases coordonnées susceptibles d'être réduites.

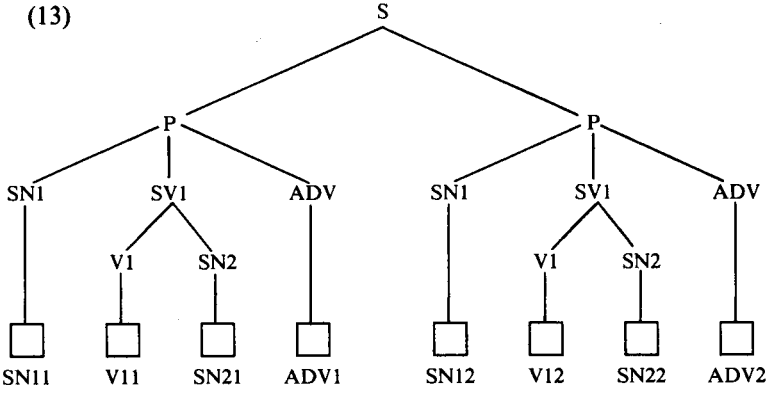
4. La procédure COORDONNE

Cette seconde procédure a pour but de générer les formes réduites. Elle contient un certain nombre de sous-procédures. La première est ANALYSE. La fonction ANALYSE, que nous avons définie, est l'inverse de la fonction CONSTRUIT. ANALYSE met en correspondance un patron (12) et une donnée. Cette donnée est une arborescence dont (11) est un exemple :

(12) [S [P [SN1 $SN11] [SV1 $SV11] [ADV $ADV1]] et [P [SN1 $SN12] [SV1 $SV12] [ADV $ADV2]]]

Les objets précédés du signe «$» désignent des variables Logo auxquelles seront affectées des valeurs. Nous pourrions représenter (12) ainsi :

(13)



Lorsque la structure produite par GÉNÈRE est mise en correspondance avec le patron (12), la liste [Gertrude] sera mise en correspondance avec $SN11, la liste [achète] avec $V11, etc. La procédure ANALYSE n'est pas strictement nécessaire ici puisque nous avons déjà généré une bonne structure; mais elle est utile si, par exemple, les données n'avaient pas été produites par GÉNÈRE.

Nous avons distingué différents types de coordination; nous les présentons en (14); si les syntagmes nominaux sujets sont identiques, les syntagmes verbaux et les adverbes différents, nous appliquons COORDONNE1; si les syntagmes nominaux sujets et les adverbes sont différents et les syntagmes verbaux identiques, nous appliquons COORDONNE2; si les syntagmes nominaux sujets et les syntagmes verbaux sont différents et les adverbes identiques, nous appliquons COORDONNE3; dans les autres cas, si les trois constituants sont différents, nous appliquons COORDONNE4. Le AND correspond au connectif logique ET, le NOT, à NON; STOP est une instruction utilisée pour mettre fin (localement) à l'exécution d'une procédure. On vérifie alors que l'explication donnée ci-dessus paraphrase le fragment de programme (14) :

(14) IF (AND : SN11 = : SN12 NOT : SV11 = : SV12 NOT
: ADV1 = : ADV2) [COORDONNE1 STOP]
IF (AND NOT : SN11 = : SN12 : SV11 = : SV12 NOT
: ADV1 = ADV12) [COORDONNE2 STOP]
IF (AND NOT : SN11 = : SN12 NOT : SV11 = : SV12
: ADV1 = : ADV12) [COORDONNE3 STOP]
COORDONNE4 STOP

Nous allons examiner successivement les quatre types de coordination.

4.1 *COORDONNE1*

Dans COORDONNE1, nous avons trois possibilités :

Première possibilité . Les verbes et les syntagmes nominaux objets sont différents; dans ce cas, nous appliquons (15) :

(15) IF AND NOT : V11 = : V12 NOT : SN21 = : SN22 [IMPRIME CONSTRUIT [S [P [SN1 $SN11] [SV1 $SV11] [ADV $ADV1]] et [P [SV1 $SV12] [ADV $ADV2]]] STOP]

La procédure IMPRIME sert à imprimer les phrases réduites. Une phrase est d'abord construite à partir de l'arborescence en supprimant les marques et en aplatissant l'arborescence. La première lettre du premier mot est ensuite remplacée par une majuscule; un point est inscrit à la fin et la phrase résultante est imprimée. Pour cette première possibilité dans COORDONNE1, (16b) correspond à (16a) :

(16) a. Raymonde achète du foie gras du Gers au restaurant du coin et Raymonde commande des chocolats noirs au sous-sol de l'église.
b. Raymonde achète du foie gras du Gers au restaurant du coin et commande des chocolats noirs au sous-sol de l'église.

