



HAL
open science

Grammaires de graphes, algorithme d'analyse : applications

Jean Azema

► **To cite this version:**

Jean Azema. Grammaires de graphes, algorithme d'analyse : applications. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1975. Français. NNT: . tel-00286260

HAL Id: tel-00286260

<https://theses.hal.science/tel-00286260>

Submitted on 9 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE
INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

POUR OBTENIR LE GRADE DE
DOCTEUR DE 3^{ème} CYCLE

Jean AZEMA

Grammaires de graphes.
Algorithme d'analyse ~ Applications.

Thèse soutenue le 6 mars 1975 devant la Commission d'Examen :

Président : J. KUNTZMANN

Examineurs : C. BENZAKEN
G. VEILLON

M. Michel SOUTIF

Présidents

M. Louis NEEL

M. Gabriel CAU

Vice-Présidents

MM. Lucien BONNETAIN

Jean BENOIT

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BEAUDOING André	Clinique de Pédiatrie et Puériculture
	BERNARD Alain	Mathématiques Pures
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BEZES Henri	Pathologie chirurgicale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BOUCHERLE André	Chimie et toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARLIER Georges	Biologie végétale
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Clinique Oto-Rhino-Laryngologique
	CHATEAU Robert	Thérapeutique (Neurologie)
	CHIBON Pierre	Biologie animale
	COEUR André	Pharmacie chimique et chimie analytique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie pathologique
	CRAYA Antoine	Mécanique
Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBERMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumo-Phtisiologie
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée

MM.	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DRUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de dermatologie et syphiligraph
	FAU René	Clinique neuro-psychiatrique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques pures
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Mathématiques appliquées
	GAVEND Michel	Pharmacologie
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	KAHANE André	Physique Générale
	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LOUP Jean	Géographie
Mlle	LUTZ Elisabeth	Mathématiques pures
	MALGRANGE Bernard	Mathématiques pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	MULLER Jean-Michel	Thérapeutique (néphrologie)
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-chirurgie
	SEIGNEURIN Raymond	Microbiologie et hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction mécanique
	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
MM.	VERAIN André	Physique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	CHEEKE John	Thermodynamique
	COPPENS Philip	Physique
	CORCOS Gilles	Mécanique
	CRABBE Pierre	CERMO
	GILLESPIE John	I.S.N.
	ROCKAFELLAR Ralph	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

Mlle	AGNIUS-DELDORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BERTRANDIAS Jean-Paul	Mathématiques pures
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
Mme	BONNIER Jane	Chimie générale
MM.	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique
	CONTE René	Physique
	DEPASSEL Roger	Mécanique des fluides
	GAUTHIER Yves	Sciences biologiques
	GAUTRON René	Chimie
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	GROULADE Joseph	Biochimie médicale
	HACQUES Gérard	Calcul numérique
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Méd. Préventive
	IDELMAN Simon	Physiologie animale
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
Mme	KAHANE Josette	Physique
MM.	KUHN Gérard	Physique
	LOISEAUX Jean	Physique nucléaire
	LUU-DUC-Cuong	Chimie organique
	MAYNARD Roger	Physique du solide
	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	PFISTER Jean-Claude	Physique du solide
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	Mathématiques appliquées
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	RICHARD Lucien	Biologie végétale
Mme	RINAUDO Marguerite	Chimie macromoléculaire
MM.	ROBERT André	Chimie papetière
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
MM.	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	AMBLARD Pierre	Dermatologie
	ARMAND Gilbert	Géographie
	ARMAND Yves	Chimie
	BARGE Michel	Neurochirurgie
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamique
M.	BOUCHARLAT Jacques	Psychiatrie adultes
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BRODEAU François	Mathématiques (IUT B)
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie papetière
	CHIAVERINA Jean	Biologie appliquée (EFP)
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CORDONNIER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	CYROT Michel	Physique du solide
	DELOBEL Claude	M.I.A.G.
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie végétale
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	FONTAINE Jean-Marc	Mathématiques pures
	GAUTIER Robert	Chirurgie générale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROS Yves	Physique (stag.)
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	KOLODIE Lucien	Hématologie
	KRAKOWIAK Sacha	Mathématiques appliquées
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LEROY Philippe	Mathématiques
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)
	MICHOULIER Jean	Physique (IUT A)
Mme	MINIER Colette	Physique
MM.	NEGRE Robert	Mécanique
	NEMOZ Alain	Thermodynamique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PERRET Jean	Neurologie
	PHELIP Xavier	Rhumatologie
	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RAMBAUD Pierre	Pédiatrie
Mme	RENAUDET Jacqueline	Bactériologie
MM.	ROBERT Jean-Bernard	Chimie-Physique

MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VROUSOS Constantin	Radiologie

MAITRES DE CONFERENCES ASSOCIES

MM.	COLE Antony	Sciences nucléaires
	FORELL César	Mécanique
	MOORSANI Kishin	Physique

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

MM.	BOST Michel	Pédiatrie
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	FAURE Gilbert	Urologie
	MALLION Jean-Michel	Médecine du travail
	ROCHAT Jacques	Hygiène et hydrologie

Fait à Saint Martin d'Hères, OCTOBRE 1974.

o

"MEMBRES DU CORPS ENSEIGNANT DE L'I.N.P.G."

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Radioélectricité
BESSON Jean	Electrochimie
BONNETAIN Lucien	Chimie Minérale
BONNIER Etienne	Electrochimie, Electrometallurgie
BRISSONNEAU Pierre	Physique du solide
BUYLE-BODIN Maurice	Electronique
COUMES André	Radioélectricité
FELICI Noël	Electrostatique
PAUTHENET René	Physique du solide
PERRET René	Servomécanismes
SANTON Lucien	Mécanique
SILBER Robert	Mécanique des fluides

PROFESSEUR ASSOCIE

M. BOUDOURIS Georges	Radioélectricité
----------------------	------------------

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuel	Electronique
BLOCH Daniel	Physique du solide et cristallographie
COHEN Joseph	Electrotechnique
DURAND François	Metallurgie
MOREAU René	Mécanique
POLOUJADOFF Michel	Electrotechnique
VEILLON Gérard	Informatique fondamentale et appliquée
ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM. BOUVARD Maurice	Génie mécanique
CHARTIER Germain	Electronique
FOULARD Claude	Automatique
GUYOT Pierre	Chimie minérale
JOUBERT Jean Claude	Physique du solide
LACOUME Jean Louis	Géophysique
LANCIA Roland	Physique atomique
LESPINARD Georges	Mécanique
MORET Roger	Electrotechnique nucléaire
ROBERT François	Analyse numérique
SABONNADIÈRE Jean Claude	Informatique fondamentale et appliquée
Mme SAUCIER Gabrièle	Informatique fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M. LANDAU Ioan Doré	Automatique
---------------------	-------------

CHARGE DE FONCTIONS DE MAITRES DE CONFERENCES

M. ANCEAU François	Mathématiques appliquées
--------------------	--------------------------

Je tiens à remercier

Monsieur le Professeur J. KUNTZMAN, qui a bien voulu me faire l'honneur de présider le Jury;

Monsieur C. BENZAKEN, qui a bien voulu juger mon travail avec beaucoup d'attention;

Monsieur G. VEILLON pour la façon dont il a bien voulu m'orienter dans ce travail et m'aider de ses conseils;

Je n'oublierai pas tous ceux avec qui j'ai travaillé, en particulier J.C. SAILLARD pour sa collaboration efficace.

Je tiens également à remercier le service de tirage, ainsi que Madame JOUFFRE pour le soin qu'elle a mis à la réalisation matérielle de ce document.

INTRODUCTION

CHAPITRE 1

	Description d'un chemin par un graphe	
1.1.	Description de dessins et reconnaissance de formes	1
1.1.1.	Méthode par classification	
1.1.2.	Méthode par décomposition	
1.2.	Représentation d'un dessin par un graphe	4
1.2.1.	Relations entre primitives	
1.2.2.	Graphe des relations	
1.2.3.	Graphe des primitives	
1.3.	Dessin automatique d'un graphe	10
1.3.1.	Dessin et terminal graphique	
1.3.2.	Dessin d'un graphe sans renseignements	
1.3.3.	Renseignements sur le graphe	

CHAPITRE 2

	Description formelle d'un graphe	
2.1.	Principes utilisés	16
2.1.1.	Motivations	
2.1.2.	Graphes et grammaires	
2.1.3.	Les différents modèles de grammaires	
2.2.	Description par une expression produite par une grammaire	18
2.2.1.	Définition	
2.2.2.	Exemples de concaténations	
2.2.3.	Autres opérations	
2.2.4.	Description du graphe	
2.2.5.	Grammaires produisant des descriptions de graphes	
2.2.6.	Applications	
2.3.	Grammaires de structure	25
2.3.1.	Principe	
2.3.2.	Description du dessin	
2.3.3.	Exemple de dérivation	
2.3.4.	Comparaison avec les grammaires de chaîne	
2.4.	Grammaires de graphes	29
2.4.1.	Principe	
2.4.2.	Exemples de grammaires de graphes et de dérivations	
2.4.3.	Comparaison avec les autres grammaires.	

CHAPITRE 3

	Graphes et Langages	
3.1.	Remarques préliminaires	39
3.2.	Définitions	39
3.2.1.	Graphe renseigné	

3.2.2.	Grammaire de graphe	
3.3.	Exemples de grammaires de graphes	56
3.3.1.	Rappels sur les graphes planaires et les graphes séparables	
3.3.2.	Grammaire de graphe produisant les graphes planaires sans isthmes	
<u>CHAPITRE 4</u>	Grammaires de graphes et analyse syntaxique	
4.1.	Nécessité de l'analyse syntaxique	80
4.1.1.	Position du problème	
4.1.2.	Problèmes se ramenant à un problème d'analyse syntaxique	
4.2.	Les principes de l'analyse syntaxique et les grammaires de graphes	81
4.2.1.	Critères de classification des algorithmes	
4.2.2.	Analyse ascendante	
4.2.3.	Analyse descendante	
4.2.4.	Choix de l'algorithme	
4.3.	Analyse syntaxique pour la grammaire de MONTANARI	88
4.3.1.	Existence d'une dérivation symétrique	
4.3.2.	Algorithme d'analyse	
4.3.3.	Amélioration de l'algorithme	
4.3.4.	Programmation	
4.3.5.	Evaluation et interprétation des résultats	
<u>CHAPITRE 5</u>	Applications des grammaires de graphes	
5.1.	Remarque préliminaire	127
5.2.	Reconnaissance d'un graphe	127
5.2.1.	Reconnaissance de formes	
5.2.2.	Planéarité	
5.3.	Production d'un graphe	128
5.3.1.	Production de types de graphes particuliers	
5.3.2.	Dessin de graphes	
5.4.	Isomorphisme	138
5.5.	Représentation interne d'un graphe	142
5.5.1.	Rappels	
5.5.2.	Représentation par une dérivation	
5.5.3.	Représentation des graphes planaires sans isthmes	
5.5.4.	Codage des cartes planaires	

I N T R O D U C T I O N

Dans le domaine du traitement automatique des dessins, l'utilisation de modèles génératifs semblables aux grammaires formelles, permet une approche originale de nombreux problèmes.

Différents travaux ont été faits dans ce sens relatifs essentiellement à des problèmes de reconnaissance de formes.

En introduisant le concept de "Grammaire de Graphe" on peut alors définir un modèle général.

Pour illustrer les possibilités d'utilisation de ces grammaires de graphes nous avons choisi une grammaire de MONTANARI produisant les graphes planaires sans isthme .

Dans le premier chapitre nous verrons comment l'on peut représenter un dessin par un graphe, ce qui pour la suite permet de ramener tout problème de traitement de dessin à un problème de traitement de graphes.

Le second chapitre est consacré à l'étude des différents modèles génératifs utilisés dans la représentation ou le traitement des graphes ou des dessins : expression produite par une grammaire de "chaîne", grammaire de structure, grammaire de graphe.

Dans le troisième chapitre, après avoir défini les grammaires de graphes et en avoir étudié les principales propriétés, on en donne plusieurs exemples . On étudie plus particulièrement la grammaire donnée par MONTANARI, pour pouvoir par la suite construire un algorithme d'analyse.

Le quatrième chapitre est donc consacré à l'étude des problèmes posés par l'analyse dans le cas des grammaires de graphes. On voit ainsi comment on peut obtenir un algorithme d'analyse efficace pour la grammaire produisant les graphes planaires sans isthme.

C'est dans le cinquième chapitre que l'on voit les possibilités d'application des grammaires de graphes : problème de reconnaissance directement lié à l'analyse syntaxique; recherche de l'isomorphisme de deux graphes; génération et dessin de graphes, avec en particulier un algorithme pour le dessin planaire des graphes planaires; possibilité d'un codage très économique d'un graphe par la suite des règles de la dérivation.

Enfin pour un graphe planaire sans isthmes nous verrons comment on peut donner le couple de permutations (σ, α) de la carte planaire correspondante (au sens de CORI [52]) et donc utiliser pour le codage le code de Lehman Lenormand.

CHAPÎTRE

1

DESCRIPTION D'UN DESSIN PAR UN GRAPHE.

1.1. DESCRIPTION DE DESSINS ET RECONNAISSANCE DE FORMES.

Pour pouvoir traiter en machine un dessin quelconque, ou une représentation d'un objet, on est amené à formaliser une description de ce dessin ou de cet objet.

Les divers modèles de description que nous connaissons, ont, la plupart du temps, été donnés par des auteurs s'intéressant à des problèmes de reconnaissance de formes [13.17.20.25].

C'est dans ce domaine, en effet, que le besoin d'outils pour le traitement automatique des dessins s'est fait le plus vite sentir, en raison des lacunes de la méthode par classification, qui était l'approche traditionnelle du problème

1.1.1. METHODE PAR CLASSIFICATION.

Principe.

Le principe de la reconnaissance des formes par classification, consiste à munir l'ensemble des formes que l'on veut reconnaître d'une relation d'équivalence,

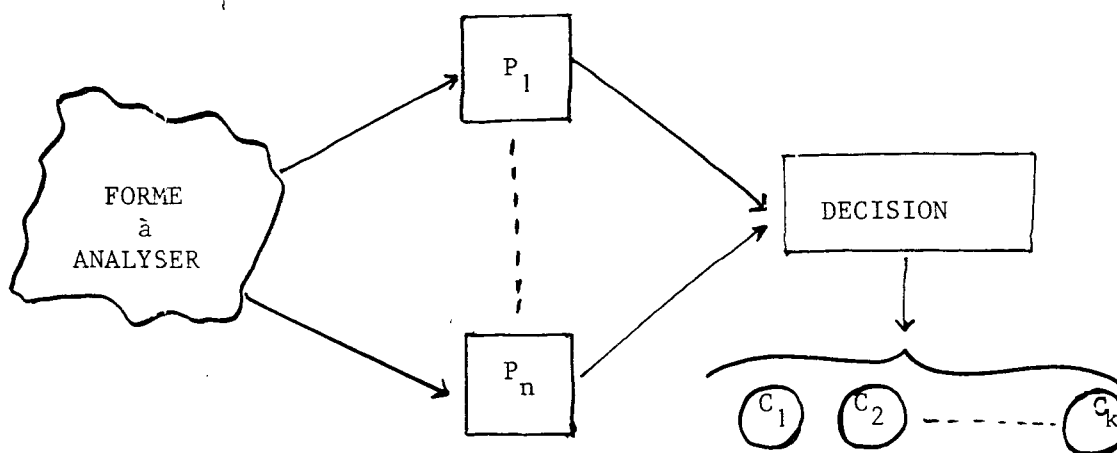
Le nom, et donc la description d'une forme reconnue par un tel système sera le nom de la classe d'équivalence à laquelle la forme appartient.

Pratiquement on procède ainsi :

Des filtres (détecteurs physiques, opérateurs....) calculent les propriétés des formes à reconnaître selon les valeurs d'attributs caractéristiques (intensité, niveau de gris, couleur, opacité ou transparence etc...)

ou en fonction de certaines configurations [20].

C'est à partir de ces propriétés que l'on décide à quelle catégorie appartient la forme.



INSUFFISANCE DE LA METHODE.

On ne peut par ce procédé, que reconnaître les formes relativement simples. Même si dans certains cas (reconnaissance de lettres) on peut décomposer la forme en éléments plus simples (segments de droite ou de courbes), ce procédé n'est pas systématique et la reconnaissance de formes se fait globalement.

Chaque système est apte à reconnaître un nombre très limité de formes, et on ne peut, en se limitant à ce procédé, envisager de pouvoir reconnaître des formes qui, bien que constituées d'éléments assez simples, ont une structure complexe, comme :

- des photographies aériennes
 - des résultats d'expérience en chambre à bulle
 - des schémas de circuits électriques
 - des plans d'architecture
 - des dessins biomédicaux
 - des représentations de molécules chimiques
 - des scènes naturelles.
- etc....

Avec de telles structures on est amené à se poser des questions telles que :

- est-ce qu'on trouve tel objet dans la forme ?
- est-ce que tel objet apparaît avec telle qualité ?
- quelle est la position de tel objet par rapport à tel autre ?
- peut-on donner **une** description complète du dessin ?
- etc....

Il s'agit alors de savoir classer la forme dans une catégorie ou dans une autre,

Pour pouvoir répondre à ces questions, on doit être à même de donner une description détaillée de la forme à analyser : nature et propriétés des objets qu'elle contient, ainsi que les relations existant entre ces objets.

On est donc amené à décomposer les formes en éléments plus petits, et à introduire un formalisme permettant de décrire les relations entre ces éléments.

1.1.2. METHODE PAR DECOMPOSITION.

PRINCIPE.

Le principe de la reconnaissance de formes par décomposition, consiste à rechercher dans une forme à structure complexe, les éléments fondamentaux, ou élémentaires, dont l'assemblage constitue la forme. Ces éléments sont souvent appelés *primitives*.

Une fois cette décomposition faite, on peut facilement avoir une représentation schématique de la forme. Les primitives sont représentées par une figure géométrique simple et il ne reste plus qu'à décrire les relations entre ces primitives.

C'est pour cette description que de nombreux modèles ont été proposés [28].

Dans ces modèles on retrouve une formalisation de la description des dessins ou d'une classe de dessins.

On peut, dans ce processus de formalisation, distinguer deux étapes, plus ou moins distinctes, suivant les modèles

- 1°/ Représentation du dessin au moyen d'un graphe
- 2°/ Description du graphe, éventuellement au moyen d'un système formel.

1.2. REPRESENTATION D'UN DESSIN PAR UN GRAPHE.

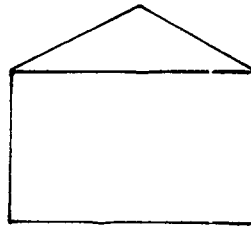
1.2.1. RELATIONS ENTRE PRIMITIVES

Soit d un dessin et soit $P = \{p_1, \dots, p_m\}$ l'ensemble des primitives contenues dans d .

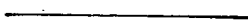
Entre les éléments de P sont définies un certain nombre de relations n -aires notées $R(x_1, \dots, x_n)$ ou $x_i \in P$ pour $1 \leq i \leq n$.

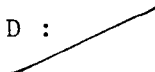
EXEMPLE,


Soit le dessin suivant :

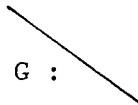


On peut y distinguer six primitives appartenant à quatre classes de primitives, à savoir

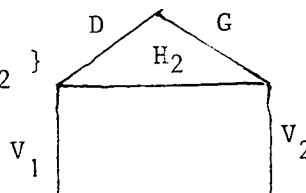
H : 

D : 

V : 

G : 

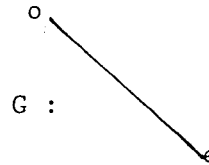
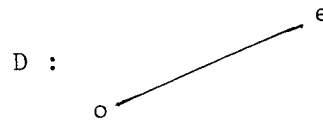
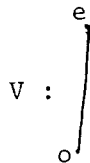
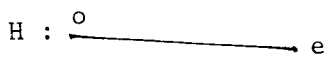
On a donc $P = \{D, G, V_1, V_2, H_1, H_2\}$



On définit une relation binaire entre ces primitives ; la concaténation

Pour cela on étiquette les deux extrémités des primitives : l'une o, l'autre e. (Dans le cas général on prendra deux points distincts faisant figure d'extrémité),

Les primitives avec leurs extrémités étiquetées sont alors :



La relation binaire de concaténation entre deux primitives P_i et P_j se note $CAT(P_i, P_j)$,

La définition de cette relation est : l'extrémité e de P_i est confondue avec l'extrémité o de P_j .

Le dessin ci-dessus sera donc décrit par les relations :

$$CAT(V_1, H_2)$$

$$CAT(V_1, D)$$

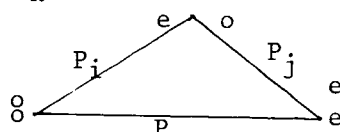
$$CAT(D, G)$$

$$CAT(H_1, V_2)$$

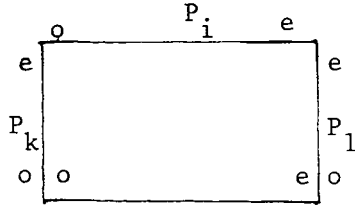
Avec la relation CAT telle qu'elle est définie, on ne peut exprimer que les extrémités de G, H_2 et V_2 sont confondues, ainsi que pour les extrémités e de V_1 et H_1 .

On doit donc définir d'autres relations de concaténation.

On peut donc définir une relation ternaire $TRI(P_i, P_j, P_k)$ signifiant que les trois primitives P_i, P_j, P_k forment un triangle de la façon suivante :



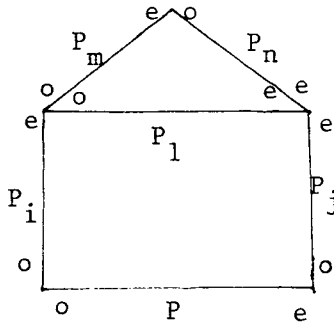
et une relation quaternaire $QUAT(P_i, P_j, P_k, P_l)$ signifiant que les quatre primitives P_i, P_j, P_k, P_l forment un quadrilatère de la façon suivante :



Le dessin est alors décrit par

$$TRI(D, G, H_2) \quad QUAT(H_2, H_1, V_1, V_2)$$

Enfin l'on peut définir une relation à six éléments MAISON $(P_i, P_j, P_k, P_l, P_m, P_n)$ ainsi définie



Le dessin sera défini par MAISON $(V_1, V_2, H_1, H_2, D, G)$.

Partant d'un dessin formé de m primitives, on peut le décrire par une relation à m éléments.

On peut également le décrire par un ensemble de relations à n éléments $2 \leq n < m$.

De toute façon, comme nous le verrons plus loin, on se ramène toujours à un ensemble de relations binaires, et ce sont ces relations que l'on a représenté à l'aide de graphes.

1.2.2. GRAPHES DES RELATIONS

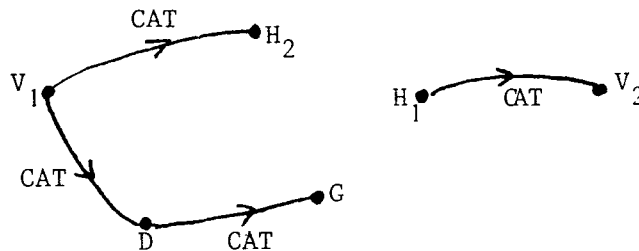
Une première façon de représenter à l'aide de graphes les relations entre les primitives d'un dessin, consiste à se définir ainsi des graphes orientés et renseignés,

Les sommets du graphe correspondent aux primitives et les arêtes expriment les relations entre ces primitives.

Il y a une arête étiquetée par R_k entre les sommets étiquetés P_i et P_j si on a la relation $R_k(P_i, P_j)$.

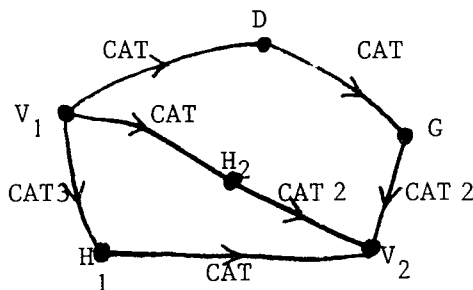
EXEMPLES

1°/ Le graphe représentant la relation CAT pour le dessin ci-dessus sera :



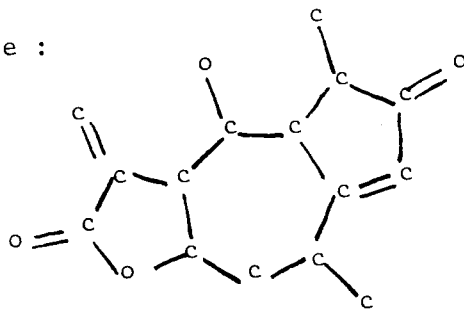
On pourra dans le même graphe exprimer d'autres relations de concaténation, notamment entre les primitives G et V_2 , et H_2 et V_2 d'une part et entre H_1 et V_1 d'autre part.

Si $CAT2(G, V_2)$, $CAT2(H_2, V_2)$ et $CAT3(V_1, H_1)$ sont ces relations, le graphe sera

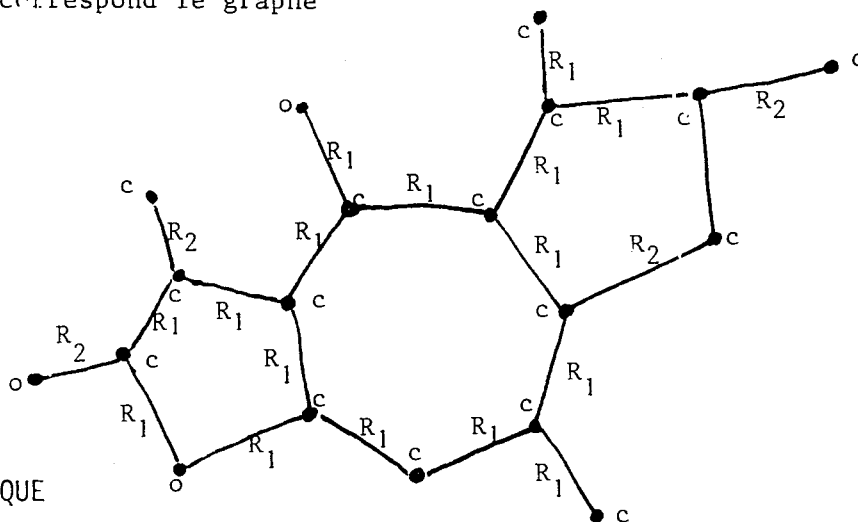


2°/ Pour décrire le schéma d'une molécule de chimie organique, on prendra comme primitives les atomes. On définit une relation binaire symétrique R_1, R_2, R_3 entre les atomes selon qu'il existe une liaison simple, double ou triple entre ces atomes.

A la molécule :



correspond le graphe



REMARQUE

Lorsqu'on a des relations entre primitives, qui ne sont pas des relations binaires, on décompose la relation en conjonction de relations binaires

$$R(P_1, \dots, P_n) \rightarrow R_1(P_{11}, P_{12}) \wedge R_2(P_{21}, P_{22}) \wedge \dots \wedge R_k(P_{k1}, P_{k2})$$

avec

$$P_{11}, P_{12}, \dots, P_{k1}, P_{k2} \in \{P_1, \dots, P_n\}$$

Par exemple

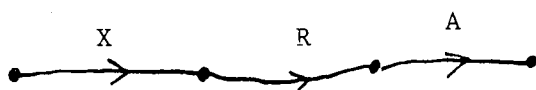
$$\text{TRI}(P_i, P_j, P_k) \rightarrow \text{CAT}(P_i, P_j) \wedge \text{CAT2}(P_j, P_k) \wedge \text{CAT3}(P_i, P_k)$$

1.2.3. GRAPHE DES PRIMITIVES

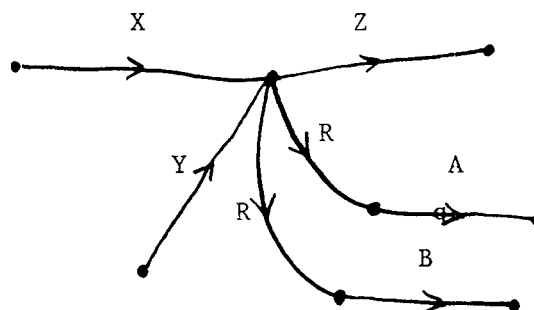
Ayant défini sur les primitives deux extrémités o et e on peut assimiler une primitive à une arête. Dans ce cas il n'est pas nécessaire de représenter les relations de concaténation puisqu'elles sont données par le graphe lui-même.

On doit cependant représenter par d'autres arêtes les autres relations, ce qui peut entraîner dans certains cas des ambiguïtés.

Par exemple, si on a les primitives A, B, X, Y, Z et les relations $CAT(X, Z)$, $CAT(Y, Z)$, $R(X, A)$, $R(Y, A)$ sachant que l'on représente $R(X, A)$ par



Le graphe correspondant sera :



graphe qui décrit également les relations $R(X, B)$ et $R(Y, A)$.

On réservera donc la représentation d'un dessin par le graphe des primitives aux seuls cas où les relations entre les primitives sont des relations de concaténation.

C'est par exemple le cas pour les schémas des molécules chimiques où les primitives sont d'une part des liaisons et d'autre part les atomes. Dans ce cas le graphe est une reproduction fidèle du dessin, et, comme nous le verrons au paragraphe suivant on peut, à partir du graphe, obtenir une représentation graphique du dessin.

Comme on peut s'en rendre compte, le problème de la représentation d'un dessin par un graphe, n'est pas un problème définitivement résolu.

En fait, pour chaque cas particulier, c'est-à-dire chaque classe de dessin à représenter, il faut choisir la solution la plus appropriée, et ce en fonction de la nature et du nombre des primitives contenues dans le dessin, ainsi que des relations entre elles.

1.3. DESSIN AUTOMATIQUE D'UN GRAPHE.

1.3.1. DESSIN ET TERMINAL GRAPHIQUE.

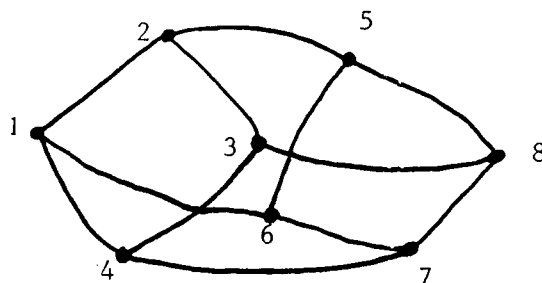
La façon de résoudre le problème du dessin sur un terminal graphique dépend :

- d'une part des informations à partir desquelles on va pouvoir élaborer le dessin
- d'autre part de la nature du terminal utilisé et des techniques de programmation dont on dispose.

EXEMPLE.

Si on veut dessiner un cube, les différentes informations dont on dispose peuvent être

- les coordonnées relatives dans l'espace des huit sommets ainsi que la donnée des couples de sommets entre lesquels il faut tracer des segments de droite
- la donnée de trois couples de plans deux à deux parallèles
- la description du graphe suivant :



avec l'indication que toutes les arêtes doivent être dessinées par des segments

de droites égaux, et que les arêtes (1,2),(3,4),(5,6),(7,8) d'une part et les arêtes (1,4),(2,3),(5,8),(6,7) d'autre part sont parallèles.

(On peut d'ailleurs à ce niveau là indiquer toutes sortes d'autres séries de contraintes : angles droits, parallélisme de faces etc...)

- description du parcours d'un chemin avec indication des longueurs et des changements de direction.
- etc....

Tout ceci pour montrer qu'il existe de multiples façons de décrire un cube, et suivant les informations données, on dispose, ou on ne dispose pas de renseignements tels que la grandeur, l'orientation, la position dans l'espace, voir la forme même du cube.

Pour le traitement du dessin proprement dit, les facilités dont on dispose sont également très variées :

- dessin des segments de droite indiquant seulement les coordonnées des extrémités, ou au contraire obligation de les dessiner points par points
- existence d'un sous-programme de dessin de parallélogramme ou même de cube
- programmes de calcul de ligne ou de surfaces cachées,
- possibilités d'interaction avec le terminal graphique, permettant de modifier le dessin du cube, de l'agrandir, de le changer d'orientation etc...

Compte-tenu de toutes ces contraintes, ou possibilités, il reste encore de nombreuses façons d'envisager le dessin du cube sur le terminal graphique avec ou sans vue perspective, avec ou sans tracé des lignes cachées, avec ou sans ombrage des faces

Pour pouvoir dessiner un graphe, on retrouve les mêmes problèmes :

- de quelles informations dispose-t-on, à la fois sur le graphe,

et sur la façon de le dessiner ?

- de quelles techniques de programmation va-t-on pouvoir disposer ?

Nous verrons d'abord le cas où les seules informations se réduisent à la donnée du graphe.

1.3.2. DESSIN D'UN GRAPHE SANS RENSEIGNEMENTS

L'information minimum dont on doit pouvoir disposer pour dessiner un graphe est la donnée du graphe, c'est-à-dire de l'ensemble des sommets et des couples de sommets entre lesquels il y a une arête.

D'un point de vue technique de programmation, on doit se limiter à dessiner les segments de droite. Rares sont les terminaux qui permettent de dessiner des courbes, on doit le faire point par point ce qui entraîne des temps de calcul et une occupation mémoire importants.

Compte-tenu de ceci la première et triviale façon de dessiner le graphe, est de disposer aléatoirement les sommets et de tracer un segment de droite entre les couples de sommets qui forment les arêtes du graphe.

Le résultat n'aura évidemment rien d'harmonieux et ne correspondra pas en général à ce qu'on attendait.

On améliore la présentation en définissant de la façon moins simpliste la disposition des sommets les uns par rapport aux autres. C'est ainsi qu'on impose une distance minimum entre les sommets, qu'on interdit l'alignement de trois sommets ou plus.... Mais le résultat reste encore décevant.

Certains algorithmes permettent de rendre beaucoup plus agréable le dessin.

On peut donner comme exemple l'algorithme de dispersion barycentrique (SAILLIARD [1]). Chaque sommet affecté d'un poids (son degré par exemple) est disposé de façon à être situé le plus possible du barycentre des autres sommet

Mais la seule façon d'obtenir un dessin du graphe conforme à des désirs

(alors que justement aucune information ne précisait quels étaient ces désirs), est de disposer d'un terminal interactif, afin de pouvoir modifier à sa guise, la disposition des sommets du graphe.

En général on utilise aussi l'interaction pour se donner la possibilité d'ajouter ou de supprimer des arêtes ou des sommets du graphe. Dans ce cas là l'aspect du dessin est moins important, car alors on dispose d'un outil puissant pour définir de nouveaux graphes ou en modifier d'anciens.

1.3.3. RENSEIGNEMENTS SUR LE GRAPHE

Comme nous l'avons vu dans les paragraphes précédents on peut se servir d'un graphe pour représenter un dessin. Quand on dessine un graphe on veut donc pouvoir retrouver le dessin qu'il représente.

On se limitera ici à ce que nous avons appelé le "graphe des primitives" dans le cas où il n'y a que des relations de concentration entre les primitives. C'est le seul cas où l'on puisse encore parler de dessin de graphe, car le passage graphe-dessin est relativement simple, il suffit de remplacer le dessin d'une arête (en l'occurrence un segment de droite) par le dessin de la primitive qu'elle représente.

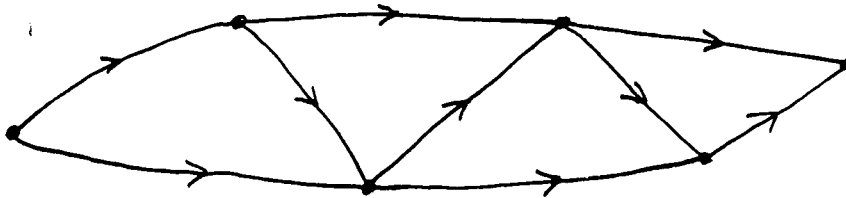
Le premier renseignement à ajouter est donc le nom de la primitive associée à chaque arête.

Mais cela n'est en général pas insuffisant.

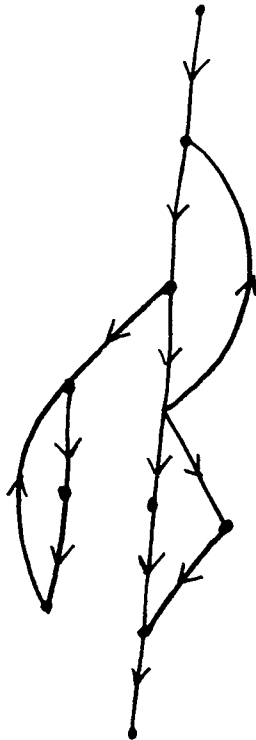
En effet on ne dessine pas de la même façon un réseau PERT, un organigramme ou un schéma de molécule chimique. Il faut tenir compte en plus des contraintes propres au type de graphe à dessiner.

Il est difficile d'isoler ces contraintes du contexte de l'élaboration du dessin. C'est pourquoi on est amené à concevoir autant d'algorithmes qu'il y a de type de graphe à dessiner.

Par exemple pour dessiner un réseau PERT on placera les sommets de gauche à droite de la source au puits.



Pour dessiner un organigramme, on les placera de haut en bas



Mais une simple rotation de 90° ne permettra pas d'utiliser le même programme de dessin d'un réseau PERT à un organigramme ou réciproquement.

CHAPÎTRE

II

DESCRIPTION FORMELLE D'UN GRAPHE

2.1. PRINCIPES UTILISES.

2.1.1. MOTIVATIONS

Les moyens de décrire un graphe et d'y associer une représentation interne pour le traitement machine, sont fort nombreux :

- matrice de connectivité ou d'incidence
- table des liaisons
- représentation avec des pointeurs : plexes
- représentation associative hashcodée
- modèles compacts comme le codage par un mot d'un langage algébrique [52].

Ces différents modèles présentent les mêmes inconvénients que la méthode de classification en reconnaissance de formes : le graphe forme un tout; il n'apparaît plus la notion de structure que l'on avait dégagée avec la décomposition en primitives.

D'autre part, ces graphes que l'on veut décrire représentent des dessins. Donc à certaines classes de dessins on devra associer certaines classes de graphes. Tout comme les dessins appartenant à une même classe sont constitués à partir du même ensemble de primitives, les descriptions des graphes correspondants, devront comporter des éléments communs. C'est pour toutes ces raisons que plusieurs auteurs ont proposé des systèmes génératifs pour décrire des graphes, ou plus généralement, des familles de graphes.

2.1.2. GRAPHE ET GRAMMAIRES

Soit \mathcal{G} une grammaire (pour le moment on ne précise ni la nature, ni la

forme de la grammaire).

