Institut National Polytechnique de Lorraine (INPL)
Ecole des Mines de NANCY (ENSMIM)

Sc N 80/16A

S C A P A

Un système de Correction Automatique des Plans de vol,
avec Apprentissage

Service Commun de la Documentation
INPL
Nancy-Brabois

UNIVERSITÉ NANCY BIBLIOTHÈQUE

THESE

presentée devant
l'Institut National Polytechnique de Lorraine
pour l'obtention du titre de

DOCTEUR-INGÉNIEUR

Spécialité : Informatique

par D. DERAMAUT
Ingénieur Civil des Mines

Soutenue le 1er Février 1980
devant la commission d'examen composée de :

D 136 036418 8

. PAIR
COULON
. HATON
. KAYSER
G. MAIGNAN

ScN 80
6 A

S C A P A

Un système de Correction
Automatique de Plans de Vol,
avec Apprentissage

Service Commun de la Documentation
INPL
Nancy-Brabois
UNIVERSITÉ NANCY I BIBLIOTHÈQUE

Cette étude a été effectuée par le Centre de Recherche en Informatique
de Nancy pour le compte d'EUROCONTROL, suivant les termes du contrat
N° C/69/CE/PS/77.

TABLE DES MATIERES

I. PRESENTATION

Introduction page 1

I.1. Les Messages "PLAN DE VOL" page 2
I.2. Le problème des erreurs page 7
I.3. Objectifs de l'étude page 10

II. LES APPROCHES POSSIBLES page 11

II.1. Une approche syntaxique générale page 13
II.2. Une approche sémantique "ad hoc" page 16
II.3. Intégration de processus correctifs par apprentissage page 18

III. SCAPA - Description Fonctionnelle

III.1. Aperçu général du fonctionnement de SCAPA page 20
III.2. Description de l'analyse correction page 22
III.3. Description de l'apprentissage page 37

IV. SCAPA - expérimentations

IV.1. Expérimentation de l'analyse correction page 69
IV.2. Expérimentation de l'apprentissage page 81
IV.3. Evaluation du système SCAPA page 89

V. CONCLUSIONS

.../...

Annexes

A1 : Contrôle de la Navigation Aérienne

A1.1. Carte de la zône de Karlsruhe
A1.2. Exemples de messages Plan de Vol
A1.3. Description O.A.C.I. (1) des champs d'un Plan de Vol.

A2 : Données de base et notations de SCAPA

A2.1. Listes des classes, des relations, des productions, des opérateurs
A2.2. Graphes des productions.

A4 : Notes techniques sur l'implantation de SCAPA

A4.1. Les modules
A4.2. L'overlay
A4.3. Les fichiers.

A5 : Jeu d'essai

A5.1. Autres exemples d'analyse-correction
A5.2. Autres exemples d'apprentissage.

Lexique

Bibliographie

1) O.A.C.I. = Organisation de l'Aviation Civile Internationale.

.../...

I. PRESENTATION

Introduction

Pour assurer la sécurité et la fluidité du trafic aérien, il est nécessaire d'organiser un contrôle de l'extérieur.
Les contrôleurs des différents centres disposent essentiellement de 2 types d'informations :

1) les informations radar, acquises localement,

2) les informations "Plan de Vol" transmises de centre en centre.

Un message "Plan de Vol" contient les principales informations statiques relatives à un vol : identification, type d'appareil, départ, arrivée, itinéraire prévu, etc...

Ces messages, mis à jour tout au long du vol permettent, une fois exploités, de prévoir et de planifier les flux d'avions pour une région.

Dans les centres de contrôle où l'importance du trafic justifie un équipement moderne, le traitement des données radar et des données "Plan de Vol" est largement automatisé et fournit aux contrôleurs des informations pertinentes de synthèse.

Dans cette automatisation des traitements, un des problèmes que l'on rencontre est celui de l'extraction des données Plan de Vol, car ces derniers contiennent fréquemment des erreurs, aléatoires ou d'origines variées. Le problème que pose cette étude est celui de la correction automatique de ces erreurs : correction des erreurs connues (fréquentes), et apprentissage progressif des corrections d'erreurs nouvelles. Sur le plan informatique, il s'agit d'un problème de compilation de texte comportant des erreurs. Dans ce domaine, les approches syntaxiques s'avèrent insuffisantes, les approches sémantiques "ad hoc" limitées. Des mécanismes d'apprentissage semblent pouvoir remédier en partie à ces points. Ces approches font l'objet du Chapitre II. Avant de préciser ce problème des erreurs (§ 2) et de délimiter nos objectifs (§ 3), nous allons détailler le rôle et l'organisation des Plans de Vols (§ 1).

.../...

I.1. Les Messages "Plan de Vol" (ou FPL (2))

La fonction majeure du Plan de Vol est de véhiculer les principales données et intentions d'un vol ; les données les plus utiles sont :

- L'indicatif d'appel (ou call-sign) ; c'est la référence principale ; elle est inchangée durant tout le trajet et sera utilisée par tous les contrôleurs ayant à désigner ce vol.

