UN MÉTALANGAGE DE GRAMMAIRES TRANSFORMATIONNELLES

APPLICATIONS AUX PROBLEMES DE TRANSFERTS ET DE GENERATION SYNTAXIQUES

G. 2300 - A

Janvier 1967

G. VEILLON

J. VEYRUNES

B. VAUQUOIS

Cette publication présente une partie des résultats acquis par l'équipe de recherche sur les modèles au cours de l'année 1966. Cette partie est essentiellement consacrée à la description de l'outil méta-linguistique utilisé pour les grammaires transformationnelles du système de traduction automatique. Ces grammaires feront l'objet de publication séparées lorsqu'elles seront suffisamment au point.

Les auteurs expriment leur gratitude à Monsieur le Professeur J. P. BENZECRI qui a bien voulu s'intéresser à cette étude et à la D.R.M.E. dont la participation financière dans le cadre du contrat 65-34-497 en a permis la réalisation.

P L A N

INTRODUCTION

Au cours de ces dernières années les grammaires transformationnelles ont occupé une place privilégiée dans la littérature relative aux systèmes de structures linguistiques, aux problèmes de génération syntaxique et aussi, éventuellement, à quelques problèmes de reconnaissance liés à la traduction automatique [1], [2], [3], [4], [5]. Ces grammaires transformationnelles et, plus généralement, les transformations brillent par le nombre de définitions dont elles ont été l'objet. En effet, un auteur qui étudie un problème linguistique adapte la notion de transformation dont il a besoin au but qu'il poursuit. Attachée à un système linguistique cette notion revet une signification particularisée par le problème traité. La transformation indique comment, quelquefois explique pourquoi tels phénomènes se présentent. A propos des problèmes que nous avons rencontrés lors de la réalisation de certains modèles du système de traduction automatique nous avons également été conduits à définir des transformations. A cette multiplicité des définitions s'oppose, en général, une certaine faiblesse des moyens d'expression et des algorithmes d'utilisation. En effet, il est souvent proposé des "règles de transformation" mais le formalisme de leur écriture ou bien n'est pas rigoureusement défini ou bien se heurte à des difficultés matérielles insurmontables si l'on veut établir toutes les règles d'un modèle réel et non d'un modèle réduit. En outre, des problèmes liés aux algorithmes d'emploi de telles grammaires transformationnelles sont peut souvent abordés. Sans prétendre résoudre tous ces problèmes, à partir des notions de transformations déjà rencontrées et de celles que notre système de traduction automatique faisait apparaitre, nous avons essayé de systématiser cette notion.

- Les "modèles transformationnels" en traduction automatique

Rappelons la place de ces modèles dans le système de traduction automatique du C.E.T.A. - L'article "syntaxe et interprétation" [6] précise le modèle de reconnaissance syntaxique des phrases de la langue source (modèle M2). Ce que l'on obtient à la sortie de ce modèle est une structure arborescente (éventuellement plusieurs) de la phrase proposée. Cette structure établit un système de liaisons entre les mots qui constituent la phrase en indiquant pour chaque liaison la règle de grammaire qui a permis de l'obtenir ainsi que l'élément considéré comme gouverneur.

Le but poursuivi par le modèle suivant (M3) est de transformer cette structure pour obtenir une évaluation sémantique des liaisons syntaxiques. On est donc conduit à substituer à une ou plusieurs règles formelles, une "étiquette" (telle que "agent", "objet", "conséquence", etc...) qui doit indiquer la fonction du mot ou groupe de mots considéré dans la phrase. En outre, pour éviter la prolifération des ambiguités de construction syntaxique, le modèle M2 peut associer à une phrase une seule structure qui, en fait, représente une famille de structures bien déterminées. En conséquence le modèle M3 doit éventuellement corriger le système de liaisons proposé par le modèle précédent pour aboutir à une articulation correcte des fonctions de chaque mot dans la phrase.

Le problème de la transformation consiste donc à remplacer une structure arborescente par une autre en changeant les noms attribués aux sommets, en supprimant ou en rajoutant certains sommets, enfin en modifiant la relation d'ordre du graphe initial.

Le modèle suivant (M'2) fait apparaître le même problème. Il s'agit maintenant de transformer le graphe associé à une phrase dans le modèle M3 (sorte de langage pivot) pour en déduire une construction syntaxique de surface dans la langue cible.

Ainsi, les transformations qui interviennent dans ces deux modèles, de même que celles qui sont mentionnées dans la littérature peuvent se ramener à un problème mathématique unique. C'est celui de l'application d'un ensemble de graphes dans un autre ensemble de graphes. Il semble d'après l'usage actuel que l'on fait des transformations, que l'on puisse limiter le problème précédent à des graphes qui sont seulement des arborescences sans perdre de généralité. Encore faut-il mentionner que ces ar escences peuvent comporter des relations d'ordre latéral, c'est-à-dire que tous les descendants directs d'un sommet quelconque sont totalement ordonnés.

Dans ce qui suit nous nous proposons d'étudier les transformations de structures arborescentes d'un point de vue formel en soulignant quelques problèmes (non résolus) et en examinant un type de langage outil pour la description de ces transformations. Enfin, la dernière partie montre l'application de ce métalangage à l'écriture des grammaires des modèles M3 et M'2.

I - TRANSFORMATIONS DE STRUCTURES ARBORESCENTES

1° - Définition des structures arborescentes

a) n-graphe

Un graphe (X,Γ) est la donnée d'un ensemble X et d'une application multivoque Γ de X dans X - On peut dire aussi que (X,Γ) est la donnée d'une partie de X^2.

Soient $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ des applications multivoques de X dans X. Nous appellerons n-graphe le (n+1)-uple : $(X, \Gamma_1, \Gamma_2, \dots, \Gamma_n)$ considéré comme le graphe union des graphes (X, Γ_1), (X, Γ_2), ..., (X, Γ_n) dans lequel tous les arcs sont désignés selon leur appartenance à l'application d'où ils proviennent

Exemple   Soient : $X = (x_1, x_2, x_3, x_4, x_5)$

$$\Gamma_1 : \begin{cases} x_1 \longrightarrow x_2, x_3 \\ x_2 \longrightarrow x_4 \\ x_4 \longrightarrow x_3, x_5 \end{cases}$$

$$\Gamma_2 : \begin{cases} x_3 \longrightarrow x_1, x_2, x_4, x_5 \\ x_2 \longrightarrow x_1, x_3, x_4 \end{cases}$$

Le 2-graphe (X, Γ_1, Γ_2) est représenté par la figure 1

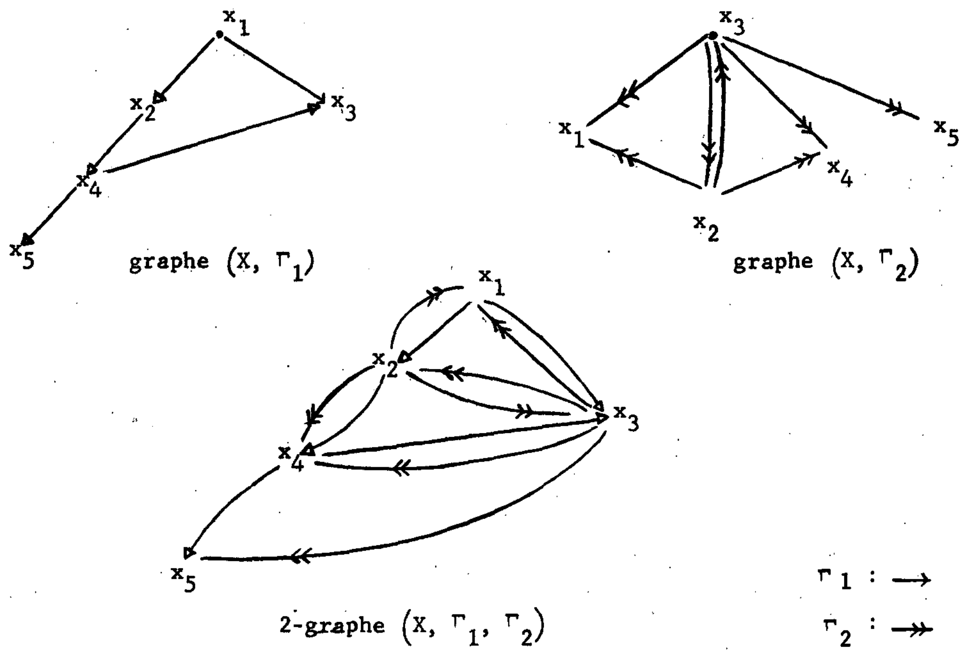


figure 1

b) Structure

Soit un n-graphe $(X, \Gamma_1, \ldots, \Gamma_n)$. Désignons par $\mathcal{V}$ un ensemble de symboles (vocabulaire ou alphabet), et soit Δ une application univoque de X dans $\mathcal{V}$.
Nous appellerons "structure", le (n+3)-uple : $(X, \mathcal{V}, \Gamma_1, \ldots, \Gamma_n, \Delta)$
On obtient ainsi la structure à partir du n-graphe en attribuant à chaque sommet x_i du n-graphe le symbole appartenant à $\mathcal{V}$ au moyen de Δ.
Ainsi des sommets distincts de la structure peuvent être désignés par le même symbole.

Exemple Reprenons le 2-graphe du paragraphe précédent et soient :
$\mathcal{V} = (a, b, c)$

Δ :

x_1	x_2	x_3	x_4	x_5
a	b	c	a	c

La structure $(X, V, \Gamma_1, \Gamma_2, \Delta)$ est indiquée sur la figure 2

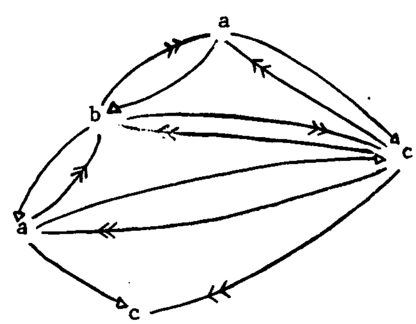


figure 2

c) Structure arborescente

On appelle "structure arborescente" une structure $\Sigma = (X, V, \Gamma_1, \Gamma_2, \Delta)$ où

- le graphe (X, Γ_1) est une arborescence, c'est-à-dire :

  1) il existe un sommet $x_o \in X$ unique tel que $\Gamma_1 x_o = \emptyset$

  2) pour tout sommet $x \neq x_o$, $\Gamma_1 x = y$, $y \in X$

  3) le graphe est connexe

- L'application Γ_1 étant déterminée, l'application Γ_2 doit satisfaire aux conditions suivantes :

Le graphe (X, Γ_2) est tel que

- à tout sous-ensemble $\Gamma_1^{-1}x$ correspond une composante connexe de (X, Γ_2)

- pour chacun de ces sous-ensembles $\Gamma_1^{-1}x$, la relation d'ordre associée à Γ_2 est totale. Ainsi, dans tout sous-ensemble $X_i = \Gamma_1^{-1}x_i$, il existe un $x_A \in X_i$ unique tel que $\Gamma_2^{-1}x_A = \emptyset$ et un $x_B \in X_i$ unique tel que $\Gamma_2\, x_B = \emptyset$

La donnée du 2-graphe (X, Γ_1, Γ_2) constitue une arborescence munie d'une relation d'ordre restreinte telle que l'a définie HOLT [7]
On peut interpréter le graphe (X, Γ_1) comme une relation de "paternité" :

"y est père de x" $\Longleftrightarrow \Gamma_1\, x = y$

De même, le graphe (X, Γ_2) exprime un ordre entre les fils d'un même père :

$\Gamma_2 x = y$ implique que x est né avant y

Un sommet de type x_A est dit "Ainé" ; un sommet de type x_B est dit "Benjamin".

<u>Exemple</u> Soit le 2-graphe (X, Γ_1, Γ_2) où

$X = (A, B, C, D, E, F, H, I)$

Γ_1 :

A	B	C	D	E	F	G	H	I
∅	A	B	B	B	A	A	G	G

Γ_2 :

A	B	C	D	E	F	G	H	I
∅	F	D	E	∅	G	∅	I	∅

représenté par la figure 3, qui est une arborescence munie d'une relation d'ordre restreinte.