La grammaire \mathcal{G} produit un langage $L(\mathcal{G})$.

Si à tout x , $x \in L(\mathcal{G})$ on peut associer, à un isomorphisme près un graphe G_x appartenant à une famille de graphes $\mathcal{F}_{\mathcal{G}}$.

et si, réciproquement, à tout graphe G , $G \in \mathcal{F}_{\mathcal{G}}$ il existe un y , $y \in L(\mathcal{G})$ unique représentant G , on dira que la famille de graphes $\mathcal{F}_{\mathcal{G}}$ est décrite par la grammaire \mathcal{G} .

Les propriétés que doivent avoir de telles grammaires sont les suivantes :

- 1°/ Permettre de décrire, avec des moyens finis, un ensemble infini de graphes.
- 2°/ Donner, par le processus même de génération, une structure du graphe décrit, correspondant à la structure du dessin représenté.
- 3°/ Donner des moyens efficaces pour la reconnaissance, l'analyse, l'interprétation et la génération des graphes et des dessins qui y sont associés.
- 4°/ Permettre un codage très économique du graphe, soit au moyen de la description elle-même, soit avec la liste des règles permettant la génération de la description.

C'est en fonction de ces propriétés, qu'ont été proposés les différents modèles de grammaires que nous allons étudier dans les paragraphes suivants.

2.1.3. LES DIFFERENTS MODELES DE GRAMMAIRES

On peut classer les modèles de grammaires décrivant une famille de graphes en trois catégories, qui, comme nous le verrons, sont incluses les unes dans les autres :

- 1°/ les grammaires de chaînes
- 2°/ les grammaires de structure
- 3°/ les grammaires de graphes.

Les grammaires de chaînes produisent des expressions linéaires (chaînes) décrivant les graphes au moyen d'opérateurs unaires et binaires.

Tel est la notation P.D.L. (Picture Description Language) de A.C. SHAW [41,42].

Les opérateurs utilisés ont été étudiés algébriquement, indépendamment de la notion de graphe (R. MOHR [45]).

Les grammaires de structure ont la particularité d'indiquer, dans les règles de production, comment construire le graphe.

C'est ainsi que NARASIMHAN [30-34] a donné une grammaire produisant les lettres de l'alphabet.

Ces grammaires ont été formalisées par FEELER [15,18] sous le nom de "Plex grammar".

Les grammaires de graphes produisent directement des graphes, l'application d'une règle de production consistant à remplacer un sous-graphe par un autre sous-graphe.

Ces grammaires ont été introduites par PFALTZ et ROSENFELD [38] qui leur ont donné le nom de "Web-grammars".

MONTANARI [29] a repris ce modèle et en a donné des applications intéressantes.

2.2. DESCRIPTION D'UN GRAPHE PAR UNE EXPRESSION PRODUITE PAR UNE GRAMMAIRE.

2.2.1. DEFINITIONS.

Soit G un graphe $G = (S, A)$

S l'ensemble des sommets

$A \subset S \times S$ ensemble des arêtes

Si $a \in A$ on a $a = (o_a, e_a)$ et on dira que o_a est l'origine de a et e_a l'extrémité de a .

A toute arête a du graphe, on associe valeur $v(a) \in V$. V est l'ensemble des valeurs associées aux arêtes du graphe.

Dans la description on identifie a par sa valeur $v(a)$.

Pour décrire le graphe on définit un certain nombre d'opérations appelées "concaténations".

Ces ne sont pas à proprement parler des lois de composition, mais ces opérations renseignent sur la façon dont deux ou plusieurs arêtes sont connectées entre elles.

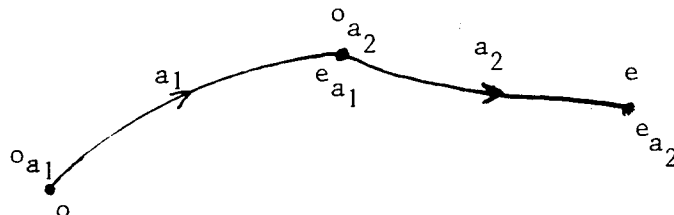
Pour pouvoir définir les concaténations comme des lois de composition, il faut donner à l'ensemble de toutes les arêtes un support sur lequel les arêtes seront équivalentes, à un déplacement près. MOHR [45].

La concaténation indique quelles sont les extrémités qui sont confondues et définit, pour l'ensemble des deux objets concaténés (arêtes ou ensemble d'arêtes déjà concaténés) une origine et une extrémité.

2.2.2. EXEMPLES DE CONCATENATIONS

Parmi toutes les définitions de concaténation possibles, nous donnerons en exemple, celles retenues par A.C. SHAW [41,42]

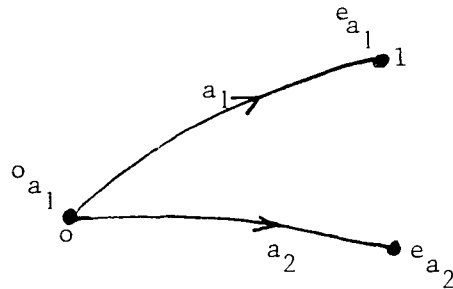
1° / $a_1 + a_2$
 e_{a_1} et o_{a_2} sont confondus, la nouvelle origine est o_{a_1} , la nouvelle extrémité e_{a_2} .



(On remarquera l'analogie avec la somme vectorielle).

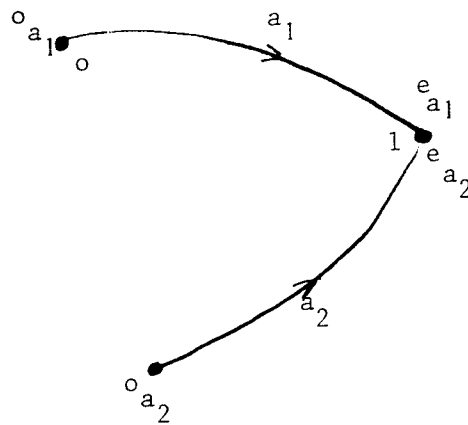
2° / $a_1 \times a_2$

0_{a_1} et 0_{a_2} sont confondus, la nouvelle origine est 0_{a_1} (ou 0_{a_2}) et la nouvelle extrémité est e_{a_1}



3° / $a_1 - a_2$

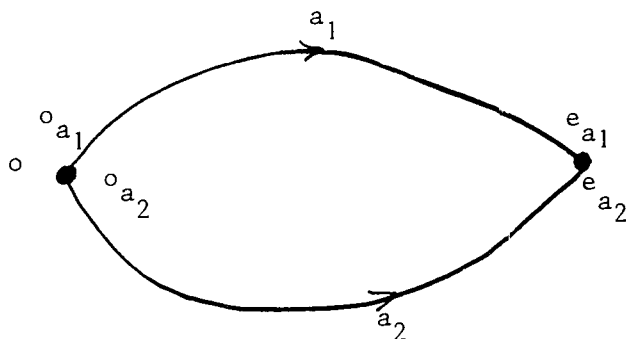
e_{a_1} et e_{a_2} sont confondus, la nouvelle origine est 0_{a_1} et la nouvelle extrémité est e_{a_1} (ou e_{a_2})



4° / $a_1 * a_2$

0_{a_1} et 0_{a_2} sont confondus d'une part, et d'autre part e_{a_1} et e_{a_2} le sont également.

La nouvelle origine est 0_{a_1} (ou 0_{a_2}), et la nouvelle extrémité est e_{a_1} (ou e_{a_2})



2.2.3. AUTRES OPERATEURS.

Pour faciliter la description du graphe, il peut être utile d'introduire d'autres opérateurs, notamment des opérateurs unaires, opérant sur une seule arête.

Comme exemple, on peut citer l'opérateur $/$, introduit par A.C. SHAW, opérateur qui permet d'échanger l'origine et l'extrémité.

Ces opérateurs peuvent également s'appliquer à des ensembles d'arêtes déjà concaténées; ensembles où l'on a défini une origine et une extrémité.

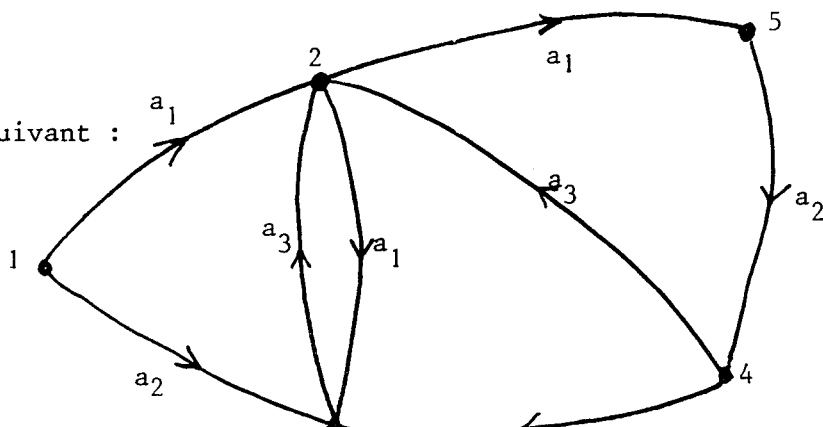
2.2.4. DESCRIPTION DU GRAPHE.

A l'aide des concaténations et des autres opérateurs, on décrit comment les différentes arêtes du graphe sont assemblées entre elles.

La représentation littérale des symboles de ces opérateurs et de leurs operands, les arêtes, donne une expression qui est la *description* du graphe.

EXEMPLE,

Soit le graphe suivant :



On peut le décrire par l'expression

$$(((((/a_2)+a_1)*a_3) * (/a_1)) * (/(((a_1+a_2) * (/a_3)) +a_2)))$$

Pour un graphe donné, la description n'est évidemment pas unique.

En revanche on démontre que tout graphe peut être décrit par une telle expression.

De plus, il n'est pas nécessaire d'utiliser tous les opérateurs précédemment définis.

C'est ainsi que, par exemple les trois opérateurs /, + et * sont suffisants pour pouvoir décrire n'importe quel graphe SHAW [41,42], MOHR [45]

2.2.5. GRAMMAIRES PRODUISANT LES DESCRIPTIONS DES GRAPHES.

On peut décrire un graphe par une expression qui est une chaîne linéaire de caractères.

A une classe de graphes donnée, on peut donc associer l'ensemble des expressions décrivant ces graphes, c'est-à-dire un ensemble de chaînes de caractères

S'il existe une grammaire \mathcal{G} produisant un langage \mathcal{L} tel que \mathcal{L} soit l'ensemble des chaînes de caractères associé à une classe de graphe \mathcal{C} , on dira que produit les descriptions des graphes appartenant à \mathcal{C} .

EXEMPLE 1-

La grammaire hors-contexte \mathcal{G} suivante, produit les descriptions de tous les graphes dont les arêtes sont étiquetées a_1, a_2 ou a_3 .

$$\mathcal{G} = (V_N, V_T, S, \mathcal{P})$$

$$\text{avec } V_N = \{S, 0_1, 0_2\}$$

$$V_T = \{(\,), /, +, *, a_1, a_2, a_3\}$$

$$\begin{aligned} \mathcal{P} : \quad & S \rightarrow (O_1 S) \\ & S \rightarrow (S O_2 S) \\ & S \rightarrow a_1 \\ & S \rightarrow a_2 \\ & S \rightarrow a_3 \\ & O_1 \rightarrow / \\ & O_2 \rightarrow + \\ & O_3 \rightarrow * \end{aligned}$$

Cette grammaire produit toutes les expressions complètement parenthésées que l'on peut construire avec les opérateurs binaires + et *, l'opérateur unaire / et les opérandes a_1 , a_2 et a_3 .

Elle produit donc les descriptions de tous les graphes dont les arêtes sont étiquetées par a_1 , a_2 et a_3

EXEMPLE 2.

On peut également produire les descriptions d'une classe de graphes beaucoup moins générale. La grammaire hors-contexte suivante produit les descriptions des graphes représentant les réseaux électriques série-parallèle n'ayant comme composants que des résistances (r) et des capacités (c)

$$\mathcal{G} = (V_N, V_T, S, \mathcal{P})$$

avec $V_N = \{S\}$

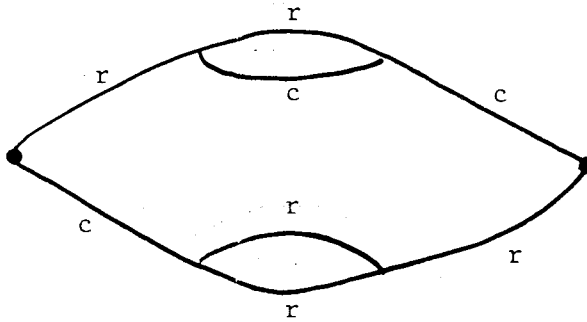
$$V_T = \{r, c, (,), *, +\}$$

$$\begin{aligned} \mathcal{P} : \quad & 1) S \rightarrow (S + S) \\ & 2) S \rightarrow (S * S) \\ & 3) S \rightarrow r \\ & 4) S \rightarrow c \end{aligned}$$

La grammaire produit l'expression suivante :

$$(r + ((r * c) + c)) * (c + ((r * r) + r))$$

qui décrit le schéma du réseau :



2.2.6. APPLICATIONS.

Les grammaires produisant les descriptions de graphes sont la plupart du temps de type hors-contexte.

On peut donc facilement réaliser des analyseurs syntaxiques reconnaissant et analysant les expressions produites par ces grammaires. Parmi les symboles que l'on retrouve dans ces expressions, il y a d'une part les opérandes représentant les primitives, et d'autre part les opérateurs qui indiquent comment les primitives sont assemblées entre elles.

Si par un procédé quelconque on peut d'une part reconnaître les primitives d'une forme et d'autre part décrire comment ces primitives sont assemblées entre elles, alors grâce à l'analyseur syntaxique on réalise un processus de reconnaissance de forme.

C'est ainsi que SCHAW analyse les photographies d'expériences de physique des particules dans une chambre à bulle [41] donnant par là une application concrète des grammaires produisant les descriptions des graphes.

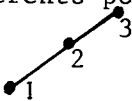
2.3. GRAMMAIRES DE STRUCTURE

2.3.1. PRINCIPE.

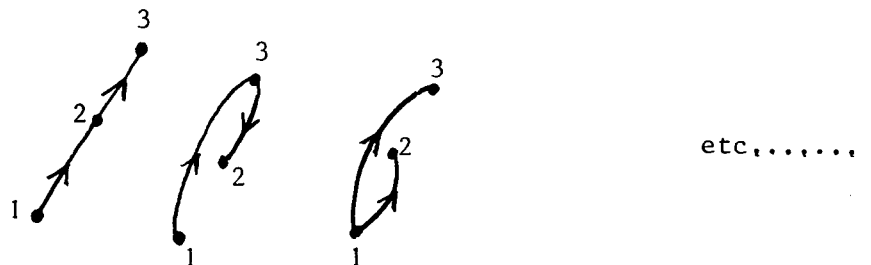
Comme pour la description des graphes avec des expressions, on utilise la notion de primitive

Mais, dans ce cas, les primitives ne sont pas directement assimilables aux arêtes du graphe. En effet, au lieu d'avoir pour chaque primitive deux points particuliers, l'un appelé origine, l'autre appelé extrémité, ce qui rend l'analogie immédiate, on pourra, dans chaque primitive, particulariser un nombre n de points, n n'étant pas nécessairement égal à deux.

C'est en ces points que sont assemblés les primitives. Pour pouvoir associer le dessin obtenu à un graphe, il faut, au niveau de chaque primitive, ajouter les arêtes reliant les différents points de connection.

Par exemple la primitive v  ayant trois points de connexion

pourra être représentée par l'un des graphes suivants



2.3.2. DESCRIPTION DU DESSIN

Dans le formalisme utilisé pour décrire un dessin à l'aide d'une grammaire de structure, les symboles terminaux représentent les primitives, et les symboles non terminaux représentent les parties du dessin de plus haut niveau

Les différents symboles de la partie droite d'une règle de production correspondent aux constituants du dessin défini par le ou les symboles de la partie gauche.

La description des concaténations des différentes parties du dessin entre elles, est donnée par une extension des règles de production : à la suite des symboles terminaux et non terminaux, en partie droite et en partie gauche, on adjoint une liste des points de connection, ainsi qu'une table de correspondance entre les points de connection de la partie droite et ceux de la partie gauche.

Dans le cas d'une règle hors contexte

$$A \rightarrow X$$

où A est un symbole non terminal et X une chaîne de symboles terminaux et de symboles non terminaux, la règle étendue sera :

$$A \begin{array}{c} \triangle \\ \triangle \end{array} A \longrightarrow X \begin{array}{c} \psi_X \\ \triangle \end{array} XA$$

Δ_A est la liste des points de connection de A

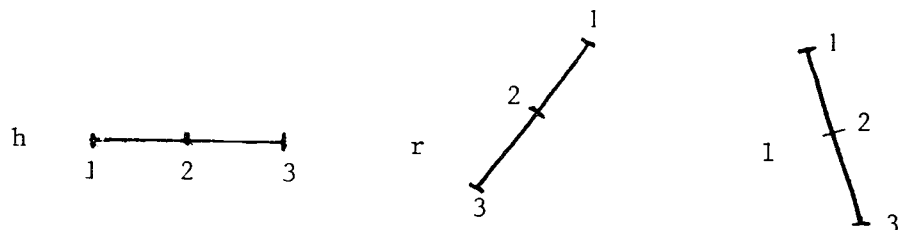
ψ_X est la liste des points de connection entre toutes les primitives ou partie du dessin représentés par les symboles de X.

Δ_{XA} est la table de correspondance entre les points de connection des éléments de X et ceux de A.

2.3.3. EXEMPLE DE DERIVATION

NARASIMHAN [30], dans le cadre d'un système de reconnaissance de caractères, donne une grammaire qui permet de décrire et de reconnaître les lettres de l'alphabet.

Par exemple pour définir la lettre A on utilise les trois primitives suivantes :



Les règles de production sont :

$$1) \quad A \longrightarrow \text{INV} . h$$

$$2) \quad \text{INV} \longrightarrow r . 1$$

L'extension de la deuxième règle est :

$$\text{INV} \begin{array}{c} \triangle \\ \text{INV} \end{array} \longrightarrow r . 1 \quad \psi_{r.1} \begin{array}{c} \triangle \\ r.1 \text{ INV} \end{array}$$

avec



$$\text{INV} = (1,2)$$

il reste dans INV deux points de connection étiquetés 1 et 2

$$\psi_{r.1} = (11)$$

r et 1 sont connectés de façon à ce que les les points de connections s de r et s de 1 soient confondus

$$\Delta_{r.1 \text{ INV}} =$$

$$1 \rightarrow 2(r)$$

le point de connection s de INV est le point de connection 2 de r

$$2 \rightarrow 2(1)$$

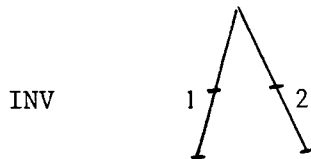
Le point de connection 2 de INV est le point de connection 2 de 1.

Pour simplifier, on peut écrire la règle de production sans qu'il n'y ait aucune ambiguïté :

$$\text{INV}(1,2) \rightarrow r.1 \quad (11;2;2)$$

où dans la parenthèse de partie droite on trouve d'abord $\psi_{r.1}$ puis pour chaque primitive de partie droite le point de connection correspondant au point de connection précisée dans la parenthèse de partie gauche.

INV est donc représenté



L'extension de la première règle est :

$$A \quad \Delta_A \rightarrow \text{INV}.h \quad \psi_{\text{INV}.h} \begin{array}{c} \triangle \\ \text{INV}.h \text{ A} \end{array}$$

Δ_A est vide, car A représentant le dessin final, ne contient plus de points de connection,

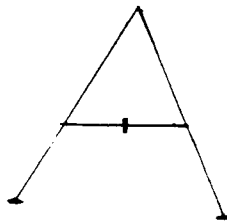
Il est donc de même pour $\triangleleft_{INV,hA}$

$\psi_{INV,h} = (11,23)$ c'est-à-dire les points de connection 1 de INV et 1 de h sont confondus, ainsi que les points de connection 2 de INV et 3 de h,

La règle de production peut donc s'écrire

$$A \longrightarrow INV,h(11,23)$$

Et le dessin produit est



2.3.4. COMPARAISON AVEC LES GRAMMAIRES DE CHAÎNE

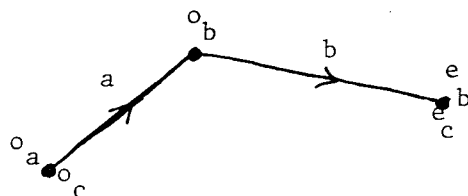
Tout ce que l'on peut décrire par une grammaire de chaîne, peut être décrit par une grammaire de structure.

En effet une primitive, dans une grammaire de chaîne, avec ses deux extrémités o et e, est une primitive, dans une grammaire de structure, ayant deux points de connections o et e.

Quant aux lois de concaténation utilisées dans les grammaires de chaînes, elles peuvent être décrites dans les grammaires de structures au moyen des extensions des règles :

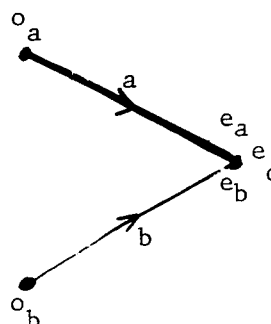
1) $c \rightarrow a+b$

devient



$$c(o_c e_c) \rightarrow a, b(e_a o_b; o_a; e_b)$$

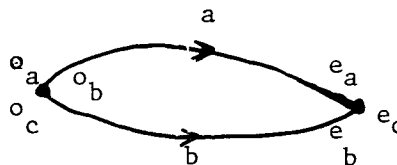
2) $c \rightarrow a - b$



devient

$$c(o_c e_c) \rightarrow a, b(e_a e_b; o_a; e_b)$$

3) $c \rightarrow a * b$



devient

$$c(o_c e_c) \rightarrow a - b(o_a o_b; e_a e_b; o_a; e_b)$$

Donc tous les dessins que l'on peut décrire par une grammaire de chaîne, peuvent être décrits par une grammaire de structure.

Réciproquement, pour tous les exemples de dessins décrits par une grammaire de structure que l'on connaît, on a trouvé des descriptions produites par une grammaire de chaîne. Mais on ne peut cependant pas généraliser et l'on pense malgré tout que le formalisme des grammaires de structure est plus puissant que celui des grammaires de chaîne

2.4. GRAMMAIRES DE GRAPHE

2.4.1. PRINCIPE

Les grammaires produisant des chaînes de caractères décrivant les graphes sont des grammaires génératives de type hors-contexte.

Les grammaires de structure n'apportent qu'une extension au niveau des règles de production.

Avec PAVLIDIS [35,36] d'une part, PFALTZ et ROSENFELD [38] d'autre part, apparaît un formalisme assez différent. En effet, ce que l'on produit, ce n'est pas une description du graphe, mais c'est le graphe lui-même.

L'axiome de départ, est un ensemble de graphes initiaux.

Tous les graphes ont leurs arêtes et leurs sommets étiquetés, les étiquettes étant des éléments des vocabulaires : vocabulaire terminal et vocabulaire non terminal.

Le graphe produit est un graphe dont toutes les arêtes et tous les sommets sont étiquetés par des éléments appartenant au vocabulaire terminal.

Si au cours d'une dérivation un graphe n'a pas tous ses sommets et toutes ses arêtes étiquetés par des symboles terminaux, on peut éventuellement appliquer une règle de production.

Dans une règle de production, il y a :

- d'une part deux graphes

le graphe de la partie gauche qui doit au moins avoir un sommet ou une arête étiqueté par un symbole non terminal

et le graphe de partie droite

- d'autre part un ensemble de règles d'imbrication et de conditions d'application de la règle.

Pour pouvoir appliquer une règle de production à un graphe G , il faut :

1°/ que la partie gauche de la règle soit un sous-graphe de G

2°/ que les conditions d'application de la règle soit remplies

L'application proprement dite de la règle consistant à remplacer le sous-graphe partie gauche par le sous-graphe partie droite, tout en respectant les règles d'imbrication.

Ce formalisme sera précisé et défini dans le chapitre suivant.

2.4.2. EXEMPLE DE GRAMMAIRE DE GRAPHE ET DE DERIVATION.

On donne, en exemple une grammaire produisant toutes les représentations des schémas des molécules chimiques possibles (indépendamment de toute existence vérifiée expérimentalement par des chimistes).

Les éléments du vocabulaire terminal sont d'une part les symboles atomique (étiquettes des sommets du graphe) et d'autre part 1,2 et 3 (étiquettes des arêtes signifiant liaison simple, double ou triple).

Les éléments du vocabulaire non terminal, ne sont que des étiquettes de sommets : M_0, M_1, M_2, M_3, M_4 .

Le graphe initial ou axiome est $\bullet M_0$.

Les règles de production sont données par les dix schémas de règle suivant

- 1) $M_i \bullet \longrightarrow M_{i+1} \text{---} 1 \text{---} M_1 \quad 0 \leq i \leq 3$
- 2) $M_i \bullet \longrightarrow M_{i+2} \text{---} 2 \text{---} M_2 \quad 0 \leq i \leq 2$
- 3) $M_i \bullet \longrightarrow M_{i+3} \text{---} 3 \text{---} M_3 \quad 0 \leq i \leq 1$

Pour ces trois schémas de règle il n'y a pas de condition d'application.

Les règles d'imbrication sont les suivantes :

- Le sommet de partie droite étiqueté $M_{i+1} (M_{i+2}, M_{i+3})$ est le même que l'unique sommet de partie gauche.
- Le sommet étiqueté $M_1 (M_2, M_3)$ est un nouveau sommet.

Les autres règles sont les suivantes :

- 4) $M_i \bullet \quad M_j \bullet \longrightarrow M_{i+1} \text{---} 1 \text{---} M_{j+i} \quad 1 \leq i \leq 3$
- 5) $M_i \bullet \quad M_j \bullet \longrightarrow M_{i+2} \text{---} 2 \text{---} M_{j+2} \quad 1 \leq i \leq 2, 1 \leq j \leq 2$
- 6) $M_1 \bullet \quad M_1 \bullet \longrightarrow M_4 \text{---} 3 \text{---} M_4$

Pour ces trois schémas de règle la condition d'application est la suivante : les sommets de la partie ne doivent pas être connexes dans le graphe à dériver.

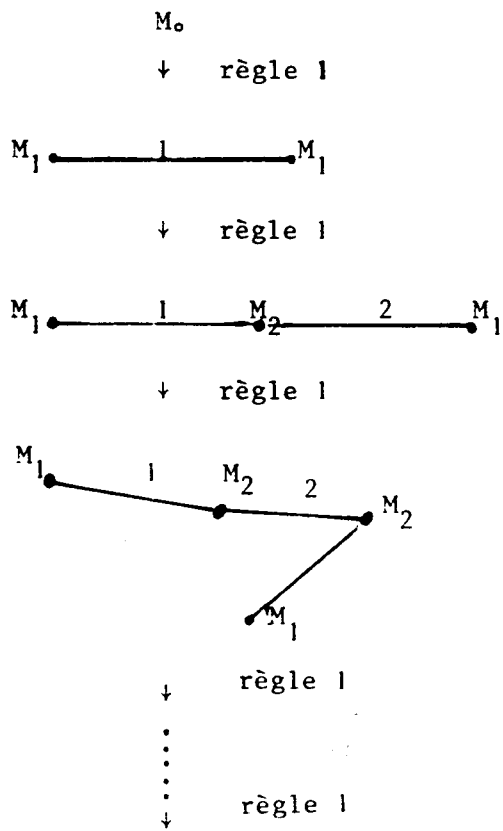
Les règles d'imbrication sont immédiates : on crée une nouvelle arête entre les deux sommets et leurs étiquettes deviennent celles de la partie de droite.

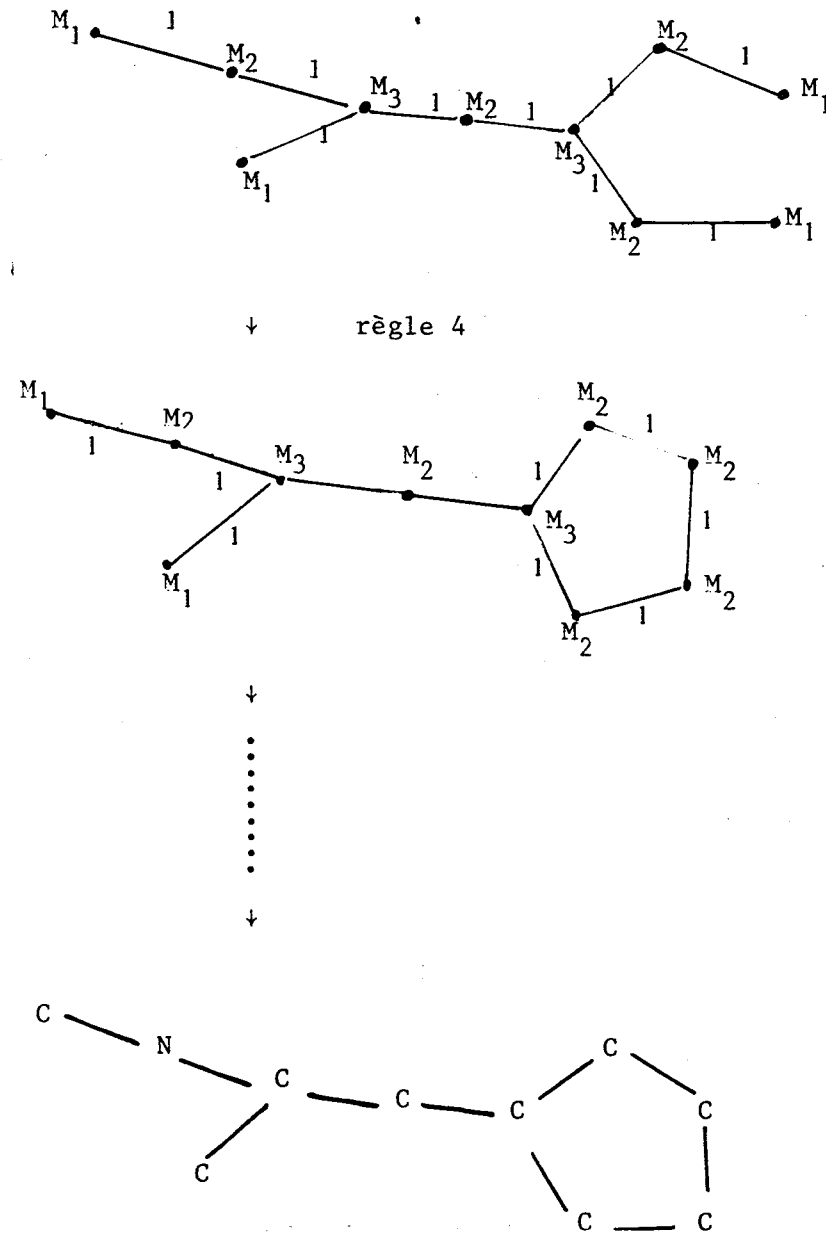
Enfin les dernières règles sont les suivantes :

- 7) $M_1 \longrightarrow$ H/Li/Na/F/Cé atomes de valence 1
- 8) $M_2 \longrightarrow$ Be/Mg/Ca/O/S..... atomes de valence 2
- 9) $M_3 \longrightarrow$ B/Al/N./p..... atomes de valence 3
- 10) $M_4 \longrightarrow$ C/Si/Ti..... atomes de valence 4

Pour ces quatre derniers ensembles de règles de production il n'y a pas de condition d'application, et les règles d'imbrication sont immédiates.

EXEMPLE DE DERIVATION





en ne dessinant pas les liaisons avec les atomes hydrogène.

Comme autre exemple de grammaire de graphe on peut donner celui de la grammaire décrivant des organigrammes dans l'article de KNUTH et FLOYD [48]

Le graphe initial est $\langle \text{programme} \rangle$ ou P.

Les étiquettes des sommets sont les suivantes :

symboles terminaux	\square	τ	DEBUT	FIN
symboles non terminaux	$\langle \text{programme} \rangle$ ou P			

<instruction> ou i

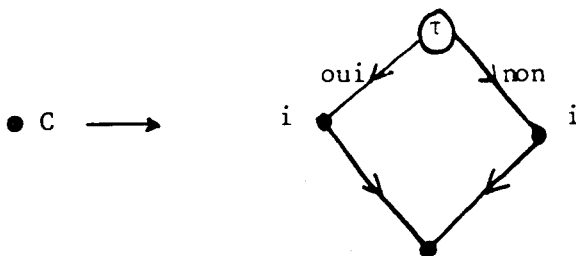
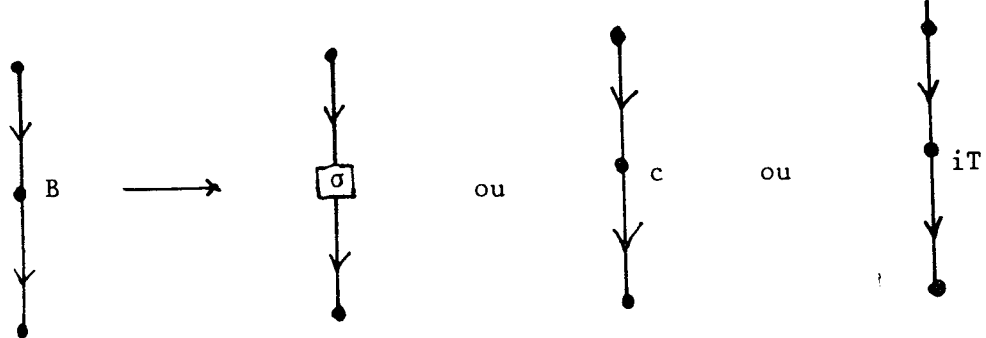
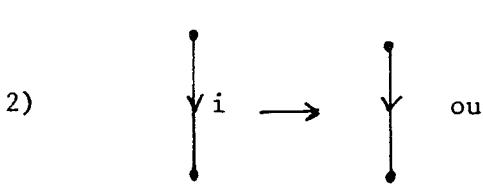
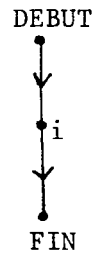
<instruction>de base ou B

<instruction conditionnelle> ou C

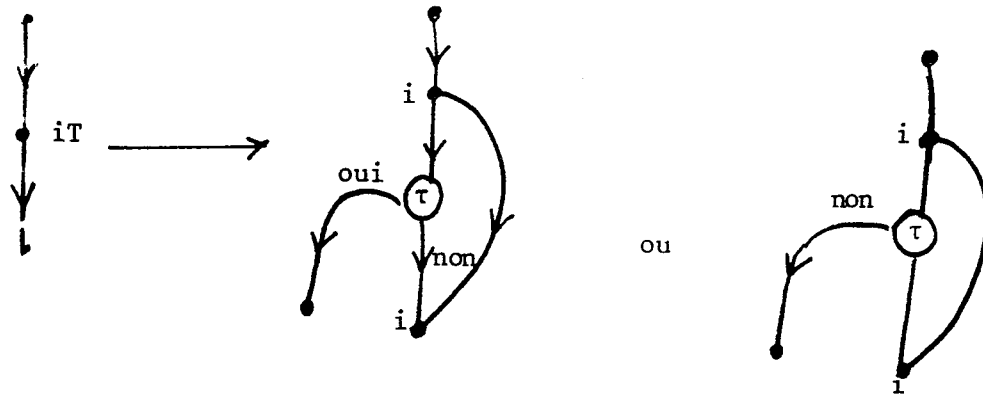
<instruction d'itération> ou iT

Les étiquettes des arêtes sont des symboles terminaux : OUI et NON.

Les règles de production sont :



6)



2.4.3. COMPARAISON AVEC LES AUTRES GRAMMAIRES.

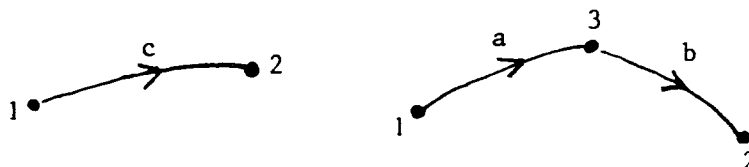
Avec le formalisme des grammaires de graphe on peut générer tous les graphes appartenant aux ensembles de graphes décrits par des grammaires de chaîne ou des grammaires de structure,

On remplace dans chaque règle de production la description du graphe ou de la partie du graphe, par le graphe lui-même.

Par exemple une règle d'une grammaire de chaîne telle que:

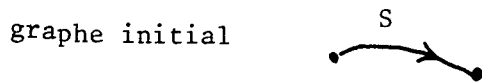


devient

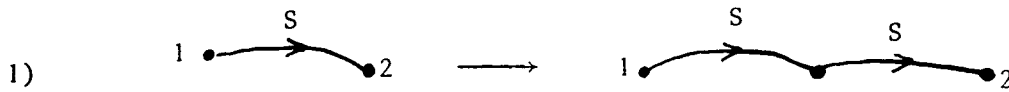


la règle d'imbrication étant que les sommets étiquetés 1 ou 2 en partie gauche et en partie droite se correspondent, le sommet étiqueté 3 étant un nouveau sommet.

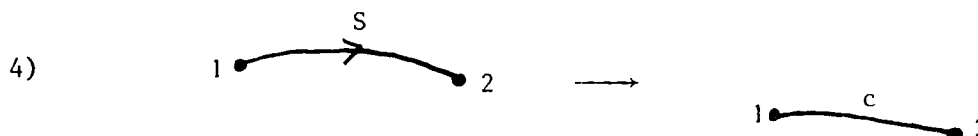
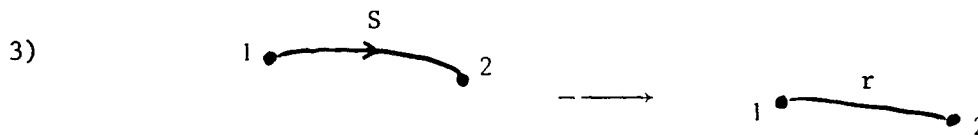
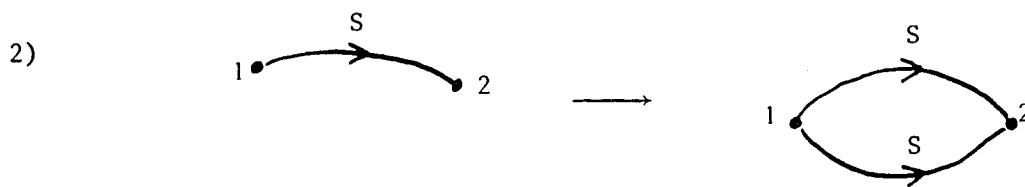
La grammaire produisant des descriptions de réseaux série-parallèle donnée en exemple au paragraphe 2.2.5. peut être transformée en une grammaire produisant directement des réseaux série-parallèle :



règles de production :



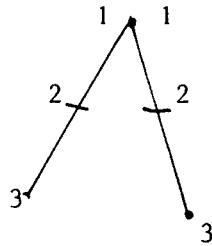
Les étiquettes des sommets indiquent les règles d'imbrication : la même étiquette en partie gauche et en partie droite indique qu'il s'agit du même sommet dans le graphe; un sommet non étiqueté en partie droite indique un nouveau sommet



Pour ces quatre règles, il n'y a aucune condition d'application.