  Exemples : AF330 Vol Air France N°330
  GAF31 Vol German Air Force N°31
  BØZØ41 Vol Militaire

- Le type d'avion : à partir des performances connues de ce type d'avion (en particulier les vitesses), on peut estimer les heures de survol des balises.

  Exemples : B707 Boeing 707
  F104 Starfighter (Lockeed)
  DC10 (Mc Donnell Douglas)

L'itinéraire prévu est décrit par :

- L'aéroport de départ et heure souhaitée de décollage

  Exemples : LFPØ1730 Départ de Paris Orly 17 h 30
  EDDF0030 Départ de Francfort 0 h 30

- Une séquence de balises et de routes aériennes : les vitesses et altitudes de croisière, et les régimes de vol, complètent cette description.

  Exemples : balises : MMD Montmédy
  LYO Lyon
  NTM Nattenheim
  MUN Munich

  routes : UG1 Route "Upper green 1"
  TB6 Route "Tacan blue 6"
  UW26N Route "Upper white 26 North"

  vitesses 0440F230 : 440 noeuds - 23 000 pieds

  et

  altitudes M082M0840 : 0,82 Machs - 8400 mètres

  régimes : I Vol aux instruments
  Z Vol à vue puis aux instruments.

.../...

2) FPL : Flight Plan

- L'aéroport d'arrivée : l'heure prévue d'atterrissage et un aéroport de remplacement.

  Exemple : LFPØ1900 LFPB : arrivée à Orly 19 h
  remplacement : Le Bourget

- Le matériel de navigation/communication à bord

- Le statut du vol (= militaire ou général ou opérationnel).

On voit que les Plans de Vol peuvent éventuellement jouer un rôle important dans des opérations de sauvetage et de recherches.

Au début de l'ATC (3), les messages Plans de Vol étaient rédigés en langage naturel avec certaines conventions assez vagues. En 1944, l'O.A.C.I. (4) proposa une description syntaxique des messages ATS (5) ; nous allons la donner ici sommairement, conformément aux dernières modifications des normes O.A.C.I. (Août 1978).

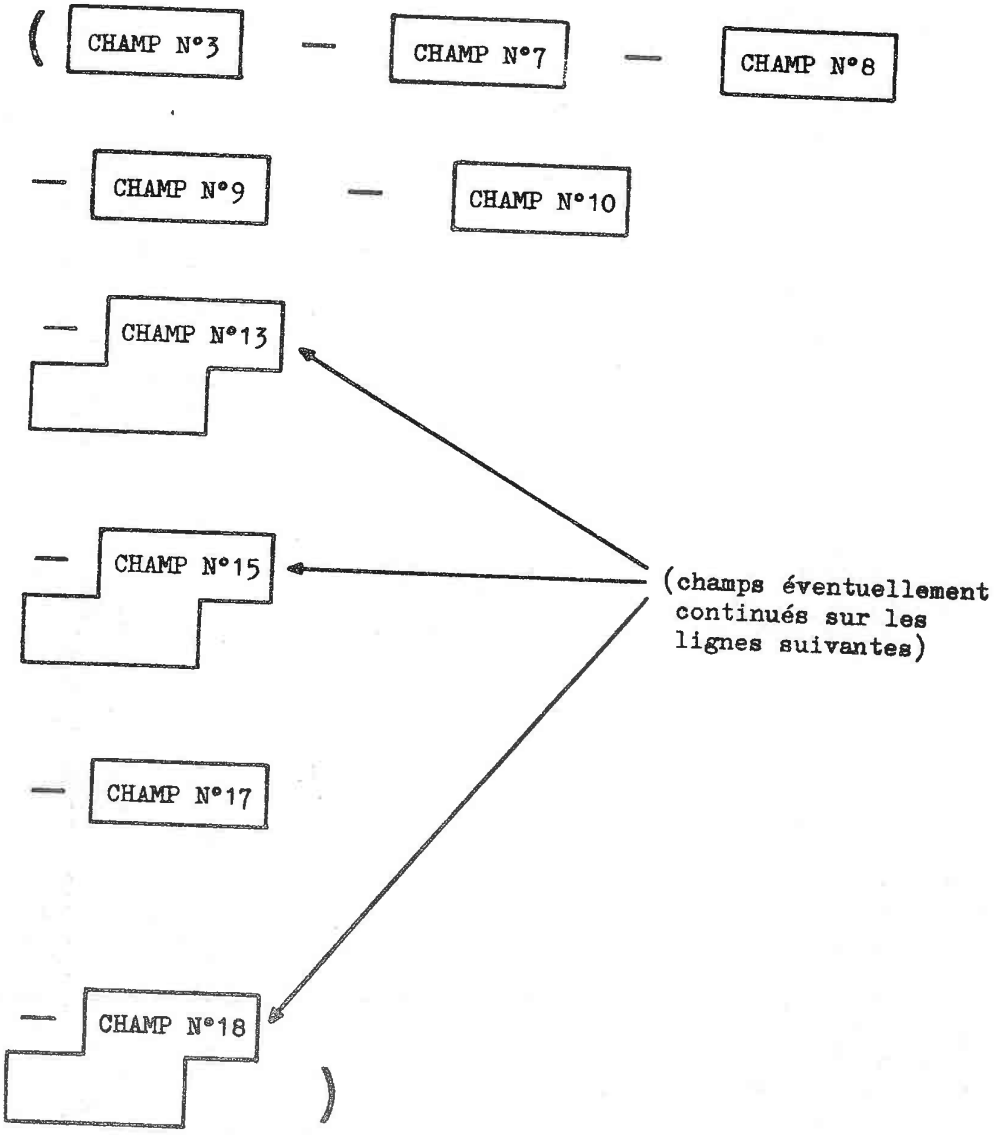
Les messages sont segmentés en champs ; chaque champ comporte une donnée ou un groupe de données cohérentes. Le type du message pré-détermine la séquence des champs. Les champs sont au nombre de 20 dont 9 composent un message FPL :