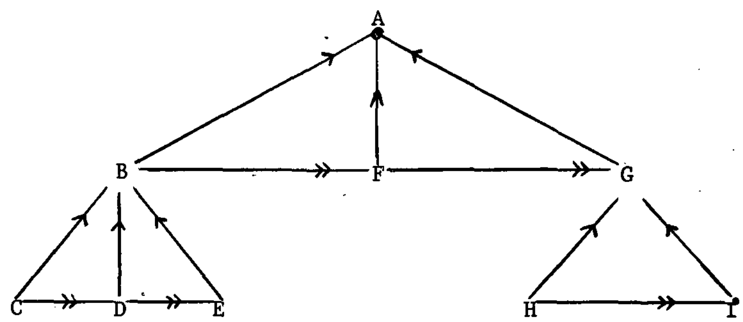


figure 3

$\Gamma_1 \rightarrowtail$
$\Gamma_2 \twoheadrightarrow$

Si nous nous donnons en outre :

$\mathcal{V} = (a, b, c, d, e)$

et Δ :

A	B	C	D	E	F	G	H	I
a	b	d	e	e	c	b	d	e

La structure $\Sigma = (X, \mathcal{V}, \Gamma_1, \Gamma_2, \Delta)$ est une structure arborescente - cf figure 4

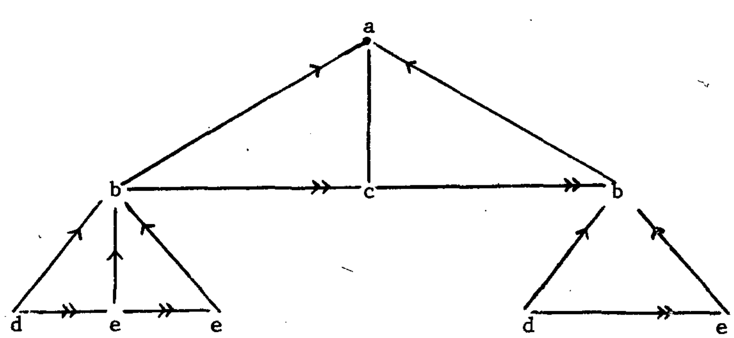


figure 4

2° - Représentation des structures arborescentes et des arborescences munies d'une relation d'ordre restreinte :

On s'attache surtout à la représentation des arborescences munies d'une relation d'ordre restreinte, celle des structures arborescentes se déduisant facilement de la première.
Pour des raisons algorithmiques, nous représenterons une telle arborescence à l'aide de trois applications biunivoques, notées α, γ et π, et définies de la façon suivante :

- α (x) = y si et seulement si $\Gamma_1 y = x$ et $\Gamma_2^{-1} y = \emptyset$ (fils ainé)

- γ (x) = y si et seulement si $\Gamma_2 x = y$ (frère cadet)

- π (x) = y si et seulement si $\Gamma_1 x = y$ et si $\Gamma_2 x = \emptyset$ (père du benjamin)

Exemple

Reprenons l'arborescence du paragraphe précédent (figure 3). Sa représentation au moyen des fonctions α, γ et π est alors celle de la figure 5 où l'on a représenté aussi la structure arborescente déjà mentionnée.

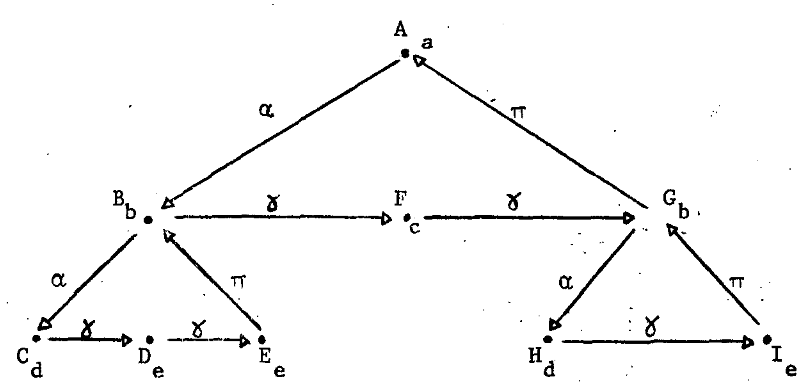


figure 5

On retrouve certaines des fonctions décrites par A. HOLT [7] pour représenter les arborescences munies d'une relation d'ordre restreinte.
On peut aussi considérer cette structure comme la donnée du 3-graphe (X, α, γ, π) du vocabulaire V et de l'application Δ ($\Delta : X \rightarrow V$)

3° - <u>Chemins et relations sur une structure arborescente</u>

D'après les définitions de α, γ, π à partir de Γ_1 et Γ_2, pour tout couple de sommets distincts (x, y), il existe au moins un chemin d'origine x et d'extrémité y et il existe un chemin unique sans circuit. Ces chemins correspondent à des compositions des fonctions α, γ, π.

Soit f une composition de α, γ, π et $x \in X$ un sommet - Si fx existe, c'est un sommet y unique de la structure ; en outre $f^{-1}y$ existe et est le seul sommet x.

Un tel chemin caractérise une relation entre deux sommets.

Nous nous proposons de déterminer les moyens de construire des relations qui possèdent les propriétés suivantes :

- des relations 1-aire correspondant à des propriétés possédées par un sommet considéré isolément.

- des relations binaires que nous rangeons dans l'une des deux classes suivantes :

- des "relations simples" - Ces relations sont telles qu'à tout x_i il corresponde au plus un y_i en relation avec x_i.

- des "relations multiples" - A tout x_i il correspond un ensemble $X_i \subset X$ de sommets y tels que $x_i R y$ $(y \in X_i)$, X_i possédant en général plus d'un élément. <u>On veut alors que la relation multiple induise un ordre total sur les éléments de X_i</u> - L'importance de cette propriété apparaîtra lorsque l'on travaillera sur la structure où plusieurs sommmets peuvent porter le même nom (symbole de V).

Toute relation peut être considérée comme un prédicat.
Si R est une relation 1-aire, R_x est un prédicat à 1 variable (x possède la propriété R).

Si R est une relation binaire, x R y est un prédicat à deux variables et l'ensemble X_i des sommets y liés à x_i par R est l'extension du prédicat $x_i Ry$.

$$X_i = \{y / x_i Ry\}$$

a) Propriétés possédées par un sommet

D'après les définitions de α, γ, π certains sommets possèdent des propriétés particulières. Ces propriétés sont caractérisées par un prédicat.

Exemples - "Etre la racine de l'arborescence" désigné par $\mathcal{P}$

Ainsi $\mathcal{P}x$ est un prédicat vrai si et seulement si x est la racine de l'arborescence :

$$\mathcal{P}x \longleftrightarrow \gamma x = \emptyset \wedge \pi x = \emptyset$$

- "Etre un fils ainé" - $\mathcal{A}$

$$\mathcal{A}x \longleftrightarrow \gamma^{-1}x = \emptyset$$

- "Etre un benjamin" - $\mathcal{B}$

$$\mathcal{B}x \longleftrightarrow \gamma x = \emptyset$$

- "N'avoir pas de fils" - $\mathcal{C}$

$$\mathcal{C}x \longleftrightarrow \alpha x = \emptyset$$

A partir de ces propriétés élémentaires on peut construire des propriétés plus compliquées

Par exemple

- "Etre fils unique" - $\mathcal{U}$

$$\mathcal{U}x \longleftrightarrow \mathcal{A}x \wedge \mathcal{B}x$$

(On peut aussi écrire $\mathcal{U}x \longleftrightarrow x = \alpha\,\pi\,x$)

- "Etre le point unique de l'arborescence" - $\mathcal{T}$

$$\mathcal{T}x \longleftrightarrow \mathcal{P}x \wedge \mathcal{C}x$$

b) <u>Construction des relations simples</u>

Le cas le plus simple de construction de telles relations est celui qui provient d'une composition déterminée de fonctions α, γ, π décrivant un chemin sans circuit.

<u>Exemple</u> : Etablir la relation : "3ième FS" où

x 3ième FS y signifie "y est le troisième fils de x"

Alors :

$$x \text{ 3ième FS } y \longleftrightarrow y = \gamma^2 \alpha x$$

En nous referrant à l'arborescence de la figure 5

A 3ième FS G est vraie et G est le seul sommet qui soit dans cette relation avec A.

Un cas plus compliqué est celui où la relation x R y est satisfaite si et seulement si, f_1, f_2, ... , f_n étant des compositions de α, γ, π, il existe au plus un j $(1 \leqslant j \leqslant n)$ tel que $y = f_j x$; le nombre n n'étant pas connu mais seulement majoré par le nombre d'éléments de X. On utilise alors des prédicats munis de quantificateurs.

<u>Exemples</u> : Etablir la relation : "FNE" où

x FNE y signifie "y est le frère ainé de x"

Alors :

$$x \text{ FNE } y \longleftrightarrow \bigvee_{r=0}^{n} y = \alpha \pi \gamma^r x$$

On voit qu'ici la relation peut être réflexive si x est lui-même un ainé. Si l'on veut éliminer ce cas, il suffit de définir FNE comme la conjonction du prédicat déjà écrit et du prédicat $y \neq x$ (ou bien $\neg \mathcal{A}(y)$)

- Etablir la relation "PERE" où

x PERE y signifie y est le père de x

$$x \text{ PERE } y \longleftrightarrow \bigvee_{r=0}^{n} y = \pi \gamma^r x$$

Comme la relation simple x AINE y (signifiant "y est le fils ainé de x") est définie par $x \text{ AINE } y \longleftrightarrow y = \alpha x$,
on peut aussi définir FNE par

$$x \text{ FNE } y \longleftrightarrow \bigvee_{z} \left[(y \text{ PERE } z) \wedge (z \text{ AINE } x)\right]$$

- Etablir la relation "FBENJ" où

x FBENJ y signifie "y est le plus jeune frère de x

$$x \text{ FBENJ } y \longleftrightarrow \bigvee_{r=0}^{n} (y = \gamma^r x \wedge \mathcal{B} y)$$

A toute relation simple R, on associe la fonction R_f définie par $y = R_f x$ si et seulement si xRy

c) Construction des relations multiples

Dans de telles relations xRy peut être vérifié par un ensemble de sommets y pour un x déterminé, et

$X_i = \{y / x_i \; R \; y\}$ doit être totalement ordonné

On donne alors le moyen d'énumérer récursivement les éléments de X_i - Dans ce but, on détermine un premier élément y_1 de la même manière qu'on a construit les relations simples - Puis on donne le moyen de calculer y_{i+1} connaissant y.

Exemples - Etablir la relation "FRERE" où

x FRERE y signifie que "y est frère plus jeune de x"

Alors :

$$\{y / x \text{ FRERE } y\} = \begin{cases} y_1 = \gamma x \\ y_{i+1} = \gamma y_i \end{cases}$$

Etablir la relation "FILS" où

x FILS y signifie que "y est fils de x"

$$\{y / x \text{ FILS } y\} = \begin{cases} y_1 = \alpha x \\ y_{i+1} = \gamma y_i \end{cases}$$

Etablir la relation "COFILS" où

x COFILS y est vraie si et seulement si

y = x ou y est un fils du père de x

$$\{y / x \text{ COFILS } y\} = \begin{cases} y_1 = \text{FNE}_f x \\ y_{i+1} = \gamma y_i \end{cases}$$

L'ordre total induit est celui donné par γ sur tous les fils du même père.

Enfin, établissons la relation "GOUV" où

x GOUV y est vraie si et seulement si

$y = \gamma^{-1}x$ (x est le cadet de y)

ou y se trouve dans la lignée des benjamins du premier y.

$$\{y \,/\, x \text{ GOUV } y\} = \begin{cases} y_1 = \gamma^{-1}x \\ y_{i+1} = \text{DFILS}_f y_i \end{cases}$$

où DFILS est une relation simple (y est le dernier fils de x)

$$x \text{ DFILS } y \longleftrightarrow \bigvee_{r=0}^{n} y = \gamma^r \alpha x \wedge \mathcal{B} y$$

Ainsi une relation multiple s'exprime au moyen de deux relations simples, c'est-à-dire au moyen de 2 fonctions f et g

$$\begin{cases} y_1 = fx \\ y_{i+1} = gy_i \end{cases}$$

L'élément y_r de rang r dans l'ensemble $X_i = \{y \,/\, xRy\}$ est alors obtenu

par $y_r = g^{r-1}fx$

Si $g^{r-1}fx$ n'existe pas, c'est que X_i contient moins de r éléments.