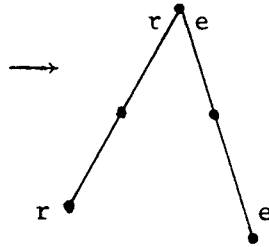
Dans le cas d'une règle d'une grammaire de structure

INV(1,2) r.1(11;2;2)



On aurait

INV₁ INV₂



et pour

A → INV, h(11, 2 3)

on aurait



A → INV₁ h h INV₂

A une primitive à n points de connections, correspond un graphe à n sommets. L'extension d'une règle de production d'une grammaire de structure précise quelles sont les règles d'imbrication de la règle de production de la grammaire de graphe correspondante.

Il n'y a donc aucune difficulté pour produire avec une grammaire de graphe, les graphes décrits par une grammaire de chaîne ou une grammaire de structure.

Réciproquement, il est plus difficile de passer d'une grammaire de graphe à une grammaire de chaîne ou une grammaire de structure.

En effet, dans les règles de production, il faut remplacer les graphes de partie droite et de partie gauche par leur description. Or pour un graphe donné, il n'existe pas une expression unique permettant de le décrire avec les opérateurs + * et /, et il en est de même en ce qui concerne les extensions des règles des grammaires de structure.

Pour chaque règle de production, il y aura plusieurs "traductions" possibles et l'ensemble de la grammaire va manquer de cohérence.

IL est également difficile d'exprimer au moyen du formalisme des grammaires de chaîne et des grammaires de structure, les conditions d'application d'une règle. De même pour respecter les règles d'imbrication, il peut être nécessaire d'introduire de nouvelles primitives qui n'apparaîtront plus dans la description finale du graphe.

Le formalisme des grammaires de graphes, est le formalisme le plus général qui permette de décrire et même de produire directement des graphes.

On peut avec les grammaires de graphe faire tout ce que l'on fait avec les autres formalismes.

On abordera dans le chapitre suivant, une étude plus approfondie de ces grammaires.

Mais, comme on le verra, ce que l'on gagne en généralité, on le perd en efficacité au niveau des applications notamment en ce qui concerne l'analyse syntaxique.

CHAPÎTRE

III



GRAPHES ET LANGAGES

3.1. REMARQUE PRELIMINAIRE.

Dans cette partie, on définit des graphes renseignés sur les sommets et sur les arêtes, et on généralise la notion de grammaire de chaîne à la notion de grammaire de graphes.

Ces grammaires correspondent aux "Web-grammars" introduites par PFALTZ et ROSENFELD [38]. On a cependant adopté des notations plus générales pouvant s'appliquer à tous types de graphes : orientés, non orientés, renseignés, etc....

L'exemple de la grammaire permettant de produire les graphes planaires sans isthmes a été étudié par MONTANARI [29].

C'est à partir de cette grammaire que dans le chapitre suivant on étudiera les problèmes de l'analyse syntaxique. On a donc repris les démonstrations concernant cette grammaire de façon à pouvoir introduire plus aisément les algorithmes d'analyse.

3.2. DEFINITIONS

3.2.1. GRAPHE RENSEIGNE

DEFINITION.

Soit V un ensemble fini appelé vocabulaire de renseignements, et soient V_S et V_A deux sous-ensembles non vides de V , non nécessairement disjoints, tels que $V_S \cup V_A = V$, appelés respectivement vocabulaire de renseignements sur les sommets et vocabulaire de renseignements sur les arêtes.

On dit que G est un graphe renseigné par V si G est un quadruplet ainsi

défini :

$$G = (S, A, f_S, f_A)$$

1°/ S est un ensemble (non nécessairement fini dans le cas général) appelé ensemble de sommets.

2°/ A est une partie de $S \times S$ appelée ensemble des arêtes

3°/ f_S est une application partout définie de S dans V_S .

4°/ f_A est une application partout définie de A dans V_A .

EXEMPLE et REPRESENTATION.

Soit $V = V_S \cup V_A$ avec $V_S = \{a, b, c, d\}$

et $V_A = \{c, d, e, f\}$

et soit

$$G = (S, A, f_S, f_A)$$

avec

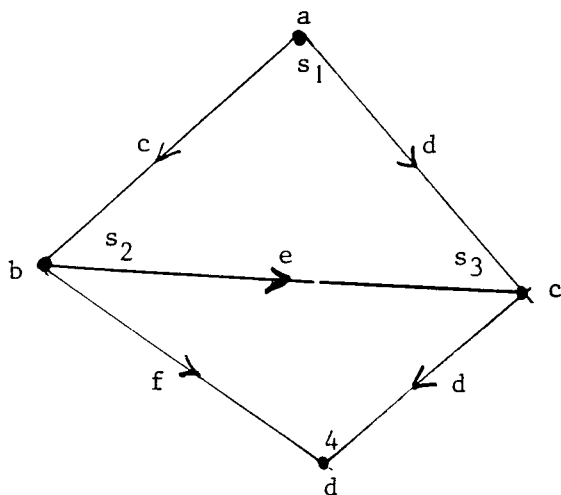
$$S = \{S_1, S_2, S_3, S_4\}$$

$$A = \{(S_1, S_2), (S_1, S_3), (S_2, S_3), (S_2, S_4), (S_3, S_4)\}$$

$$\begin{array}{l} f_S \text{ ainsi définie} \\ f_S(S_1) = a \\ f_S(S_2) = b \\ f_S(S_3) = c \\ f_S(S_4) = d \end{array}$$

$$\begin{array}{l} f_A \text{ ainsi définie} \\ f_A((S_1, S_2)) = c \\ f_A((S_1, S_3)) = d \\ f_A((S_2, S_3)) = e \\ f_A((S_2, S_4)) = f \\ f_A((S_3, S_4)) = d \end{array}$$

Le graphe G sera représenté de la façon suivante :



CAS PARTICULIERS DE GRAPHE S RENSEIGNES.

Si $\text{Card}(V_A)=1$ et $\text{Card}(V_S)>1$, le graphe est renseigné uniquement sur les sommets,

Si $\text{Card}(V_S)=1$ et $\text{Card}(V_A)>1$, le graphe est renseigné uniquement sur les arêtes.

Si $\text{Card}(V_S)=1$ et $\text{Card}(V_A)=1$ le graphe n'est pas renseigné.

La plupart des graphes utilisés dans les applications, peuvent être aisément définis comme étant des graphes renseignés :

- réseaux PERT
- représentation de schémas de circuits électriques
- représentation de schémas de molécules chimiques
- etc.....

SOUS-GRAPHE RENSEIGNE.

DEFINITION [46]

Soit G un graphe renseigné par $V \ G = (S, A, f_A, f_S)$ et G' un autre graphe renseigné par le même V $G' = (S', A', f'_S, f'_A)$. On dit que G' est un sous-graphe renseigné de G si

$$1^\circ / S' \subseteq S$$

$$2^\circ / A' = (S' \times S') \cap A$$

$$3^\circ / \text{pour tout } s \in S' \quad f'_{S'}(s) = f_S(s)$$

$$4^\circ / \text{pour tout } a \in A' \quad f'_{A'}(a) = f_A(a)$$

On dit que G' est un graphe partiel de G si

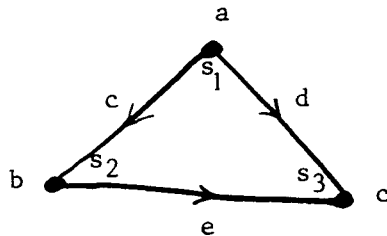
$$1^\circ / S' = S$$

$$2^\circ / A' \subset A$$

$$3^\circ / \& 4^\circ / \text{ comme ci-dessus}$$

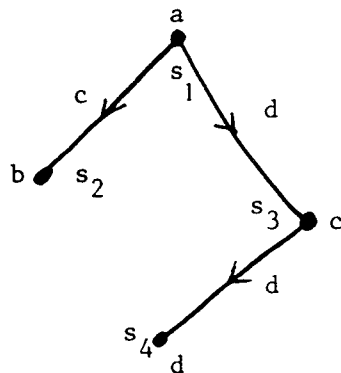
EXEMPLES

Si G est le graphe précédemment donné en exemple,
le graphe



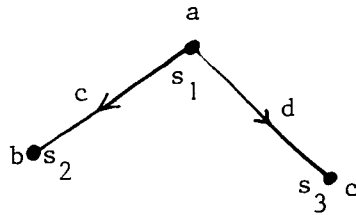
est un sous-graphe renseigné de G , $S' = \{s_1, s_2, s_3\}$

Le graphe



est un graphe partiel de G . $A' = \{(s_1, s_2), (s_1, s_3), (s_3, s_4)\}$.

Enfin le graphe



est un sous-graphe partiel de G , avec $S' = \{s_1, s_2, s_3\}$

et $A' = \{(s_1, s_2), (s_1, s_3)\}$

GRAPHE RENSEIGNE NON ORIENTE.

DEFINITION

Soit G un graphe renseigné par V $G = (S, A, f_s, f_A)$

On dit que G est non orienté si, quel que soit $s_i \in S$ et quel que soit $s_j \in P$ on a :

$$1^\circ / (s_i, s_j) \in A \iff (s_j, s_i) \in A$$

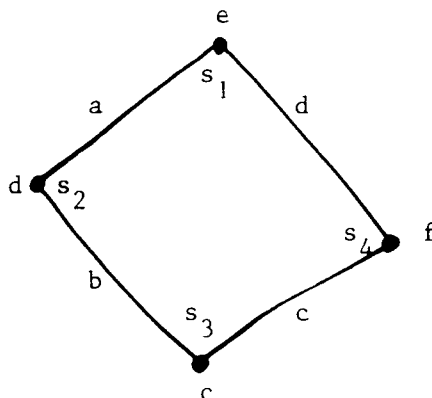
$$2^\circ / (s_i, s_j) \notin A \iff (s_j, s_i) \notin A$$

$$3^\circ / (s_i, s_j) \in A \implies f_A((s_i, s_j)) = f_A((s_j, s_i))$$

On représentera les deux arêtes (s_i, s_j) et (s_j, s_i) par une seule arête, en indiquant qu'une seule fois le renseignement commun.

Pour simplifier, on appellera par la suite graphe, un graphe renseigné non orienté, fini.

Exemple de graphe



3.2.2. GRAMMAIRE DE GRAPHE

DEFINITION

Une grammaire de graphe \mathcal{G} est un triplet

$$\mathcal{G} = (V, I, R) \quad \text{ou}$$

1°/ V est un ensemble fini, non vide, appelé vocabulaire de renseignements.

V est constitué des sous-ensembles $V_S, V_A, V_i, V_N, V_{SN}, V_{ST}, V_{AN}, V_{AT}$ ainsi définis :

$$V_S \cup V_A = V$$

V_S est un ensemble, non-vide, appelé vocabulaire de renseignements sur les sommet

$$V_S = V_{SN} \cup V_{ST} \quad \text{avec } V_{SN} \text{ non vide}$$
$$\text{et } V_{SN} \cap V_{ST} = \emptyset$$

V_A est un ensemble, non vide, appelé vocabulaire de renseignements sur les arête

$$V_A = V_{AN} \cup V_{AT} \quad \text{avec } V_{AN} \cap V_{AT} = \emptyset$$

$$V_T = V_{ST} \cup V_{AT} \quad \text{est le vocabulaire des renseignements terminaux,}$$

ou vocabulaire terminal

$$V_N = V_{SN} \cup V_{AN} \quad \text{est le vocabulaire des renseignements non termi-}$$

naux, ou vocabulaire non terminal.

(Par la suite on représentera par des lettres de l'alphabet minuscules les éléments du vocabulaire terminal et par des lettres majuscules les éléments du vocabulaire non terminal).

2°/ I est un graphe, appelé graphe initial. C'est un graphe à un seul sommet renseigné par "S", $S \in V_{SN}$ (V_{SN} n'est pas vide, par définition, il contient au moins S).

$$I : S$$

3°/ R est un ensemble fini de règles de production. Une règle de production r est un triplet

$$r = (\alpha, \beta, \gamma)$$

ou $\alpha = (S_\alpha, A_\alpha, f_{S_\alpha}, f_{A_\alpha})$ et $\beta = (S_\beta, A_\beta, f_{S_\beta}, f_{A_\beta})$

sont deux graphes renseignés par V ,

γ est une application (pas nécessairement partout définie) de S dans S .

Application d'une règle de production

Soit $r = (\alpha, \beta, \gamma)$ une règle de production et soit $G = (S, A, f_S, f_A)$ un graphe.

Pour pouvoir appliquer la règle r au graphe G , il faut :

1°/ que α soit un sous-graphe partiel de G .

2°/ que tous les sommets de α pour lesquels l'application γ n'est pas définie, aient le même degré que les sommets correspondants de G . En effet si ces sommets avaient dans G des degrés supérieurs, les arêtes dont ces sommets étaient l'une des extrémités, et qui n'appartenaient pas au sous-graphe α , n'auraient après application de la règle pas d'extrémités correspondantes.

L'application de la règle r au graphe G , consiste à transformer ce graphe G en un graphe G' ainsi défini :

$$G' = (S', A', f_{S'}, f_{A'}) \quad \text{avec}$$

1°/ S' est formé de tous les sommets de G sauf des sommets du sous-graphe α , auxquels on ajoute ceux de β . Les sommets de S , appartenant au sous-graphe α , pour lesquels l'application γ est définie, sont remplacés par les sommets correspondants (par l'application γ) de β .

2°/ A' est formé, d'une part des arêtes appartenant à A , mais n'appartenant pas à A_α , et d'autre part des arêtes appartenant à A_β .

Certaines arêtes, appartenant pourtant à la fois à A et à A' n'auront pas les mêmes extrémités dans les deux cas.

a) Si s_i et s_j appartiennent à S sans appartenir à S_α alors

$$(s_i, s_j) \in A \Rightarrow (s_i, s_j) \in A'$$

b) Si s_i et $s_j \in S$, $s_i \notin S_\alpha$ et $s_j \in S_\alpha$ alors $(s_i, s_j) \in A \Rightarrow (s_i, \gamma(s_j)) \in A'$

c) Si s_i et $s_j \in S$, $s_i, s_j \in S_\alpha$ et si $(s_i, s_j) \in A$ sans que $(s_i, s_j) \in A_\alpha$ alors

$$(\gamma(s_i), \gamma(s_j)) \in A'$$

d) dans ce dernier cas si $(s_i, s_j) \notin A_\alpha$ alors $(\gamma(s_i), \gamma(s_j)) \in A_\beta \Rightarrow (\gamma(s_i), \gamma(s_j)) \in A'$

e) si $s_i \in S$, $s_i \in S_\alpha$, $s_j \notin S$, $s_j \in S_\beta$ alors $((s_i), s_j) \in A_\beta \Rightarrow (\gamma(s_i), s_j) \in A'$

f) si $s_i, s_j \notin S_\beta$, $s_i, s_j \in S_\beta$ alors si $(s_i, s_j) \in A_\beta$ alors $(s_i, s_j) \in A'$.

3°/ Pour tout $s \in S'$

$$f_{S'}(s) = f_S(s) \quad \text{si } s \in S \text{ et } s \notin S_\alpha$$

$$f_{S'}(s) = f_{S_\beta}(s) \quad \text{si } s \in S_\beta$$

4°/ Pour tout $(s'_i, s'_j) \in A'$

$$f_{A'}((s'_i, s'_j)) = f_A((s_i, s_j)) \quad \text{si } (s_i, s_j) \in A, (s_i, s_j) \notin A_\alpha$$

$$f_{A'}((s'_i, s'_j)) = f_{A_\beta}((s_i, s_j)) \quad \text{si } (s_i, s_j) \in A_\beta$$

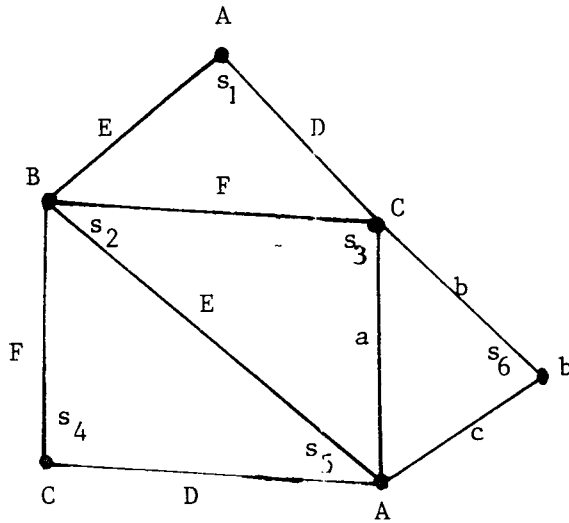
s_i et s_j étant les correspondants dans S , S_α ou S_β de s'_i et s'_j .

REMARQUE

Dans les cas b) et c) ci-dessus on voit que si s_i (ou s_j) $\in S$, s_i (ou s_j) $\in S_\alpha$ pour que l'on puisse appliquer la règle, il faut que l'application γ soit définie en s_i (ou s_j) dans le cas où s_i (ou s_j) est l'extrémité d'une arête n'appartenant pas à A_α . Si l'application γ n'est pas définie en de tels sommets, il ne faut pas qu'il existe de telles arêtes. C'est pourquoi tous les sommets de G' pour lesquels l'application γ n'est pas définie doivent avoir le même degré que les sommets correspondants de G .

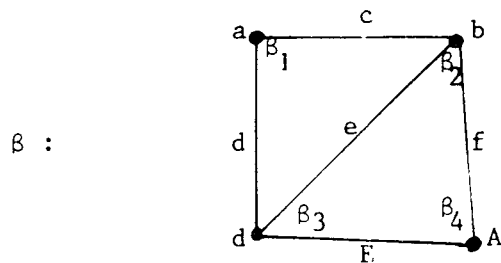
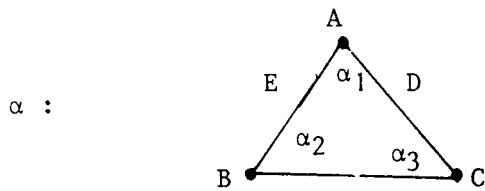
EXEMPLE D'APPLICATION.

Soit G le graphe suivant :



et soit $r = (\alpha, \beta, \gamma)$

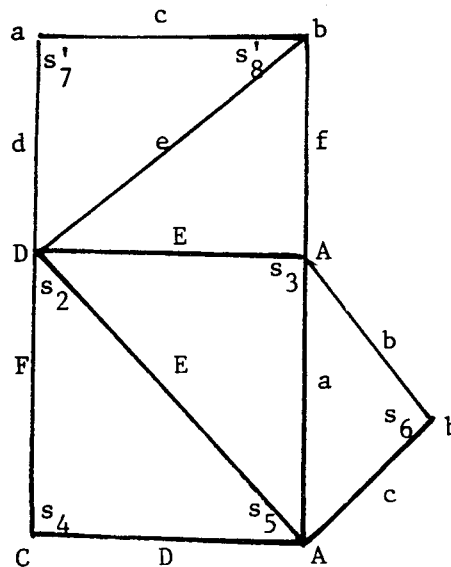
avec



γ ainsi définit : $\gamma(\alpha_2) = \beta_3$; $\gamma(\alpha_3) = \beta_4$

α est un sous-graphe partiel de G ; une première fois si on considère les sommets s_1, s_2, s_3 et les arêtes $(s_1, s_2), (s_1, s_3), (s_2, s_3)$, et une seconde fois si on considère les sommets s_5, s_2, s_4 et les arêtes $(s_5, s_2), (s_5, s_4), (s_2, s_4)$.

L'application γ n'est pas définie pour le seul sommet α_1 de α . Il faut donc le sommet correspondant dans le graphe G ait le même degré, c'est-à-dire 2. Ce n'est le cas que pour le premier des deux sous-graphes, et c'est donc là que l'on va appliquer le règle, et l'on obtient le graphe G' ;



NOTATIONS

On notera $G \xrightarrow[r]{} G'$ l'application de la règle r à G .

On dira que F dérive de E , et on notera $E \xrightarrow[g^*]{} F$ si il existe une séquence de graphes $G_1, G_2, \dots, G_n, G_{n+1}$ et une suite de règles de \mathcal{G} r_1, r_2, \dots, r_n tels que $E = G_1$ $F = G_{n+1}$ et pour tout $i = 1, \dots, n$

$$G_i \xrightarrow[r_i]{} G_{i+1}$$

La suite des règles (r_1, \dots, r_n) s'appelle une dérivation

3.2.3. LANGAGE PRODUIT PAR UNE GRAMMAIRE.

DEFINITION.

Soit \mathcal{G} une grammaire de graphe

$$\mathcal{G} = (V, I, R)$$

Le langage produit par la grammaire \mathcal{G} est un ensemble de graphes $L_{\mathcal{G}}$

$$L_{\mathcal{G}} = \{ G = (S, A, f_S, f_A) \mid \begin{array}{l} (I \xrightarrow[\mathcal{G}]{*} G) \\ \wedge (\text{pour tout } s \in S, f_S(s) \in V_{ST}) \\ \wedge (\text{pour tout } a \in A, f_A(a) \in V_{AT}) \end{array} \}$$

GRAMMAIRES de GRAPHES EQUIVALENTES

DEFINITION

On dit que deux grammaires de graphes sont équivalentes si elles produisent le même langage.

GRAMMAIRE de GRAPHE SOUS-CONTEXTE.

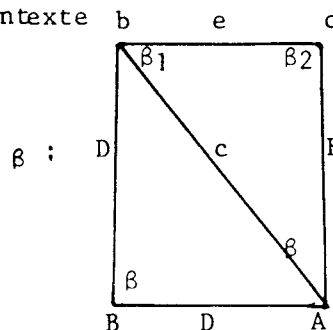
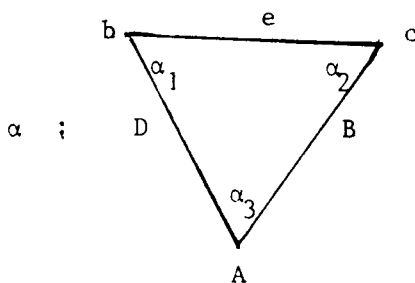
DEFINITION.

Soit $\mathcal{G} = (V, I, R)$ une grammaire de graphe.

On dit que \mathcal{G} est une grammaire de graphe sous-contexte, si pour toute règle $r \in R$, $r = (\alpha, \beta, \gamma)$ il existe un sommet $p \in S_{\alpha}$ tel que le sous-graphe obtenu en enlevant au graphe α le sommet p soit un sous-graphe de β , que γ restreinte à l'ensemble des sommets de ce sous-graphe, soit l'application identique (γ est donc définie pour tous ces sommets) et que $\text{card}(S_{\alpha}) \leq \text{card}(S_{\beta})$

EXEMPLE

La règle suivante est une règle sous-contexte



$$\begin{aligned} \gamma(\alpha_1) &= \beta \\ \gamma : \gamma(\alpha_2) &= \beta \\ \gamma(\alpha_3) &= \beta \end{aligned}$$

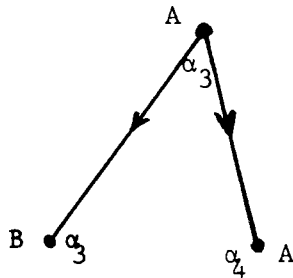
PROPRIETES DES GRAMMAIRES DE GRAPHE SOUS-CONTEXTE

1°/ L'ensemble des sommets créés au cours d'une dérivation peut être représenté par une arborescence.

En effet, chaque fois que l'on applique une règle r , il n'y a qu'un seul sommet qui est "modifié".

On donnera comme descendants de ce sommet, tous les nouveaux sommets apparaissant dans S_β et qui n'appartenaient pas à S_α .

Par exemple en appliquant la règle ci-dessus, on aurait :



2°/ Le problème est de savoir si un graphe donné appartient ou non au langage produit par une grammaire donnée est calculable.

En effet au cours de la dérivation, ni le nombre de sommets, ni le nombre d'arêtes du graphe que l'on construit ne décroît.

Les vocabulaires de renseignements sur les sommets et sur les arêtes étant finis, le nombre de règles de production de la grammaire étant également fini il n'y a qu'un nombre fini de dérivations permettant de produire des graphes ayant le même nombre d'arêtes et le même nombre de sommets que le graphe considéré.

REMARQUE.

Dans le cas des grammaires de chaîne, on a la propriété suivante: Pour toute grammaire où toutes les règles de production ont une partie droite de longueur plus grande que celle de la partie gauche, il existe une grammaire sous-cont

te équivalente [47].

Cette propriété, qui est en quelque sorte, la réciproque de la propriété 2, n'est pas vraie pour les grammaires de graphe.

Cela est dû au fait que, lorsque une même règle r peut s'appliquer en plusieurs endroits du graphe G (il existe plusieurs sous-graphes partiels de G isomorphes à α) on ne peut pas, comme on peut le faire avec les grammaires de chaînes, appliquer indifféremment en l'un quelconque de ces endroits la règle, pour finalement produire le même graphe.

Dans le cas des grammaires de chaînes on dit que la grammaire est ambiguë et pour différencier les dérivations on introduit des notions telles que dérivation la plus à gauche.

Pour les grammaires de graphes on ne peut pas parler de dérivation la plus à gauche car il n'existe, en général, aucune relation d'ordre entre les sommets ou les arêtes du graphe.

Mais surtout, le fait d'appliquer une règle à un sous-graphe plutôt qu'à un autre peut avoir des conséquences pour la suite. On peut par exemple en appliquant la règle au premier sous-graphe modifier le second et de ce fait on ne pourra plus appliquer la règle au second sous-graphe.

Donc on ne peut pas dire qu'en remplaçant une règle par une séquence de règles sous-contexte (par exemple mais en fait c'est général) dans une grammaire on obtient une grammaire équivalente.

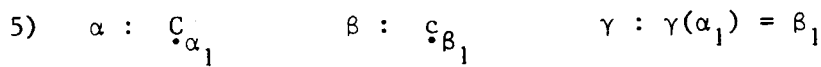
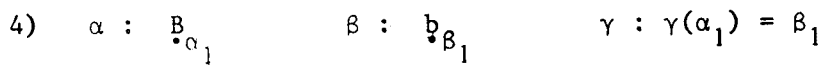
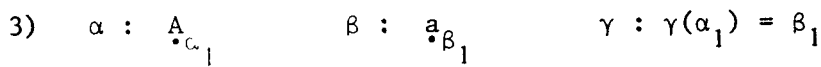
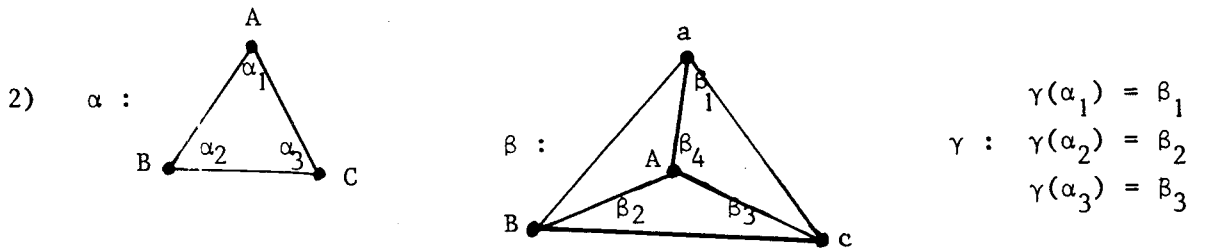
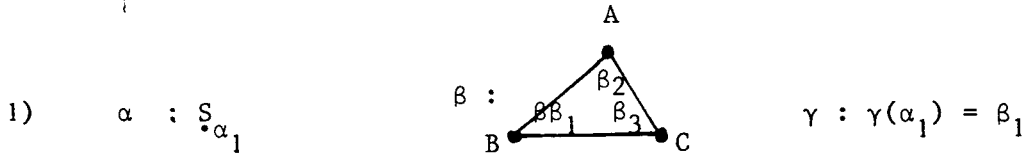
EXEMPLE DE GRAMMAIRE DE GRAPHE SOUS-CONTEXTE.

$$\begin{aligned}
 g &= (V, I, R) \\
 \text{avec } V &= V_S \cup V_A \\
 V_S &= \{S, A, B, C, a, b, c\} \\
 V_{SN} &= \{S, A, B, C\} \\
 V_{ST} &= \{a, b, c\}
 \end{aligned}$$

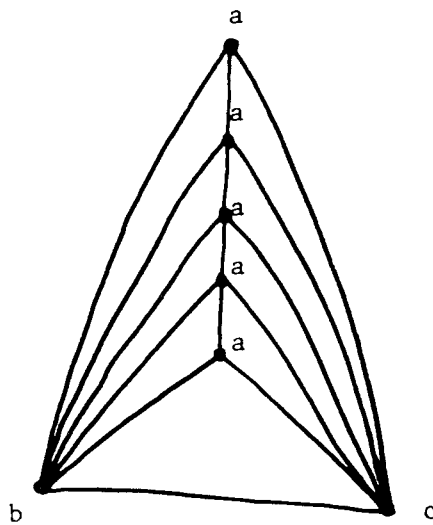
$$V_A = V_{AT} = \{e\} \quad V_{AN} = \emptyset$$

I : S

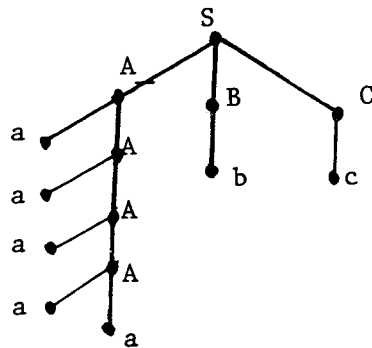
R ensemble des règles de production



Exemple de graphe g n r 



On obtient ainsi une classe de graphes maximaux planaires à n sommets ($n \geq 3$)
 si $n > 4$ deux sommets ont un degré égal à trois
 deux sommets ont un degré égal à $n-1$
 les $n-1$ autres sommets ont un degré égal à quatre
 L'arborescence représentant l'ordre de génération des sommets est la suivante:



(on ne représente que les renseignements des sommets).

GRAMMAIRE DE GRAPHE HORS-CONTEXTE

DEFINITION

Soit $\mathcal{G} = (V, I, R)$ une grammaire de graphe.

On dit que \mathcal{G} est une grammaire de graphe hors-contexte, si pour toute règle r de R $r = (\alpha, \beta, \gamma)$, α est un graphe à un seul sommet p et tel que

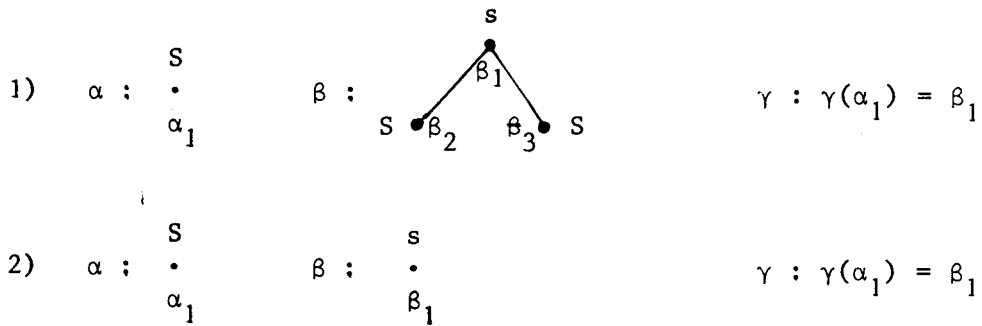
$$f_{S, \alpha}(p) \in V_{SN}$$

EXEMPLES DE GRAMMAIRE DE GRAPHE HORS-CONTEXTE

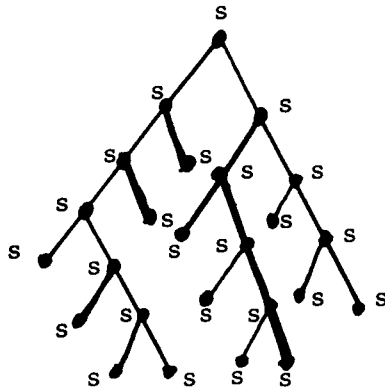
1) Grammaire permettant de produire les arbres binaires

$$\begin{aligned} \mathcal{G} &= (V, I, R) \\ V &= V_S \cup V_A \\ V_S &= \{S, s\} \quad V_{SN} = \{S\} \quad V_{ST} = \{s\} \\ V_A &= V_{AT} = \{e\} \quad V_{AN} = \emptyset \\ I &= S \end{aligned}$$

R ensemble des règles de production :



Exemple de graphe généré



2°/ Grammaire permettant de produire les arborescences représentant les dérivations d'une grammaire de chaîne hors-contexte

$$\mathcal{G} = (V, I, R)$$

Soit $G = (V_N, V_T, S, \mathcal{P})$ une grammaire de chaîne hors-contexte :

V_N : vocabulaire non terminal

V_T : vocabulaire terminal

$S \in V_N$

\mathcal{P} : ensemble des règles de productions du type

$$A_i \rightarrow a_i b_i$$

ou $C_j \rightarrow D_j E_j$

avec $A_i, B_i, C_j, D_j, E_j \in V_N$ et $a_i \in V_T$

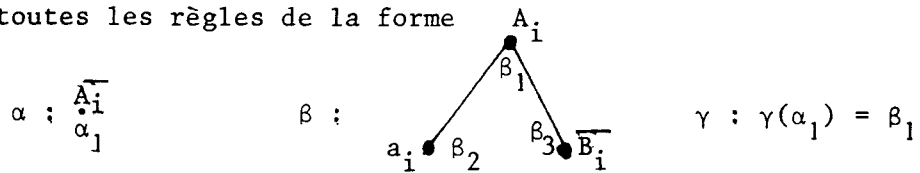
\mathcal{G} est ainsi définie :

$$V = V_S \cup V_A$$

$$\begin{aligned}
 V_S &= V_{SN} \cup V_{ST} \\
 V_{SN} &= \overline{V}_N = \{ \overline{X} / X \in V_N \} \\
 V_{ST} &= V_T \cup V_N \\
 V_A &= V_{AT} = \{e\} \quad V_{AN} = \emptyset \\
 I &: S
 \end{aligned}$$

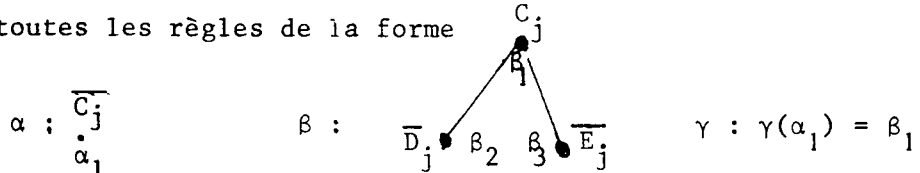
R ensemble des règles de productions comprend :

1°/ toutes les règles de la forme



si $A_i \rightarrow a_i B_i$ est une règle appartenant à \mathcal{P} .

2°/ toutes les règles de la forme



si $C_j \rightarrow D_j E_j$ est une règle appartenant à \mathcal{P} .

GRAPHE INDIRECTEMENT PRODUIT PAR UNE GRAMMAIRE

DEFINITION

Soit $G = (S, A, f_S, f_A)$ un graphe appartenant au langage $L_{\mathcal{G}}$ produit par une grammaire de graphe \mathcal{G} .

Soient V_{ST} et V_{AT} les vocabulaires terminaux de la grammaire. Pour tout $s \in S$, $f_S(s) \in V_{ST}$ et pour tout $a \in A$ $f_A(a) \in V_{AT}$.

Soient V'_{ST} et V'_{AT} deux sous-ensembles de V_{ST} et V_{AT}

$$V'_{ST} \subseteq V_{ST} \quad V'_{AT} \subseteq V_{AT}$$

On dit que le sous-graphe partiel de G , $G' = (S', A', f_S, f_A)$ si il existe est indirectement produit par la grammaire \mathcal{G} si :

1°/ pour tout $s \in S'$ $f_S(s) \in V'_{ST}$

et pour tout $s \in S$ et $s \notin S'$ alors $f_S(s) \in V'_{ST}$

2°/ pour tout $a \in A'$ $f_A(a) \in V'_{AT}$

et pour tout $a \in A$ et $a \notin A'$ alors $f_A(a) \in V'_{AT}$

REMARQUE.

Si la grammaire \mathcal{G} ne produit que des graphes renseignés sur les sommets (ce qui est le plus courant), le graphe G' indirectement produit par \mathcal{G} est le sous-graphe de G restreint aux sommets renseignés par des éléments de V'_{ST} et aux arêtes reliant de tels sommets.

La notion de graphe indirectement produit par une grammaire a été introduite pour pouvoir conserver les propriétés de calculabilité de certaines grammaires (tout comme pour les grammaires on introduit des marqueurs en bout de chaîne pour les grammaires de chaîne). On évite ainsi, d'avoir à supprimer certains sommets ou certaines arêtes "auxiliaires" utilisées au cours de la dérivation.

Nous verrons dans les paragraphes suivants les exemples de telles grammaires

3.3. EXEMPLES DE GRAMMAIRES DE GRAPHES

3.3.1, RAPPELS SUR LES GRAPHES PLANAIRES ET LES GRAPHES SEPARABLES.

On peut donner plusieurs définitions des graphes planaires, qui sont toutes équivalentes.

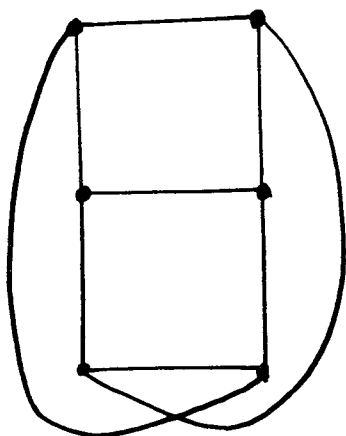
DEFINITION TOPOLOGIQUE.