Deuxième possibilité. Les verbes sont différents et les syntagmes nominaux objets sont identiques; dans ce cas, nous appliquons (17) :

(17) IF NOT : V11 = : V12 [MAKE "ADV2 VIRGULE : ADV2 IMPRIME CONSTRUIT [S [P [SN1 $SN11] [V1 $V11] [ADV $ADV1]] et [P [V1 $V12] [ADV $ADV2]] [SN2 $SN21]] STOP]

La procédure VIRGULE permet de «coller» une virgule à la droite d'un constituant; dans ce cas (18b) correspond à (18a) :

(18) a. Raymonde achète du foie gras du Gers au restaurant du coin et Raymonde commande du foie gras du Gers au sous-sol de l'église.
b. Raymonde achète au restaurant du coin et commande au sous-sol de l'église, du fois gras du Gers.

Troisième possibilité. Les verbes sont identiques et les syntagmes nominaux objets sont différents; dans ce cas, nous appliquons (19) :

(19) IF NOT : SN21 = :SN22 [IMPRIME CONSTRUIT [S [P [SN1 $SN11] [SV1 $SV11] [ADV $ADV1]] et [P [SN2 $SN22] [ADV $ADV2]]] STOP]

À (20a) correspond (20b) :

(20) a. Raymonde achète du foie gras du Gers au restaurant du coin

et Raymonde achète des chocolats noirs au sous-sol de l'église.

b. Raymonde achète du foie gras du Gers au restaurant du coin et des chocolats noirs au sous-sol de l'église.

4.2 *COORDONNE2*

Ici nous avons deux formes réduites possibles :

Première forme réduite, (21) :

(21) IMPRIME CONSTRUIT [S [P [*SN1 $SN11*] [*SV1 $SV11*] [*ADV $ADV1*]] et [P [SN1 $SN12] [ADV $ADV2]]]

À (22a) correspond (22b) :

(22) a. Gertrude achète un gâteau aux amandes au sous-sol de l'église et Raymonde achète un gâteau aux amandes à la cafétéria de l'université.

b. Gertrude achète un gâteau aux amandes au sous-sol de l'église et Raymonde à la cafétéria de l'université.

Deuxième forme réduite, (23) :

(23) MAKE "ADV2 VIRGULE : ADV2
MAKE "V11 PLURIEL : V11
IMPRIME CONSTRUIT [S [P [SN1 $SN11] [ADV $ADV1]] et [P [SN1 $SN12] [ADV $ADV2]] [SV1 [V1 $V11] [SN2 $SN21]]]

La procédure PLURIEL est utilisée pour construire le pluriel d'un verbe. Puisque tous les verbes choisis sont du premier groupe, à la troisième personne du singulier, la procédure PLURIEL ne fait qu'ajouter la terminaison «-nt» au verbe. L'exemple (24) correspond donc aussi à (22a)

(24) Gertrude au sous-sol de l'église et Raymonde à la cafétéria de l'université, achètent un gâteau aux amandes.

4.3 *COORDONNE3*

Dans COORDONNE3, trois possibilités se présentent :

Première possibilité. Les verbes et les syntagmes nominaux objets sont différents; dans ce cas, nous appliquons (25) :

(25) IF AND NOT : V11 = : V12 NOT : SN21 = : SN22 [MAKE "SV12 VIRGULE : SV12 IMPRIME CONSTRUIT [S [P [SN1 $SN11] [SV1 $SV11]] et [P [SN1 $SN12] [SV1 $SV12]] [ADV $ADV1]] STOP]

(26b) correspond à (26a) :

(26) a. La fille achète des chocolats noirs au sous-sol de l'église et Michèle apporte un gâteau aux amandes au sous-sol de l'église.
b. La fille achète des chocolats noirs et Michèle apporte un gâteau aux amandes, au sous-sol de l'église.

Deuxième possibilité. Les verbes sont différents et les syntagmes nominaux objets sont identiques; dans ce cas, deux phrases réduites sont possibles selon que nous appliquons la première partie de (27) ou la seconde :

(27) IF NOT : V11 = : V12 [MAKE "V12 VIRGULE : V12 MAKE "ADV1 VIRGULE : ADV1 IMPRIME CONSTRUIT [S [P [SN1 $SN11] [V1 $V11]] et [P [SN1 $SN12] [V1 $V12]] [ADV $ADV1] [SN2 $SN21]] MAKE "SN21 VIRGULE : SN21 IMPRIME CONSTRUIT [S [P [SN1 $SN11] [V1 $V11]] et [P [SN1 $SN12] [V1 $V12]] [SN2 $SN21] [ADV $ADV2]] STOP]

Les exemples (28b) ou (28c) peuvent donc correspondre à (28a) :

(28) a. La fille achète des chocolats noirs au sous-sol de l'église et Michèle apporte des chocolats noirs au sous-sol de l'église.
b. La fille achète et Michèle apporte, au sous-sol de l'église, des chocolats noirs.
c. La fille achète et Michèle apporte, des chocolats noirs, au sous-sol de l'église.