Pour que chaque élément ne soit énuméré qu'une fois, il est nécessaire que

$$\bigwedge_{i=1}^{n} \bigwedge_{p=1}^{n} y_{i+p} \neq y_i$$

4° - Reconnaissance d'une figure dans une structure arborescente

a) Schéma de figure

Soient R_1, R_2, ... , R_n des relations binaires (simples ou multiples). Soient P_1, ... P_m des propriétés.

On définit un schéma de figure F comme une expression formelle construite sur le vocabulaire terminal $\{R_1, \dots, R_n, P_1, \dots, P_m, (,)\}$ de la manière suivante

(1) Si R_i est une relation alors (R_i) est un schéma de figure

(2) Si P_i est une propriété, alors (P_i) est un schéma de figure

(3) Si F est un schéma de figure et R_i une relation alors $(R_i F)$ est un schéma de figure

(4) Si F_1 et F_2 sont deux schémas de figure, alors $F_1 F_2$ est un schéma de figure

(5) Tous les schémas de figure sont donnés par (1),(2),(3) et(4).

En utilisant le symbolisme de définition d'ALGOL on a :

$$\langle R \rangle ::= R_1 \mid R_2 \mid \dots \mid R_n$$

$$\langle P \rangle ::= P_1 \mid P_2 \mid \dots \mid P_m$$

$$\langle F \rangle ::= \underset{(1)}{(\langle R \rangle)} \mid \underset{(2)}{(\langle P \rangle)} \mid \underset{(3)}{(\langle R \rangle \langle F \rangle)} \mid \underset{(4)}{\langle F \rangle \langle F \rangle}$$

Exemple

L'expression : $(R_{i_1}(R_{i_2}(P_{j_1}))(R_{i_3})(R_{i_4}))(R_{i_5}(P_{j_2}))$

où $1 \leqslant i_k \leqslant n$ pour $1 \leqslant k \leqslant 5$ et $1 \leqslant j_\ell \leqslant m$ pour $1 \leqslant \ell \leqslant 2$

est un schéma de figure F.

En effet

$\left(P_{j_1}\right)$ est un schéma de figure par la règle de construction (2)

$\left(R_{i_2}\left(P_{j_1}\right)\right)$ -------------------------------------- (3)

$\left(R_{i_3}\right)$ -------------------------------------- (1)

$\left(R_{i_2}\left(P_{j_1}\right)\left(R_{i_3}\right)\right)$ -------------------------------------- (4)

$\left(R_{i_4}\right)$ -------------------------------------- (1)

$\left(R_{i_2}\left(P_{j_1}\right)\left(R_{i_3}\right)\left(R_{i_4}\right)\right)$ -------------------------------------- (4)

$\left(R_{i_1}\left(R_{i_2}\left(P_{j_1}\right)\left(R_{i_3}\right)\left(R_{i_4}\right)\right)\right)$ -------------------------------------- (3)

$\left(P_{j_2}\right)$ -------------------------------------- (2)

$\left(R_{i_5}\left(P_{j_2}\right)\right)$ -------------------------------------- (3)

$\left(R_{i_1}\left(R_{i_2}\left(P_{j_1}\right)\left(R_{i_3}\right)\left(R_{i_4}\right)\left(R_{i_5}\left(P_{j_2}\right)\right)\right)\right)$ -------------------------------------- (4)

Considérons maintenant une arborescence munie d'une relation d'ordre restreinte $\mathcal{A}$.
Soient X l'ensemble des sommets de $\mathcal{A}$, x_o un sommet quelconque ($x_o \in X$) et F un schéma de figure.
On dit que x_o admet le schéma de figure F (dans $\mathcal{A}$) si et seulement si :

α) la proposition F (x_o) est vraie, F (x_o) étant la proposition déduite du prédicat F (x) défini de la manière suivante :

Soient F_1, F_2, F_3 des schémas de figure

Si F_1 est construit au moyen de la règle (1)

$$F_1 = (R_i) \qquad \text{alors} \quad F_1\ (x) \longleftrightarrow \bigvee_y xR_i y \qquad x, y \in X$$

Si F_1 est construit au moyen de la règle (2)

$$F_1 = (P_j) \qquad \text{alors} \quad F_1\ (x) \longleftrightarrow P_j\ (x) \qquad x \in X$$

Si F_2 est construit au moyen de la règle (3)

$$F_2 = (R_i\ F_1) \qquad \text{alors} \quad F_2\ (x) \longleftrightarrow \bigvee_y \left[xR_i y \wedge F_1\ (y) \right] \qquad x, y \in X$$

Enfin si F_3 est construit au moyen de la règle (4)

$$F_3 = F_1\ F_2 \qquad \text{alors} \quad F_3\ (x) \longleftrightarrow F_1\ (x) \wedge F_2\ (x) \qquad x \in X$$

Ainsi le prédicat associé au schéma de figure F donné en exemple est :

$$F(x) \longleftrightarrow \bigvee_{x_1} \left[xR_{i_1} x_1 \wedge \bigvee_{x_2} \left[x_1 R_{i_2} x_2 \wedge P_{j_1} (x_2) \right] \wedge \bigvee_{x_3} x_1 R_{i_3} x_3 \wedge \bigvee_{x_4} x_1 R_{i_4} x_4 \right] \wedge \bigvee_{x_5} \left[xR_{i_5} x_5 \wedge P_{j_2} (x_5) \right]$$

$$\text{où} \quad x, x_1, x_2, x_3, x_4, x_5 \in X$$

(β) les sommets qui interviennent en tant que variables liées dans les quantificateurs sont tous les éléments différents de X.
Pour formaliser cette deuxième condition, on remarque que chaque relation introduit un sommet ; donc si F comporte p relations, $x_1, \ldots, x_p$ doivent être tous différents les uns des autres.

Alors soit le prédicat $D_2\ (u_1, u_2) \longleftrightarrow u_1 \neq u_2$

Par récurrence on construit :

$$D_{k+1}\ (u_1, \ldots, u_k, u_{k+1}) \longleftrightarrow D_k\ (u_1, \ldots, u_k) \wedge \bigwedge_{j=1}^{k} D_2\ (u_j, u_{k+1})$$

Enfin on pose : $\quad D_1$ (u) toujours vrai.

Ainsi, F étant un schéma de figure à p relations, la proposition "x_o admet le schéma de figure F (dans $\mathcal{A}$)" $\longleftrightarrow F(x_o) \wedge D_{p+1}(x_o, x_1, \dots, x_p)$

L'extension de ce prédicat est l'ensemble des $x \in X$ qui rendent ce prédicat vrai.

b) <u>Figures dans une arborescence munie d'une relation d'ordre restreinte</u>

Désignons toujours par $\mathcal{A}$ cette arborescence ; X est l'ensemble de ses sommets et $x_o \in X$ et F un schéma de figure comportant n relations.

Au prédicat d'1 variable F (x) on associe le prédicat à n + 1 variables

$F^{(n+1)}(x, x_1, \dots, x_n)$ déduit de F (x) en supprimant les quantificateurs qui lient les variables $x_1, \dots, x_n$.

On appelle φ une figure de schéma F et de pivot x_o dans $\mathcal{A}$ tout élément de l'ensemble des (n+1)-uples $(x_o, x_1, \dots, x_n)$ où $x_o, x_1, \dots, x_n \in X$ et $F^{(n+1)}(x_o, x_1, \dots, x_n)$ est vrai.

On peut alors définir un ordre total sur les figures de schéma F et de pivot x_o de la manière suivante

Soient $\varphi = (x_o, x_1, \dots, x_n)$ et $\varphi' = (x_o, x'_1, \dots, x'_n)$ deux telles figures

Soit k le plus petit indice tel que $x_k \neq x'_k$ ($k \neq o$ puisque x_o est pivot, et $k \leq n$ puisque φ et φ' sont différentes)

A x_k et x'_k est associée la relation R_{i_k} telle que il existe un $j < k$ avec

$$x_j = x'_j \quad \text{et} \quad x_j \, R_{i_k} \, x_k \text{ dans } \varphi \text{ et } x_j \, R_{i_k} \, x'_k \text{ dans } \varphi'.$$

La relation R_{i_k} est évidemment une relation multiple (sinon $x_k = x'_k$) ; on peut alors écrire x_k et x'_k sous la forme

$$x_k = g^r f \, x_j \quad \text{et} \quad x'_k = g^{r'} f \, x_j$$

On ordonne φ et φ' par la relation $\varphi < \varphi'$ si et seulement si $r < r'$

Montrons que cette relation sur les figures est un ordre total :

Il est facile de voir que cette relation est non réflexive (tous les x_k sont égaux) et antisymétrique (elle est déduite de la relation d'ordre sur r).

Montrons qu'elle est transitive :

En effet, si l'on a $\varphi < \varphi'$ et $\varphi' < \varphi''$, alors

soient k et ℓ les plus petits indices tels que :

$$x_k \neq x'_k \quad \text{et} \quad x'_\ell \neq x''_\ell$$

si min (k, ℓ) = k, alors k est le plus petit indice tel que $x_k \neq x''_k$,

car $x'_k = x''_k$, et l'on a $\varphi < \varphi''$ pour la même raison que $\varphi < \varphi'$

si min (k, ℓ) = ℓ, alors $x_\ell \neq x''_\ell$ car $x_\ell = x'_\ell$ et $\varphi < \varphi''$.

Il est bien évident que cet ordre est total, puisqu'il est défini pour tout couple de figures.

<u>Exemple</u>

Reprenons le schéma de figure F proposé au paragraphe 4 a

dans lequel R_{i_1} est la relation FILS

R_{i_2} -------------- FILS

R_{i_3} -------------- FRERE

R_{i_4} -------------- COUSING

R_{i_5} -------------- CADET

P_{j_1} est la propriété "n'avoir pas de fils" $\mathcal{P}$ (paragraphe 3 a)

P_{j_2} est la propriété "être benjamin" $\mathcal{B}$ (paragraphe 3 a) x COUSING y étant vraie si et seulement si y est un cousin à gauche de x et définie par :

$$y_1 = \alpha\gamma^{-1}\ \mathrm{PERE}_f\ x$$

$$y_{i+1} = \begin{cases} \gamma y_i & \text{si } \gamma \neq \emptyset \\ \alpha\gamma^{-1}\pi y_i & \text{si } \gamma = \emptyset \end{cases}$$

Alors le schéma de figure s'écrit :

$$F\ :\ \big(\mathrm{FILS}(\mathrm{FILS}(\mathcal{C}))\ \big(\mathrm{FRERE}\big)\ \big(\mathrm{COUSING}\big)\big)\big(\mathrm{CADET}(\mathcal{B})\big)$$

Soit l'arborescence $\mathcal{A}$ suivante :

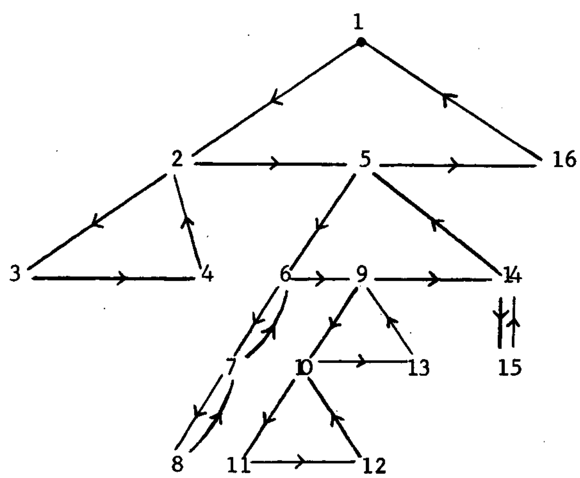


figure 6

Si l'on se donne comme pivot le sommet 5, nous obtiendrons dans l'ordre les figures suivantes :

(5, 9, 13, 14, 3, 16)
(5, 9, 13, 14, 4, 16)

Il peut se faire qu'un schéma de figure en contienne un autre.
Soit par exemple

$$F'\ :\ \big(\mathrm{FILS}(\mathrm{FILS}(\mathcal{C}))\big)\big(\mathrm{FILS}(\mathrm{COUSING})\big)\big(\mathrm{CADET}(\mathcal{B})\big)$$

Du même pivot, le sommet 5, on obtient, dans l'ordre, les figures

(5, 9, 13, 6, 3, 16)
(5, 9, 13, 6, 4, 16)
(5, 9, 13, 14, 3, 16)
(5, 9, 13, 14, 4, 16)
(5, 14, 15, 6, 3, 16)
(5, 14, 15, 6, 4, 16)
(5, 14, 15, 9, 3, 16)
(5, 14, 15, 9, 4, 16)

qui contiennent les 2 figures précédentes

Remarque : Etant donné un sous-ensemble de n + 1 points quelconque d'une arborescence, $(o \leqslant n \leqslant \mathrm{card}\ (X))$, ce sous-ensemble est une figure, car on peut toujours trouver un schéma de figure correspondant.

c) Figures dans une structure arborescente

Soit S une structure arborescente ; S est donnée par une arborescence munie d'une relation d'ordre restreinte $\mathcal{A}$, et par un vocabulaire V et une application Δ de X dans V. $S = (\mathcal{A}, V, \Delta)$.