Un graphe est planaire si on peut le dessiner sur le plan ou sur la sphère de façon à ce qu'on n'ait aucune intersection d'arête en dehors des sommets

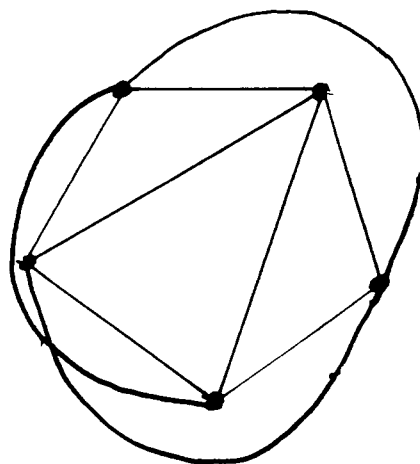
DEFINITION DE KURATOWSKY

Un graphe est planaire si et seulement si, il ne contient pas de sous-

graphe "partiel" (au sens large, c'est-à-dire que les arêtes sont, soit des arêtes du graphe, soit des chaînes) ayant l'une des formes suivantes :



$K_{3,3}$



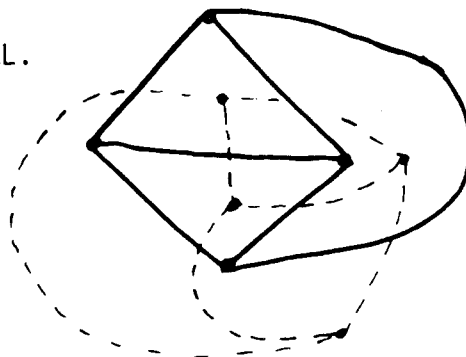
K_5

DEFINITION DE MAC LANE

Un graphe est planaire si et seulement si, on peut trouver une collection de cycles de base tels que chaque arête est prise au plus deux fois dans cette collection.

DEFINITION DE WHITNEY. Un graphe est planaire si et seulement si, il a un dual au sens de Whitney [57]. Ce dual est un graphe, qui n'a de sens que pour les graphes planaires, est un graphe obtenu à partir d'une représentation plane du graphe. A chaque face (surface du plan limitée par des arêtes du graphe) on associe un sommet, et il y a une arête entre deux sommets si les deux faces sont adjacentes.

EXEMPLE DE GRAPHE PLANAIRE ET DE SON DUAL.



PROPRIETE,

Soit n_s le nombre de sommets d'un graphe planaire, et n_a son nombre d'arêtes, n'_s et n'_a ceux du dual, on a :

$$n_a = n'_a,$$
$$n_s + n'_s = n_a + 2$$

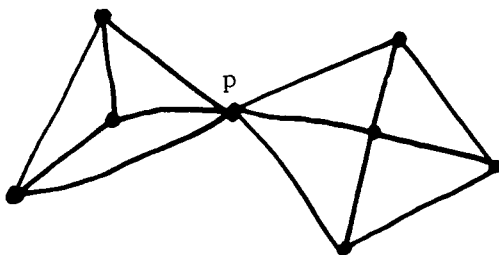
GRAPHE SEPARABLE,

DEFINITION,

Un graphe connexe G est séparable si il contient deux sous-graphes F_1 et F_2 ayant au moins une arête, et tels que :

- toute arête de G appartient soit à F_1 , soit à F_2
- F_1 et F_2 ont exactement un sommet en commun appelé sommet de coupure

EXEMPLE DE GRAPHE SEPARABLE



p est le sommet de coupure

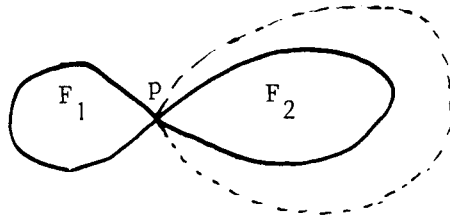
Si un graphe connexe n'a pas ces propriétés, il est dit "non séparable".

Si un graphe séparable est planaire, on a la propriété suivante :

Il existe un sommet P tel que sa séquence de faces contienne plus d'une fois la même face. (On appelle séquence de faces d'un sommet P d'un graphe planaire, la suite des faces que l'on rencontre en tournant autour de P dans le sens des aiguilles d'une montre, et en partant de l'une quelconque des faces).

La réciproque est également vraie, c'est-à-dire que cette propriété caractérise les graphes planaires séparables,

La démonstration est immédiate, en construisant une courbe fermée passant par P , entièrement contenue dans une face et séparant le graphe en deux sous-graphes,



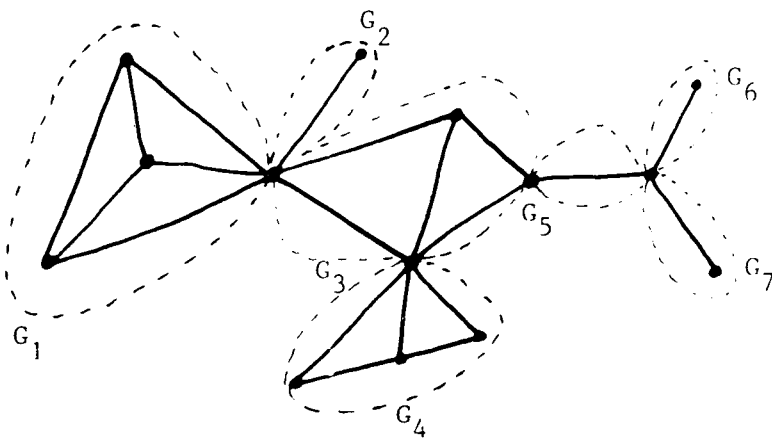
DECOMPOSITION DES GRAPHES SEPARABLES,

Si un graphe est séparable, on peut le décomposer en deux sous-graphes F_1 et F_2 qui n'ont, en commun, qu'un seul sommet.

Ces deux sous-graphes, ou l'un d'entre eux, peuvent à leur tour, être séparables et décomposés en deux, et ainsi de suite, jusqu'à ce que l'on n'ait plus que des graphes non séparables.

Un graphe séparable peut donc être décomposé en n sous-graphes non séparables,

EXEMPLE



La structure du graphe ainsi décomposée, est une structure d'arbre.

GRAPHE AVEC ISTHMES

DEFINITION

On dit qu'un graphe est un graphe avec isthmes, si dans sa décomposition en sous-graphes non séparables, un au moins de ces sous-graphes est réduit à une arête appelée isthme.

Si ce n'est pas le cas on dit que le graphe est sans isthmes.

PROPRIETES

- un isthme est une arête qui n'appartient à aucun cycle.
- dans le cas des graphes planaires, on démontre qu'un graphe planaire est un graphe avec isthmes, si et seulement si, il existe une arête (p,q) adjacente des deux cotés à la même face.

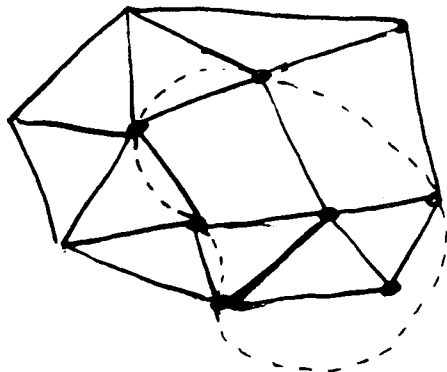
COURBE DE COUPURE D'UN GRAPHE PLANAIRE

DEFINITION

On appelle courbe de coupure d'un graphe planaire, si elle existe, une courbe plane fermée, ayant les propriétés suivantes :

- 1°/ La courbe ne coupe la représentation planaire du graphe qu'aux sommets,
- 2°/ Les différents segments de la courbe (parties de la courbe comprises entre deux sommets) appartiennent à des faces différentes.

EXEMPLE



Une courbe de coupure d'un graphe planaire, induit une partition de l'ensemble des sommets (resp. des faces) du graphe en trois classes :

- 1°/ La classe des sommets (resp. des faces) frontières : ceux qui sont sur la courbe
- 2°/ la classe des sommets (resp. des faces) intérieurs : ceux qui sont à l'intérieur de la courbe
- 3°/ la classe des sommets (resp. des faces) extérieurs : ceux qui sont à l'extérieur de la courbe

De même une courbe de coupure d'un graphe planaire induit une partition de l'ensemble des arêtes du graphe en deux classes :

- 1°/ la classe des arêtes intérieures : celles qui sont à l'intérieur de la courbe
- 2°/ la classe des arêtes extérieures : celles qui sont à l'extérieur de la courbe.

Nous démontrons maintenant une propriété des graphes planaires sans isthmes et des courbes de coupure. Cette propriété sera utilisée d'une part dans le paragraphe suivant dans l'exemple d'une grammaire produisant les graphes planaires sans isthmes, et d'autre part dans le chapitre suivant, dans le cas de la construction d'un analyseur de tels graphes.

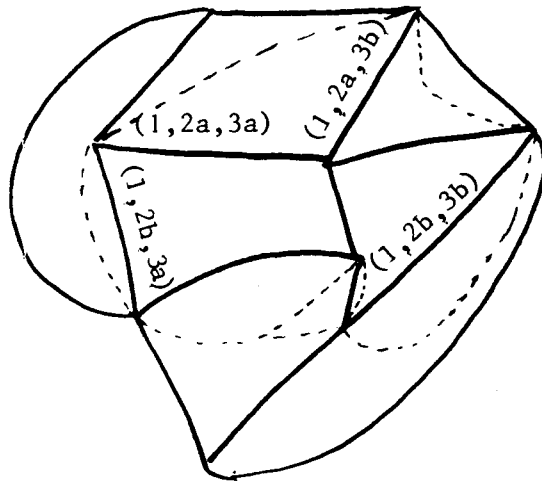
LEMME

Etant donné un graphe planaire G sans isthmes et sans boucles, et étant donné une courbe de coupure de ce graphe planaire, il est toujours possible de trouver une arête intérieure ayant les propriétés suivantes :

- 1°/ Une des extrémités de l'arête est un sommet frontière et une des faces adjacente à l'arête est une face frontière
- 2°/ L'autre extrémité est
 - a) soit un sommet intérieur
 - b) soit un sommet frontière consécutif du premier sommet sur la courbure de coupure

3°/ L'autre face est

- a) soit une face intérieure
- b) soit une face frontière adjacente à la lère face

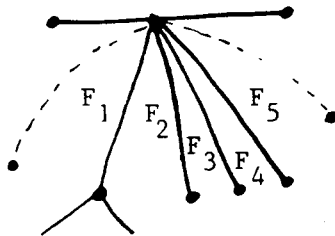


Exemple sur lequel on peut voir les 4 cas possib.

DEMONSTRATION

Pour tous les sommets frontières, il existe au moins une arête vérifiant la propriété 1°/ car sinon, deux segments consécutifs seraient contenus dans la même face.

Au moins une de ces arêtes (et même deux s'il y a plus d'un arc intérieur connecté à ce sommet) est adjacente à une face frontière

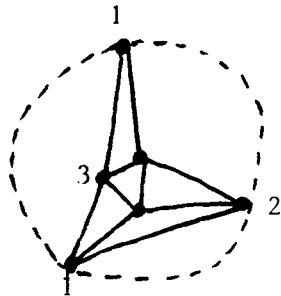


F_1 et F_5 sont des faces frontières

comme on peut le voir en examinant la séquence des faces associée à ce sommet

Supposons que le nombre de sommets frontière soit inférieur ou égal à trois.

Alors les conditions 2°/ et 3°/ sont satisfaites pour certaines de ces arêtes



Dans ce cas le premier sommet est adjacent aux autres sommets sur la courbe de coupure.

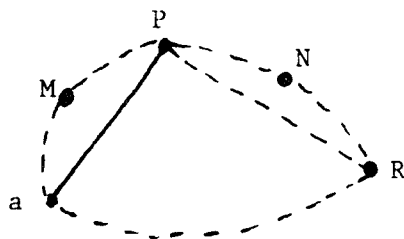
Donc l'autre extrémité de l'arête est, soit un sommet intérieur cas 2°/a) (sommet 3 sur l'exemple ci-dessus) soit un sommet frontière consécutif cas 2°/b) (sommet 2 sur l'exemple),

Il en est de même pour les faces,

Donc la face N est adjacente à P et à un sommet R qui n'est pas adjacent à P.

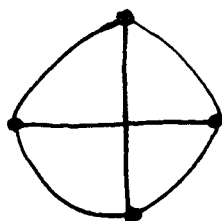
Ainsi on peut construire un nouveau graphe planaire en ajoutant à G l'arête (P,R).

Si les conditions 2 et 3 ne sont pas satisfaites on peut construire un nouveau graphe planaire G' dans lequel tout sommet frontière est relié par une arête intérieure à au moins un sommet frontière non adjacent (si c'est la condition 2) qui n'est pas vérifiée pour chaque sommet frontière, c'est déjà le cas).



Il ne reste plus qu'à montrer que G' n'est pas planaire ce qui amène la contradiction.

Dans le cas où il y a quatre sommets frontière c'est évident :



Si le nombre de sommets frontière est $n(n \geq 4)$, considérons l'arête (P, Q)

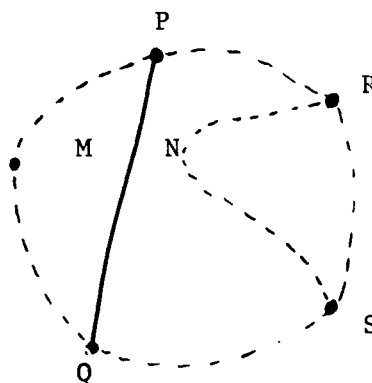
En enlevant les arêtes redondantes, c'est la seule arête intérieure connectée soit avec P soit avec Q (ou les deux), prenons P .

Si R et S sont les sommets frontières, adjacents à P, Q ne peut être adjacents aux deux à la fois (car sinon $n = 4$),

Supposons maintenant que le nombre de sommets frontière est au moins de 4.

Supposons également qu'aucune arête satisfaisant à la condition 1) ne satisfasse aux conditions 2 et 3.

Si une arête satisfait à la condition 1, mais ne satisfait pas à la condition 2, cela signifie que cette arête relie deux sommets frontières non consécutifs sur la courbe de coupure



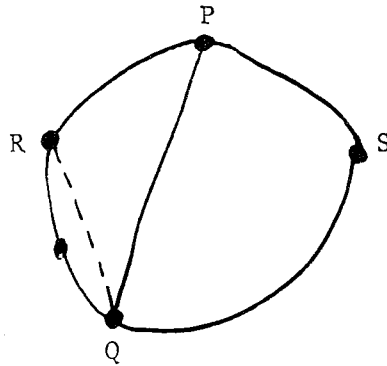
Soit (P, Q) une arête satisfaisant à la condition 1 (P est un sommet frontière, M est une face frontière)

Si (P, Q) ne satisfait pas à la condition 3 cela signifie que l'autre face n'est pas adjacente à la face N sur la courbe de coupure.

Soient R et S les extrémités du segment de la courbe de coupure traversant

la face R et S sont des sommets frontières.

On a $R \neq P$ et $S \neq P$ et, soit R, soit S, (ou les deux), prenons n'est pas adjacent à P, car si P,R,S étaient deux par deux adjacents, ils seraient les seuls sommets frontières.



Soit Q non adjacent à R. On peut alors enlever l'arête(P,Q) et la remplacer par l'arête (R,Q). On a ainsi le même problème pour n-1 sommets frontières. On recommence le processus jusqu'à ce qu'on n'ait plus que 4 sommets frontières, et l'on a ainsi établi la contradiction.

3.3.2, GRAMMAIRE DE GRAPHE PRODUISANT LES GRAPHS PLANAIRES SANS ISTHMES.

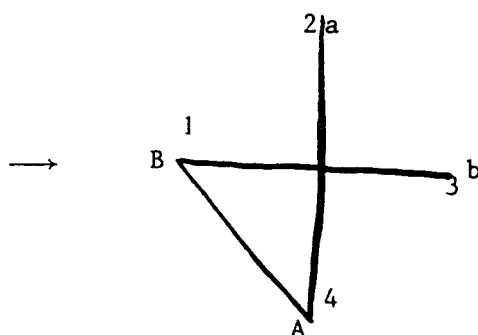
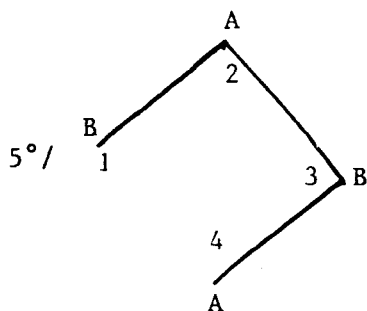
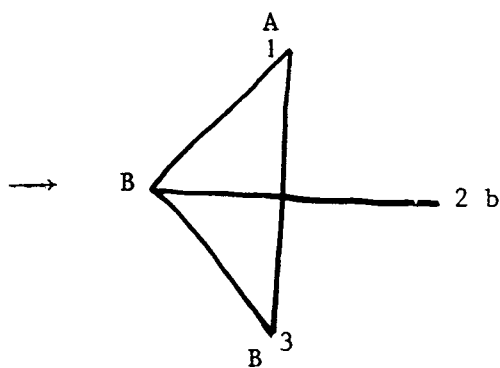
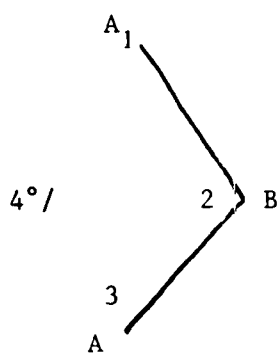
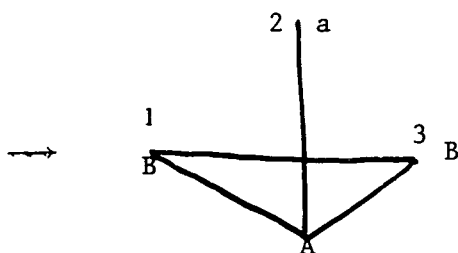
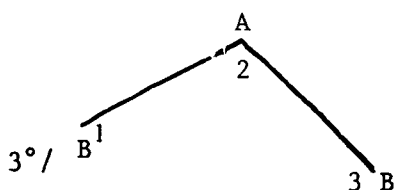
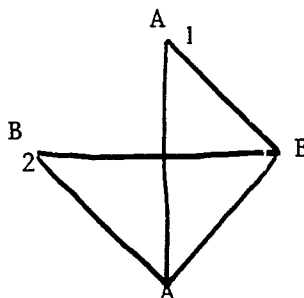
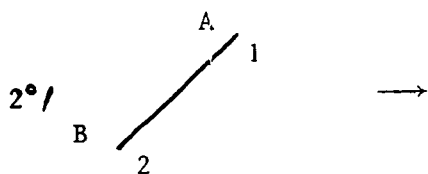
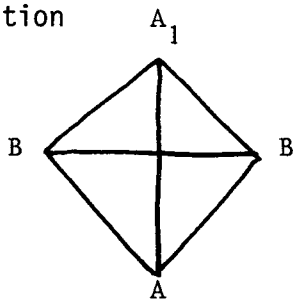
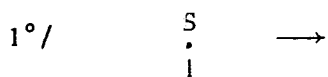
THEOREME

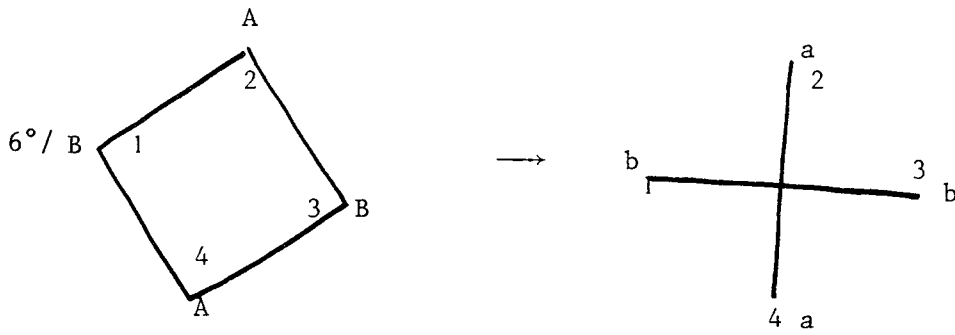
La grammaire de graphe suivante, génère indirectement tous les graphes planaires sans isthmes.

Elle génère exactement ces graphes et leur dual.

$$\begin{aligned} \mathcal{G} &= (V, I, R) \\ V &= V_S \cup \{e\} \quad (\text{graphes enseignés sur les sommets}) \\ V_S &= V_{ST} \cup V_{SN} \\ V_{ST} &= \{a, b\} \quad V_{ST} = \{a\} \\ V_{SN} &= \{S, A, B\} \\ I &: \mathcal{S} \end{aligned}$$

Ensemble des règles de production

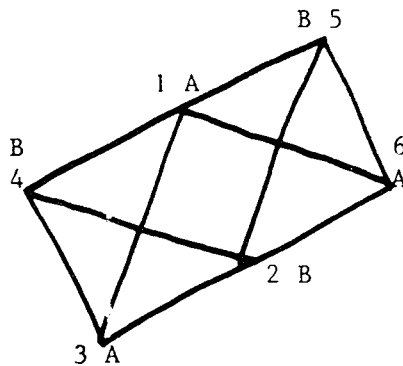
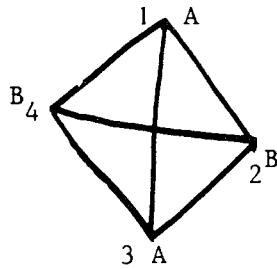
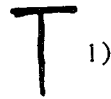


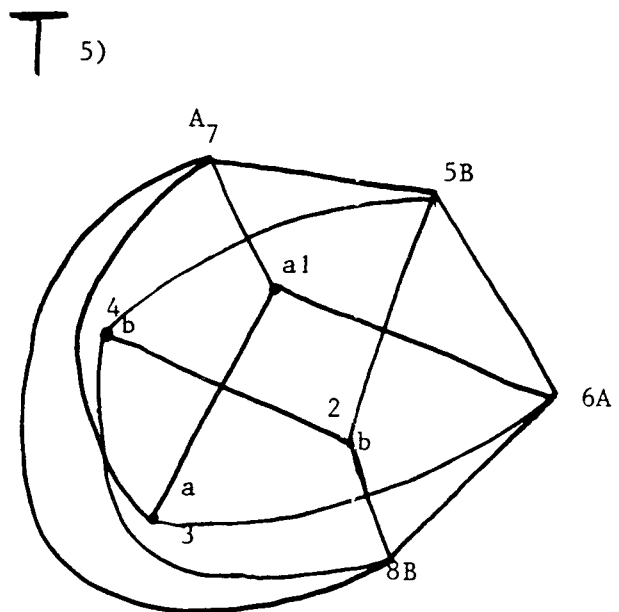
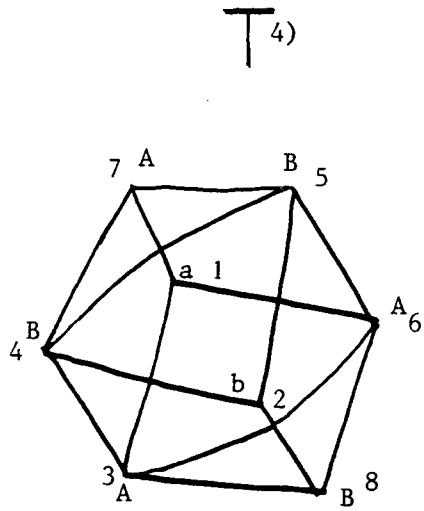
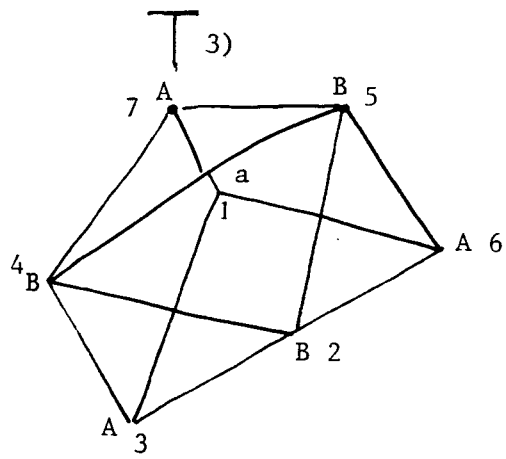


Exemple de dérivation

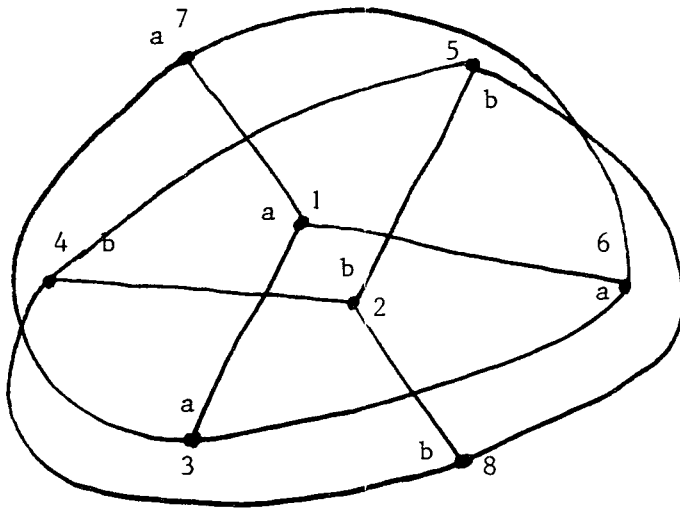
(on numérote les sommets au fur et à mesure de leur création et ce numéro reste le même ; le sous-graphe de partie gauche est surchargé)

1 · S

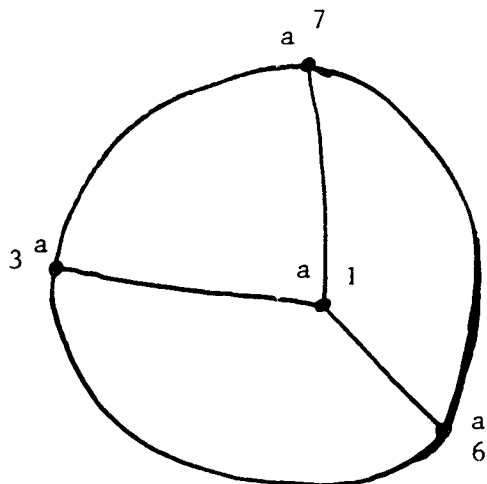




T_6



Le graphe indirectement g n r  est



DEMONSTRATION

On démontre d'abord que la grammaire génère indirectement tous les graphes planaires sans isthmes.

Soit G un graphe planaire sans isthmes quelconque. Considérons une représentation planaire de ce graphe.

On veut construire une copie de ce graphe et de son dual en appliquant les règles de la grammaire.

On cherche une séquence de courbes de coupure du graphe telles que la partie extérieure du graphe représente la partie égale déjà copiée. On montre alors par induction qu'il est toujours possible de construire le prochain pas de la copie en appliquant une des règles de la grammaire.

A un "pas" dans la copie on a les propriétés suivantes :

1°/ Les sommets extérieurs sont renseignés par le symbole "a" pour le graphe et "b" pour le dual.

2°/ Les sommets frontières sont renseignés par le symbole "A" pour le graphe et "B" pour le dual.

3°/ La courbe de coupure existe dans la copie, c'est un cycle d'arêtes reliant alternativement les sommets renseignés par A et ceux renseignés par B. Nous l'appellerons cycle des symboles non terminaux.

4°/ On vérifie aisément en regardant les règles de production de la grammaire

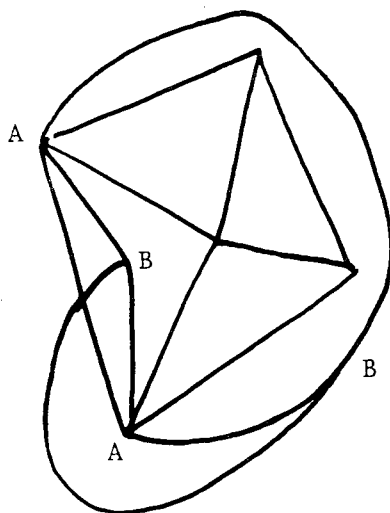
- a) il n'y a pas d'arête reliant deux sommets renseignés par le symbole "A" (resp."B") : de telles arêtes n'existent ni dans les graphes de la partie droite, ni dans les graphes de la partie gauche, et l'on n'en crée jamais en appliquant l'une des six règles.
- b) Tout sommet renseigné par le symbole "A" (resp."B") est connecté à exactement deux sommets renseignés par le symbole "B"(resp"A") (pour les mêmes raisons).

Pour trouver une première courbe de coupure, on considère la face extérieure de la représentation planaire et l'une quelconque des arêtes adjacentes à cette face.

La courbe de coupure contiendra tout le graphe sauf cette arête. Elle passera par la face extérieure (représentée par un sommet renseigné par le symbole "B") et l'autre face adjacente à l'arête choisie (représentée également par le symbole "B"). Les deux sommets renseignés par le symbole A sont les deux extrémités de l'arête.

La règle de la grammaire correspondant à cette construction est évidemment la règle 1.

Il ne reste plus pour compléter le schéma qu'à construire l'arête reliant entre eux les deux sommets renseignés par le symbole "B".



Dans la suite on considérera que les sommets renseignés par les symboles B et b correspondent aux faces frontières et extérieures. On se retrouve alors dans les mêmes hypothèses que pour le lemme 1, On peut donc trouver à chaque pas une arête intérieure (arête du graphe qui n'a pas encore été copiée) ayant l'une des 4 propriétés trouvées. A ces 4 types d'arêtes on peut associer respectivement la construction des quatre règles 2/, 3/, 4/ et 5/ comme nous allons le voir cas par cas.

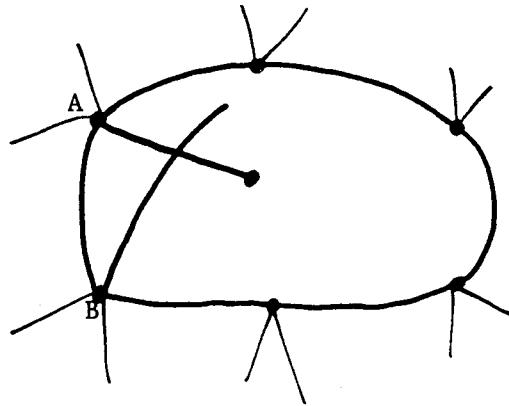
1, CAS DES PROPRIETES 1, 2a, 3a ;

On peut trouver une arête intérieure ayant les propriétés suivantes :

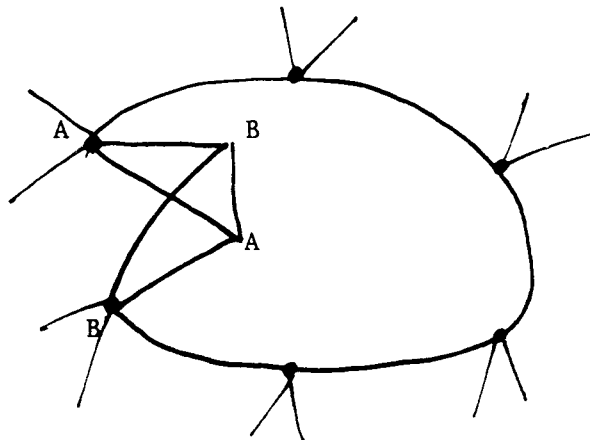
- Une des extrémités est un sommet frontière (sommet renseigné par le symbole A) et une des faces adjacentes à l'arête est une face frontière (représentée par un sommet renseigné par le symbole B).

L'autre extrémité est un sommet intérieur (c'est donc un nouveau sommet que l'on va copier), l'autre face est une face intérieure (c'est également une nouvelle face).

On peut schématiser ce cas ainsi :



La règle à appliquer pour recopier l'arête est la règle 2/. On obtient alors



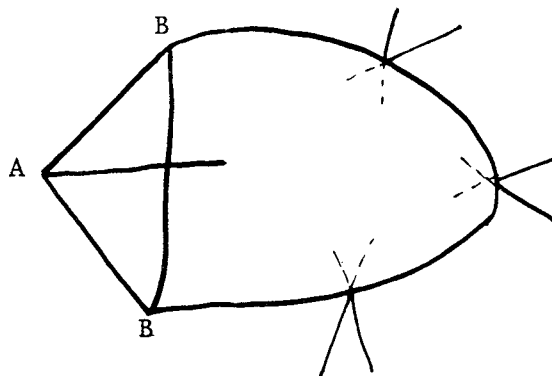
2ème CAS PROPRIETES 1, 2a, 3b

On peut trouver une arête intérieure ayant les propriétés suivantes :

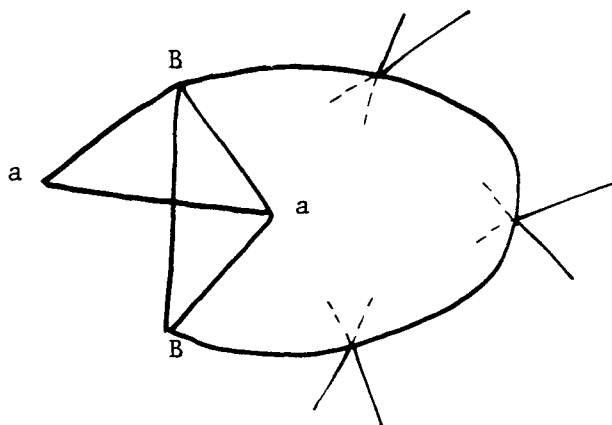
Une des extrémités de l'arête est un sommet frontière (A) et une des faces adjacentes à l'arête est une face frontière (B)

L'autre extrémité est un sommet intérieur (nouveau sommet à recopier)
l'autre face est une face frontière adjacente à la première (B).

On a donc le schéma suivant



La règle à appliquer pour recopier l'arête est la règle 3) et on obtient



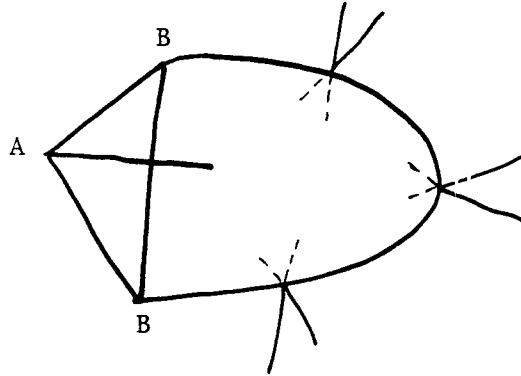
3ème CAS PROPRIETES 1, 2b, 3b

On peut trouver une arête intérieure ayant les propriétés suivantes :

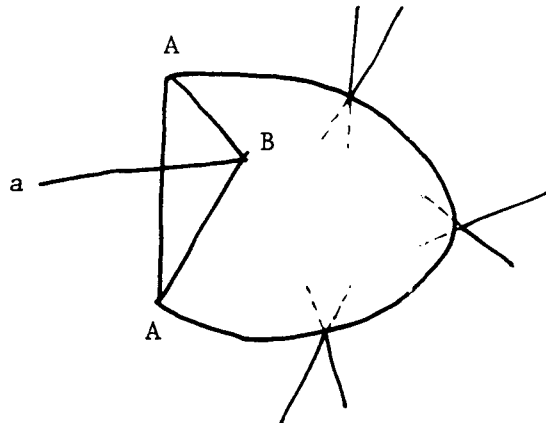
Une des extrémités de l'arête est un sommet frontière (A) et une des faces adjacentes à l'arête est une face frontière (B).

L'autre extrémité est un sommet frontière (A) consécutif du premier sur la coupe de coupure. L'autre face est une face intérieure (nouvelle).

On a donc le schéma suivant



La règle à appliquer pour recopier l'arête est la règle 4) et on obtient



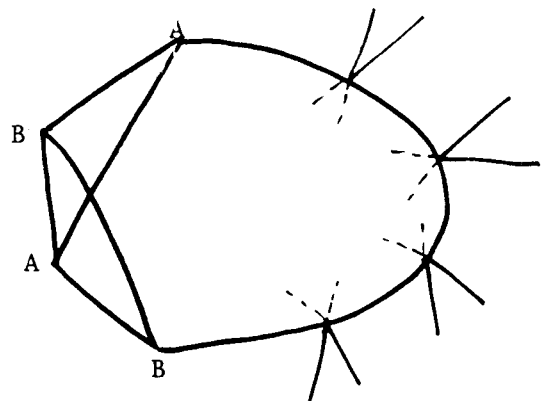
4ème CAS PROPRIETES 1, 2b, 3b

On peut trouver une arête intérieure ayant les propriétés suivantes :

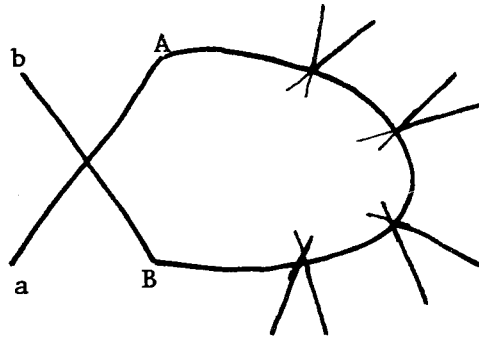
Les deux extrémités sont des sommets frontière (A) consécutifs

Les deux faces adjacentes sont des faces frontière (B)

On obtient le schéma suivant



La règle à appliquer est la règle 5) s'il y a plusieurs arêtes intérieures ou la règle 6) s'il ne reste plus qu'une arête à recopier. On obtient



Dans le cas où il ne reste qu'une arête à recopier, on retrouve obligatoirement les hypothèses du 4ème cas et on applique la règle 6). Le graphe est ainsi complètement recopié et l'on a de même construit le dual.

REMARQUE

Ce processus de "recopiage" du graphe arête par arête n'est pas qu'un simple artifice de programmation. Comme nous le verrons dans le chapitre suivant on peut par ce processus élaborer un algorithme d'analyse du graphe d'après la grammaire.

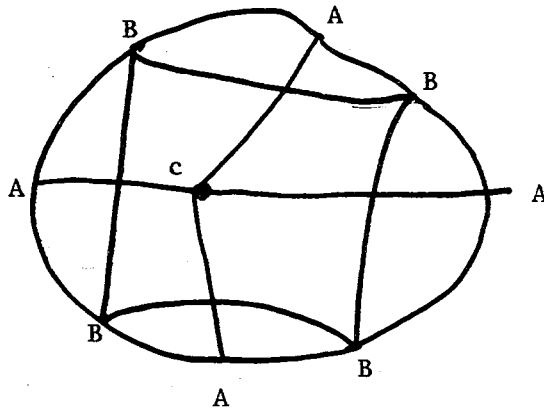
On démontre ensuite que la grammaire ne génère que les graphes planaires, c'est-à-dire que à tous les stades de la dérivation après avoir appliqué une règle la partie du graphe obtenue et son dual peut être dessinée sur le plan.

Soit G_i le graphe obtenu après l'application de i règles de la grammaire.