Troisième possibilité. Les verbes sont identiques et les syntagmes nominaux objets, différents; ici encore, deux formes réduites sont possibles selon que nous appliquons la première partie de (29) ou la seconde :

(29) IF NOT : SN21 = : SN22 [IMPRIME CONSTRUIT [S [P [SN1 $SN11] [SV1 $SV11] [ADV $ADV1]] et [P [SN1 $SN12] [SN2 $SN22]]] MAKE "SN22 VIRGULE : SN22 IMPRIME CONSTRUIT [S [P [SN1 $SN11] [SV1 $SV11]] et [P [SN1 $SN12] [SN2 $SN22]] [ADV $ADV1]] STOP]

À (30a) peuvent correspondre (30b) ou (30c) :

(30) a. La fille achète des chocolats noirs au sous-sol de l'église et Michèle achète un gâteau aux amandes au sous-sol de l'église.
b. La fille achète des chocolats noirs au sous-sol de l'église et Michèle un gâteau aux amandes.
c. La fille achète des chocolats noirs et Michèle un gâteau aux amandes, au sous-sol de l'église.

4.4 *COORDONNE4*

Les syntagmes nominaux sujets, les adverbes et les syntagmes verbaux sont différents. L'élément identique se trouve dans le syntagme verbal; ce peut être le verbe ou le syntagme nominal objet.

Première possibilité. Les verbes sont identiques; nous appliquons (31) :

(31) IF : V11 = V12 [IMPRIME CONSTRUIT [S [P [SN1 $SN11] [SV1 $SV11] [ADV $ADV1]] et [P [SN1 $SN12] [SN2 $SN22] [ADV $ADV2]]] STOP]

À (32a) correspond (32b) :

(32) a. Jacinthe achète des chocolats noirs au restaurant du coin et Michèle achète du foie gras du Gers au sous-sol de l'église.
b. Jacinthe achète des chocolats noirs au restaurant du coin et Michèle du foie gras du Gers au sous-sol de l'église.

Deuxième possibilité. Les syntagmes nominaux objets sont identiques; nous appliquons (33) :

(33) IF : SN21 = : SN22 [MAKE "ADV2 VIRGULE : ADV2 IMPRIME CONSTRUIT [S [P [SN1 $SN11] [V1 $V11] [ADV $ADV1]] et [P [SN1 $SN12] [V1 $V12] [ADV $ADV2]] [SN2 $SN21]] STOP]

et (34b) correspond à (34a) :

(34) a. Jacinthe achète des chocolats noirs au restaurant du coin et Michèle commande des chocolats noirs au sous-sol de l'église.
b. Jacinthe achète au restaurant du coin et Michèle commande au sous-sol de l'église, des chocolats noirs.

5. Conclusion

La programmation en Logo de ces différentes règles de coordination et une étude attentive des résultats nous ont permis de conclure que dans tous les cas les règles programmées rendent compte des règles du français. Logo nous semble être un outil précieux pour explorer les règles parfois complexes d'une langue. Le couple de procédures CONSTRUIT — ANALYSE nous a permis d'introduire en Logo une notation commode pour décrire les opérations d'analyse et de synthèse des arborescences. Ces procédures pourraient être utilisées pour faciliter la réalisation d'autres programmes qui manipulent des éléments linguistiques.

Louisette Émirkanian
Département de linguistique
UQAM

Lorne H. Bouchard
Département de mathématiques
et d'informatique
UQAM

Références

ABELSON, H. et A. diSessa (1980) *Turtle Geometry*, Cambridge MA, MIT Press.

ÉMIRKANIAN, L. (1979) *La coordination en français*, thèse de troisième cycle, Université de Provence.

PAPERT, S. (1980) *Mindstorms. Children, Computers, and Powerful Ideas*, New-York, Basic Books.

SHEIL, B. (1984) «Power Tools for Programmers», dans *Interactive Programming Environments*, D. Barstow, H. Shrobe et E. Sandewall (Eds), New York, McGraw-Hill, pp. 19-30.

WINSTON, P. (1977) *Artificial Intelligence*, Reading MA, Addison-Wesley.