F étant un schéma de figure, à chaque relation R_{i_k} de F on associe un symbole $a_{i_k} \in V \cup \{*\}$

On associe également un symbole $a_o \in V \cup \{*\}$ à la place du pivot.

Soit $x_o \in X$ un pivot, on appelle figure de schéma F et de pivot x_o dans S, toute figure de schéma F et de pivot x_o dans $\mathcal{A}$ telle que pour tout k $o \leqslant k \leqslant n$ $\Delta x_{i_k} = a_{i_k}$ ou $a_{i_k} = *$

Exemple

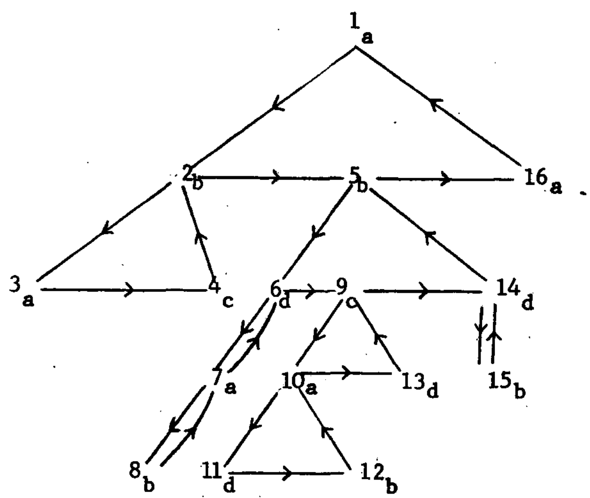


figure 7

Le vocabulaire $V = \{a, b, c, d\}$ et l'application Δ est représentée sur la figure.
Considérons le schéma de figure F' (paragraphe 4 b) qui donnait 8 figures de pivot x_5 dans $\mathcal{A}$
Alors l'expression :

$$* \left(\text{FILS } c\left(\text{FILS } *(\mathcal{C})\right)\right)\left(\text{FILS } d\left(\text{COUSING } c\right)\right)\left(\text{CADET } a\left(\mathcal{B}\right)\right)$$

appliquée au pivot x_5 fournit dans S les deux figures

(5, 9, 13, 6, 4, 16)
(5, 9, 13, 14, 4, 16)

5° - Transformations

a) Familles d'arborescences

Soit un ensemble X et deux sous ensemble X', X'' tels que $X' \subset X$, $X'' \subset X$ et $X' \cap X'' = \emptyset$. Soient deux arborescences $\mathcal{A}' = (X', \Gamma'_1, \Gamma'_2)$ et $\mathcal{A}'' = (X'', \Gamma''_1, \Gamma''_2)$

Nous appellerons famille d'arborescences l'ensemble $\{\mathcal{A}', \mathcal{A}''\}$ que nous pouvons noter : $(X' \cup X'', \Gamma_1, \Gamma_2$ avec $\Gamma_1 x = \Gamma'_1 x$ si $x \in X'$

et $\Gamma_1 x = \Gamma''_2 x$ si $x \in X''$

b) Décomposition d'une arborescence

Soient une arborescence (X, Γ_1, Γ_2) et un sommet x, on se propose de définir les deux applications suivantes :

$$
\begin{array}{lll}
\text{Si} & y \neq x & \Gamma'_1 y = \Gamma_1 y \\
 & y = x & \Gamma' y = \emptyset \\
 & \left.\begin{array}{l} y \neq x \\ y \neq \Gamma_2^{-1} x \end{array}\right\} & \Gamma'_2 y = \Gamma_2 y \\
 & y = x & \Gamma'_2 y = \emptyset \\
 & y = \Gamma_2^{-1} x & \Gamma'_2 y = \Gamma_2 x
\end{array}
$$

si $y = \Gamma_2^{-1} x$

alors $(X, \Gamma'_1, \Gamma'_2)$ est une famille de deux arborescences $\mathcal{A}', \mathcal{A}''$, telles que $X' \cup X'' = X$; Γ'_1 et Γ'_2 répondent aux axiomes de définition des arborescences, le sommet x étant tel que $\Gamma'_1 x = \Gamma'_2 x = \emptyset$ (racine)

Nous appellerons détachement de l'arborescence de racine x (DETA (x)) une telle opération. Elle a pour effet de décomposer une arborescence en deux arborescences.

En utilisant les fonctions α, γ et π, DETA (x) peut se décrire ainsi :

si $\alpha(y) \neq x$, $\alpha'(y) = \alpha(y)$

si $\alpha(y) = x$, $\alpha'(y) = \gamma(x)$

si $\gamma(y) = x$ et si $\gamma(x) \neq \emptyset$, $\gamma'(y) = \gamma(x)$

si $\gamma(x) = \emptyset$, $\gamma'(y) = \emptyset$, $\pi'(y) = \pi(x)$

si $\gamma(y) \neq x$, $y \neq x$ $\gamma'(y) = \gamma(y)$

si $y = x$ $\gamma'(x) = \emptyset$

si $y \neq x$, $\pi'(y) = \pi(y)$, si $y = x$, $\pi'(x) = \emptyset$

Exemple :

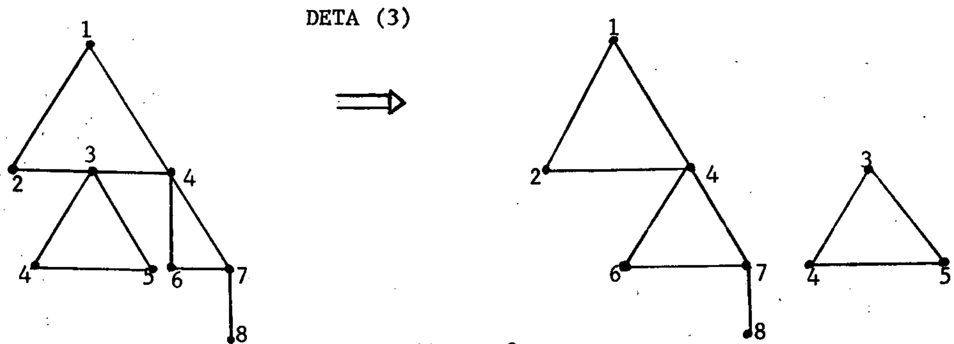
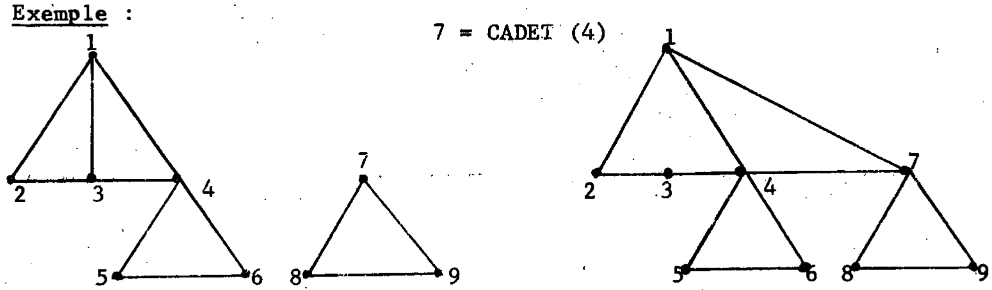


figure 8

c) Insertion d'arborescence

Soit une famille de 2 arborescences $\{\mathcal{A}', \mathcal{A}''\}$ de racines x' et x". On se propose de construire une arborescence unique en insérant le sommet x", racine de $\mathcal{A}''$, comme sommet de $\mathcal{A}'$. Pour cela, on définit l'insertion par une fonction f et un sommet z de X'. L'arborescence résultante sera telle que x" = f (z), et que toutes les relations d'ordre partiel associées à Γ_1 et Γ_2 soient conservées. La fonction f doit permettre de placer x" dans $\mathcal{A}'$. x" étant tel que $\gamma(x'') = \pi(x'') = \emptyset$, f permet de définir $\gamma(x'')$ ou $\pi(x'')$, sans altérer $\alpha(x'')$.
On se resteindra donc à des insertions de x" comme sommet terminal de $\mathcal{A}'$ ce qui exclut, en particulier, la fonction PERE ; le positionnement de x" = PERE_f (z) implique de définir, en outre, la relation d'ordre entre z et les FILS de x".

Exemple :



$\gamma_1, \alpha_1, \pi_1$

$\gamma_2(z) = x''$

$\gamma_2(x'') = \gamma_1(z)$ si $\gamma_1(z) \neq \emptyset$

$\pi_2(x'') = \pi_1(z)$ si $\gamma_1(z) = \emptyset$

figure 9

d) Transformations synthétiques

Aux transformations élémentaires de détachement et d'insertion sont adjointes des transformations synthétiques ne concernant qu'une seule arborescence.

Les transformations synthétiques définies actuellement sont au nombre de deux :

L'échange de deux sommets, ou de deux sous arborescences, noté x_1 ECH x_2. correspond aux fonctions α', γ', π' déduites ainsi de α, γ, π

- Si $\alpha(x) \neq x_1, x_2$, $\alpha'(x) = \alpha(x)$

  Si $\alpha(x) = x_1$ $\quad \alpha'(x) = x_2$

  Si $\alpha(x) = x_2$ $\quad \alpha'(x) = x_1$

  Si $\gamma(x) \neq x_1, x_2$ $\quad \gamma'(x) = \gamma(x)$

  $\gamma(x) = x_1$ $\quad \gamma'(x) = x_2$

  $\gamma(x) = x_2$ $\quad \gamma'(x) = x_1$

  Si $x \neq x_1, x_2$ $\quad \pi'(x) = \pi(x)$

  $\gamma'(x) = \gamma(x)$

  Si $x = x_1$, $\gamma(x_1) \neq \emptyset$, $\gamma'(x_2) = \gamma(x_1)$, $\pi'(x_2) = \emptyset$; $\gamma(x_1) = \emptyset$, $\pi'(x_2) = \pi(x_1)$

  Si $x = x_2$, $\gamma(x_2) \neq \emptyset$, $\gamma'(x_1) = \gamma(x_2)$, $\pi'(x_1) = \emptyset$; $\gamma(x_2) = \emptyset$, $\pi'(x_1) = \Pi(x_2)$

L'échange n'est défini que si les sommets x_1 et x_2 ne sont pas comparables selon Γ_1.