Les sommets renseignés par les symboles "A" et "B" sont en alternance sur un cycle que nous avons appelé "cycle des symboles non terminaux". Les autres sommets (renseignés par les symboles "a" et "b") sont à l'extérieur du cycle.

A partir de G_i , on construit G_i' en ajoutant à l'intérieur du cycle un sommet renseigné par le symbole "c" et connecté à tous les sommets renseignés par "A",

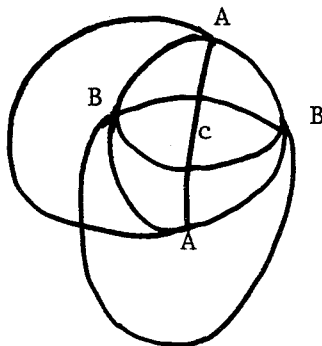
Les sommets renseignés par "B" sont connectés deux par deux s'ils sont adjacents à un même sommet "A".



On a comme hypothèse que le sous-graphe formé des sommets renseignés par les symboles "A", "a" et "c" et les arêtes les connectant, est un graphe planaire sans isthme.

Le sous-sgraphe formé des sommets renseignés par les symboles "B" et "b" et des arêtes les connectant est le dual du premier.

Cette hypothèse est vraie si on n'a appliqué que la règle 1 :



Supposons qu'elle soit vraie pour $G_i^!$ et appliquons à G_i l'une quelconque des règles 2/ à 6/

On obtient le graphe G_{i+1} .

Dans tous les cas on peut construire le graphe $G_{i+1}^!$ dont le sous-graphe

formé des sommets renseignés par les symboles "A", "a" et "c" et les arêtes les connectant est planaire sans isthme,

Avec la règle 2/ le nombre de sommets renseignés par les symboles "A" et de ceux renseignés par le symbole "B" est augmenté de un.

Pour construire G'_{i+1} on ajoute une arête "A-c" entre deux arêtes "A-c" G'_{i+1} reste planaire. Cette arête est adjacente à deux faces différentes donc G'_{i+1} reste sans isthme.

Pour les règles 3/ et 4/ la disposition des sommets sur le cycle des symboles non terminaux reste identique.

Pour la règle 5/ il y a un symbole A et un symbole B de moins, on enlève donc une arête "A-c".

Enfin après avoir appliqué la règle 6/ il n'y a plus de cycles des symboles non terminaux, donc G'_{i+1} , on a enlevé deux arêtes "A-c" qu'on a remplacées par une seule arête "a-a".

G_{i+1} est donc planaire sans isthme.

REMARQUES.

1°/ La règle 1/ est la première règle à appliquer (c'est la seule règle ayant le graphe initial ou partie gauche) mais on ne peut l'appliquer qu'une fois (il n'y a aucun graphe de partie droite ayant un sommet renseigné par le symbole "S").

2°/ La règle 6/ est la dernière règle à appliquer et on ne peut l'appliquer qu'une fois ; c'est la seule règle n'ayant pas de sommets renseignés par des symboles non terminaux dans le graphe de partie droite. C'est également la seule règle où le graphe de partie gauche est la totalité du cycle des symboles non terminaux. Donc après avoir appliqué la règle 6/ il n'y a plus de symboles non terminaux et la dérivation est terminée.

3°/ Chaque fois que l'on applique une règle, on crée une nouvelle arête.

Donc la longueur d'une dérivation (nombre de règles à appliquer pour générer un graphe) est égale au nombre d'arête du graphe indirectement généré (qui est également égal au nombre d'arêtes du dual).

4°/ RELATION D'EULER

Soit m le nombre d'arêtes du graphe.

n_a le nombre de sommets

n_b le nombre de sommets du dual.

n_B est le nombre de sommets renseignés par le symbole "B"

a tout stade de la dérivation on pose $f_i = n_{B_i} + n_{a_i} + n_{b_i}$

si l'on applique l'une des règles 2 à 5 f_i est augmenté de 1 : $f_{i+1} = f_i + 1$.

Si on applique la règle 1 ou 6 f_i est augmenté de 2

Or ces deux règles ne sont appliquées que une fois chacune, les quatre autres étant au total appliquées $m-2$ fois.

$$\text{On a donc } (m-2) + 2 \times 2 = n_a + n_b$$

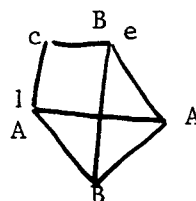
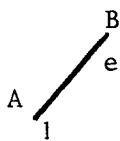
soit la relation d'Euler

$$m = n_a + n_b - 2$$

AUTRES GRAMMAIRES DE GRAPHES

En modifiant les règles de la grammaire précédente ou en ajoutant d'autre on peut générer indirectement d'autres classes de graphes planaires :

Graphes planaires non séparables en modifiant la règle 2°/



Tous les graphes planaires en ajoutant les règles :

$$7^\circ / \begin{array}{c} \cdot \\ \text{a} \end{array} \longrightarrow \begin{array}{c} \text{a} \text{---} \text{a} \\ \cdot \end{array} \quad \text{et} \quad 8^\circ / \begin{array}{c} \cdot \\ \text{a} \end{array} \longrightarrow \begin{array}{c} \text{a} \text{---} \text{S} \\ \cdot \end{array}$$

(voir MONTANARI [29])

CHAPÎTRE

IV

GRAMMAIRES DE GRAPHES ET ANALYSE SYNTAXIQUE

4.1. NECESSITE DE L'ANALYSE SYNTAXIQUE

4.1.1. POSITION DU PROBLEME

Soit $\mathcal{G} = (V, I, R)$ une grammaire de graphes et soit $L_{\mathcal{G}}$ le langage produit par \mathcal{G} .

$L_{\mathcal{G}}$ est un ensemble de graphes. Etant donné un graphe quelconque G , l'analyse syntaxique permet de résoudre les deux problèmes suivants :

1°/ Est-ce que G appartient à $L_{\mathcal{G}}$?

2°/ Le problème de l'appartenance de G à $L_{\mathcal{G}}$ étant résolu, quelle est la dérivation ou quelles sont les dérivations de \mathcal{G} qui permettent de produire G ?

4.1.2. PROBLEMES SE RAMENANT A UN PROBLEME D'ANALYSE SYNTAXIQUE

De nombreux problèmes concernant les graphes peuvent se ramener à un problème d'analyse syntaxique

1°/ Etant donné une classe de graphe C , si il existe une grammaire \mathcal{G} telle que le langage produit par \mathcal{G} , $L_{\mathcal{G}}$ coïncide exactement avec C , alors le problème de l'appartenance d'un graphe quelconque G à la classe C , c'est-à-dire à $L_{\mathcal{G}}$ est donc un problème d'analyse syntaxique.

2°/ Etant donné un graphe G appartenant au langage $L_{\mathcal{G}}$ produit par une grammaire \mathcal{G} , la dérivation (suite des règles, et pour chaque règle renseignements sur "l'emplacement" du sous-graphe de partie gauche par rapport à l'ensemble du graphe) qui permet de produire G , détermine de façon non ambiguë ce graphe.

Tout comme une matrice renseignée, ou un tableau d'arêtes, la dérivation, obtenue par l'analyse syntaxique, peut être une représentation ou une description du graphe.

Dans certains cas, comme nous le verrons pour la grammaire de Montanari, cette représentation peut être très économique sinon le plus économique, et on peut envisager son utilisation pour le codage en machine des descriptions des graphes.

3°/ Soient deux graphes G et G' appartenant au langage produit par une grammaire G . Si il existe une dérivation D qui permet de produire à la fois G et G' , alors les deux graphes G et G' sont isomorphes.

Pour montrer que deux graphes sont isomorphes, on peut donc, d'abord faire l'analyse syntaxique du premier graphe afin de trouver la dérivation D qui permet de la produire, puis à partir de cette dérivation essayer de produire le second graphe.

Dans certaines conditions on peut également montrer l'homomorphisme de deux graphes.

Dans le chapitre 5 on reviendra sur ces différentes applications de l'analyse syntaxique.

4.2. LES PRINCIPES DE L'ANALYSE SYNTAXIQUE ET LES GRAMMAIRES DES GRAPHES.

4.2.1. CRITERES DE CLASSIFICATION DES ALGORITHMES [49]

Les principaux algorithmes d'analyse syntaxique ont été développés pour des langages de type hors-contexte. Les critères généralement adoptés pour la classification de ces algorithmes sont les suivants :

- 1 - l'algorithme est-il ad'hoc, c'est-à-dire propre à un seul langage, ou est-il paramétrique, c'est-à-dire la syntaxe du langage est-il un paramètre de l'algorithme ?
- 2 - l'analyse est-elle unique (une seule dérivation fournie) ou multiple (plusieurs dérivations) ?
- 3 - l'algorithme est-il déterministe ou pas ?

- 4 - l'algorithme est-il ascendant, descendant ou mixte ?
- 5 - l'analyse sur la chaîne se fait-elle de gauche à droite ou de droite à gauche ?
- 6 - à chaque pas de l'algorithme, analyse-t-on un caractère ou n caractères ?
- 7 - l'algorithme est-il général ou particulier, c'est-à-dire doit-on mettre la grammaire sous une forme particulière (forme normale, forme standard de Greisbach... pour les grammaires de type hors-contexte) ?
- 8 - l'algorithme est-il relatif ou non, c'est-à-dire peut-on adjoindre des renseignements à l'algorithme ?

Les critères 1,2,3 et 8 - c'est-à-dire algorithme ad-hoc ou paramétrique, analyse unique ou multiple, algorithme déterministe ou non, et enfin sélectif ou non - sont très généraux et peuvent donc s'appliquer aux algorithmes d'analyse syntaxique de grammaires de graphes.

Le critère sur le sens de l'analyse de gauche à droite ou de droite à gauche n'a pas de sens en ce qui concerne l'analyse des graphes, en effet on ne peut pas définir, d'une façon générale une relation d'ordre total que ce soit sur l'ensemble des sommets d'un graphe ou sur l'ensemble des arêtes d'un graphe.

Quant au nombre de caractères à analyser à chaque pas de l'algorithme, il s'agira évidemment du nombre de sommets et du nombre d'arêtes du graphe.

Les algorithmes d'analyse de graphes pourront également être généraux ou particuliers suivant qu'il faudra apporter des transformations à la grammaire de graphes ou pas.

Nous verrons dans les paragraphes suivants à quoi correspondent les critères d'analyse ascendante et d'analyse descendante pour les grammaires de graphes.

4.2.2. ANALYSE ASCENDANTE

PRINCIPE.

On part du graphe à analyser et en "remontant" la dérivation on essaie d'arriver à l'un des graphes initiaux de la grammaire.

A chaque étape, on cherche donc, s'il existe un sous-graphe du graphe en cours d'analyse, sous-graphe isomorphe à l'un des graphes de la partie droite des règles de la grammaire.

Si un tel sous-graphe existe, on "applique à l'envers" la règle correspondante, (si c'est possible), c'est-à-dire qu'on remplace le graphe de partie droite par le graphe de partie gauche, en respectant la correspondance entre les sommets et les arêtes.

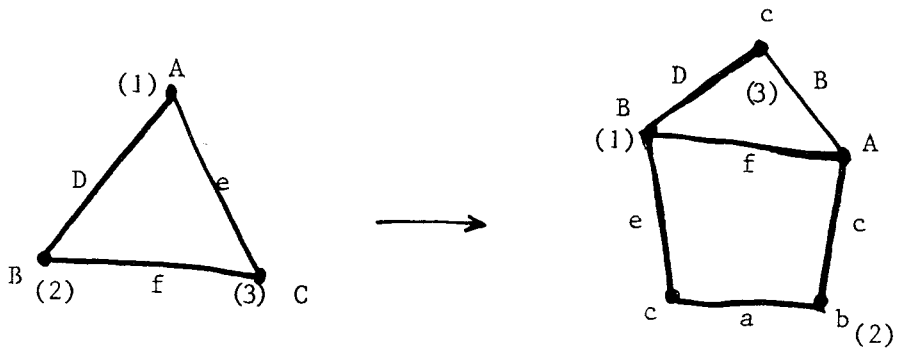
On recommence un tel processus jusqu'à ce que le graphe en cours d'analyse soit lui-même isomorphe à l'un des graphes initiaux de la grammaire. Dans ce cas le graphe appartient bien au langage généré par la grammaire et l'on obtient la dérivation en prenant en sens inverse l'ensemble des règles utilisées au cours de l'analyse.

Si au cours de l'analyse on ne trouve pas de sous-graphe isomorphe à un graphe de la partie droite, c'est que le début de l'analyse est erroné, il faut essayer une autre voie. Si toutes les voies ont été essayées sans succès, c'est que le graphe n'appartient pas au langage produit par la grammaire.

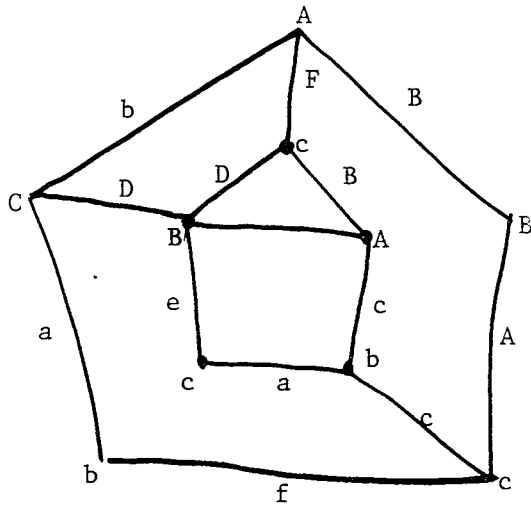
C'est la façon dont on essaie successivement ou parallèlement toutes les voies d'analyse qui fait la particularité des algorithmes.

EXEMPLE

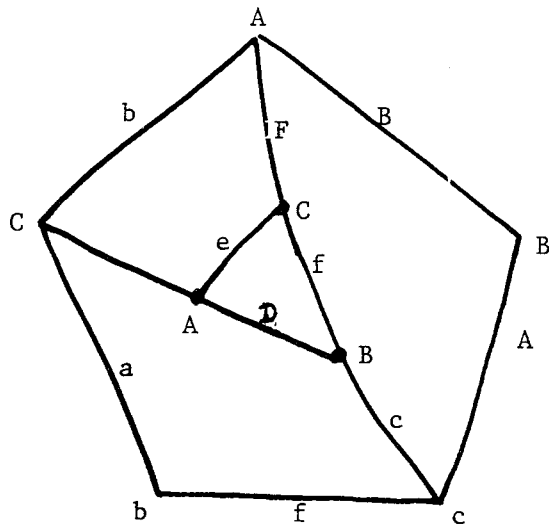
Supposons que dans la grammaire de graphes on ait la règle suivante :



Les chiffres entre parenthèses indiquent la correspondance entre les sommets
 Si au cours de l'analyse on a le graphe suivant



En appliquant à l'envers la règle précédente on obtient :



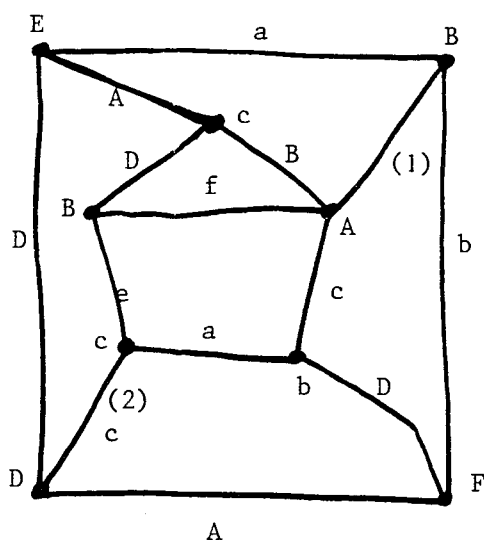
REMARQUE

Il n'est toujours pas possible, ayant trouvé un sous-graphe isomorphe à un graphe de partie droite, "d'appliquer la règle à l'envers".

En effet, la correspondance entre les sommets du graphe de partie droite et du graphe de partie gauche, n'est pas une bijection.

Si l'application n'est pas surjective, et si le degré d'un sommet qui n'est l'image d'aucun autre est plus grand dans le graphe que celui du même sommet dans le sous-graphe de partie droite, alors les arêtes partant de ce sommet n'appartenant pas à ce sous-graphe ne pourront être connectés à aucun sommet.

C'est le cas, si voulant appliquer à l'envers la règle de l'exemple précédent, on avait le graphe suivant :



Les sommets étiquetés A et C dans le graphe de partie droite ne sont l'image d'aucun sommet du graphe de partie gauche, donc dans la substitution des arêtes (1) et (2) ne pourront être connectées d'aucun sommet.

D'un autre côté si l'application n'est pas injective, un même sommet peut être l'image de plusieurs autres, il y aura donc plusieurs façons de connecter les arêtes ayant un de ces sommets comme extrémité, et n'appartenant pas au sous-graphe. Il faudra alors envisager toutes les possibilités ce qui évidemment alourdit considérablement l'algorithme.

CONCLUSION

La performance d'un algorithme d'analyse ascendant est limitée par le nombre d'essais à faire pour couvrir toutes les possibilités d'application des règles à chaque étape. Il peut d'une part y avoir plusieurs graphes de parties droites des règles de grammaire isomorphes à plusieurs sous-graphes du graphe en cours d'analyse.

D'autre part si l'application entre les sommets du graphe de partie droite et ceux de partie gauche, n'est pas injective, il peut y avoir plusieurs façons d'appliquer la règle. (D'où l'intérêt de n'utiliser que les grammaires pour lesquelles, dans toutes les règles d'application, cette application est injective).

L'intérêt d'un algorithme d'analyse ascendant, est que partant du graphe à analyser, il est constamment guidé par ce **graphe** qui se modifie tout au cours de l'analyse.

4.2.3. ANALYSE ASCENDANTE

PRINCIPE

On part de l'un des graphes initiaux, et l'on essaie, après applications successives des règles de la grammaire de produire le graphe à analyser.

La façon d'appliquer une règle ne pose dans ce cas aucun problème car dans la règle d'application sont définies les conditions d'imbrication et il n'y a donc aucune ambiguïté.

A chaque pas de l'algorithme, il faut choisir d'une part la règle à appliquer et trouver d'autre part dans le graphe déjà généré le ou les sous-graphes isomorphes au graphe de la partie gauche de la règle.

Mise à part quelques considérations sommaires sur le nombre de sommets ou d'arêtes créés ou le degré des sommets, le choix de la règle à appliquer est très arbitraire, car la donnée d'un graphe produit par une grammaire ne donne que très peu de renseignements sur la façon dont ce graphe a été produit. Il

faudra donc envisager à chaque pas toutes les possibilités d'application de règle de production.

CONCLUSION

A la limite on peut presque dire qu'un algorithme d'analyse descendant consiste à énumérer tous les graphes produits par la grammaire jusqu'à ce qu'on ait trouvé le graphe à analyser.

Pas plus que l'algorithme d'analyse ascendante, cet algorithme est en général efficace : ce sont des algorithmes essentiellement combinatoires car on essaie toutes les solutions possibles.

De plus les principales opérations à effectuer sont des recherches d'isomorphismes ou d'homomorphismes entre graphes, recherches qui sont également combinatoires.

Bien que faciles à décrire et à réaliser, ces algorithmes ne peuvent s'appliquer au cas général en raison du nombre d'opérations à effectuer, nombre croissant exponentiellement avec la complexité du graphe à analyser.

4.2.4. CHOIX DE L'ALGORITHME

Si, d'un point de vue théorique, rien n'empêche d'écrire des algorithmes d'analyse paramétriques, en raison de leur caractère combinatoire de tels algorithmes sont impossibles à mettre en oeuvre.

Pour réduire le nombre d'opérations à effectuer on doit nécessairement tenir compte des particularités de la grammaire, et pour cela on réalisera des algorithmes ad'hoc.

Ces algorithmes devront autant que possible être mixtes, c'est -à-dire ascendants pour que l'analyse soit "guidé" par le graphe à analyser et descendants pour enlever toute ambiguïté au niveau de l'application des règles.

Nous verrons dans les paragraphes suivants, comment en étudiant la grammaire

de Montanari, on peut réaliser un algorithme répondant à ces caractéristiques.

4.3. ANALYSE SYNTAXIQUE POUR LA GRAMMAIRE DE MONTANARI.

4.3.1. EXISTENCE D'UNE DERIVATION SYMETRIQUE

La première particularité de la grammaire de Montanari que l'on met en évidence est l'existence d'une dérivation symétrique : pour toute dérivation D permettant de produire un graphe donné G , il existe une dérivation symétrique D' permettant de produire le même graphe que G , mais en créant les arêtes dans l'ordre inverse de celui où elles ont été créées par D .

Grâce à cette propriété, on vérifiera que tout algorithme ascendant (toujours pour la grammaire de Montanari) est équivalent à un algorithme descendant et réciproquement.

Mettons d'abord en évidence l'existence de la dérivation symétrique.

NOTATIONS.

Soit G un graphe indirectement généré par la grammaire de Montanari.

Si G a n arêtes, on notera (r_1, r_2, \dots, r_n) la suite des règles de la grammaire qui permettent de générer indirectement G .

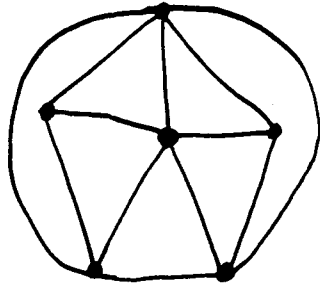
On notera G_i le sous-graphe produit après l'application des règles r_1, r_2 , jusqu'à r_i (dans le même ordre), A_i le sous-graphe de G_i restreint aux sommets étiquetés "a" et "A" et aux arêtes reliant de tels sommets, et B_i , celui restreint aux sommets étiquetés "b" et "B". On a évidemment $A_n = G$.

On notera $\overline{A_i}$ le complémentaire de A_i par rapport à G . (En considérant les graphes A_i et G comme n'étant pas étiquetés).

$\overline{A_i}$ est la partie du graphe G qui n'a pas encore été générée.

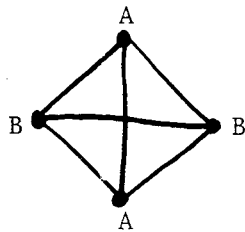
EXEMPLE

Le graphe G suivant



est produit en appliquant successivement les règles (1,2,2,2,3,4,4,5,5,4,5,6)

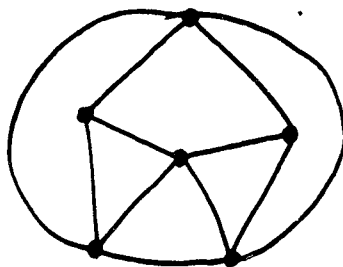
Après application de r_1 , c'est-à-dire la règle 1, on a le graphe



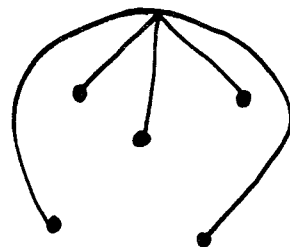
Le graphe A_1 est donc



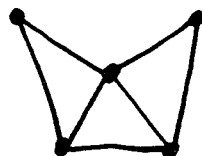
et $\overline{A_1}$ est



On vérifie de même que le graphe A_5 par exemple est

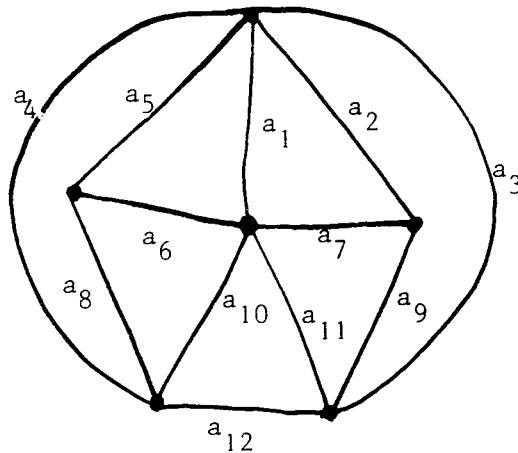


Alors que $\overline{A_5}$ est le graphe



Soit $D = (r_1, \dots, r_n)$ une dérivation, on note a_i l'arête de G créée par l'application de la règle i .

En reprenant l'exemple précédent, les arêtes du graphe G seraient ainsi étiquetées



Ces différentes notations seront utiles pour démontrer le théorème dont l'énoncé est le suivant :

THEOREME

A toute dérivation $D = (r_1, r_2, \dots, r_n)$, chaque r_i étant l'une des six règles de la grammaire de Montanari, on peut associer une "dérivation symétrique" $D' = (r'_1, r'_2, \dots, r'_n)$.

Ainsi définie : pour tout i $1 \leq i \leq n$ $r'_{n-i+1} = f(r_i)$

avec

$f(1) = 6$	$f(4) = 4$
$f(2) = 5$	$f(5) = 2$
$f(3) = 3$	$f(6) = 1$

La dérivation D' a les propriétés suivantes :

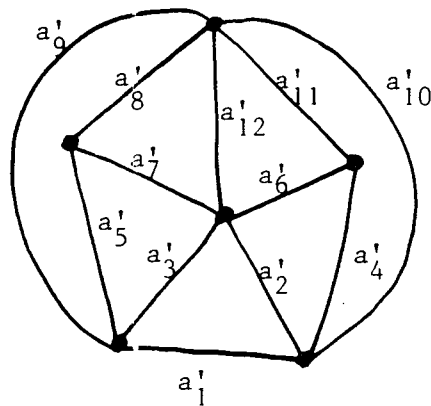
- 1°/ L'application successive des règles de la dérivation D' permet de produire le même graphe G que celui produit par D .
- 2°/ Les arêtes de G sont créées dans l'ordre inverse en dérivant selon D' que celui obtenu en dérivant selon D .
- 3°/ L'ensemble des sous-graphes de G obtenus à chaque application d'une règle, selon D d'une part et selon D' d'autre part, sont deux à deux complémentaires, en les associant dans l'ordre où ils sont produits

pour D dans l'ordre inverse pour D',

EXEMPLE

Toujours pour le graphe précédent, la dérivation symétrique de la dérivation (1,2,2,2,3,4,4,5,5,4,5,6) est la dérivation (1,2,4,2,2,4,4,3,5,5,5,6)

Les arêtes sont créées selon D' dans l'ordre a'_1, a'_2, \dots ainsi



DEMONSTRATION

Pour toutes les dérivations, la première règle qu'on doit appliquer est la règle 1 car c'est la seule qui ait le graphe initial en partie gauche. La dernière règle à appliquer est obligatoirement la règle 6 car c'est la seule n'ayant pas de symboles non terminaux en partie droite.

Donc quel que soit D on a $r_1=1$ et $r_n=6$.

On peut alors écrire pour D' $r'_1=1$ et $r'_n=6$

D'autre part pour $1 < i < n$ on ne peut en aucun cas avoir $r_i=1$ car une fois appliquée la règle 1 le symbole non terminal "S" ne se rencontre plus.

De même on ne peut avoir $r_i=6$ (toujours pour $1 < i < n$) car après application de la règle 6, il n'y a plus de symboles non terminaux, dont la règle 6 est obligatoirement la dernière règle appliquée.

Dans tous les cas on a bien

$f(1) = 6$
 et $f(6) = 1$

Pour la suite de la démonstration on ne considérera que les règles r_i pour $1 < i < n$ sachant que r_i ne peut être que l'une des règles 2,3,4 ou 5.

On rappelle que G_i est produit après application des règles r_1, r_2, \dots, r_i et que a_i est l'arête créée par l'application de la règle r_i .

Les graphes A_i et B_i sont deux sous-graphes de G_i et ils n'ont aucun sommet ni aucune arête en commun.

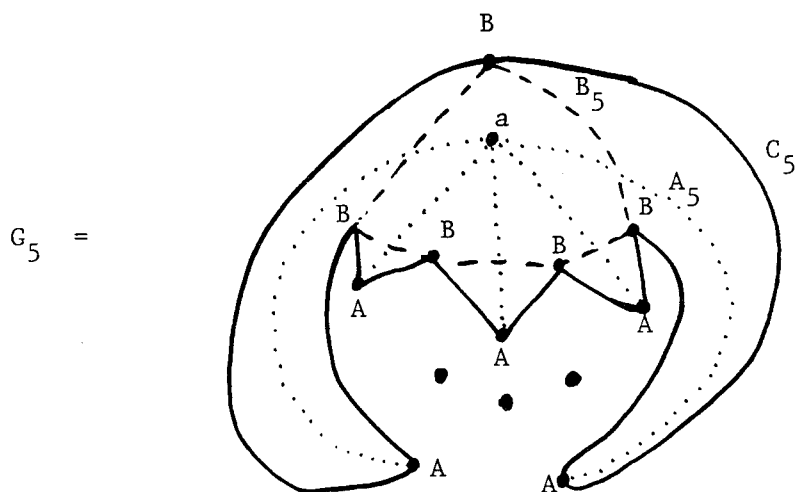
Le graphe G_i est formé du graphe A_i , du graphe B_i et d'un nombre pair d'arêtes reliant un sommet étiqueté "A" à un sommet étiqueté "B". On appelle C_i le sous-graphe de G_i formé de tous les sommets étiquetés "A" ou "B" et des arêtes les reliant, arêtes n'appartenant ni à A_i ni à B_i .

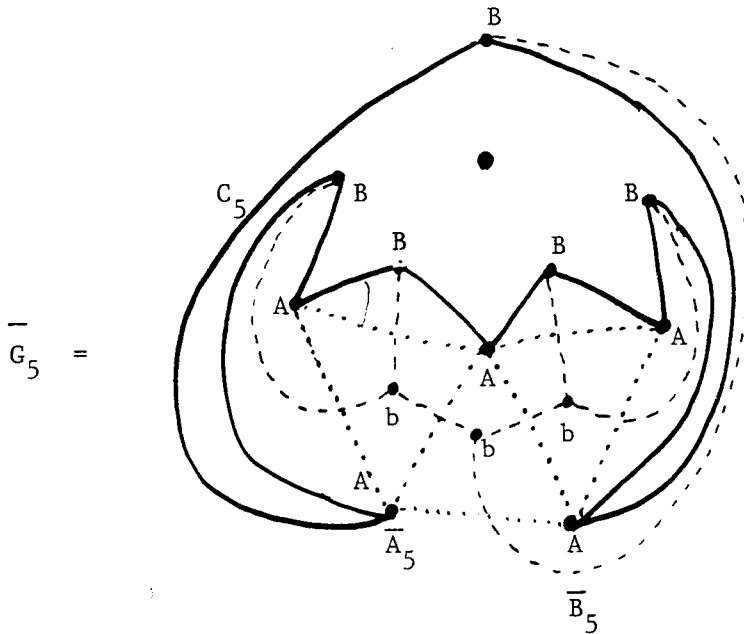
On définit ainsi le graphe \overline{G}_i .

\overline{G}_i est formé des graphes \overline{A}_i et \overline{B}_i (\overline{B}_i complémentaire de B_i par rapport à B_n le dual de G) ainsi que du graphe C_i .

Ceci est possible car les sommets étiquetés par "A" (resp. "B") sont les seuls sommets appartenant à la fois à A_i et à \overline{A}_i (resp. B_i et \overline{B}_i)

On voit toujours pour l'exemple cité précédemment comment est construit le graphe \overline{G}_5 à partir des graphes \overline{A}_5 et \overline{B}_5 et C_5 .





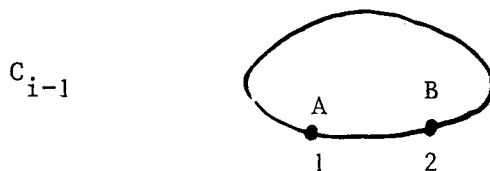
On écrira $G_i = A_i \cup B_i \quad C_i$

$\overline{G}_i = \overline{A}_i \cup \overline{B}_i \quad C_i$

On va montrer qu'il existe une bijection sur l'ensemble des règles $\{2,3,4,5\}$ telle que si l'on applique $f(r_i)$ à \overline{G}_{i-1} on obtient $\overline{G}_i = 1$.

1er cas $r_i = 2$

On numérote 1 et 2 les deux sommets de C_{i-1} qui font partie du graphe de partie gauche de la règle 2



En appliquant la règle 2, on crée un nouveau sommet 3 étiqueté par "A" et une arête a_i entre 1 et 3.

Donc A_i s'obtient à partir de A_{i-1} en ajoutant le sommet 3 et l'arête a_i entre 1 et 3.

On crée également un nouveau sommet 4 étiqueté par "B" ainsi que l'arête b_i entre 2 et 4.

B_i est formé de B_{i-1} auquel on a adjoint le nouveau sommet 4 et une arête entre 2 et 4.

Pour avoir C_i à partir de C_{i-1} on ajoute les sommets 3 et 4 et on remplace l'arête (1,2) par les arêtes (1,4), (4,3), (3,2).

Le graphe \overline{G}_i précédemment défini, est formé de l'union des graphes $\overline{A}_i, \overline{B}_i$ et C_i , \overline{A}_i et \overline{B}_i étant les complémentaires des graphes A_i et B_i tels que nous les avons construits à partir de A_{i-1} et B_{i-1} .

Si maintenant, on applique la règle 5 à \overline{G}_i avec comme graphe de partie gauche, le graphe formé des arêtes (1,4), (4,3) et (3,2) ainsi que de leurs extrémités, on obtient le graphe formé de l'union des trois graphes suivants :

- le graphe \overline{A}_i auquel on adjoint une arête entre les sommets 1 et 3, l'arête a_i . Comme A_i étant l'union de A_{i-1} et de l'arête a_i , l'union de \overline{A}_i et de l'arête a_i est le graphe $\overline{A_{i-1}}$.

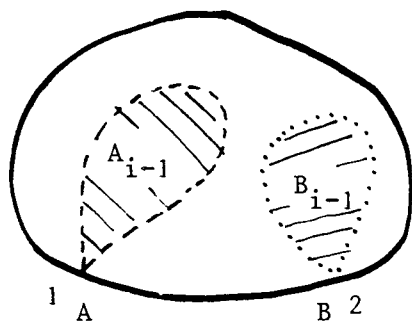
- le graphe \overline{B}_i auquel on adjoint une arête entre les sommets 2 et 4, l'arête b_i , c'est-à-dire le graphe $\overline{B_{i-1}}$

- le graphe C_i où l'on supprime les sommets 3 et 4 ainsi que les arêtes (1,4), (4,3), (3,2) remplacées par l'arête (1,2)

L'union de ces trois graphes forme par définition le graphe $\overline{G_{i-1}}$.

On peut schématiquement représenter le graphe G_{i-1} , le même graphe auquel on a appliqué la règle 2, c'est-à-dire le graphe G_i ainsi :

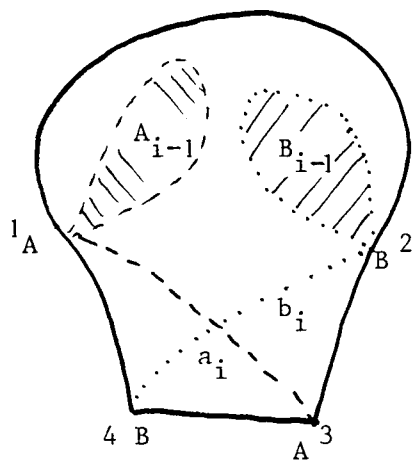
G_{i-1}



R2

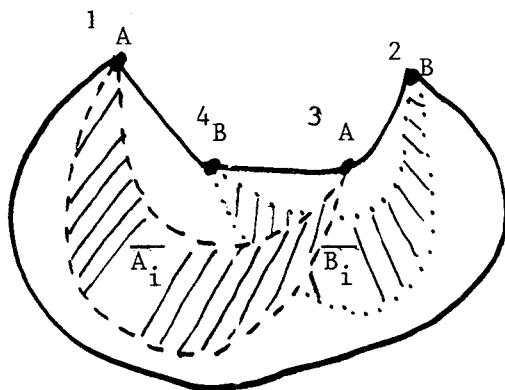


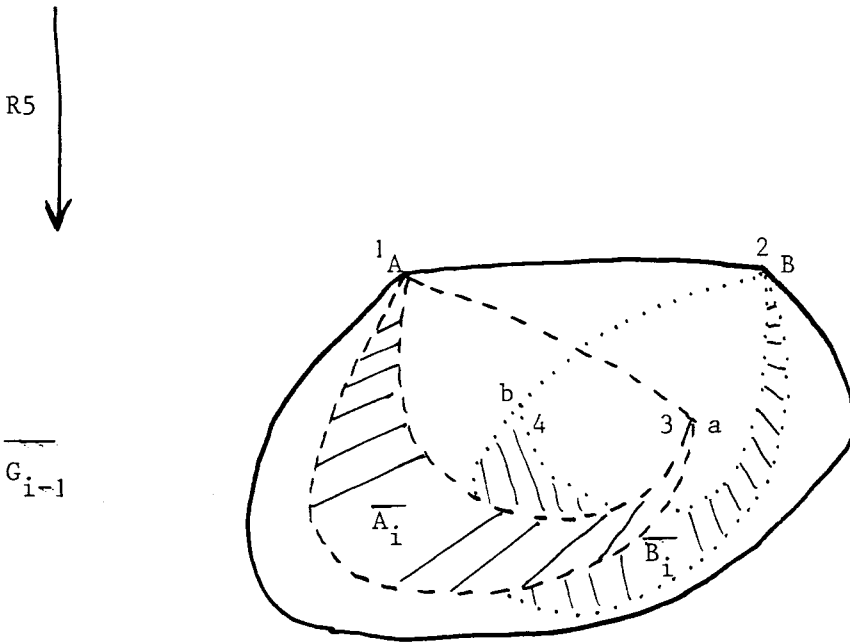
G_i



De même pour le graphe \overline{G}_i et l'application à ce graphe de la règle 5

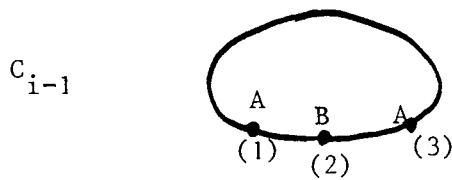
\overline{G}_i





2ème cas $r_i = 3$

On numérote 1,2,3 les trois sommets de C_{i-1} qui font partie du graphe de partie gauche de la règle 3



En appliquant la règle 3 on crée une arête a_i entre les sommets étiquetés par "A" 1 et 3.

Donc à partir de A_{i-1} on obtient A_i en ajoutant la seule arête a_i entre 1 et 3.

On crée un nouveau sommet 4 étiqueté par "B" et on a l'arête b_i entre 2 et 4.

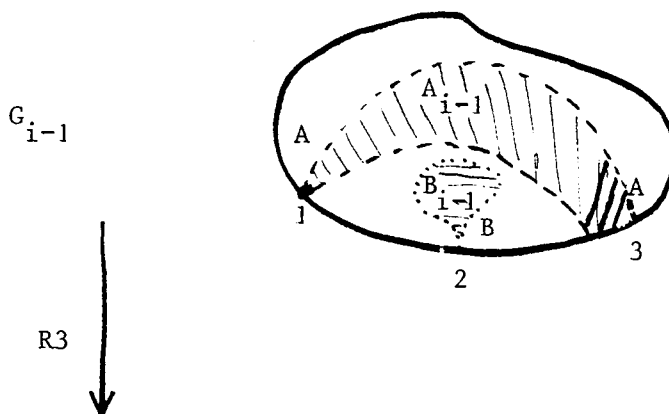
B_i est formé de B_{i-1} auquel on a adjoint un nouveau sommet 4 et une arête entre 2 et 4.

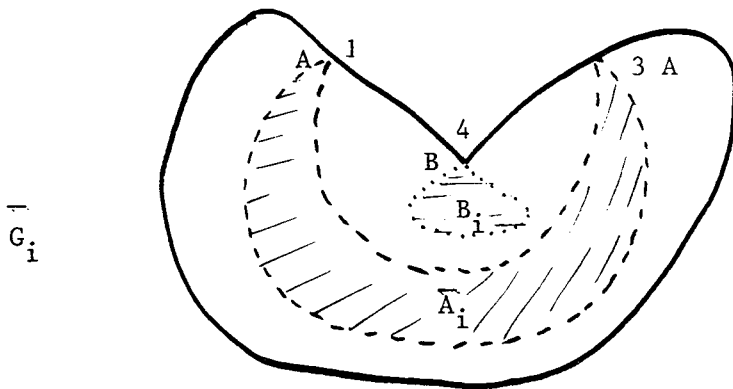
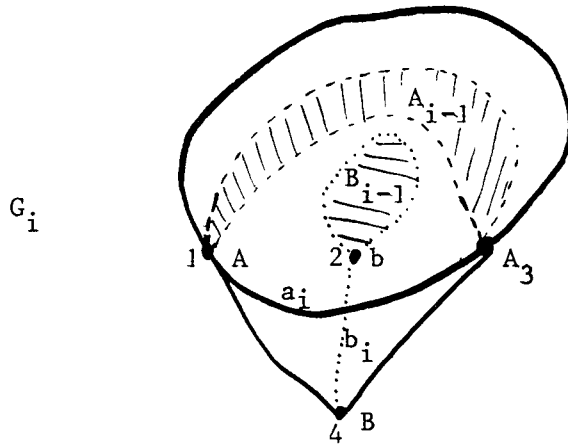
Pour avoir C_i à partir de C_{i-1} on remplace le sommet 2 par le sommet 4 et les arêtes (1,2) et (2,3) par les arêtes (1,4) et (4,3).

Si maintenant on applique toujours la règle 3 à $\overline{G_i}$ avec comme graphe de partie gauche le graphe formé par les arêtes (1,4) (4,3) et de leurs extrémités les sommets 1,4 et 3, on obtient le graphe formé de l'union de trois graphes suivants :

- le graphe $\overline{A_i}$ auquel on adjoint une arête entre les sommets 1 et 3, l'arête a_i , c'est-à-dire le graphe $\overline{A_{i-1}}$.
- le graphe $\overline{B_i}$, auquel on adjoint un sommet 2, une arête entre 4 et 2, l'arête b_i , c'est-à-dire le graphe $\overline{B_{i-1}}$.
- le graphe C_i où l'on remplace le sommet étiqueté par "B" 4 par le sommet 2 et les arêtes (1,4) et (4,3) par les arêtes (1,2) et (2,3), c'est-à-dire le graphe C_{i-1} .

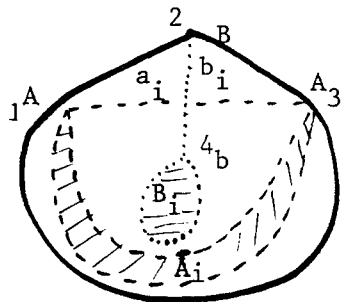
On a bien le graphe $\overline{G_{i-1}}$





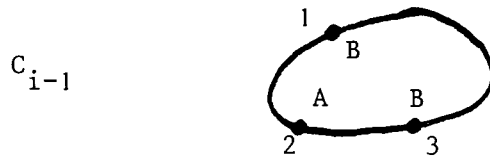
R3 ↓

\overline{G}_{i-1}



3ème cas $r_i = 4$

On numérote les 1,2,3 les trois sommets de C_{i-1} qui font partie du graphe de partie gauche de la règle 4



En appliquant la règle 4 on crée une arête b_i entre les sommets 1 et 3.

Donc B_i est formé de B_{i-1} et de l'arête b_i entre 1 et 3.

On crée également un nouveau sommet étiqueté par "A" numéroté ici 4, ainsi que l'arête a_i entre 2 et 4.

A_i est formé de A_{i-1} du nouveau sommet 4 et de l'arête a_i entre 2 et 4.

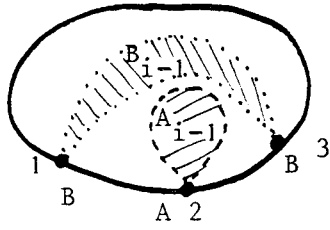
C_i s'obtient à partir de C_{i-1} en remplaçant le sommet 2 par le sommet 4 et les arêtes (1,2) et (2,3) par les arêtes (1,4) et (4,3)

Si on applique la règle 4 à $\overline{G_i}$ avec comme graphe de partie gauche le sous-graphe de C_i formé des arêtes (1,4),(4,3) et de leurs extrémités 1,4 et 3, on obtient le graphe formé de l'union des trois sous-graphes suivants :

- le graphe $\overline{A_i}$ auquel on adjoint un nouveau sommet numéroté 2, et une arête entre 4 et 2, l'arête a_i , c'est-à-dire le graphe $\overline{A_{i-1}}$.
- le graphe $\overline{B_i}$ auquel on adjoint une arête entre les sommets 1 et 3, l'arête b_i , c'est-à-dire le graphe $\overline{B_{i-1}}$.
- le graphe C_i où l'on a remplacé le sommet étiqueté par "A" 4 par le sommet 2 et les arêtes (1,4) et (4,3) par les arêtes (1,2) et (2,3), c'est-à-dire le graphe C_{i-1} .

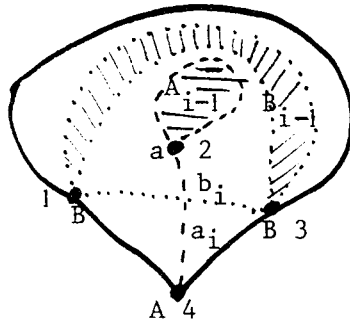
L'union de ces 3 graphes forme bien le graphe $\overline{G_{i-1}}$.

G_{i-1}

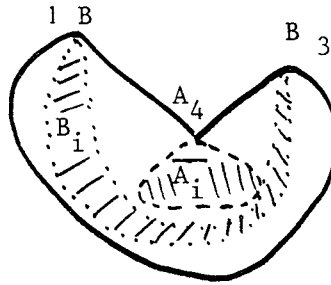


R4

G_i

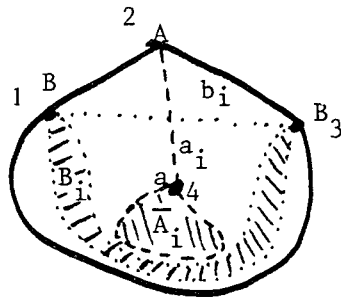


$\overline{G_i}$



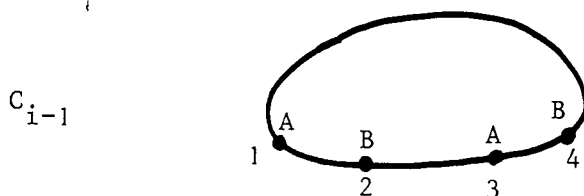
R4

$\overline{G_{i-1}}$



4ème cas $r_i = 5$

On numérote 1,2,3,4 les quatre sommets de C_{i-1} qui font partie du graphe de partie gauche de la règle 5



En appliquant la règle 5 on crée une arête a_i entre les sommets étiquetés par "A" 1 et 3.

Donc A_i s'obtient à partir de A_{i-1} en ajoutant l'arête a_i entre 1 et 3.

De même on crée l'arête b_i entre les sommets étiquetés par "B" 2 et 4.

B_i s'obtient à partir de B_{i-1} en ajoutant l'arête b_i entre 2 et 4.

Pour avoir C_i à partir de C_{i-1} on supprime les sommets 2 et 3 ainsi que les arêtes (1,2), (2,3) et (3,4) que l'on remplace par l'arête (1,4).

Si on applique la règle 2 au graphe \overline{G}_i avec comme graphe de partie gauche l'arête (1,4) et les sommets 1 et 4 on obtient le graphe formé de l'union des trois graphes suivants.

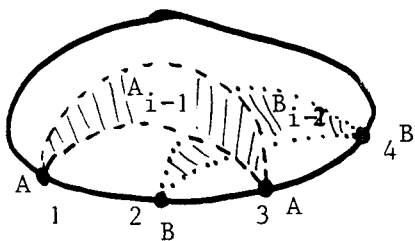
- le graphe A_i auquel on adjoint un nouveau sommet 3 et une arête entre 1 et 3, l'arête a_i , c'est-à-dire le graphe $\overline{A_{i-1}}$.

- le graphe B_i auquel on adjoint un nouveau sommet 2 et une arête entre 4 et 2, l'arête b_i , c'est-à-dire le graphe $\overline{B_{i-1}}$.

- le graphe C_i où l'on ajoute les sommets 2 et 3 et où l'on remplace l'arête (1,4) par les arêtes (1,2), (2,3) et (3,4).

- l'union de ces trois graphes forme bien le graphe $\overline{G_{i-1}}$

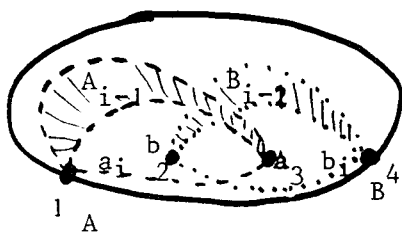
G_{i-1}



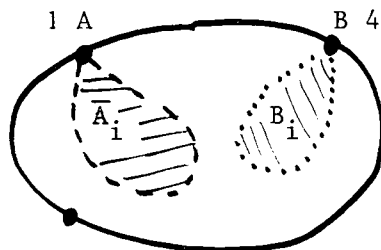
R5



G_i



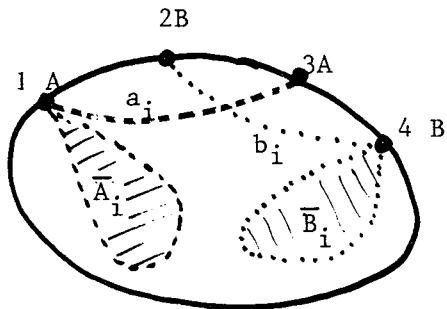
\overline{G}_i



R2



\overline{G}_{i-1}



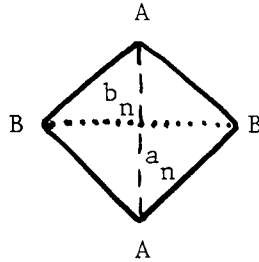
Nous avons vu d'une part le cas où $i=1$ et $i=n$ et d'autre part le cas où $1 < i < n$.

L'application f est bien définie pour l'ensemble des règles 1,2,3,4,5,6

$$\begin{array}{ll} f(1) = 6 & f(4) = 4 \\ f(2) = 5 & f(5) = 2 \\ f(3) = 3 & f(6) = 1 \end{array}$$

Soit D une dérivation $D = (r_1, r_2, \dots, r_n)$ permettant de générer le graphe G .
 G est le sous-graphe A_n du graphe G_n .

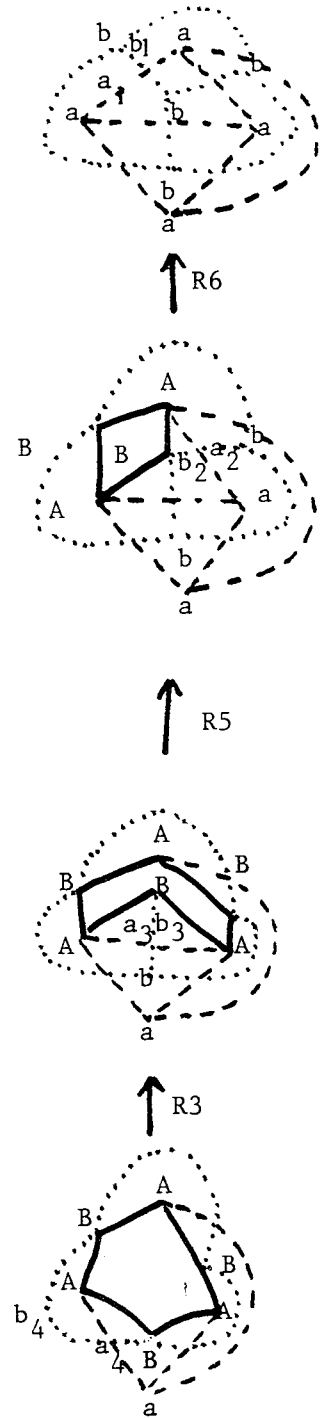
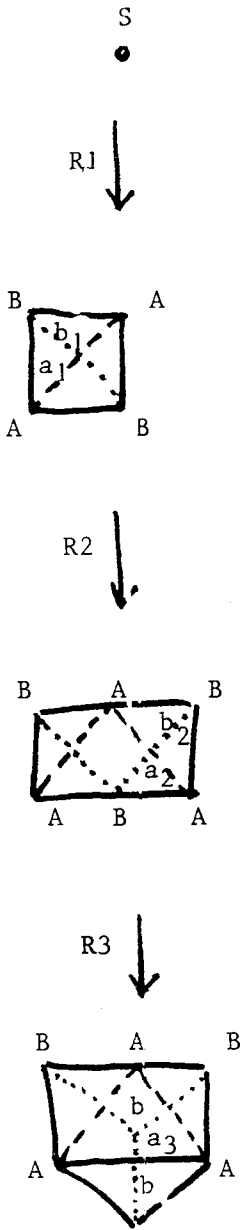
Si en appliquant la règle 1 on crée les arêtes a_n et b_n , le graphe produit est le graphe G_{n-1} .

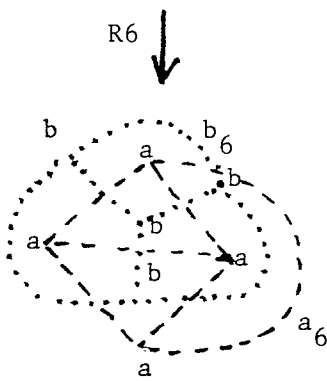
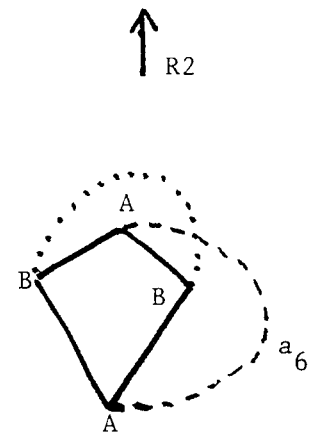
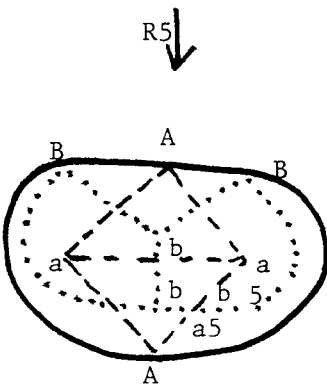
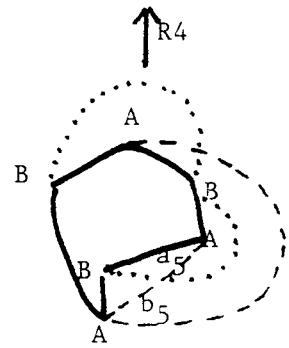
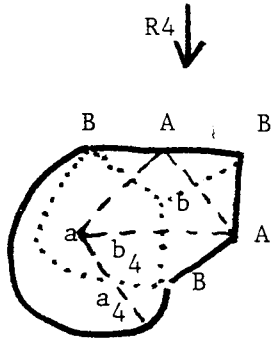


En appliquant à ce graphe la règle $f(r_{n-1})$ on produit le graphe $\overline{G_{n-2}}$. On applique successivement les règles $f(r_{n-2}), \dots, f(r_2)$. On obtient le graphe $\overline{G_1}$. On applique enfin à ce graphe $\overline{G_1}$ la règle 6 en créant les arêtes a_1 et b_1 , le graphe produit est le graphe G_n . Donc le graphe G est indirectement produit en appliquant successivement les règles de la dérivation $(f(r_n), f(r_{n-1}), \dots, f(r_2), f(r_1))$ appelée dérivation symétrique de la dérivation D .

Cette dérivation vérifie comme nous le venons de le démontrer les trois propriétés énoncées précédemment.

Exemple d'une dérivation et de sa symétrique (La première dérivation est écrite de haut en bas, la dérivation symétrique de bas en haut).





S

4.3.2. ALGORITHME D'ANALYSE

Etant donné qu'il existe une dérivation symétrique, on peut à la fois combiner un algorithme d'analyse ascendante et un algorithme d'analyse descendante.

Il est même impossible de dire si l'algorithme est ascendant ou descendant.

En effet si on considère que l'algorithme est descendant, partant du graphe initial \mathcal{S} pour arriver au graphe G en suivant la dérivation D , on peut considérer que cet algorithme part du graphe G pour remonter au graphe initial \mathcal{S} en suivant en sens inverse la dérivation symétrique D' .

Tout dépend de la façon dont on interprète les différentes étapes de l'algorithme. Dans la suite nous décrivons l'algorithme à la façon d'un algorithme descendant, mais on aurait aussi bien pu prendre la convention opposée

Grâce à cette intéressante propriété on peut cumuler les avantages des algorithmes ascendants et descendants tout en minimisant les inconvénients.

Le choix de la règle à appliquer à chaque étape ainsi que le choix du sous-graphe de partie gauche n'est cependant pas unique. Afin de pouvoir envisager toutes les solutions possibles on utilise la technique classique du retour en arrière (back-track).

PRINCIPE DE L'ALGORITHME

Partant du graphe initial, on veut construire le graphe d'analyse, en essayant systématiquement d'appliquer toutes les règles de la grammaire, en appliquant ces règles partout où cela est possible.

Lorsque les conditions d'application d'une règle sont réalisées, on n'applique cette nouvelle règle que si l'arête créée appartient effecti-

vement au graphe à analyser. Le sous-graphe, limité aux sommets étiquetés par "a" ou "A" et aux arêtes les reliant, du graphe construit au cours de l'analyse est à tout moment également un sous-graphe du graphe à analyser.

Si après avoir fait systématiquement tous les essais possibles on ne peut reconstruire le graphe, c'est que le graphe ne peut être produit par la grammaire, donc ce n'est pas un graphe planaire sans isthmes.

PREMIERE EVALUATION

Soit G le graphe à analyser, G ayant n sommets et m arêtes.

La longueur de la dérivation permettant de produire G est égale à m.

Chaque fois que l'on applique l'une des six règles, on construit une nouvelle arête. Avant d'appliquer la ième règle on a d'une part le choix entre les six règles, d'autre part le choix entre (m-i+1) arêtes à construire.

Le nombre d'essais possibles est donc borné par $6(m-i+1)$. Pour l'ensemble de l'analyse le nombre d'essais est formé par

$$\prod_{i=1}^{i=m} 6(m-i+1) = 6^m m!$$

Cette borne est très grande et l'on voit qu'un tel algorithme, bien que facile à programmer serait inutilisable dès que le nombre d'arêtes est supérieur à quelques unités. (Pour m=6 la borne est égale à 33.592.320)

Mais comme nous allons le voir, en étudiant les règles de la grammaire, et surtout les conséquences de leur application, sur la suite de la dérivation, on peut considérablement améliorer l'algorithme et ainsi limiter le nombre des essais.

4.3.3. AMELIORATION DE L'ALGORITHME

On étudie d'abord les règles 1 et 6 et on en déduit quelques améliorations concernant le début et la fin de l'analyse.

On étudie ensuite les règles 3,4 et 5 et leurs conséquences sur la suite de la dérivation.

Enfin on étudie la règle 2 qui se révèle être celle dont l'application a plus de conséquences sur la suite de l'analyse.

REGLE 1

La règle 1 est la première règle à appliquer et elle ne peut s'appliquer qu'en début de la dérivation.

L'arête créée en appliquant la première règle doit être choisie parmi les m arêtes du graphe.

En fait on démontre que l'on peut choisir arbitrairement l'une quelconque des m arêtes, sans avoir ensuite à essayer les $m-1$ autres.

En effet il suffit de se référer à la démonstration du théorème montrant que la grammaire de Montanari, produisait les graphes planaires sans isthmes

Le principe de la démonstration consistait à essayer de recopier un graphe planaire sans isthmes en appliquant les règles de la grammaire. Pour cela on commençait par recopier une arête du pourtour de la face extérieure, en appliquant la règle 1. Partant de cette arête, on montrait ensuite que si le graphe était planaire sans isthmes, on pouvait recopier tout le graphe en appliquant les règles de la grammaire.

Or une des propriétés des graphes planaires est que l'on peut toujours dessiner le graphe de façon à ce que l'une quelconque des faces soit la face extérieure.

On voit donc que l'on peut choisir n'importe quelle arête du graphe

comme première arête à recopier.

REMARQUE.

Le choix de cette première arête n'est pas aussi arbitraire que cela du point de vue de la performance de l'algorithme.

En effet, c'est autour de cette arête que sera recopié le graphe, et les prochaines arêtes à recopier seront les arêtes connexes. On a donc intérêt à limiter le nombre d'essais entre ces arêtes, **et pour cela choisir** comme première arête, une arête dont l'un au moins des sommets extrémité, ait un degré minimum.

REGLE 6

Cette règle ne peut être utilisée qu'une fois dans la dérivation et c'est la dernière règle.

Par conséquent, tant qu'il reste au moins deux arêtes ou plus à recopier, on n'a pas à essayer d'appliquer la règle 6, et s'il ne reste qu'une seule arête à recopier, on ne doit essayer d'appliquer que la règle 6.

REGLES 3,4,5.

On va montrer que si on a la possibilité d'appliquer l'une des règles 3,4,5, on peut le faire sans que cela ait de conséquences pour la suite de la dérivation.

On précise d'abord ce que l'on entend par "ne pas avoir de conséquences pour la suite de la dérivation".

Soit $D : (r_1, \dots, r_i, \dots, r_n)$ une dérivation permettant de produire un graphe G .

Alors, on dira que $D_i = (r_1, \dots, r_i)$ est un début de dérivation correcte pour G . L'application de r'_{i+1} n'a pas de conséquence pour la suite de la

dérivation, si $D'_{i+1} = (r_1, \dots, r_i, r'_{i+1})$ est un début de dérivation correcte, autrement dit s'il existe r'_{i+2}, \dots, r'_m tels que $D' = (r_1, \dots, r_i, r'_{i+1}, \dots, r'_m)$ permette de produire G .

On rappelle que G_i est le graphe obtenu après application des règles r_1, r_i et que A_i est le sous-graphe de G_i restreint aux sommets étiquetés par "A" et "a" et aux arêtes les reliant.

Si r_{i+1} est l'une des règles 3, 4 ou 5 et si A_{i+1} est un sous-graphe de G qui a les propriétés suivantes : les sommets étiquetés par "a" ont le même degré que les sommets correspondants de G , les sommets étiquetés par "A" ayant un degré strictement inférieur que celui des sommets correspondants alors $D_{i+1} = (r_1, \dots, r_i, r_{i+1})$ est un début de dérivation correcte.

Ce résultat est d'abord dû au fait que lorsque l'on recopie l'arête produite par la règle r_{i+1} , le graphe reste planaire et sans isthme, ainsi que nous l'avons vu dans la démonstration du théorème montrant que la grammaire de Montanari produit les graphes planaires sans isthmes.

D'autre part, cela tient au fait que contrairement à ce qui se passe lorsque l'on peut appliquer la règle 2, il n'y a pas d'ambiguïté sur le choix de l'arête à recopier.

REGLE 3

Le sommet du graphe de partie gauche étiqueté par "A" a dans A_i un degré égal à celui du sommet correspondant de G , moins un.

Il ne reste donc qu'une seule arête connectée à ce sommet et non encore recopiée. Il suffit alors de vérifier que l'autre extrémité n'est pas un sommet de G_i .

REGLE 4 et REGLE 5

On doit recopier une arête entre les deux sommets étiquetés par "A" du sous-graphe de partie gauche.

S'il y a plusieurs arêtes, entre ces deux sommets, qui n'ont pas été recopiées (arêtes multiples) le choix de l'une quelconque de ces arêtes, ne pose aucun problème car elles jouent toutes le même rôle, et on pourra successivement les recopier en appliquant la règle 4 pour les premières, la règle 4 ou la règle 5 pour la dernière.

S'il n'y a qu'une seule arête, c'est évidemment celle-là qu'il faudra recopier.

REMARQUE.

Lorsqu'on a la possibilité d'appliquer indifféremment la règle 3, la règle 4 ou la règle 5 (l'application de chaque règle concernant évidemment une arête différente), pour améliorer la performance de l'algorithme, il convient d'appliquer d'abord la règle 5.

En effet, pour chercher si l'on peut appliquer une règle, on utilise des considérations sur les degrés des sommets renseignés par un symbole non terminal, et sur l'existence ou non, d'arêtes entre ces sommets.

Or la règle 5 diminue de deux le nombre de ces sommets, ce qui diminue, par la suite, d'autant le nombre de recherche à effectuer.

REGLE 2

Les seules possibilités de choix qu'il reste, se situent au niveau de l'application de la règle 2.

En effet le sommet du graphe de partie gauche étiqueté par "A", est l'extrémité de plusieurs arêtes qui n'ont pas encore été recopiées et dont l'autre extrémité n'est, également, pas encore recopiée.

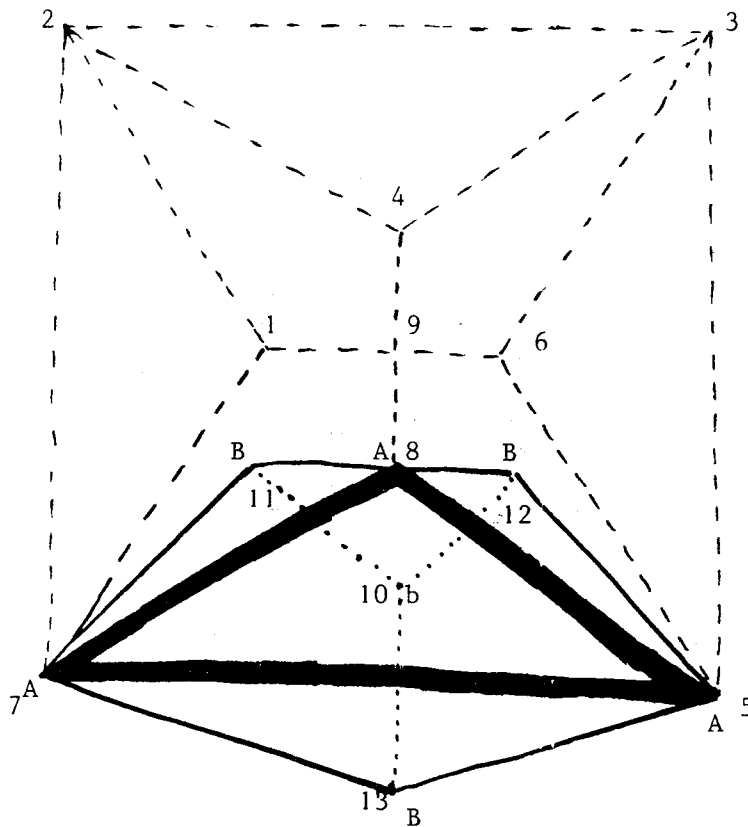
L'arête choisie appartiendra au pourtour de la face représentée par le sommet étiqueté par "B" dans le graphe de partie gauche.

Donc à moins de connaître à l'avance quelles sont les faces du graphe, il n'est pas possible de déterminer l'arête à recopier. On devra donc faire autant d'essais qu'il y a d'arêtes non encore recopiées dont le sommet considéré est l'une des extrémités.

(Si l'on connaît quelles sont les faces du graphe, c'est que l'on sait que le graphe est planaire. Dans ce cas le problème de l'analyse ne se pose pas du tout de la même façon et l'on peut facilement écrire un algorithme dont la durée de son exécution est directement proportionnelle au nombre d'arêtes du graphe.)

EXEMPLE

Prenons le cas du graphe suivant ou l'on a surchargé les arêtes déjà recopiées



On peut appliquer la règle 2 en prenant comme graphe de partie gauche le graphe



On peut alors recopier soit l'arête (7,1) soit l'arête (7,2) et on ne peut savoir à priori laquelle de ces deux arêtes appartient au pourtour de la face représentée par le sommet 11 et auquel appartient déjà l'arête (7,8).

REMARQUES

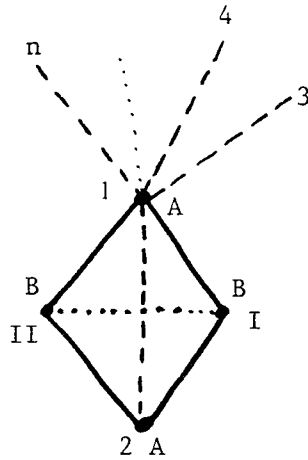
1)

Si après avoir essayé de recopier toutes les arêtes dont un sommet étiqueté par "A" est l'une des extrémités, on n'a pas trouvé de bonne dérivation, il est inutile d'essayer d'appliquer la règle 2 (aussi bien d'ailleurs que les autres règles) à un autre sous-graphe. En effet, si l'on pouvait appliquer la règle 2, on était dans les conditions 1,2a,3a du lemme et comme on l'a vu dans la démonstration, si le graphe était planaire sans isthmes, il était impossible de le recopier.

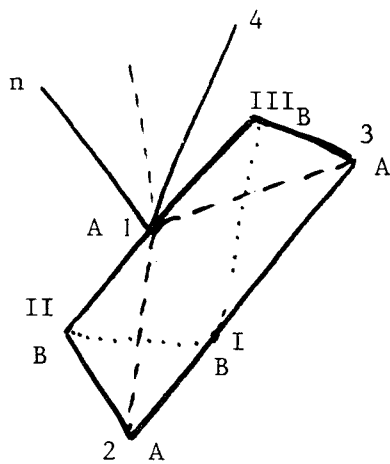
2)

Dans le cas où l'on a recopié qu'une seule arête donc après avoir appliqué la règle 1), si on ne peut appliquer la règle 2 (ce qui est toujours le cas si le graphe n'a pas d'arêtes multiples), alors il n'est pas nécessaire d'essayer toutes les arêtes. En effet, il y a deux faces de part et d'autre de la première arête et il y a pour chaque face une arête connectée à l'une des extrémités de la première arête, et appartenant au pourtour de la face. A ce niveau de l'analyse, les deux faces ne sont pas différenciées, et chaque arête peut donc être indifféremment associée avec

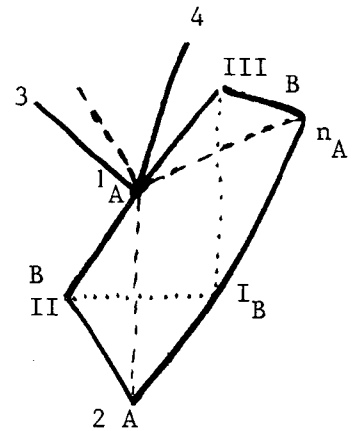
l'une ou l'autre face. Le nombre d'essais à faire est alors égal au nombre d'arêtes libres moins une.



Si les arêtes (1,2) et (1,3) appartiennent au pourtour de la même face et que (1,2), (1,n) appartiennent au pourtour d'une autre face, on peut indifféremment appliquer la règle 2 ainsi :



ou bien

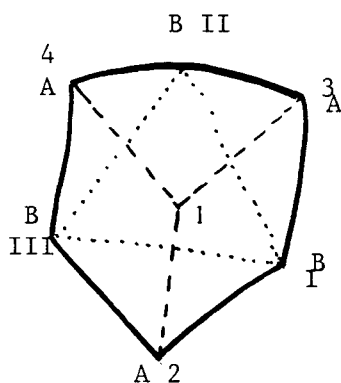


3)

Comme on l'a vu, on peut appliquer la règle 2 à n'importe quel sous-graphe $A \text{ --- } B$ du graphe produit. Pour diminuer le nombre d'essais, on prendra donc le sommet étiqueté par "A" où le nombre d'arêtes libres est minimum. (Ce minimum n'est jamais inférieur à 2, car s'il était égal à 1, on pourrait appliquer la règle 4).

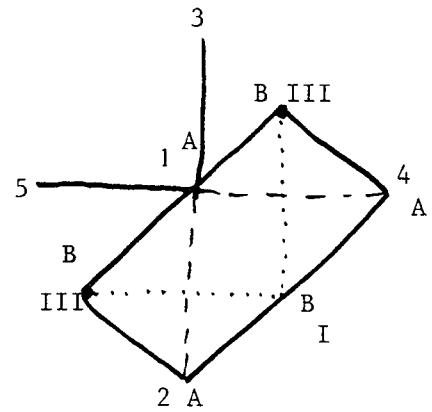
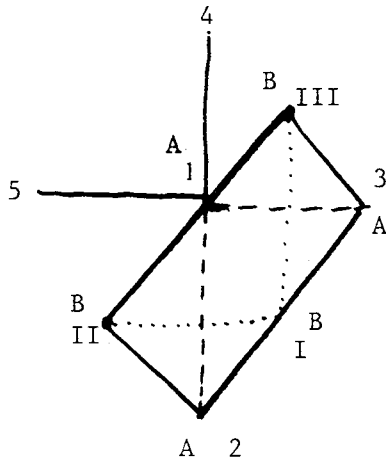
Dans le cas de la remarque précédente, le premier sommet choisi est le sommet ayant dans le graphe le degré minimum. Or pour les graphes planaires, ce degré est au plus égal à 5, donc le nombre de choix à faire pour appliquer la première fois la règle est au plus de 3.

Si le degré minimum du graphe est égal à 3, il n'y a aucun choix



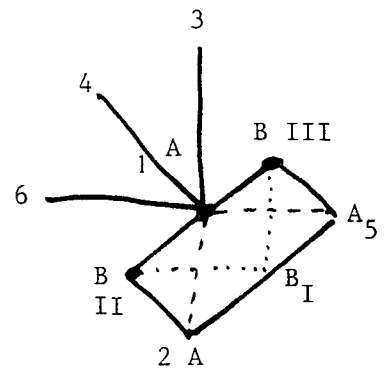
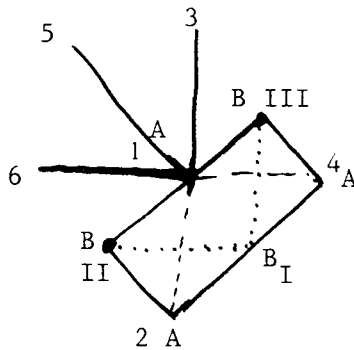
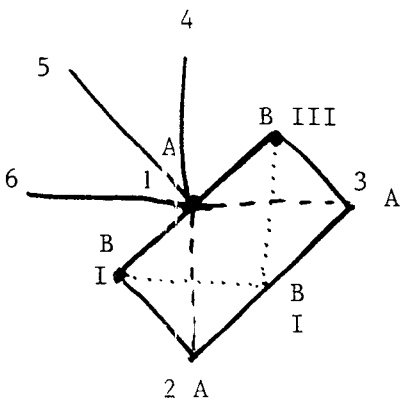
On applique successivement la règle 1 qui recopie (1,2), puis la règle 2 qui recopie (1,3), enfin la règle 3 qui recopie (1,4).

Si le degré minimum est égal à 4, il y a au plus deux choix (dans la mesure où le premier choix s'est avéré mauvais).



On a appliqué d'abord la règle 1 en recopiant (1,2), puis la règle 2 en recopiant soit (1,3) soit (1,4).

Enfin si le degré minimum est égal à 5, il y a au plus trois choix



On applique d'abord la règle 1 en recopiant (1,2), puis la règle 2 en recopiant soit (1,3), puis si ce choix s'avère mauvais (1,4), enfin si on ne parvient toujours pas à trouver une bonne dérivation (1,5). Si c'est encore sans succès, il est inutile d'essayer (1,6), le graphe ne peut être produit par la grammaire.

ALGORITHME AMELIORE.

On décrit pas à pas l'algorithme, en tenant compte de toutes les améliorations précédentes.

La technique du retour en arrière est obtenue grâce à une procédure réursive ANALYSE écrite en pseudo-algol.

On utilise également la procédure $P\emptyset S(R_i)$, procédure booléenne dont la valeur est *vraie* si on a la possibilité d'appliquer la règle R_i , et la procédure APPLI (R_i) qui permet d'appliquer effectivement la règle R_i .

NAR est le nombre d'arêtes que l'on a déjà recopiées

NA est le nombre d'arêtes du graphe.

Le programme d'analyse est le suivant :

début

APPLI (R1);

NAR ← 1;

si ANALYSE (NAR) alors écrire " le graphe est planaire"

sinon écrire " le graphe n'est pas planaire ";

fin

La procédure ANALYSE est la suivante :

logique procédure réursive ANALYSE (NAR)

début

tant que (NAR < NA-1) et ($P\emptyset S(R_5)$ ou $P\emptyset S(R_3)$ ou $P\emptyset S(R_4)$) faire

début

si $P\emptyset S(R_5)$ alors APPLI(R5)

sinon si $P\emptyset S(R_3)$ alors APPLI(R3)

sinon si $P\emptyset S(R_4)$ alors APPLI(R4);

NAR \leftarrow NAR + 1;

fin;

si (NAR < NA-1) alors

début

ARRETES (NAR,A,N);

IMPASSE \leftarrow vrai ;

tant que I \leq N et IMPASSE faire

si PØS(R2,A(I)) alors

début

APPLI(R2)

NAR \leftarrow NAR + 1;

si ANALYSE (NAR) alors IMPASSE \leftarrow faux;

fin;

si I > N alors ANALYSE \leftarrow faux ;

fin;

sinon si PØS(R6) alors ;

début

APPLI(R6)

ANALYSE \leftarrow vrai;

fin;

sinon ANALYSE \leftarrow faux;

fin;

La procédure ARRETES (NAR,A,N) recherche parmi tous les sommets étiquetés par "A" celui d'où part le moins d'arêtes non encore recopiées.