Exemples :

1)

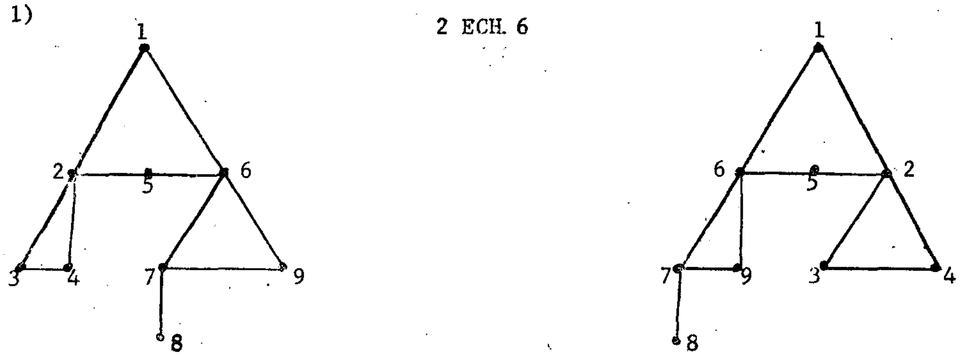


figure 10

2) Dans le cas où il existe une relation de hiérarchie (Γ_1 de l'arborescence) entre x_1 et x_2, l'opération d'échange donnera un résultat aberrant, comme le montre l'exemple suivant :

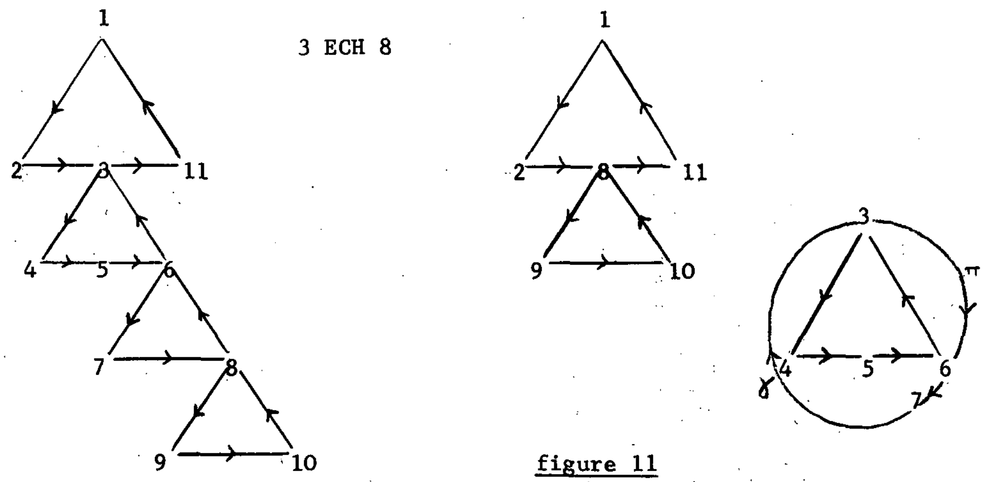


figure 11

On voit que l'arborescence n'est plus connexe. La partie correspondant au sommet de hiérarchie supérieure est séparée de l'arborescence principale. Ce n'est pas une arborescence, car il n'existe pas de racine (sommet tel que $\gamma = \pi = \emptyset$). Le résultat n'est donc ni une arborescence ni une famille d'arborescences.
Remarquons néanmoins qu'une opération de détachement sur le sommet 3 aura pour effet d'en faire la racine d'une arborescence normale.

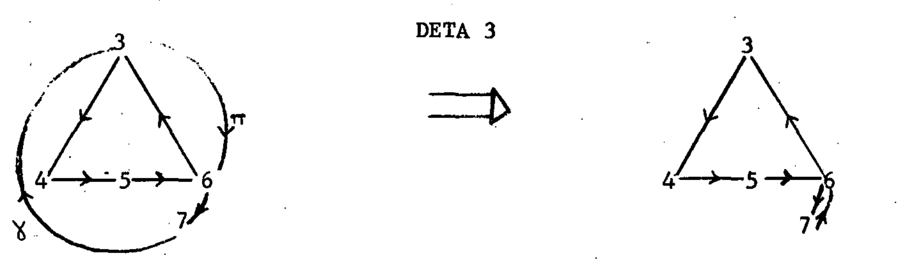


figure 12

L'autre transformation (Absorption ABS) est une suppression de sommet sans suppression de l'arborescence associée. Les relations d'ordre sur Γ_1 et Γ_2 sont préservées de la manière suivante :

soient α, γ, π et α, γ', π' les relations transformées :

ABS (x)

$$\alpha'(x) = \gamma'(x) = \pi'(x) = \emptyset$$

$$\text{Si} \quad \alpha(y) = x \;,\; \alpha(x) \neq \emptyset \;,\; \alpha'(y) = \alpha(x)$$

$$\alpha(x) = \emptyset \;,\; \gamma(x) \neq \emptyset \;,\; \alpha'(y) = \gamma(x)$$

$$\alpha(x) = \emptyset \;,\; \gamma(x) = \emptyset \;,\; \alpha'(y) = \emptyset$$

$$\alpha(y) \neq x \;,\; \alpha'(y) = \alpha(y)$$

$$\text{Si} \quad \gamma(y) = x \;,\; \alpha(x) \neq \emptyset \;,\; \gamma'(y) = \alpha(x)$$

$$\alpha(x) = \emptyset \;,\; \gamma(x) \neq \emptyset \;,\; \gamma'(y) = \gamma(x)$$

$$\alpha(x) = \emptyset \;,\; \gamma(x) = \emptyset \;,\; \pi'(y) = \pi(x)$$

Exemple :

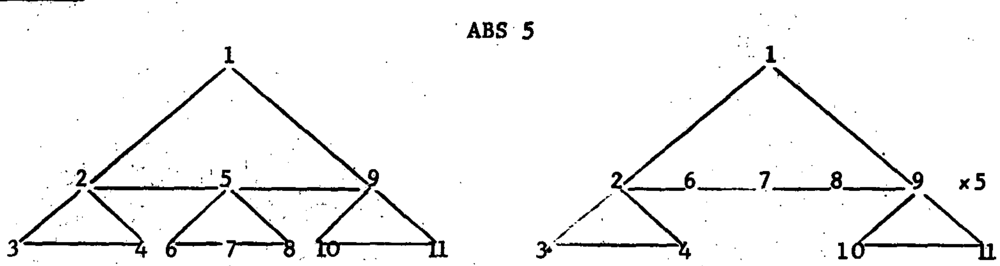


figure 13

e) Transformations .

Nous appellerons transformation une composition d'opérations de transformations élémentaires sur une arborescence donnée. Le résultat de la transformation sera une famille d'arborescences.
L'arborescence principale sera celle qui aura pour racine le même sommet que l'arborescence de départ.
Pour conserver la notion d'arborescence principale, on considère systématiquement que celle-ci est munie d'un sommet racine ω sur lequel toute transformation est interdite. Pour cela ne restreigne en rien les transformations possibles, on ne lui fait jouer aucun rôle dans la structure et on lui donne un fils unique qui est la racine véritable de

l'arborescence.

Les transformations d'insertion seront notées sans y adjoindre le détachement nécessaire, implicitement défini - On notera chaque transformation de la même manière que des relations :

Exemple : " x AINE y " signifie suppression de y et insérer y comme AINE de x .

Par abus du langage, on notera de même l'échange : " x ECH y "

Et les transformations ne portant que sur un seul sommet : " Ø DETA y "

" Ø ABS y "

La composition de transformations élémentaires pourra se noter par une liste (programme) à exécuter.

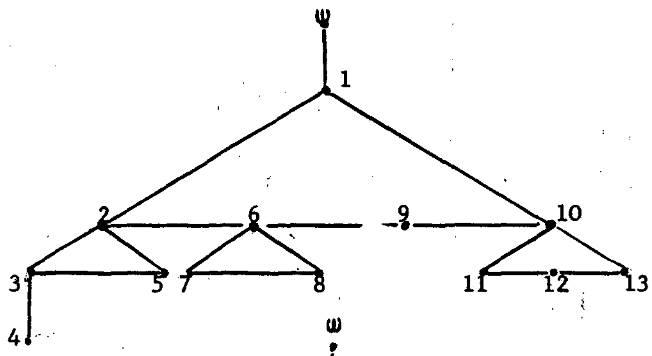
En considérant qu'à une arborescence donnée $\mathcal{A} = (X, \Gamma_1, \Gamma_2)$ on peut associer une famille d'arborescences (Y, Γ_1, Γ_2) en prenant l'ensemble Y infini tel que $X \subset Y$ et que $\wedge x \in Y - X, \alpha(x) = \gamma(x) = \pi(x) = \emptyset$ il sera toujours possible d'adjoindre un nouveau sommet dans l'arborescence principale par insertion. Il suffira de pouvoir définir un identificateur pour chaque sommet (on pourra par exemple, leur associer un entier : si CARD (X) = n, on notera (n+1),(n+2)... etc les sommets de Y - X).

<u>Exemple de transformation</u>

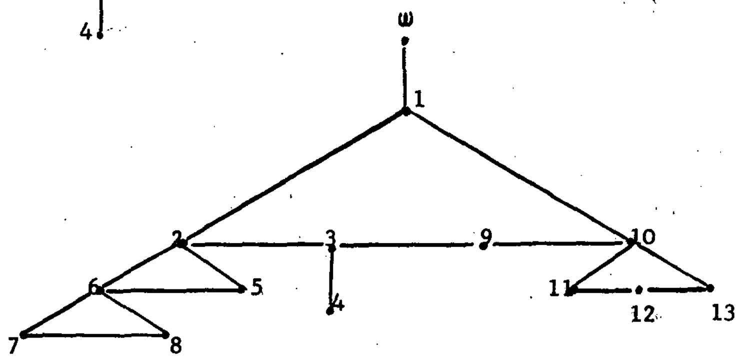
6 ECH 3 } noté aussi 3 SUBST 6
Ø DETA 3 } substitution de z par 6

Ø ABS 6
4 SUBST 2
11 SUBST 3
1 SUBST 11
11 AINE 6
11 AINE 4

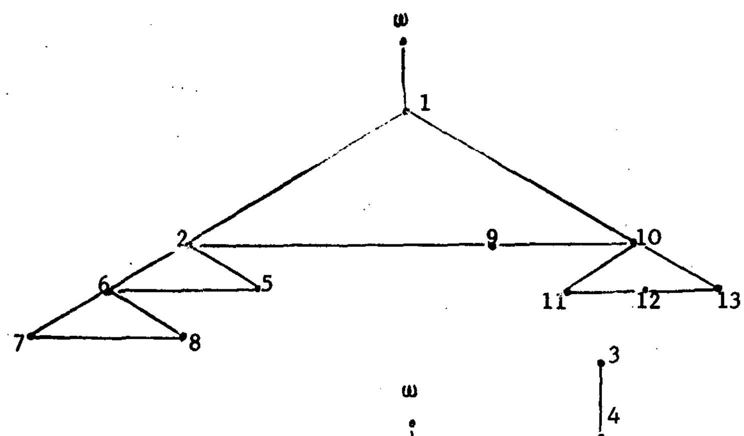
Structure de départ :



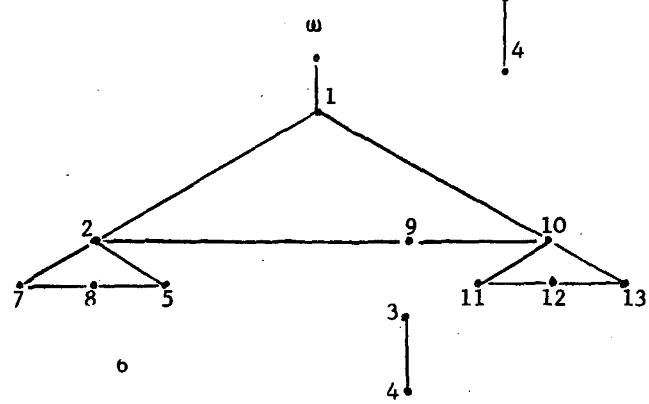
6 ECH 3

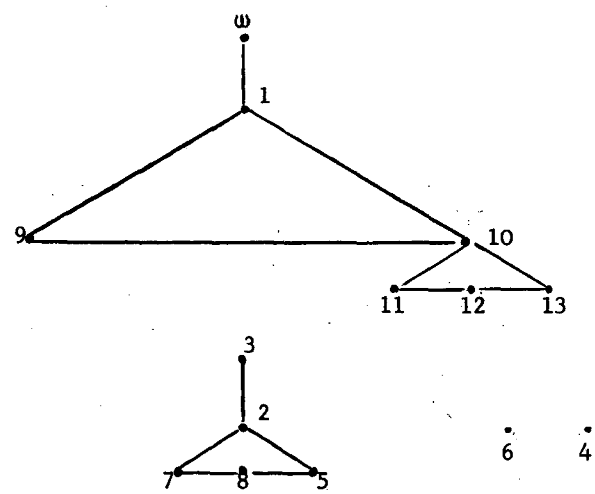


Ø DETA 3

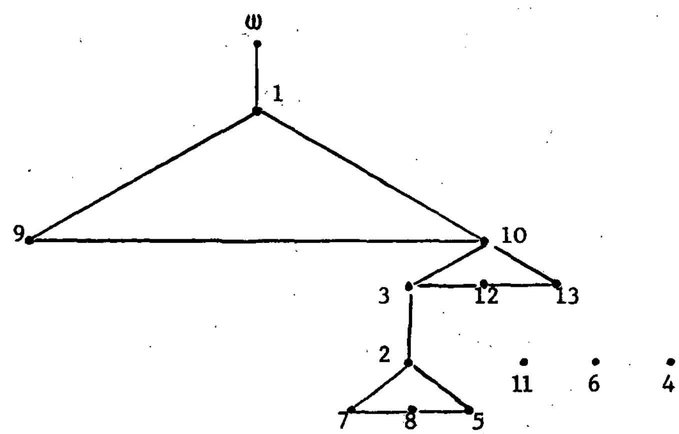


Ø ABS 6





4 SUBST 2



11 SUBST 3

qui conduit à :

1 SUBST 11

11 AINE 6

11 AINE 4

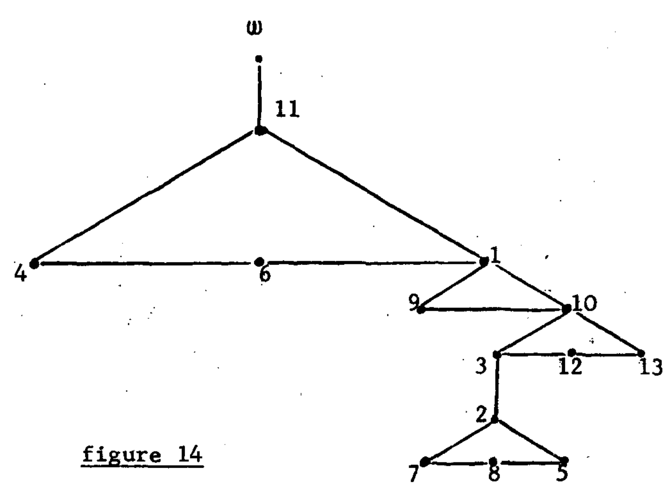


figure 14

6° - Grammaires transformationnelles

a) Règle de transformation sur une structure :

Soit une structure $S = (\mathcal{A}, V, \Delta)$, une règle de transformation est définie par une partie gauche (identification) et une partie droite (transformation)

α) Partie gauche :

La partie gauche a pour but de repérer une figure de pivot donné dans S (4-c).
Elle est constituée par un schéma de figure F muni de symboles du vocabulaire comme il est indiqué en 4-c.
A un pivot donné, un schéma de figure peut faire correspondre plusieurs figures de la structure. Par convention, à une partie gauche de règle on associe la figure minimale conformément à l'ordre total défini en 4-b.
Soit $(x_o, x_1, \dots, x_n)$ cette figure, chacun des sommets est repéré par son indice i $(o \leqslant i \leqslant n)$.

β) Partie droite :

Elle comprend une transformation (5-e) et une application δ.
Les opérandes des transformations élémentaires sont désignés par des entiers j positifs ou nuls.
Pour tout j, $o \leqslant j \leqslant n$, les sommets sont ceux de la figure
Pour tout $j > n$, il s'agit de sommets n'appartenant pas à X (5-e)
L'application δ définit une nouvelle affectation des symboles du vocabulaire sur les sommets spécifiés en partie droite.

γ) Application d'une règle de transformation :

Soient : une structure $S = (\mathcal{A}, V, \Delta)$, de racine ω, une règle de transformation et un pivot x_o.
Le résultat de l'application de la règle est ainsi défini :

1) Si S n'admet aucune figure de pivot x_o conforme à la partie gauche, la structure est inchangée.

2) Si il existe une figure, on ne considère que celle qui est minimale, et le résultat de l'application de la règle sera une

structure $S' = (\mathcal{A}', V, \Delta')$ de racine ω. L'arborescence $\mathcal{A}'$ est obtenue par application de la transformation indiquée en partie droite.
L'application Δ' est telle que $\Delta'x = \Delta x$ si δx n'est pas définie, et $\Delta'x = \delta x$ si δx est définie en partie droite.

δ) Ecriture et exemple :

La partie gauche est notée selon les conventions définies au paragraphe 4-c.

La partie droite comporte la transformation et les affectations de symboles, notées par l'opérateur := ; il est possible de recopier le symbole affecté à un sommet sans en préciser la valeur, en notant le numéro de sommet en guise de symbole.
Reprenons l'exemple donné en 4-c

Partie gauche : * (FILS c(FILS *($\mathcal{C}$)))(FILS d CØUSING c))(CADET a (⊥))

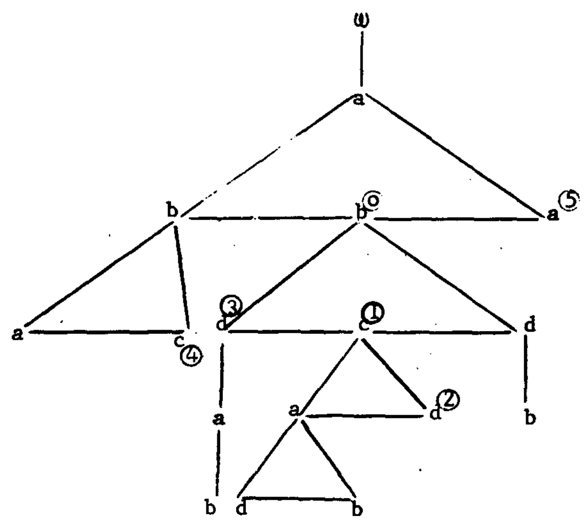


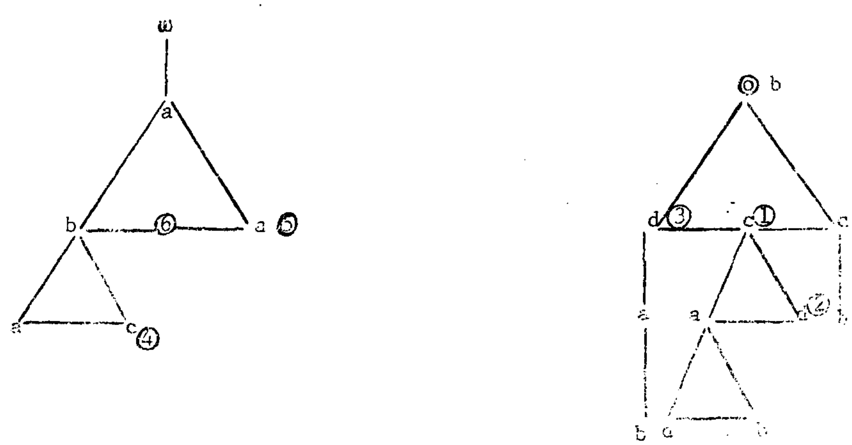
figure 15

Partie droite

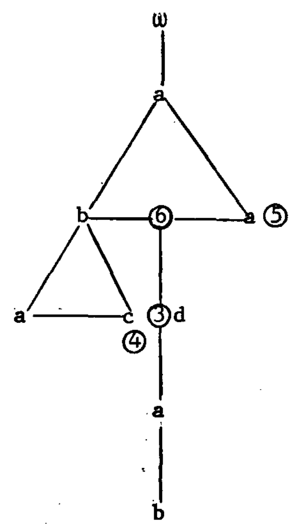
0 ECH 6
6 AINE 3
3 BENJ 0
ABS 0
3 CADET 1
6 := 0
3 := d
4 := a

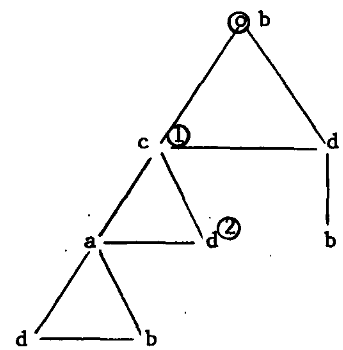
L'arborescence étant donnée, la partie gauche, avec la règle du minima nous sélectionne les 6 sommets marqués.
La transformation aura pour effet de recopier en FILS de 3 tous ses frères non repérés, et à lui conserver les frères repérés.

0 ECH 6 : on échange le sommet pivot et un sommet isolé : d'où deux arborescences.

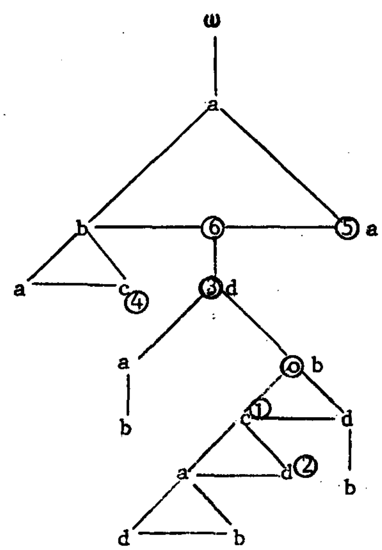


6 AINE 3 : on ramène 3 dans l'arborescence principale

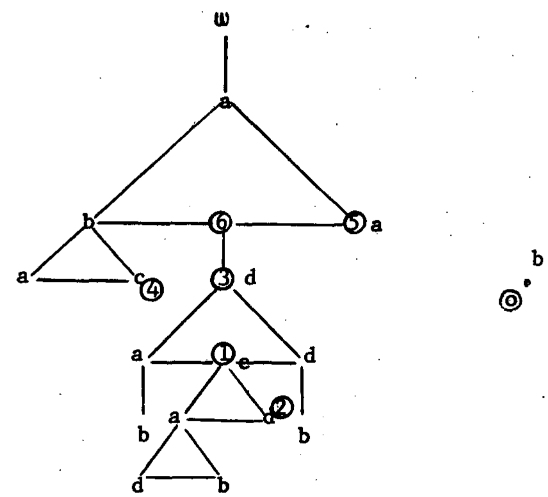




3 BENJ 0 :



ABS O :



3 CADET 1

6 := 0

3 := d

4 := a

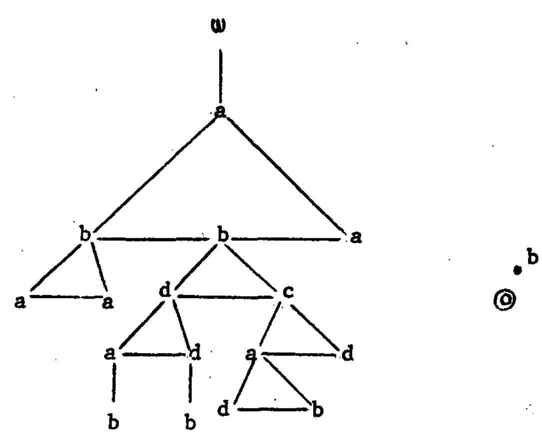


figure 16

b) Grammaire transformationnelle :

Nous appellerons grammaire transformationnelle une liste de règles de transformation.
Une arborescence étant donnée, il peut exister plusieurs dérivations finies pour une grammaire donnée, selon le choix des pivots et l'ordre d'application des règles ; chaque dérivation finie correspond à l'obtention d'une arborescence finale stable, (sur laquelle aucune règle ne peut s'appliquer). Il est cependant possible, moyennant un certain nombre de restrictions, de définir une dérivation unique à partir d'une arborescence donnée ; nous en verrons un exemple dans l'étude des problèmes de mise en oeuvre.

II - P R O G R A M M A T I O N

Nous donnerons ici une description sommaire d'un programme permettant d'exploiter une grammaire transformationnelle.