A est l'ensemble de ces arêtes et N leur nombre. Si NAR est égal à 1

ce nombre est diminué de un.

$A(I)$ représente l'une de ces arêtes et la procédure $P\emptyset S(R2, A(I))$ indique si l'on peut appliquer la règle 2 en recopiant l'arête $A(I)$.

4.3.4. PROGRAMMATION.

La donnée du programme d'analyse est évidemment le graphe à analyser. La représentation interne du graphe peut être soit une matrice de connectivité, soit un tableau des liaisons, soit, si le langage de programmation le permet, une structure de liste.

Le graphe produit peut être décomposé en trois sous-graphes :

1°/ Le sous-graphe formé par les sommets étiquetés par les symboles "A" et "a", ainsi que les arêtes les reliant, le sous-graphe A_i . C'est la partie du graphe à analyser déjà recopiée.

2°/ Le sous-graphe formé par les sommets étiquetés par les symboles "B" et "b", ainsi que les arêtes les reliant, le sous-graphe B_i . C'est la partie correspondant au graphe dual.

3°/ Le sous-graphe formé des sommets étiquetés par "A" et "B" et des arêtes ayant une extrémité "A" et une extrémité "B", le sous-graphe C_i , ou cycle des symboles non-terminaux.

Ces trois sous-graphes peuvent très utilement, être représentés indépendamment les uns des autres.

Le premier sous-graphe est un sous-graphe du graphe à analyser, on peut donc, pour le représenter, utiliser la représentation interne du graphe à analyser, soit en marquant les arêtes déjà recopiées, soit en les supprimant au fur et à mesure. Ceci est possible car on ne s'intéresse qu'aux arêtes qui n'ont pas encore été recopiées, d'autre part si on a gardé une trace des règles que l'on a appliquées, on peut facilement reconstituer la

partie du graphe déjà recopié.

Pour ne pas avoir à les recalculer chaque fois, il est utile de garder en mémoire les degrés des sommets du graphe à analyser ou de la partie du graphe qui n'a pas encore été recopiée. Dans le cas d'une représentation sous forme d'une matrice de connectivité, on peut par exemple utiliser la diagonale de la matrice.

Le deuxième sous-graphe n'est pas indispensable pour l'analyse. On le construit donc que si l'on veut disposer du graphe dual.

Le troisième sous-graphe est un cycle. On peut donc le représenter sous la forme d'un tableau linéaire.

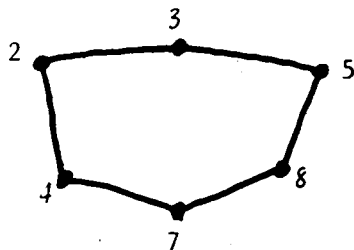
C'est une partie de ce cycle qui constitue le graphe de partie gauche des règles 2,3,4 et 5. (La totalité pour la règle 6).

On peut donc caractériser de façon non ambiguë quel est le graphe de la partie gauche de la règle en donnant l'indice du premier sommet de ce graphe dans le tableau à condition de toujours parcourir le cycle (donc le tableau) dans le même sens.

Les sommets du cycle sont en alternance des sommets étiquetés par "A" et des sommets étiquetés par "B". En prenant comme premier sommet, un sommet étiqueté par A, tous les sommets étiquetés par "A" auront un indice impair, et ceux étiquetés par "B" un indice pair.

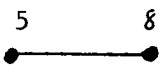
EXEMPLE.

Le cycle des symboles non terminaux suivant



pourra être représenté par

2	3	5	8	7	4
1	2	3	4	5	6

Et si  est le graphe de partie gauche de la règle 2, on dira qu'on applique la règle 2 en 3.

Il ne reste plus qu'à décrire les procédures APPLI et PØS précédemment définies.

La procédure APPLI (Ri) permet d'effectuer les modifications nécessaires sur les différents sous-graphes et les variables du programme lorsque l'on applique la règle Ri.

Cela consiste essentiellement à

- ajouter une arête au premier sous-graphe, c'est-à-dire à en supprimer une dans l'ensemble des arêtes non encore recopiées
- ajouter également une arête au second sous-graphe
- modifier le troisième sous-graphe suivant la règle :

règle 1 : création des 4 sommets

règle 2 : insertion de deux nouveaux sommets

règle 3 & 4 : remplacement d'un sommet par un autre

règle 5 : suppression de deux sommets

règle 6 : suppression de tout le cycle

La procédure PØS(Ri) recherche si il existe une partie du cycle des symboles non terminaux qui peut-être le graphe de partie gauche de la règle Ri. Ensuite on regarde si on peut effectivement appliquer cette règle :

Les conditions sont :

- pour la règle 2

Il existe une arête non encore recopiée, dont une extrémité est le sommet étiqueté par "A" du graphe de partie gauche, et l'autre sommet n'est pas encore recopié.

- pour la règle 3

il existe une arête non encore recopiée entre les deux sommets du graphe de partie gauche étiquetés par "A",

ces deux sommets étant les extrémités d'autres arêtes non encore recopiées.

- pour la règle 4

Le sommet du graphe de partie gauche étiqueté par "A" n'est l'extrémité que d'une seule arête non encore recopiée. L'autre extrémité de cette arête n'est pas recopiée elle aussi.

- pour la règle 5

Il existe une arête non encore recopiée entre les deux sommets du graphe de partie gauche renseignés par "A". Un seul de ces sommets est l'extrémité d'autres arêtes non encore recopiées.

4.3.5. EVALUATION DE L'ALGORITHME ET INTERPRETATION DES RESULTATS.

Il est très délicat d'évaluer un tel algorithme. En effet le nombre d'opérations à effectuer pour analyser un graphe, dépend non seulement du graphe donné, mais aussi de la façon dont ce graphe est représenté.

En effet, lorsque l'on veut appliquer la règle 2, on a le choix entre plusieurs arêtes, et une seule de ces arêtes permettra d'obtenir une bonne dérivation. Comme rien, à priori, ne distingue ces arêtes, c'est donc la façon dont le graphe est représenté qui sera déterminante.

Le nombre de choix, lui, dépend du graphe à analyser. Il est fonction d'une part du nombre de fois où l'on ne peut appliquer que la règle 2, à l'exclusion de toute autre règle, et d'autre part, chaque fois que l'on doit appliquer cette règle, du nombre d'arêtes entre lesquelles on doit choisir

1°/ nombre d'occurrences de la règle 2 dans une dérivation.
Soit m le nombre d'arêtes d'un graphe pouvant être produit par la grammaire.

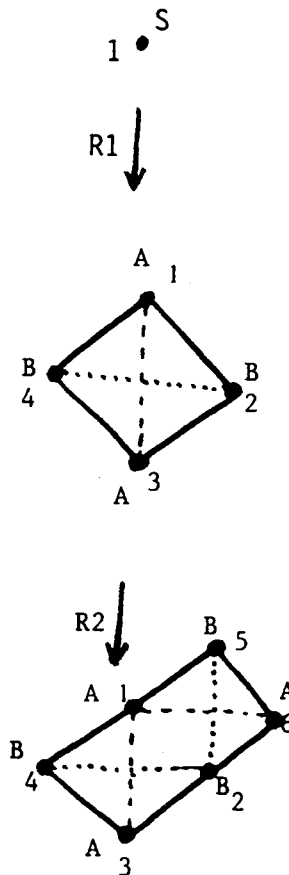
On sait que la longueur de la dérivation est égale au nombre d'arêtes du

graphe. La règle 2 ne peut se trouver ni en début de dérivation, ni en fin.

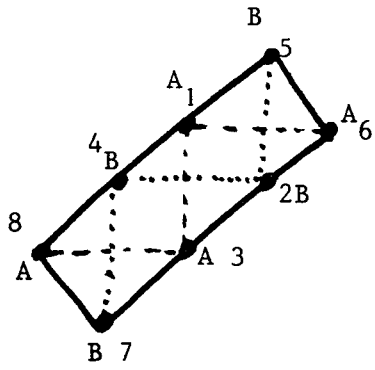
D'autre part, lorsque l'on applique la règle 2, le nombre de sommets du cycle des symboles non terminaux est augmenté de 2, et il n'y a que l'application de la règle 5 qui permet de diminuer, également de 2, le nombre de ces sommets. Au début ce nombre est nul, la règle 1 en crée quatre. Avant d'appliquer la dernière règle, la règle 6, il doit également rester quatre sommets, comme seules règles 2 et 5 modifient le nombre de ces sommets, on en déduit que dans toute dérivation, on trouve autant de fois la règle 2 que la règle 5.

Le nombre maximum d'occurrences de la règle 2 dans une dérivation est donc $\frac{n-2}{2}$.

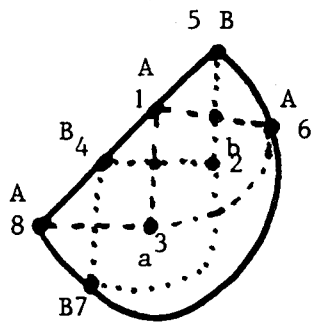
EXEMPLE DE DERIVATION OU LA BORNE EST ATTEINTE :



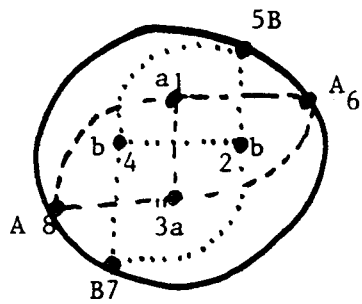
R2 ↓

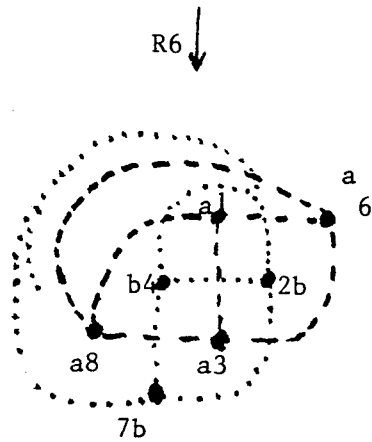


R5 ↓



R5 ↓





Dans le graphe produit, il y a $m = 6$ arêtes et on a appliqué $\frac{m-2}{2} = 2$ fois la règle 2.

Il existe d'autres dérivations, permettant de produire le même graphe et pour lesquelles le nombre d'occurrences de la règle 2 n'est pas égale à $(m-2)/2$, par exemple la dérivation $(R1, R2, R3, R4, R5, R6)$

On remarquera que dans l'analyse, on aurait d'abord trouvé cette dérivation, car on essaie d'abord les règles R3 et R4 avant d'essayer d'appliquer la règle R2.

Pour les graphes planaires sans isthmes, le maximum d'occurrences de la règle 2 dans une dérivation n'est atteint que dans certaines dérivations de certains graphes. Le nombre d'occurrences sera donc pratiquement, toujours inférieur à $\frac{m-2}{2}$.

Si on analyse un graphe quelconque, il se peut que l'on essaie plus de $\frac{m-2}{2}$ fois d'appliquer la règle 2, dans ce cas il est inutile de poursuivre plus loin l'analyse, on sait que le graphe ne peut être produit par la grammaire, donc que le graphe n'est pas planaire sans isthmes.

2°/ Nombre d'arêtes auxquelles on peut appliquer la règle 2.

Le nombre d'arêtes auxquelles on peut appliquer la règle 2 est au minimum de 2, car sinon on pourrait appliquer la règle 3. Au maximum, il est égal au degré du sommet renseigné par A moins 1.

Il y a au minimum deux sommets renseignés par A dans le cycle des symboles non terminaux. Soit d le degré maximum de tous les sommets du graphe (exception faite du sommet de degré maximum) où l'on prend le degré moins un, car on y a déjà appliqué la règle 1).

Le nombre d'arêtes auxquelles on peut appliquer la règle 2 est au maximum d-1, donc le nombre maximum de choix possibles est

$$(d-1)^{\lfloor (m-2)/2 \rfloor}$$

Il est évident que cette borne n'est presque jamais atteinte. On peut donc par exemple remarquer que si l'on peut appliquer plusieurs fois de suite la règle 2 avec le même sommet étiqueté par "A" dans le graphe de partie gauche, le nombre de choix diminue de 1 à chaque fois.

D'autre part, le nombre de fois où l'on ne peut appliquer que la règle 2, n'est qu'exceptionnellement égale à $(m-2)/2$, et si le début de la dérivation est mauvais, il ne sera pas toujours nécessaire d'essayer les m règles de la dérivation, le nombre de choix sera encore diminué.

CHAPÎTRE

V

CHAPITRE V

APPLICATIONS DES GRAMMAIRES DE GRAPHS

5.1.. REMARQUE PRELIMINAIRE.

Dans ce chapitre on essaie d'envisager les différentes applications possibles des grammaires de graphes.

Si certaines applications sont immédiates, d'autres sont beaucoup plus délicates à mettre en oeuvre, voir même hypothétiques.

Quant à l'intérêt de ces applications, il est également très divers, il réside bien souvent, beaucoup plus dans l'originalité apportée à la résolution d'un problème, que dans l'apport d'une solution efficace !

5.2. RECONNAISSANCE D'UN GRAPHE

5.2.1. RECONNAISSANCE DE FORMES.

Comme nous l'avons vu dans le premier chapitre, c'est pour résoudre des problèmes de reconnaissance de formes qu'ont été introduites les premières grammaires de graphes.

Les principales réalisations concernaient :

- la reconnaissance des caractères (NARISIMHAM [30-33])
- la reconnaissance des chromosomes (LEDLEY [26])
- l'analyse des photographies d'expériences en chambre à bulle (SCHAW [41-43]).

Dans ces applications, les grammaires de graphes étaient représentées par des grammaires de chaînes de type hors-contexte. L'analyse syntaxique était alors classique, la reconnaissance du caractère du vocabulaire terminal étant remplacé par la reconnaissance d'une primitive de la forme.

L'utilisation de grammaires de graphes plus générales permettrait de reconnaître des classes beaucoup plus vastes de formes. On se heurte alors à la difficulté de l'analyse syntaxique.

5.2.2. PLANEARITE.

Etant donnée une classe de graphes C , si il existe une grammaire G telle que le langage produit par G , L_G coïncide avec C , alors le problème de l'appartenance d'un graphe G à C se ramène à l'analyse syntaxique du graphe G selon la grammaire G .

En utilisant l'analyseur syntaxique décrit au chapitre précédent, on peut donc déterminer si un graphe quelconque est planaire sans isthme. (Etant donné qu'il est très aisé de décomposer un graphe avec isthmes en plusieurs graphes sans isthmes, on peut donc plus généralement déterminer si un graphe est planaire).

Cependant cette façon de procéder n'est pas, dans le cas général, la plus performante. Si n est le nombre de sommets du graphe, les meilleurs algorithmes de recherche de planéarité nécessitent un nombre d'opérations proportionnel à n^2 . En 1971 HOPCROFT et TARZAN ont publié un algorithme [23] où ce nombre était proportionnel à $n \log(n)$, puis en 1971 et 1973, ils ont publié deux algorithmes [50] où le nombre d'opérations est directement proportionnel à n .

5.3. PRODUCTION D'UN GRAPHE

5.3.1. PRODUCTION DE TYPES DE GRAPHES PARTICULIERS.

Le problème de la production d'un graphe (ou d'un dessin, ou d'une forme) d'un type particulier est le problème inverse de celui de la reconnaissance de formes.

Il suffit de connaître une grammaire de graphes G dont le langage engendré, L_G , soit exactement l'ensemble des graphes du type désiré.

Avec la grammaire de l'exemple donné au paragraphe 2.4.2. on peut produire n'importe quel schéma de molécules chimiques.

La notion de grammaire de graphe est une notion très générale (ou très peu contraignante si on préfère !) la construction de la grammaire désirée est donc en principe relativement aisée.

5.3.2. DESSIN DE GRAPHERS

Sachant produire un graphe d'un type donné il peut être intéressant d'avoir une représentation graphique (ou dessin) de ce graphe.

Pour cela à la production, donc au niveau de chaque règle de production, il faut associer un algorithme de représentation graphique, c'est-à-dire indiquer d'une part, l'emplacement des sommets les uns par rapport aux autres, et d'autre part la façon dont l'on doit dessiner les sommets et les arêtes.

Par exemple pour la grammaire donnée au paragraphe 2.4.2. si on veut obtenir le dessin du schéma d'une molécule chimique, la principale difficulté sera de définir l'emplacement des sommets [53]. En effet la grammaire ne donne aucun renseignement de nature topologique.

En revanche le dessin des sommets et des arêtes sera simple : symbole chimique pour le sommet, liaison simple, double ou triple pour les arêtes.

Avec la grammaire de MONTANARI, produisant les graphes planaires sans isthmes, on pourra obtenir un dessin planaire d'un graphe planaire.

Les algorithmes dans ce domaine sont fort nombreux [50.54.55], mais dans ce cas on peut s'imposer une contrainte supplémentaire, à savoir l'emplacement des sommets sur le plan est donné par avance (C'est une contrainte que l'on rencontre souvent dans la pratique, par exemple sur une plaque de circuit imprimé où l'emplacement des différents modules est imposé pour des raisons technologiques).

Le principe de l'algorithme est le suivant :

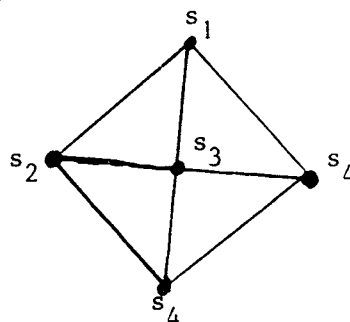
au fur et à mesure de l'application des règles de la dérivation on dessine le graphe ainsi :

le cycle des symboles non terminaux définit une surface à l'intérieur de laquelle se trouve la partie du graphe déjà dessiné. Les sommets du graphe non encore "créés" se trouvent tous à l'extérieur du cycle.

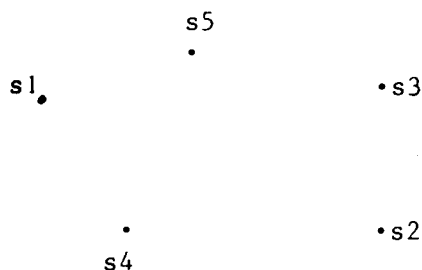
On vérifie que, lorsque l'on applique l'une des six règles de la grammaire, on peut toujours en modifiant le dessin du cycle des symboles non terminaux, dessiner la nouvelle arête de façon à ce qu'elle ne coupe aucune des arêtes déjà dessinées.

EXEMPLE

On veut dessiner le graphe



Les sommets étant disposés ainsi

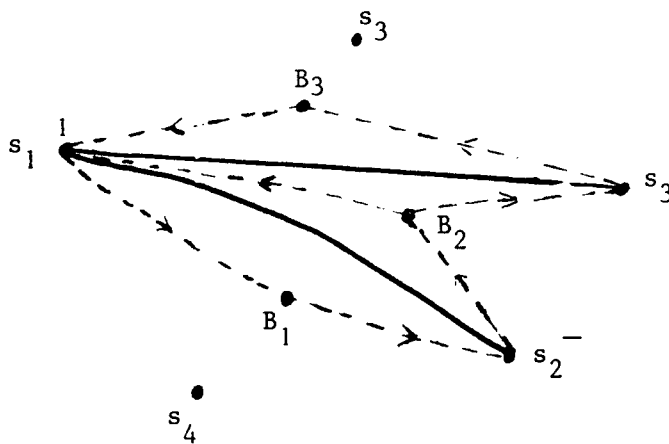


Une dérivation permettant de produire le graphe est la suivante

R1, (R2,4), (R3,3), (R4,6), (R3,5), (R4,4), (R5,2), R6

On trace l'arête s_j, s_k à l'intérieur de la nouvelle surface précédemment définie et on modifie le cycle des symboles non terminaux en remplaçant la partie (B_i, s_j) par la nouvelle courbe.

EXEMPLE:



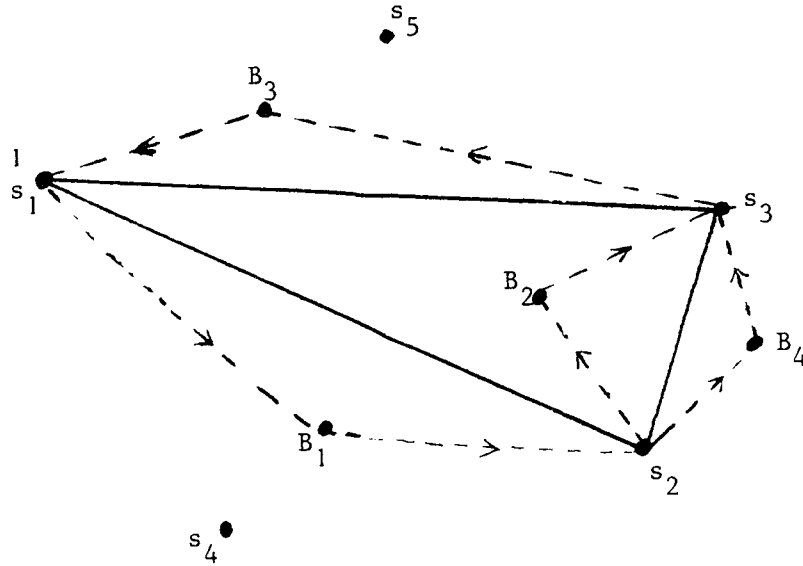
2°/ L'indice du premier sommet du graphe de partie gauche est impair. Si s_i et B_j sont les deux sommets du graphe de partie gauche on trace une courbe allant de s_i à B_j passant par s_k le nouveau sommet à créer. Le principe est le même que dans le cas précédent.

REGLE 3

Soient s_i, B_j et s_k les trois sommets. Il est toujours possible de tracer une courbe allant de s_i à s_k , à l'extérieur de la surface définie par le cycle des symboles non terminaux et telle que la surface définie par cette courbe et la partie du cycle (s_i, B_j) et (B_j, s_k) ne contienne aucun sommet non encore créé.

A l'intérieur de cette surface, on dessine l'arête s_i, s_k . On ajoute sur la courbe le sommet du graphe dual et dans le cycle des symboles non terminaux la partie $(s_i, B_j), (B_j, s_k)$ est remplacé par la nouvelle courbe.

EXEMPLE



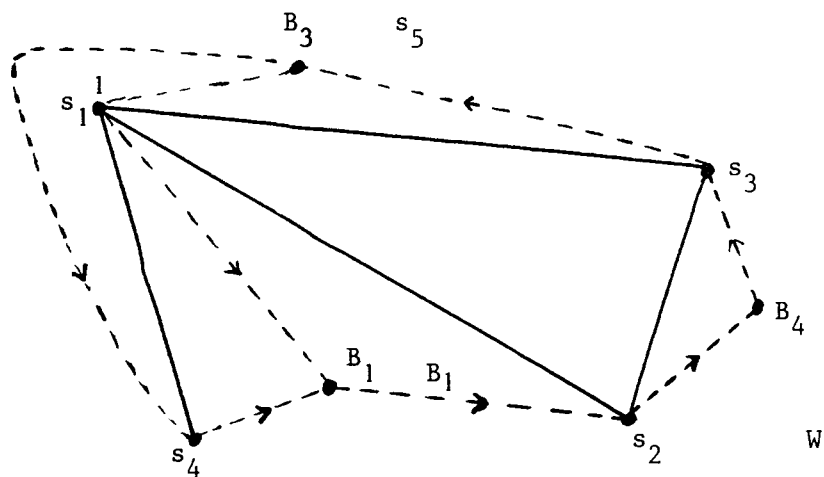
REGLE 4

Soient B_i, s_j, B_k les trois éléments du graphe de partie gauche, et soit s_1 le nouveau sommet à créer.

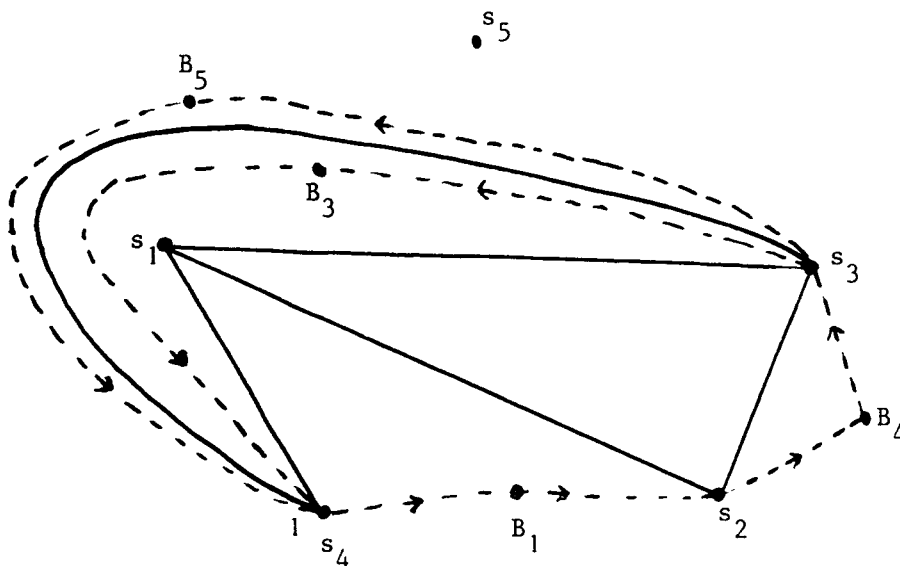
On peut toujours tracer une courbe allant de B_i à B_k passant par s_1 , à l'extérieur de la surface délimitée par le cycle des symboles non terminaux et telle que la surface comprise entre cette courbe et la partie du cycle des symboles non terminaux $(B_i, s_j), (s_j, B_k)$ ne contienne aucun sommet non encore créé.

On dessine l'arête s_j, s_1 à l'intérieur de cette surface, et dans le cycle des symboles non terminaux, on remplace la partie $(B_i, s_j)(s_j, B_k)$ par la nouvelle courbe.

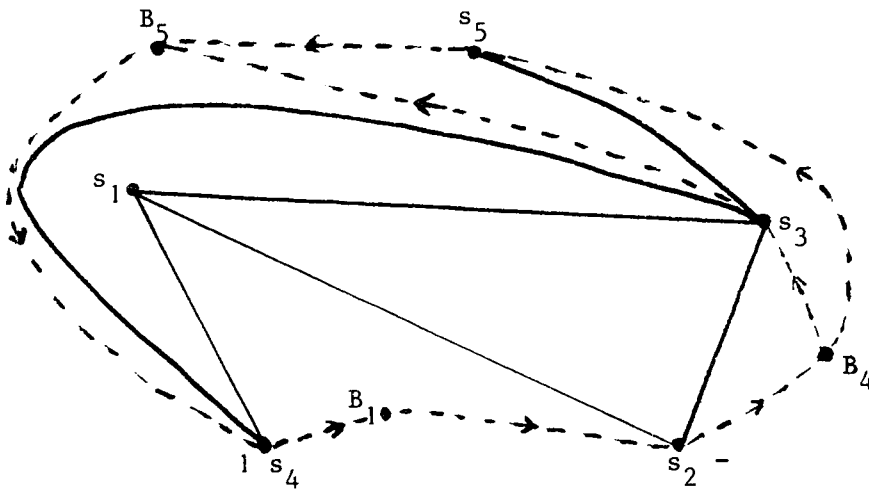
EXEMPLE



L'application de la règle (R3,5) donne



L'application de la règle (R4,4) donne



REGLE 5

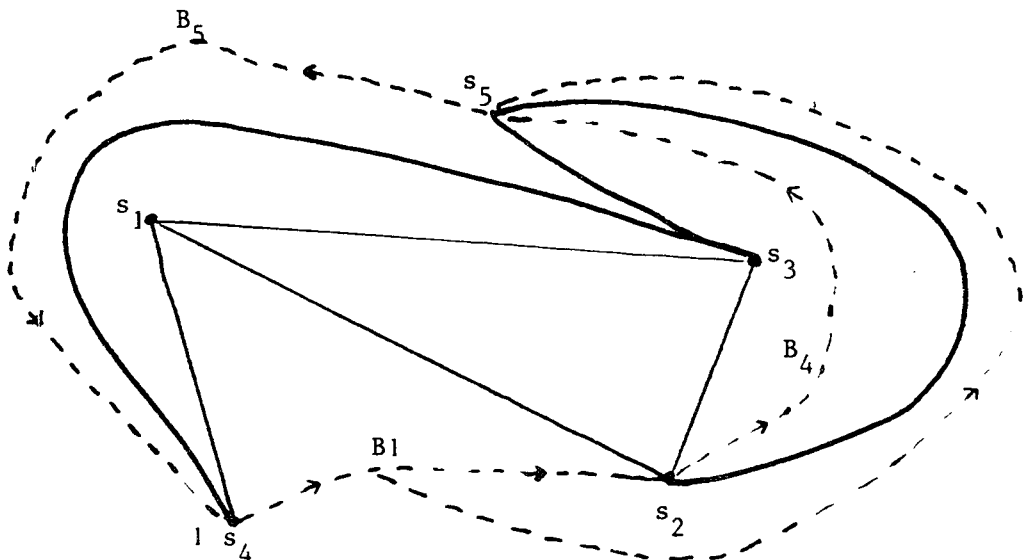
Deux cas peuvent se présenter

1°/ L'indice du premier sommet du graphe de partie gauche est pair

Soient B_i, s_j, B_k, s_l les quatre sommets du graphe de partie gauche. On peut toujours tracer à l'extérieur de la surface délimitée par le cycle des symboles non terminaux, une courbe allant de B_i à s_l de façon à ce que la surface comprise entre cette courbe et la partie du cycle $(B_i, s_j), (s_j, B_k), (B_k, s_l)$ ne contienne aucun sommet non encore créé.

On trace l'arête (s_j, s_l) à l'intérieur de cette surface. Dans le cycle des symboles terminaux, la nouvelle courbe remplace la partie $(B_i, s_j), (s_j, B_k), (B_k, s_l)$

EXEMPLE



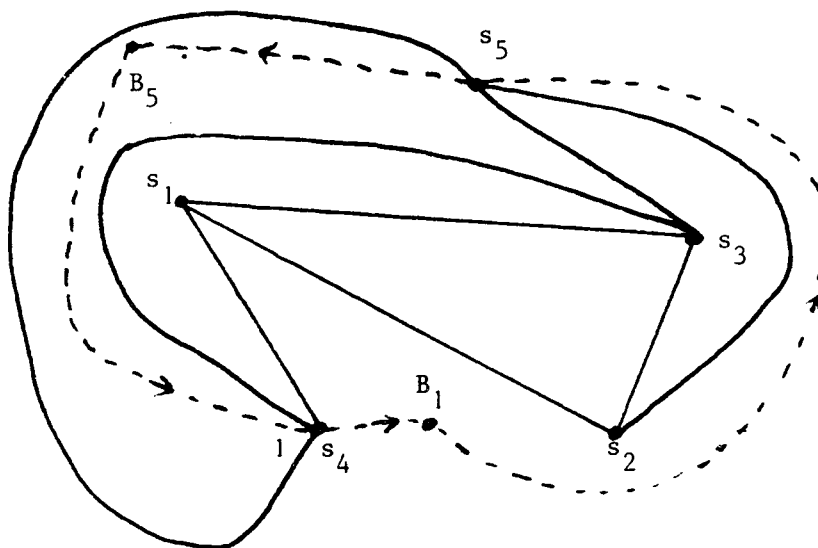
2°/ L'indice du premier moment du graphe de partie gauche est impair.

Si s_i, B_j, s_k, B_l les quatre sommets du graphe de partie gauche, comme précédemment on trace une courbe allant de s_i à B_l et à l'intérieur de la surface ainsi formée on dessine l'arête (s_i, s_k) .

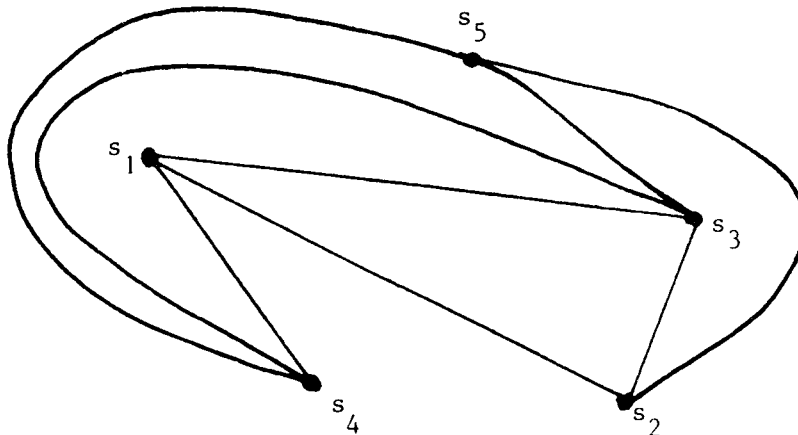
REGLE 6

La règle 6 est la dernière à appliquer. Il suffit alors de dessiner la dernière arête, à l'extérieur de la surface formée par le cycle des symboles non terminaux, et de supprimer ce cycle. On a bien un dessin planaire du graphe, les positions des sommets du graphe ayant été fixés par avance.

EXEMPLE



Le dessin définitif du graphe sera



REMARQUE

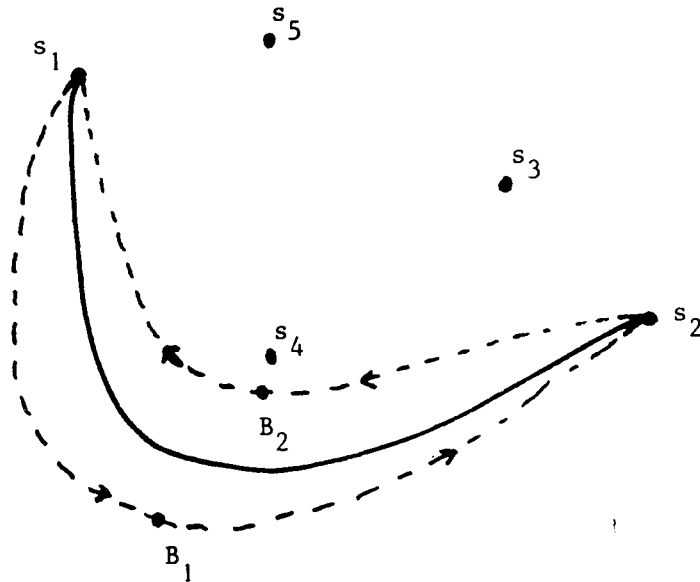
Pour un graphe donné et pour des emplacements de ses sommets donnés, il y a plusieurs façons de le dessiner avec cet algorithme.

Il peut y avoir plusieurs dérivations permettant de produire le même graphe et donc l'ordre dans lequel on dessine les arêtes peut être différent.

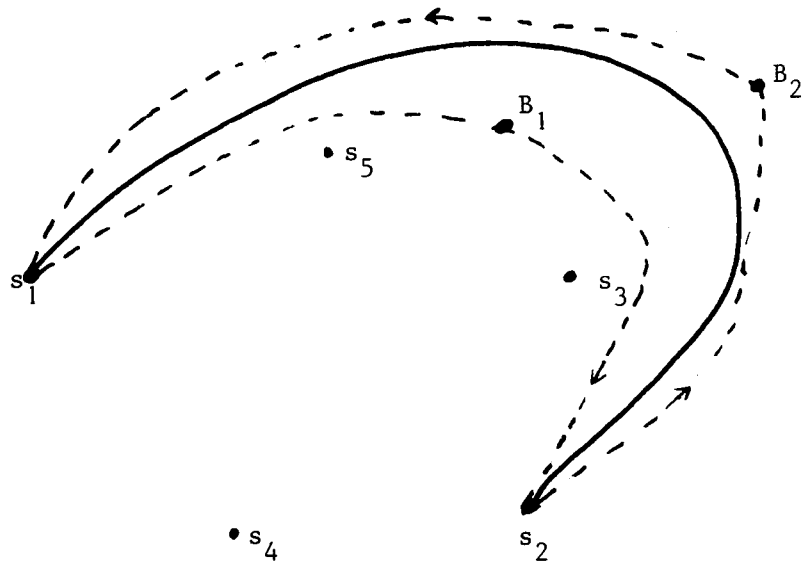
D'autre part pour une dérivation donnée, il y a plusieurs façons possibles de dessiner le cycle des symboles non terminaux, notamment au moment d'appliquer la première règle. La suite du dessin sera modifiée en conséquence

EXEMPLE.

Toujours pour le même graphe on aurait pu dessiner ainsi le premier cycle des symboles non terminaux et donc la première arête.



ou bien



etc.....

5.4 ISOMORPHISME

Soient deux graphes G_1 et G_2 renseignés par le même vocabulaire

$$G_1 = (S_1, A_1, f_{S_1}, f_{A_1})$$

$$G_2 = (S_2, A_2, f_{S_2}, f_{A_2})$$

On dit que G_1 et G_2 sont isomorphes si, et seulement si, il existe une application bijective ϕ de S_1 dans S_2

$$\phi : S_1 \longrightarrow S_2$$

telle que :

Pour tout $x_1, y_1 \in S_1$ et pour tout $x_2, y_2 \in S_2$

$$(x_1, y_1) \in A_1 \iff (\phi(x_1), \phi(y_1)) \in A_2$$

ou

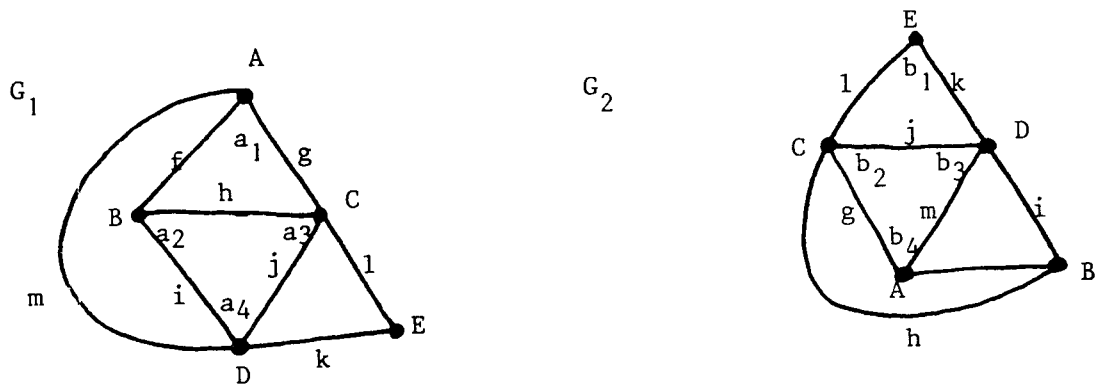
$$(x_2, y_2) \in A_2 \iff (\phi^{-1}(x_2), \phi^{-1}(y_2)) \in A_1$$

Pour les graphes renseignés, que pour tout $x_1, y_1 \in S_1$

$$f_{S_1}(x_1) = f_{S_2}(\phi(x_1))$$

si $(x_1, y_1) \in A_1$, $f_{A_1}(x_1, y_1) = f_{A_2}((\phi(x_1), \phi(y_1)))$

EXEMPLE



Les deux graphes sont isomorphes.