1° - Représentation d'une structure

Un sommet est représenté par un ensemble de pointeurs (adresses) correspondant aux diverses fonctions. En pratique, compte tenu de propriétés liées au calculateur utilisé (7044 IBM), les fonctions γ et π sont représentées par un même pointeur ; en effet, $\gamma(x) \neq \emptyset \longrightarrow \pi(x) = \emptyset$, $\pi(x) \neq \emptyset \longrightarrow \gamma(x) = \emptyset$. A ce pointeur est associé un indicateur booléen qui définit sa nature, correspondant en fait à la propriété Benjamin. Symétriquement, il a paru commode de définir un troisième pointeur contenant γ^{-1} ou α^{-1} , et un indicateur caractérisant la propriété Ainé. Un sommet peut alors se définir par un Symbole S, un pointeur α , un indicateur B, un pointeur (γ, π), un indicateur A et un pointeur $(\gamma^{-1}, \alpha^{-1})$

avec $\bar{B}\ (x) \longrightarrow \gamma(x) \neq \emptyset\ ,\ \alpha(x) = \emptyset$

et $\bar{A}\ (x) \longrightarrow \gamma^{-1}(x) \neq \emptyset\ ,\ \alpha^{-1}(x) = \emptyset$

La racine de l'arborescence (ω) aura alors les propriétés

$$A, B,\ \gamma(x) = \emptyset,\ \gamma^{-1}(x) = \emptyset$$

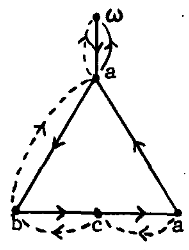
$$\pi(x) = \emptyset, \alpha^{-1}(x) = \emptyset$$

Nous représenterons schématiquement un sommet de la façon suivante :

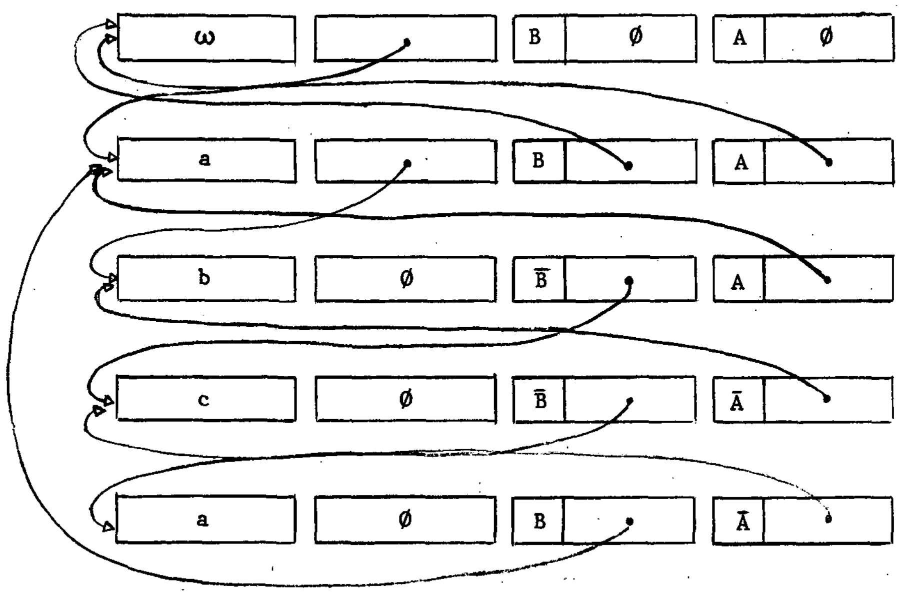
S	α	B	γ, π	A	γ^{-1}, π^{-1}

Exemple :

La structure arborescente suivante,



sera notée :



Cette représentation n'exclut pas, évidemment, l'adjonction d'autres informations sur un sommet donné, et, en particulier, de 3 pointeurs caractérisant son appartenance à une autre arborescence.

2° - Description de la reconnaissance

a) Balayage général :

La grammaire constitue en fait un programme formé par la suite séquentielle des programmes que sont les règles.

Schématiquement, le programme associé à une règle R_i recherche, pour un sommet pivot x donné, si la règle R_i est applicable.

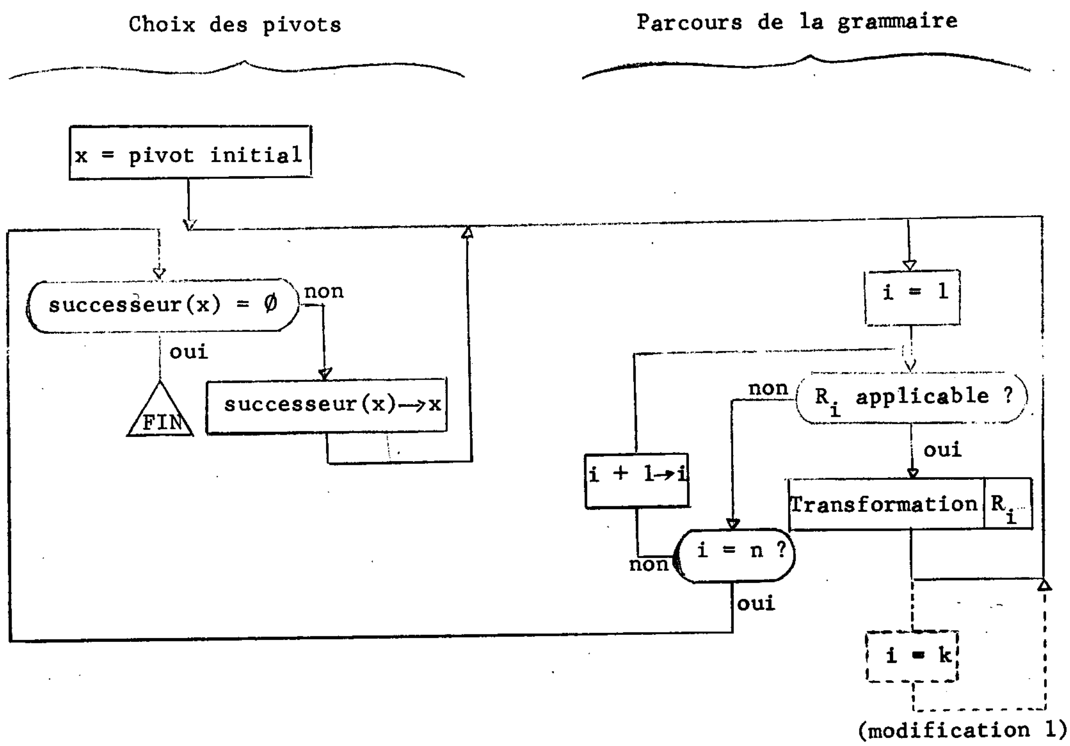
Si la règle n'est pas applicable, le même essai sera renouvelé avec R_{i+1} jusqu'à épuisement des règles de la grammaire.

Si une règle R_i est applicable, la transformation associée est réalisée et le processus complet est repris, pour le même pivot, à partir de la première règle. On voit que l'ordre séquentiel entre les règles définit un ordre de priorité d'application entre celles-ci.

Si aucune règle ne s'applique à un pivot donné, ce pivot est complètement analysé et le contrôle est donné à l'algorithme de choix des pivots, pour en déterminer un autre. Cet algorithme parcourt successivement tous les sommets de l'arborescence dans un ordre fixé (en général, l'ordre habituel de description préfixée d'une arborescence).

Plusieurs modifications peuvent accélérer et rendre plus souple le procédé, en particulier : 1) associer à chaque règle R_i le numéro de la prochaine règle à appliquer si elle est acceptée.

2) Définir des règles de changement de pivot permettant de contrôler le balayage par la grammaire.



b) <u>Recherche d'une figure correspondant à une partie gauche de règle donnée</u>

Le schéma de figure est défini par une expression parenthésée contenant des noms de relations et les noms de symboles qui leurs sont associés (6-a-α).

Soit l'exemple suivant :

$\ast$ (FILS a (FRERE b) (FILS c)) (FILS d)

A chaque relation correspond l'appel à un sous programme dont les paramètres sont le symbole associé et le nombre de parenthèses (n) fermées qui séparent la relation de la relation précédente.

Ainsi, l'exemple précédent se traduira par :

FILS (a,o)
FRERE (b,o)
FILS (c,1)
FILS (d,2)

Le sous programme relation a pour rôle de déterminer le sommet x à partir duquel on doit appliquer la relation, et de déterminer le premier y tel que xRy et que le symbole associé soit égal au symbole demandé.
En pratique, le déroulement du programme utilise une zone de mémoire, dite Pile de contrôle, dans laquelle sont stockées des informations relatives à la règle en cours d'application. A chaque sommet y sélectionné par la règle correspond une adresse dans la pile et un groupe de trois pointeurs : un pointeur vers le sommet x à partir duquel y a été obtenu (Antécédent), un pointeur vers l'adresse exacte de y dans la structure, et un pointeur contenant l'adresse d'appel du sous programme qui a localisé y.
Si P_1, P_2, P_3, P_4 sont ces adresses dans l'exemple donné, nous aurons :

Pile de contrôle

	ANTE.	STRUCT.	APPEL	
	0	S_o	0	sommet 0 (pivot)
P_1 FILS (a,o)	•	S_1	P_1	sommet 1
P_2 FRERE (b,o)	•	S_2	P_2	sommet 2
P_3 FILS (c,1)	•	S_3	P_3	sommet 3
P_4 FILS (d,2)	•	S_4	P_4	sommet 4
	•	0	0	

où S_o, S_1, S_2, S_3, S_4 sont les adresses des sommets dans la structure.

La référence ANTE est mise à jour à chaque nouvelle relation en utilisant le nombre n de parenthèses fermées. Elle est systématiquement initialisée, en admettant n = 0, à la valeur correspondant au sommet précédent dans l'ordre séquentiel.

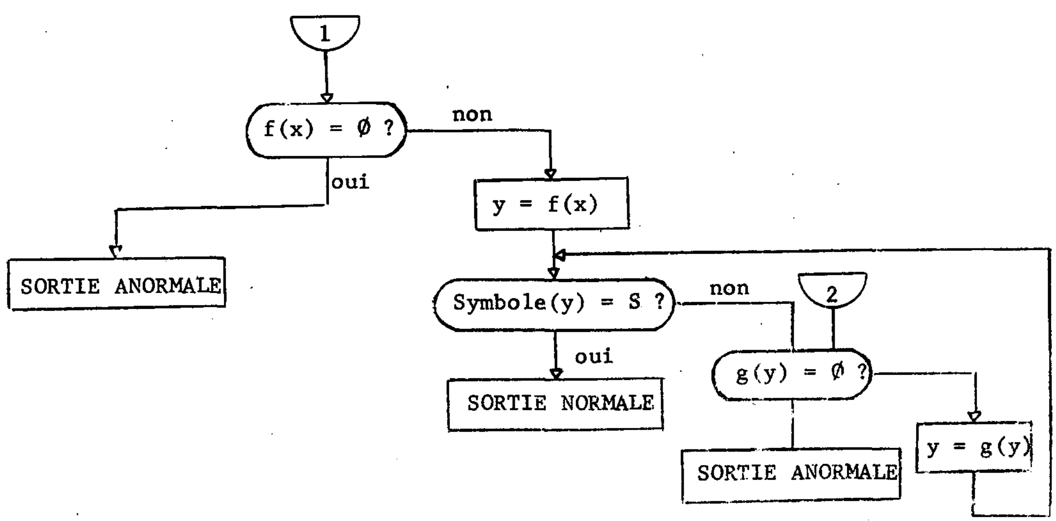
Ainsi, un éventuel sommet S_5 aurait déjà S_4 pour antécédent si n = 0, et si n = 1.

La référence APPEL est nécessaire pour la recherche systèmatique. Ainsi, si l'on avait obtenu S_1 et S_2 à partir de S_o, et si l'on ne trouvait pas de sommet S_3, il faudrait reprendre la recherche à partir d'un autre sommet S'_1.

De S_3, ANTE permet de retrouver S_1, et de réappliquer la relation P_1 au-delà de S_1.

On sait que chaque relation est définie par deux fonctions f et g.

Un organigramme approximatif serait le suivant :



L'entrée 1 correspond au premier appel de la relation. L'entrée 2 correspond au cas où l'on revient sur la relation donnée pour déterminer un autre chemin.

L'entrée 2 fait suite à une sortie anormale dans une autre relation dont le sommet a pour antécédent le sommet dont on repart en 2. Dans l'exemple, P_1 sera appelée par l'entrée 1 au départ, ainsi que P_2 et P_3. Si P_3 donne une sortie anormale, on reprendra P_1 par l'entrée 2, puis à nouveau P_2 et P_3 par l'entrée 1.

3° - <u>Application des transformations</u>

Cette partie ne présente pas de difficultés majeures. A chaque transformation élémentaire correspond un sous programme, dépendant de 3 paramètres en général :

- Le numéro d'ordre du sommet y concerné par la transformation
- Le numéro d'ordre du sommet permettant d'affecter une position à y (ce numéro est vide pour un détachement ou une absorption).
- Le nouveau symbole S' à affecter à y.

Les numéros d'ordre sont utilisés comme des références dans la pile utilisée en partie gauche. La définition d'un nouveau sommet se fait en lui attribuant un numéro d'ordre supérieur à ceux prévus dans le schéma de figure. On obtient ainsi une référence dans la pile à laquelle il ne correspond aucun sommet, ce que l'on interprête en adjoignant un nouveau sommet à la structure.

III - APPLICATION AUX PROBLEMES DE TRADUCTION AUTOMATIQUE

1° - Le rôle des grammaires transformationnelles dans le système de T.A.

L'objectif du modèle de reconnaissance syntaxique (M2), ainsi que la grammaire et les algorithmes qui le réalisent ont fait l'objet d'une publication antérieure [6] [8]

Rappelons cependant que à chaque phrase du texte initial, le modèle M2 fournit une ou plusieurs structures qui se caractérisent de la manière suivante : c'est une structure arborescente dans laquelle à chaque occurrence correspond un sommet ; chaque symbole du vocabulaire V comprend le nom d'une règle de construction syntaxique r (telle que BC$\succ\!\!-$A) et un ensemble d'informations caractérisant l'unité lexicale (variables grammaticales, codes de référence au lexique, etc...)

Entre deux sommets x et y la relation x FILS y a lieu si et seulement si les occurrences associées sont liées par une règle syntaxique r dont x est le gouverneur. Alors r fait partie du symbole de y. En outre, même si l'algorithme de réduction des constituants discontinus a dû être employé dans la phase M2, l'ordre entre les fils d'un même père reste défini par l'ordre sur les occurrences.

<u>Exemple</u> :

Nous traiterons complètement la phrase suivante :

$\#\!\!\!/$ ЭТА ФОРМУЛА ОБЬЯСНЯЕТ ЦАСТОЕ ВОЗНИКНОВЕНИЕ ДЕЙТРОНОВ

В ЯДЕРНИХ РЕАКЦИЯХ.

($\#\!\!\!/$ cette formule explique fréquente apparition [de] deuterons dans [les] nucléaires réactions).

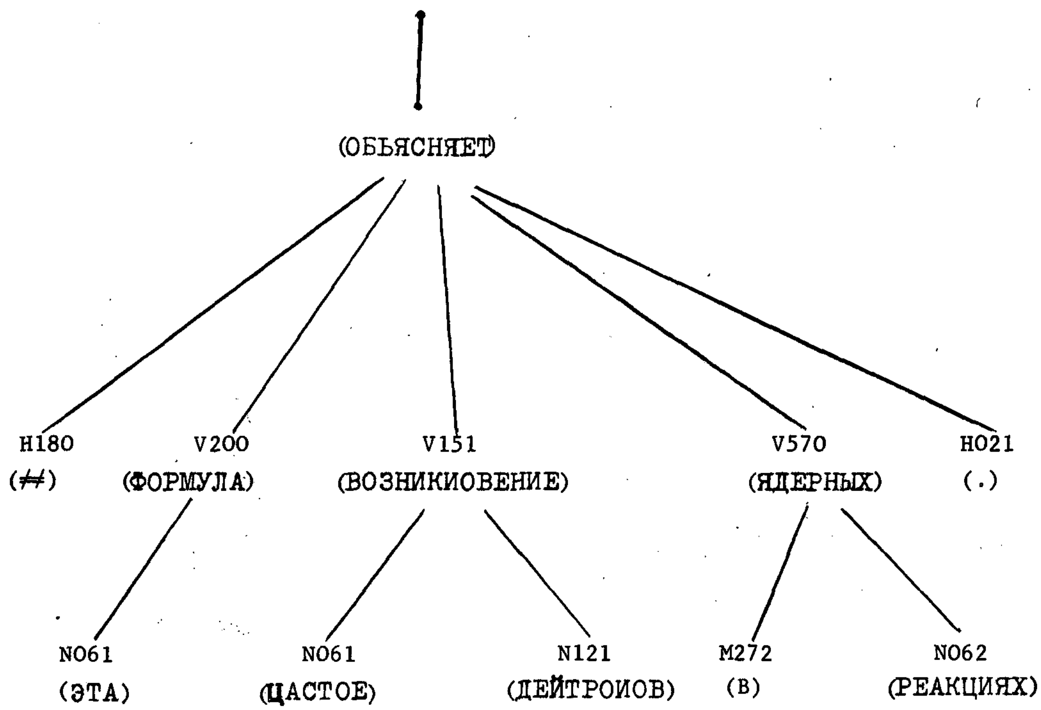


figure 17

En réalité, la structure indiquée représente deux structures compatibles avec la grammaire de M2 :

- elle-même
- celle dans laquelle le groupe prépositionnel est gouverné par ВОЗНИКНОВЕНИЕ (apparition), qui est la structure correcte.

A chaque structure ainsi définie on superpose une autre structure arborescente : la structure de chaîne des occurrences.
Il sera alors possible d'écrire des schémas de figure comportant des relations sur l'une quelconque de ces deux structures ; De plus, le chemin entre deux sommets exprimé par une relation peut être une composition des fonctions α, γ et π sur des structures différentes.

Le passage à la construction syntaxique française se fait au moyen d'une structure intermédiaire (dans le langage pivot) ce qui implique l'emploi de deux modèles transformationnels successifs.

2° - <u>Application au modèle M3</u>

L'objectif du modèle d'étiquetage (M3) consiste à :

- Déterminer la structure syntaxique correcte à partir de la famille de structures implicitement proposées au moyen de transformations sur la structure arborescente initiale.

- Substituer aux noms de règles syntaxiques une étiquette caractérisant la signification de la liaison entre deux mots de manière à éliminer les propriétés formelles de la langue source.

- Supprimer les sommets (occurrences) qui sont relatives à des mots considérés comme outils syntaxiques dans la langue source (par exemple : certaines ponctuations, prépositions fortement régies, etc...)

- Dupliquer certains sommets lorsque les occurrences associées possèdent plus d'une fonction syntaxique.

- Créer de nouveaux sommets lorsque la langue emploie une tournure synthétique.

La phrase de l'exemple précédent va être représentée dans M3 par la structure donnée en figure.

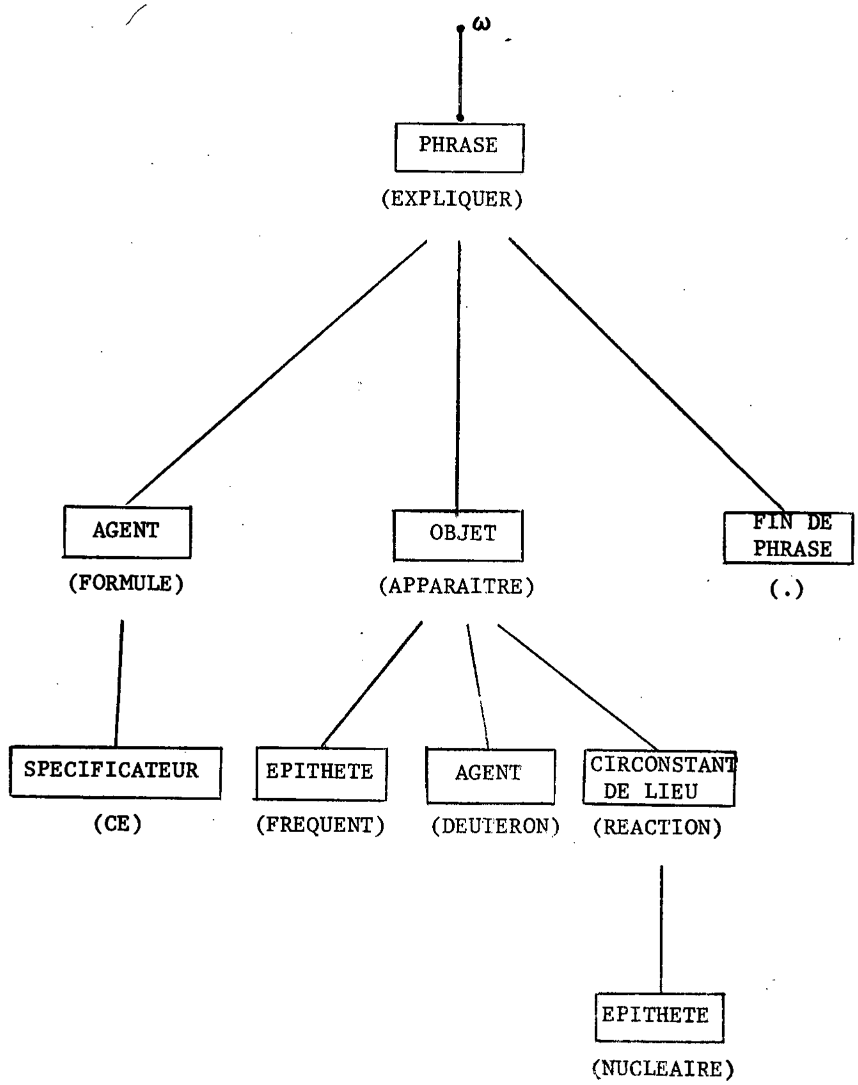


figure 18

Comme chaque symbole du vocabulaire porté par la structure initiale comprend un nom de règle syntaxique et une quantité appréciable d'informations, il est préférable de ne considérer d'abord que le nom de règle pour la recherche d'une figure obéissant à un schéma donné. Les informations composant le reste du symbole seront traitées en tant que valeurs de variables associées à la classe de symboles représentés par le même nom de règle.

Ainsi la partie gauche d'une règle de transformation se divise en deux sections :

- Un schéma de figure
- Des valeurs de variables sur les sommets de la figure identifiée, et, éventuellement, de nouvelles relations faisant apparaître de nouveaux sommets.

<u>Exemple</u> : * (FILS N121) est un schéma de figure sur la structure arborescente qui constitue la première section

a : NON-REFLECHI $\cap$ INTRANSITIF
b : REFLECHI $\cap$ INTRANSITIF
c : TRANSITIF-CONSTANT
d : NON-REFLECHI $\cap$ TRANSITIF-VARIABLE
e : REFLECHI $\cap$ TRANSITIF-VARIABLE
f : QUALITE
g : UNITE
h : ANIME $\cup$ PERSONNIFIE
i : INANIME $\cap$ NON-DENOMBRABLE

qui constituent la deuxième section

Selon différents assemblages de conditions sur les sommets de la figure on peut imaginer que des transformations différentes doivent être effectuées.
Aussi, la règle de transformation peut elle comporter plusieurs parties droites "conditionnelles".
La continuaticn du même exemple fournit les parties droites suivantes

- $(a[o])\quad (d[o] \cap h\ [1] \Longrightarrow 1$:= AGENT
- $(b[o]) \cup (e[o] \cap h\ [1] \Longrightarrow 1$:= AGENT
  2 := OBJET (PRONOMINAL-REFLECHI)
  0 AINE 2
- $(c[o]) \cup (d[o] \cap i\ [1]) \cup (d[o]) \cup (e[o]) \Longrightarrow 1$:= OBJET
- $f[o] \Longrightarrow 1$:= SUBJECT
- $(g[o] \cap i\ [1] \Longrightarrow 1$:= MATIERE
- $* \Longrightarrow 1$:= DETERMINATIF

Dans une telle règle la figure comporte 2 points : le premier point noté 0, le deuxième point noté 1 doit être fils du point 0 et porter le numéro de règle N121.
On trouve une telle figure dans l'exemple traité en partant du point correspondant au mot ВОЗНИКНОВЕНИЕ
Le point o est donc ВОЗНИКНОВЕНИЕ
Le point 1 est alors ДЕЙТРОНОВ

On obtient alors l'étiquette "AGENT" pour ДЕЙТРОНОВ qui est en effet l'agent de l'action "apparaître".

Dans le cas où le deuxième groupe de propriétés est réalisé, on construit un point supplémentaire noté 2 auquel on affecte l'étiquette "OBJET (PRONOMINAL-REFLECHI)" et on le place dans la structure en tant que "fils ainé" du point 0.

L'algorithme d'exploitation de la grammaire cherche à traiter les figures issues de la racine de la structure initiale, puis progresse par "noyaux" successifs. Les règles de la grammaire sont ordonnées. Une dérogation à l'ordre de progression dans les noyaux successifs a lieu dans deux cas :

- On rencontre un mot faisant partie d'une tournure idiomatique (priorité absolue)
- On rencontre une coordination (priorité relative)

Lorsqu'il ne reste plus des points "étiquetés" l'algorithme de transformation s'arrête.

B I B L I O G R A P H I E

[1] - N. CHOMSKY - Aspects of the theory of syntax (1965)
The M.I.T. Press

[2] - Z. HARRIS - String analysis of sentence structure (1962)
MOUTON & C°

[3] - E. S. KLIMA - Relatedness between grammatical systems
Language (1964) volume 40 n° 1

[4] - S. KUNO - A system for transformational analysis
(Congrès NEW-YORK - mai 1965)

[5] - PETRICK - A recognition procedure for transformational grammars

[6] - B. VAUQUOIS, G. VEILLON, J. VEYRUNES - Syntaxe et Interpretation
(Congrès NEW-YORK - mai 1965)

[7] - HOLT - A mathematical and applied investigation of tree structures for computer syntactic analysis
Thèse Univ. of Pennsylvania

[8] - G. VEILLON, J. VEYRUNES - Etude de la réalisation pratique d'une grammaire "Context-Free" et de l'algorithme associé
Document G-2001-1 - avril 1964