La bijection entre les deux ensembles de sommets est la suivante :

- $a_1 \leftrightarrow b_4$
- $a_2 \leftrightarrow b_5$
- $a_3 \leftrightarrow b_2$
- $a_4 \leftrightarrow b_3$
- $a_5 \leftrightarrow b_1$

Le problème de la recherche de l'isomorphisme de deux graphes, est un problème très important.

Par exemple, dans toute représentation d'un graphe, on établit implicitement un ordre sur l'ensemble des sommets ou des arêtes.

Si on utilise une représentation par une matrice de connectivité, l'ordre sur les sommets sera celui des lignes ou des colonnes associées à chaque sommet dans la matrice.

L'ordre des arêtes sera celui obtenu en parcourant la matrice ligne par ligne par exemple.

Pour les deux graphes précédents, on aurait les matrices

	a_1	a_2	a_3	a_4	a_5		b_1	b_2	b_3	b_4	b_5
a_1		1	1	1		b_2		1	1		
a_2	1		1	1		b_2	1		1	1	1
a_3	1	1		1	1	b_3	1	1		1	1
a_4	1	1	1		1	b_4		1	1		1
a_5			1	1		b_5		1	1	1	

Ces deux matrices représentent le même graphe. Dans la première l'ordre des sommets est a_1, a_2, a_3, a_4, a_5 ou b_4, b_5, b_2, b_3, b_1 .

Dans la seconde matrice l'ordre est a_5, a_3, a_4, a_1, a_2 ou b_1, b_2, b_3, b_4, b_5 .

Cet ordre sur les sommets ou sur les arêtes, ne dépend que de la façon dont on représente le graphe. Il est donc, en général, tout à fait aléatoire

Pour trouver si deux représentations de graphes, représentant effectivement le même graphe, on est amené à rechercher si les deux graphes représentés sont isomorphes.

Les algorithmes de recherche d'isomorphisme sont essentiellement combinatoires et bien peu sont performants [56].

Avec des grammaires de graphes, on peut concevoir des algorithmes procédant différemment.

Soit G une grammaire de graphe produisant une classe de graphes \mathcal{G} . Si

deux graphes G_1 et G_2 appartenant à \mathcal{C} sont tels qu'ils puissent être produit par une même dérivation de G , alors ils sont isomorphes.

En utilisant la grammaire de MONTANARI, on peut rechercher l'isomorphisme de deux graphes planaires sans isthmes.

Dans un premier temps, en utilisant l'algorithme décrit au chapitre 4, on fait l'analyse syntaxique de l'un des deux graphes. On obtient une dérivation.

Dans un deuxième temps, on essaie avec cette dérivation de produire le deuxième graphe.

Si (s_{i1}, s_{j1}) est la première arête produite par la première règle de la dérivation du premier graphe alors on devra essayer d'appliquer cette règle à toutes les arêtes (s_{i2}, s_{j2}) du deuxième graphe telle que le degré de s_{i2} soit égal au degré de s_{i1} et celui de s_{j2} à celui de s_{j1} .

On essaie d'appliquer ensuite les autres cycles de la dérivation. Ce n'est que pour appliquer la règle 2 (comme nous l'avons vu dans l'algorithme d'analyse), qu'il peut y avoir plusieurs choix possibles, on devra tous les envisager.

L'algorithme de recherche d'isomorphisme est alors le suivant:

D est le tableau dans lequel sont rangées les NL règles de la dérivation du premier graphe.

A est le tableau des NA arêtes du second graphe qui pourraient être produite par la première règle

I ← 1

tant que non DERIV(A(I)) faire I ← I + 1;

Si I < NP alors écrire "Les deux graphes sont isomorphes"

sinon écrire "Les deux graphes ne sont pas isomorphes"

logique procédure DERIV(A)


```
APPLI (R1,A);  
IM ← 1;  
DERIV ← SUI DERIV(IM);  
logique procédure recursive SUI DERIV (IM)  
tant que IM < NL et R(D(IM)) faire IM ← IM+1;  
SUITDERIV ← IM = NL;
```

R(I) est une logique procédure qui essaie d'appliquer la règle I, l'applique si c'est possible et retourne la valeur vrai, retourne la valeur faux sinon.

En particulier pour R(2), si A(2) est l'ensemble des N2 arêtes où l'on peut appliquer la règle 2 on a :

```
I ← 1  
tant que non SUITDERIV2(IM,A2(I)) faire I ← I+1;  
R ← I ≠ N2;
```

avec

logique procédure SUITDERIV2(IM,A)

```
APPLI (R2,A);  
IM ← IM+1;  
SUITDERIV2 ← SUITDERIV(IM);
```

5.5. REPRESENTATION INTERNE D'UN GRAPHE.

5.5.1. RAPPELS

Soit G un graphe ayant n sommets et m arêtes. Les principales façons de représenter le graphe G en mémoire sont les suivantes :

Représentation par une table à une dimension :

chaque entrée de la table correspond à une arête du graphe, représentée par le numéro du sommet origine, le numéro du noeud extrémité, l'identification de l'arête.

La taille mémoire occupée est de $3 \times m$ emplacements mémoire, ou seulement $2 \times m$ emplacements si on supprime l'identification de l'arête.

Représentation par une table à deux dimensions :

Chaque ligne correspond à un sommet origine et chaque colonne à un sommet extrémité. A l'intersection de la ligne et de la colonne, on trouve l'identification de l'arête correspondante.

La taille mémoire occupée est de n^2 emplacements mémoire, ($m+1$)² on adopte la représentation duale, où les lignes et les colonnes sont associées aux arêtes, alors que le numéro du sommet se trouve à l'intersection).

Représentation avec des pointeurs :

Soit avec des plexes (pour chaque sommet autant d'entrées que d'arêtes dont ce sommet est origine, pour chaque entrée, identification de l'arête et pointeur sur le plexe du sommet extrémité) soit avec des enregistrements chaînés.

La taille mémoire occupée varie avec les graphes pour les plexes, et est de $3n$ emplacements pour les enregistrements chaînés.

Il existe beaucoup d'autres représentations internes des graphes tels que les codages par exemple le code DARC [11] et surtout les représentations associatives hashcodée (on associe à chaque sommet les arêtes dont il est l'origine au moyen d'une table hashcodée).

On utilise l'une ou l'autre de ces représentations suivant l'usage que l'on veut faire du graphe. De toute façon la quantité d'information nécessaire pour représenter un graphe, n'est jamais inférieure à $2 \times m$ où m est le nombre d'arêtes.

5.5.2. REPRESENTATION PAR UNE DERIVATION.

Soit $g = (V, I, R)$ une grammaire de graphe et G un graphe appartenant à Lg .
Si $D = (r_1, \dots, r_n)$ $r_i \in R$ est une dérivation qui permet de produire G et si
 $E = (S_1, \dots, S_2)$ est la suite des ensembles des sommets des sous-graphes
de partie gauche lorsqu'on applique les règles r_1, \dots, r_n , alors le couple
 $\langle D, E \rangle$ permet de différencier sans aucune ambiguïté le graphe G , c'est donc
une représentation de G .

Cette représentation, peut dans certains cas être très économique. En
effet en plus de l'information "explicite" c'est-à-dire la donnée du couple
 $\langle D, E \rangle$, il y a une information "implicite", la donnée de la grammaire g .

Les principaux inconvénients de cette représentation sont d'une part
son manque de généralité : pour une grammaire g donnée on ne peut représen-
ter que les graphes appartenant à Lg , et d'autre part, sauf cas particulier,
son inaptitude à tout traitement sur le graphe.

On n'utilisera donc cette représentation que lorsque l'on veut stocker
une grande quantité de graphes, dans le cas où cette représentation est très
économique par rapport à toutes les autres, ou bien si cette représentation
est appropriée à un traitement particulier (c'est-à-dire une application où
l'on utilise directement les propriétés de la grammaire).

EXEMPLE

Soit la grammaire $g = (V, I, R)$ permettant de produire tous les graphes
non-orientés sans boucles

$$V = \{S, s, e\} ; V_S = \{S, s\} ; V_{SN} = \{S\} ; V_{ST} = \{s\} ; V_A = \{e\} ; V_{SN} = \phi ; V_{ST} =$$

On ne représente par le renseignement sur les arêtes

$$I : S$$

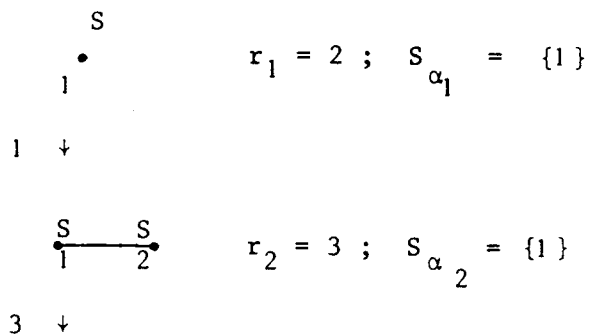
R ensemble des règles de production

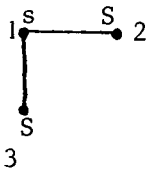
- 1°/ $\alpha : \begin{matrix} S \\ \cdot \\ \alpha_1 \end{matrix}$ $\beta : \begin{matrix} s \\ \cdot \\ \beta_1 \end{matrix}$ $\gamma : \gamma(\alpha_1) = \beta_1$
- 2°/ $\alpha : \begin{matrix} S \\ \cdot \\ \alpha \end{matrix}$ $\beta : \begin{matrix} S & S \\ \cdot & \cdot \\ \beta_1 & \beta_2 \end{matrix}$ $\gamma : \gamma(\alpha_1) = \beta_1$
- 3°/ $\alpha : \begin{matrix} S \\ \cdot \\ \alpha_1 \end{matrix}$ $\beta : \begin{matrix} S & S \\ \cdot & \cdot \\ \beta_1 & \beta_2 \end{matrix}$ $\gamma : \gamma(\alpha_1) = \beta_1$
- 4°/ $\alpha : \begin{matrix} S \\ \cdot \\ \alpha_1 \end{matrix}$ $\beta : \begin{matrix} s & s \\ \cdot & \cdot \\ \beta_1 & \beta_2 \end{matrix}$ $\gamma : \gamma(\alpha_1) = \beta_1$
- 5°/ $\alpha : \begin{matrix} S & S \\ \cdot & \cdot \\ \alpha_1 & \alpha_2 \end{matrix}$ $\beta : \begin{matrix} S & S \\ \cdot & \cdot \\ \beta_1 & \beta_2 \end{matrix}$ $\gamma : \begin{matrix} \gamma(\alpha_1) = \beta_1 \\ \gamma(\alpha_2) = \beta_2 \end{matrix}$
- 6°/ $\alpha : \begin{matrix} S & S \\ \cdot & \cdot \\ \alpha_1 & \alpha_2 \end{matrix}$ $\beta : \begin{matrix} S & s \\ \cdot & \cdot \\ \beta_1 & \beta_2 \end{matrix}$ $\gamma : \begin{matrix} \gamma(\alpha_1) = \beta_1 \\ \gamma(\alpha_2) = \beta_2 \end{matrix}$
- 7°/ $\alpha : \begin{matrix} S & S \\ \cdot & \cdot \\ \alpha & \alpha \end{matrix}$ $\beta : \begin{matrix} s & s \\ \cdot & \cdot \\ \beta_1 & \beta_2 \end{matrix}$ $\gamma : \begin{matrix} \gamma(\alpha_1) = \beta_1 \\ \gamma(\alpha_2) = \beta_2 \end{matrix}$

Il y a redondance entre les règles, mais c'est pour que les dérivations soient plus courtes.

Comme on ne crée jamais plus d'un sommet à la fois, on numérote les sommets 1,2,... au fur et à mesure qu'ils sont créés.

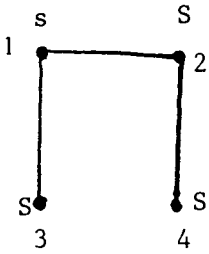
On donne un exemple de dérivation





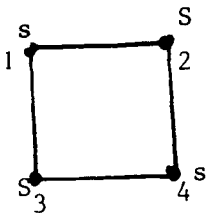
$$r_3 = 2, S_{\alpha_3} = \{2\}$$

2 ↓



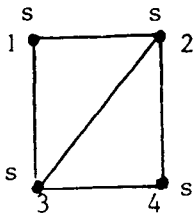
$$r_4 = 6; S_{\alpha_4} = \{3,4\}$$

6 ↓



$$r_5 = 7; S_{\alpha_5} = \{2,3\}$$

7 ↓



Le graphe ainsi produit pourra être représenté par le couple $\langle D, E \rangle$ avec

$$D = (2, 3, 2, 6, 7)$$

$$\text{et } E = (\{1\}, \{1\}, \{2\}, \{3,4\}, \{2,3\})$$

Ce que l'on peut également représenter par le tableau linéaire suivant où l'on alterne le numéro de la règle de production, notée en négatif, et la suite des sommets correspondants :

$(-2,1,-3,1,-2,2,-6,3,4,-7,2,3)$

5.5.3. REPRESENTATION DES GRAPHES PLANAIRES SANS ISTHMES.

Pour pouvoir effectivement, représenter un graphe par une dérivation, il faut évidemment pouvoir disposer de cette dérivation, donc avoir un analyseur syntaxique.

Nous avons vu au chapitre précédent comment on pouvait construire un tel analyseur pour la grammaire de MONTANARI permettant de produire les graphes planaires sans isthmes.

Nous allons voir comment, à l'aide de la dérivation ainsi obtenue, on peut représenter de façon très économique les graphes planaires sans isthmes.

La représentation du graphe est formée du couple $\langle D, E \rangle$.

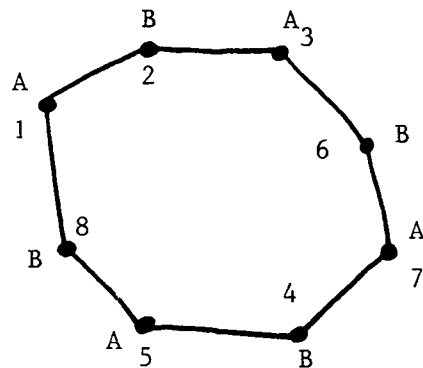
La dérivation D commence par la règle 1 et se termine par la règle 6, il n'est donc pas nécessaire de les représenter, on peut donc se contenter de la suite des $m-2$ règles (si m est le nombre d'arêtes du graphe) (r_2, \dots, r_{m-1}) où r_i est l'un des règles 2,3,4 ou 5.

E est la suite des ensembles des sommets des sous-graphes de partie gauche. Or ces sous-graphes sont eux-mêmes des sous-graphes du cycle des symboles non-terminaux.

On ne numérottera donc au cours de la dérivation que les sommets étiquetés par un symbole non terminal.

De plus si l'on prend la convention de parcourir ce cycle toujours dans le même sens, étant donné que sachant la règle à appliquer on sait le nombre de sommets du sous-graphe de partie gauche, il suffit donc de donner la position du premier sommet dans le cycle.

EXEMPLE



Si on prend comme origine du cycle le sommet étiqueté par "A" de numéro 1, et que l'on parcourt le cycle dans le sens des aiguilles d'une montre, appliquer la règle 3 en 4 signifie que le sous-graphe de partie gauche est le sous-graphe dont les sommets sont 6,7 et 4.

En appliquant la règle 1 on crée le cycle des symboles non terminaux et on fixe arbitrairement en l'un des deux sommets étiquetés par "A" l'origine du cycle.

Quand on applique la règle 2, on intercale deux nouveaux sommets. Quand on applique la règle 3, on remplace un sommet étiqueté par "A" par un autre. Si l'ancien sommet était l'origine du cycle, le nouveau sommet devient la nouvelle origine.

Pour la règle 4, on remplace un sommet étiqueté par "B" par un autre.

Enfin lorsqu'on applique la règle 5, on supprime deux sommets. Si le sommet étiqueté par "A" supprimé était l'origine, par convention, on prend comme nouvelle origine le premier sommet étiqueté par "A" suivant (toujours en parcourant le cycle dans le même sens) l'ancienne origine.

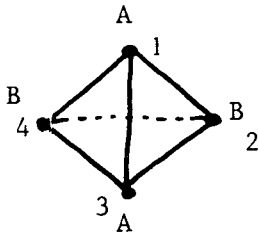
EXEMPLE

S



1 ↓

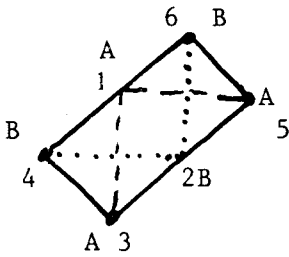
cycle : (1,2,3,4)



2 ↓

on applique 2 en la position 1

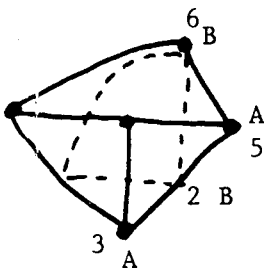
cycle : (1,6,5,2,3,4)



3 ↓

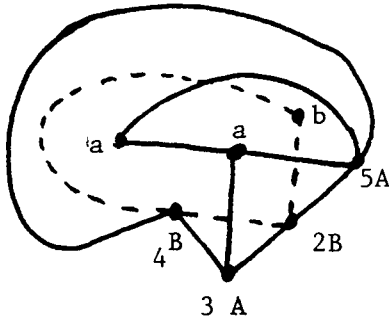
on applique 5 en la position 6

cycle : (5,2,3,4)

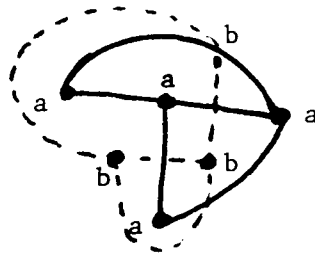


5 ↓

5 †



6 †



Une représentation du graphe (et aussi de son dual) pourra donc être

(2,1) , (3,6) , (5,6).

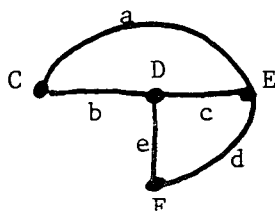
On peut sans aucune difficulté, associer à cette représentation des renseignements sur les arêtes et les sommets du graphe.

Etant donné que chaque fois que l'on applique une règle on crée une arête il suffit de ranger les renseignements concernant les arêtes dans l'ordre de création de ces dernières.

Il en est de même pour les sommets car on ne crée jamais plus d'un sommet à la fois, exception faite pour la règle 1 où l'on en crée deux. Dans ce dernier cas on prendra comme convention d'associer le premier renseignement au

sommet origine du cycle des symboles non terminaux, comme cela il n'y a aucune ambiguïté.

Par exemple le graphe renseigné :



pourra être représenté par :

(2,1),(3,6),(5,6) pour sa structure

(e,c,b,a,d) pour les renseignements sur les arêtes

(D,F,E,C) pour les renseignements sur les sommets.

On sait que pour pouvoir représenter avec des chiffres binaires un ensemble de n informations, l'on doit utiliser $\log_2 N$ chiffres binaires où N est tel que $2^{i-1} < n \leq N = 2^i$.

Si on veut représenter l'ensemble des graphes à n sommets, on doit dénombrer le nombre de tels graphes.

Deux graphes G_1 et G_2 à n sommets sont différents si les ensembles A_1 et A_2 de leurs arêtes sont différents, l'ensemble des sommets S étant identique pour les deux graphes.

Or $A \subset S \times S$ et donc le nombre d'ensembles d'arêtes différents est égal au cardinale $\mathcal{P}(S \times S)$, c'est-à-dire $2^{\text{card}(S \times S)}$ soit $2^{(n^2)}$.

Il y a 2^{n^2} graphes à n sommets et pour pouvoir représenter tous les graphes à n sommets on doit utiliser $\log_2 2^{n^2} = n^2$ chiffres binaires.

C'est ce nombre qui est nécessaire pour une représentation par une matrice de connexion.

Dans la représentation des graphes planaires sans isthmes de n sommets et m arêtes par la dérivation selon la grammaire de MONTANARI, on doit re-

présenter $m-2$ couples (r_i, s_i) r_i est l'une des règles 2,3,4, ou 5, donc deux chiffres binaires seulement sont nécessaires pour le représenter.

s_i est un numéro de sommet qui ne peut prendre au plus que $2n$ valeurs différentes (cas où tous les sommets sont sur le cycle des symboles non terminaux).

Donc s_i peut être représenté par $\log_2(2N)$ chiffres binaires soit $1 + \log_2 N$ ou N est tel que $2^{i-1} \leq N < 2^i$.

Le nombre de chiffres binaires pour représenter le graphe est donc égal à

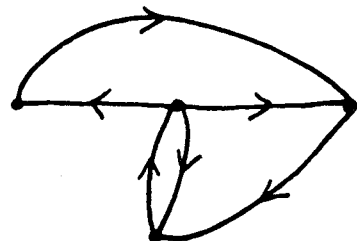
$$(m-2)(3 + \log_2 N) = m \log_2 N + 3m - 2 \log_2 N - 6.$$

On peut comparer cette représentation à la représentation des graphes de n sommets et m arêtes par la table des arêtes. On a dans ce cas besoin de $m \times 2 \times \log_2 N$ chiffres binaires.

Cependant la représentation des graphes par la dérivation selon la grammaire de MONTANARI ne permet pas de représenter tous les graphes planaires n sommets et m arêtes.

On ne représente pas les graphes orientés. On peut le faire en ajoutant des renseignements sur les arêtes. En effet, comme on l'a vu, on peut numéroter, au fur et à mesure de leur création, les sommets du graphe, donc on peut définir une relation d'ordre entre les sommets du graphe. Pour toute arête (i, j) $i < j$ du graphe non orienté on associera le renseignement '+', '-', ou '=' indiquant dans le graphe orienté on a l'arête (i, j) , (j, i) ou les deux.

Par exemple le graphe orienté suivant



sera représenté par la dérivation $(2,1)$, $(3,6)$, $(5,6)$ à laquelle on adjoint les renseignements $(=, +, +, -, -)$.

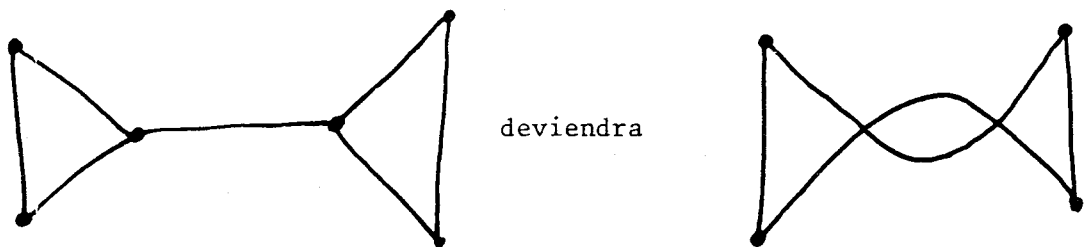
Il faut alors ajouter $2m$ chiffres binaires pour la représentation.

On peut également représenter les boucles sur les sommets c'est-à-dire les arêtes (i,i) en ajoutant un renseignement sur les sommets, soit n chiffres binaires.

Pour représenter les graphes planaires sans isthmes orientés avec boucles il faut

$$m \log_2 N + 5m + n - 2 \log_2 N - 6 \text{ chiffres binaires.}$$

Si le graphe est avec isthmes, il suffit de transformer chaque isthme en une arête double



Donc au plus (dans le cas d'un arbre) on devra doubler le nombre d'arêtes et ajouter un renseignement sur ces arêtes pour indiquer qu'elles sont doublées

Le nombre maximum de chiffres binaires est nécessaire pour représenter de cette façon un graphe planaire quelconque sera :

$2m \log_2 N + 12m - n - 2 \log_2 N - 6$
--

5.5.4. CODAGE DES CARTES PLANAIRES.

La notion de carte (ou d'hypercarte), permet d'obtenir une définition purement combinatoire des graphes (ou des hypergraphes) [52].

Une hypercarte est un couple (σ, α) de permutations sur un ensemble B de brins, tel que le groupe engendré opéré transitivement sur B.

Si α est une involution sans point fixe, l'hypercarte (σ, α) est une carte. (Les orbites de σ sont les sommets, les orbites de α , les arêtes

Une carte est une carte planaire si son genre est nul :

$$g(\sigma, \alpha) = 1 + \frac{1}{2} [Z(\alpha) - Z(\sigma) - Z(\alpha\sigma)] = 0$$

où $Z(\theta)$ désigne le nombre de cycles de la permutation .

Pour toutes les cartes planaires où l'on a particularisé un brin (cartes planaires pointées) on peut construire un code représentant ce carte. Ce code a pour longueur le nombre de brins de la carte. Avec Leh et Lenormand, on peut même transformer ce code en un code de même longueur construit à partir d'un alphabet de quatre lettres seulement.

Le problème est donc, pour un graphe donné de trouver les permutations (σ, α) associées, ce qui en fait se ramène une recherche de planéarité

Pour les graphes planaires sans isthmes, on peut utiliser la grammairie de Montanari, et même directement construire σ et α à partir de la dérivation :

Une dérivation commence par la règle 1, puis pour un graphe de m arêtes, m-2 couples (R_i, s_i) où R_i est l'une des règles 2,3,4 ou 5 et s_i l'indice du premier sommet du graphe de partie gauche dans le code des symboles non terminaux, puis à la fin on trouve la règle 6.

B contiendra 2m brins. On crée deux brins pour chaque règle de la dérivation, notés $2i-1$ et $2i$ pour $i=1, \dots, m$. Ces deux brins forment une arête, donc α sera le produit des permutations circulaires $(2i-1, 2i)$ pour $i=1, \dots, m$.

σ sera formé des permutations circulaires associées aux sommets. Pour chaque règle on ajoutera un brin dans deux permutations circulaires de façon à ce que sur le dessin planaire du graphe on parcourt les arêtes associées dans le sens trigonométrique. Dans le cas où le brin n'est ni le premier, ni le dernier de la permutation, on procède ainsi :

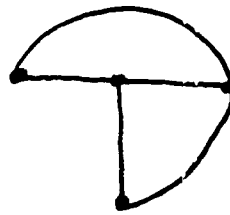
Règle 2 . Le brin $2i-1$, est ajouté à droite de la partie de la permutation déjà construite, si s_i est pair, à gauche si s_i est impair, pour l'"ancien" sommet.

Règle 4 En parcourant le cycle des symboles non terminaux dans le sens trigonométrique, on ajoute le brin $2i-1$ à gauche dans la permutation circulaire associée au premier sommet du graphe de partie gauche, et le brin $2i$ pour le dernier sommet.

Pour la règle 5 si s_i est pair on ajoute $2i-1$ à gauche, et si s_i est impair on ajoute $2i$ à droite sommet qui reste non terminal. (L'autre brin étant le dernier de la permutation associée à l'autre sommet).

Exemple

Soit le graphe



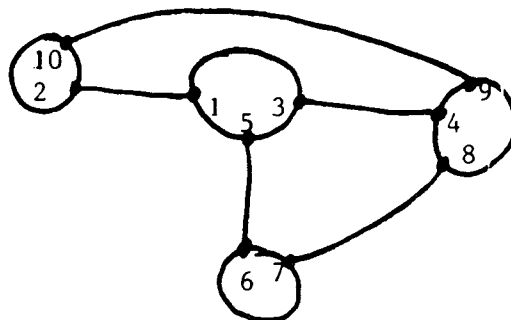
sa dérivation est $1, (2,1), (3,6), (5,6), 6$

α sera $(1,2) (3,4) (5,6) (7,8) (9,10)$

σ sera ainsi construite :

règle	cycle des symboles non terminaux	permutations com
règle 1	$(1, \dots (2, \dots$	
règle 2 en 1	$(3, 1, \dots (4, \dots (2, \dots$	
règle 3 en 4	$(6, \dots (4, \dots (2, \dots$	$(3, 1, 5)$
règle 5 en 6	$(4, 8, \dots (2, \dots$	$(3, 1, 5) (6, 7)$
règle 6		$(4, 8, 9) (2, 10) (3, 15)$

Carte que l'on peut représenter



Le cycle ξ est $(1, 2, 10, 9, 4, 8, 7, 6, 5, 3)$

Le code de CORI est $x_0 \overline{x_3} x_3 \overline{x_2} x_2 x_2 \overline{x_1} x_1 \overline{x_0} x_0$

et le code Lehman-Lenormand $x x y x y \overline{x} \overline{x} \overline{x} \overline{y} \overline{y}$

Conclusion.....

Avec le formalisme des grammaires de graphes, on dispose d'un outil très intéressant dans le domaine du traitement automatique des graphes ou des dessins. Mais comme tout formalisme très général, il est souvent difficile de pouvoir l'utiliser de façon efficace. C'est par exemple le cas des algorithmes d'analyse syntaxique qui sont essentiellement combinatoires donc peu performants.

Nous avons pourtant vu, comment, avec une grammaire de graphe particulière (produisant des graphes planaires sans isthmes), on pouvait aborder de façon très différente des problèmes "classiques" tels que la recherche de la planéarité, l'isomorphisme de graphes, ou le dessin. Si ces algorithmes ne sont pas toujours aussi efficaces que ceux connus par ailleurs, comme ils procèdent du même modèle, leur association peut être très heureuse : par exemple avec l'utilisation d'une dérivation pour le codage du graphe, puis à partir de cette dérivation, dessin du graphe ou recherche d'isomorphisme.

C'est dans ce sens qu'il serait fructueux d'étudier d'autres types de grammaires de graphes. Pour une catégorie de graphes ou de dessins donnée, on peut ainsi aborder globalement les problèmes de reconnaissance, de codage, de dessin etc.....

Les grammaires de graphes sont également des objets mathématiques dont une étude approfondie pourrait également être très intéressante. De nombreux problèmes restent encore ouverts, et leur étude permettrait de mieux comprendre le fonctionnement des algorithmes et de pouvoir ainsi les améliorer. Quel est, par exemple, le lien entre les grammaires de graphes et les grammaires de chaîne ou les autres systèmes génératifs ?

On peut également se demander si le graphe est bien la structure algébrique la plus indiquée, et si on peut généraliser la notion de grammaire formelle (ou autre système génératif) à d'autres structures pour lesquelles des problèmes tels que l'analyse syntaxique conduiraient à des algorithmes plus efficaces.

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

1. R.H.ANDERSON
"Syntax-directed recognition of two-dimensional mathematics"
Ph.D.Thesis, Div.Engg. and App.Phys. Harvard Univ.-Janv.1968
2. R. BELLMAN K. COOKE et J. LOCKETT
"Algorithms, graphs and computers".Stevens and Co - 1970
3. D.G. BOBROW
"Natural language Interaction Systems". Picture language Machines
S.Kaneff Ed. p: 31-62 - 1970.
4. M.B. CLOWES
"Perception, picture processing and computers"
Machine Intelligence 1 Eds : Collins and Michie p.181-198 - 1967
5. M.B. CLOWES
"A generative Picture grammar"
Camberra : Div.of computing Research, CSIRO Seminar paper 6 - 1969
6. M.B. CLOWES
" Transformational grammars and the organization of pictures "
Automatic interpretation and classification of Images p.43-78
Ed. A. Grassoli - 1969.
7. M.B. CLOWES
"Pictorial relationships - a syntactic approach" Machine Intel-
ligence 4 - 1969
8. M.B. CLOWES
"Picture Syntaxe"
Picture Language Machines p.119-159 - 1970 S. Kaneff Ed.
9. D.G.CORNEIL et C.C.GOTLIEB
"An efficient algorithm for graph isomorphism"
J.A.C.M. vol.17 n°1 p.51-64 - 1970
10. S.DACEY
"The syntax of a triangle and some other figures"
Pattern Recognition 2, p.11-31 - 1970

11. J.C.SAILLARD et J.SARRET
"Le programme DESMAG"
Convention DGRST n°67-01-055-351/CCM 61 pages Juillet 1969
12. E.W. ELCOK et A.M. MURRAY
"automatic discription and recognition of bo rd patterns in Go-Moku"
Machine Intelligence 2 p.75-88-1968
13. T.G. EVANS
"Descriptive pattern analysis techniques"
Automatic classification and interpretation of images p.79-95 Grasselli
Ed. PISA - 1968
14. T.G. EVANUS
" A grammar-controlled Pattern analyser" IFIP Congress 1968 p.H152-
H 157 - 1968.
15. J. FEDER
"Languages of incoded line patterns"
Information and Control 13 p. 230-244 - 1968
16. J. FEDER
"plex languages"
Information Sci. 3 p. 225-241 - 1971
17. K.S. FU et P.H. SWAIN
"On syntactic pattern recognition"
COINS 69 Symposium, software Engineering Vol.II p.155-172.
18. A.GRASSELI
"On the automatic classification of finger prints"
Proc.International conference of methodologies of Pattern Recognition,
Hawaï - 1968
19. A. GUZMAN
"Decomposition of a visual scene into bodies"
Automatic interpretation and classification of Images p.243-276
Grasselli Ed. 1969
20. W.H. HIGHLEYMAN
"Linear decision functions with application to pattern recognition"
Optical character recognition p.249-286 - 1962.

21. L HODES
 "A programming system for the one-line analysis of biomedical images"
 CACM 13, 279-283 - 1970
22. J. HOPCROFT, R. TARJAN
 " A V^2 algorithm for determining isomorphism of planar graphs".
 Information Preccessing letters 1 p.32-34 1971
23. J. HOPCROFT et R. TARJAN
 "Planarity testiny in $v \text{ Log } (v)$ steps"

 Technical report 201 Computer Science Department Stanlfort Univer-
 sity - 1971
24. R.A. KIRCH
 "Computer interpretation of English text and picture patterns"
 Trans IEEE E.C.13 p.363-376 - 1964
- 25 V.A. KOVALESKY
 "Present and future of pattern recognition theory" Proc.IFIP Congress
 65 Vol.1 p.37-43 - 1965
26. R.S. LEDLEY
 "pattern recognition studies in the biomedical sciences" Spr ng
 J.C.C. 28 p.411-430 - 1966
27. D.L. MILGRAM & A. ROSENFELD
 "Array automata and array grammars". Techn. rep.n°70 141 computer Sci-
 Center. University of Maryland 1970.
- 28 W.F. MILLER & A.C.SHAW
 "Linguistic methods in picture processing-a survey". Process F.J.C.C.
 Vol.33 p. 279-290 1968
29. U.G. MONTANARI
 "Separable graphs, planar graphs and Web-grammars" Information and
 Control vol.16 p. 243-267 1970
30. R. NARASIMHAN
 "A linguistic approach to pattern recognition". Technical report 121
 Digital computer laboratory. University of Illinois 1962
31. R. NARASIMAHN
 "Labelling schemata and syntactic description of pictures" Information

and Control Vol.7 p.151-179 1964

32. R. NARASIMHAN
"Syntax directed interpretation of classes of pictures" C.A.C.M. 9,
p.166-173 - 1966.
33. R. NARSIMHAN
"On the description, generation and recognition of classes of pictures"
Automatic interprétation and classification of images p.1-42 Grasseli
Ed. 1969.
34. R. NARASIMHAN
"Picture langages"
Picture langage machines p.1-25 Kaneff Ed. 1970
35. T. PAVLIDIS
"Analysis of set patterns"
Patern recognition 1 p.165-178 - 1968
36. T. PAVLIDIS
"Linear and context free graph grammars"
J.A.C.M. Vol.19 n°1 p: 11-22 1972
37. J.L.PFALTZ & A. ROSENFELD
"Sequential operations in digital picture processing" J.A.C.M. Vol.13
p.471-494 1966.
38. J.L.PFALTZ
"Web grammars and picture description". Tech.rep.70-138 Comp.Sci.Center
University of maryland - 1970.
39. J.L.PFALTZ
"Graph structures"
J.A.C.M. Vol.19 n°3 p.411-422 - 1972
40. A.ROSENFELD
"Isotomic grammars, parallel grammars and picture grammars"
Machine Intelligence 6 p.281-294 - 1971
41. A.C. SHAW
"A proposed language for the formal description of picture".
Standford linear accelerator center G SG 28 1967.

42. A.C. SHAW
"A formal picture description scheme as a basis for picture processing systems".
Information and control Vol.14 p.9-52 1969
43. A.C. SHAW
"Parsing of graph representable pictures"
J.A.C.M. Vol.17 p.453-481 - 1970
44. I.E. SUTHERLAND
"Schetchpad. a man.machine graphical communication system"
AFIPS conf.Proc.23 p.329.346 - 1963
45. R. MOHR
"Modèle algébrique pour l'analyse syntaxique de figures"
Thèse 3ème cycle NANCY I 1973
46. C. BERGE
"Graphes et hypergraphes"
Dunod 1970
47. N. CHOMSKY
"On certain formal propertics of grammars"
Inform.Control 2p. 137-167 - 1959
48. D.E.KNUTH -R.W.FLOYD
"Notes on avoiding "GO TO" statements".
Inform.Proc.Letters 1 p.23-31 - 1971
- 49.C. PAIR
"Cours d'analyse syntaxique"
Ecole d'été de Neuchatel 1972
50. J.HOPCROFT et R. TARJAN
"efficient planarity testing"
JACM Vol.21 n°4 p.549-568 1974
- 51.W.T.TUTTE
"On the enumeration of planar maps"
Bul. Amer.Math.Soc. 74 p.64-74 1968
- 52.R.CORI
"Un code pour les graphes planaires et ses applications"
Thèse d'état PARIS VII 1974

53.J. AZEMA J.C.SAILLARD

"Visualisation interactive de graphes"

Contrat 71-34-197 DRME 1973

54.G.DEMOUCRON Y.MALGRANGE R. PERTUISET

"Reconnaissance et construction de représentation planaires topologiques"

RF RO 30 p.34-47 1964

55.A.LEMPEL S.EVEN I.CEDERBAUM

"An algorithm for planarity testing of graphs"

Dans théorie des graphes Rome I.C.C. Dunod p.215-232 1967

56.C.TURCAT

"Isomorphisme, immersion et recouvrement de graphes"

Thèse 3ème cycle GRENOBLE 1 1974

57 H.WHITNEY

" Non-separable and planar graphs"

Trans.of the American Math.Soc.n°34 p.339-362 1932.