1) Champ N° 3 : type des message (ici "FPL")
2) Champ N° 7 : identification (call-sign) et données radar secondaire
3) Champ N° 8 : règles et type du vol
4) Champ N° 9 : nombre et type d'avions et classe de turbulence
5) Champ N° 10 : équipement radio et radar secondaire
6) Champ N° 13 : aérodrome et heure de départ ; estimées aux frontières des régions concernées
7) Champ N° 15 : route prévue
8) Champ N° 17 : aérodrome et heure d'arrivée ; aérodrome de remplacement
9) Champ N° 18 : informations complémentaires.

.../...

3) ATC = Air Traffic Control
4) OACI = Organisation de l'Aviation Civile Internationale (en Anglais ICAO)
5) ATS = Messages "Air Traffic System" dont font partie les messages "FPL"

La disposition des champs est la suivante :



La syntaxe de chacun de ces champs figure à l'Annexe A1.3.

.../...

Donnons quelques exemples de messages FPL :

```
nnnnzczc dub529 221822
ff eddyzq edjbyv edduzq lowwzq lyzazq lybezq lbsfzq ltbazq ltaczq
221817 ehamzp
(fpl-gf248-i
-vc10-r/r/k
-eham2115 lbsf2311 ltba2332 ltac2350 orbb0059
-0487f330 nyk ub1 v1w direct kft ub5 sm1 ua4 dim/k900s1005 a4
rad/0477f370 va4n ist vg8 srt vr21 az r21 ib r21 alpha basrah ba r21
-omdb0305 otbd
-sel/amgj reg/a40vc)
```

```
nnnnzczc dub530 221824
ff eddxyi edzzzq eddyzq edduzq lszhzq lszhzt
221823 edvvzp
(fpl-sr3507-i
-dc9-r/r/k
-edvv1930
-0420f290 vv86 pa9 ffm ug31 tgo sig sha
-lszh2030 lfsb
-prt/red/rmk/permission nr l 3/24.20.10-006b)
```

```
nnnnzczc dub531 221827
ff eddxyi edzzzq edduzq lowwzq lhbpzq lrbbzq lbwnzq
221824 eddkzp
(fpl-hf469-i
-b727-r/r/k
-eddk1925 lrbb2052 lbwn2125 ltba2149 ltac2210
-0457f330 dk4 pb1 ffm pg1 erl ug1 lnz ug1 g1 a17 r39 b27 g1 vg1
delta yaa vg8 yanke buk
-ltac2230 ltba
-reg/dahlr)
```

```
nnnnzczc dub532 221832
ff edduzq edduyu eddyyu eddyzq educyx edjcyv edadyo
221828 edamyo
(fpl-xerox20-ia
-f4-r/r/j
-edam0700
-0480f250 am01 brl/0510f310 ffm ua9 wrb/0510f330 dle/0510f350 ged
0510f370 erl ug1 rdg/0510f350 igl tgo/0400f200 hdl
-edam0910 edad
-(date of flt 23 apr)opr/usmil oatms/rmks bz flt/req radar vectors
 dep to ffm
-fuel/0300 pob/2)
```

.../...

Déchiffrons, à titre d'exemple, le plan de vol suivant :

(fpl-ba888-i
-b747-r/ri/k
-egll1850 lbsf2102 ltba2117 ltac2149 orbb2247 obbi0009
-0496f330 dvr ug1 ntm ub6 tgo ub1w rtt ub1w v1w direct
kft ub5 smi ua4 dim/k905s1005 a4 rad/0496f330 va4n ist vg8
srt vr21 az r21 ba b55
-obbi0024 omdb
-sel/baf reg/gawnb)

C'est un vol de la British Airways (BA) N° 888 désirant voler aux instruments (I)

L'avion est un Boeing (B747) ayant l'équipement noté (R/RI/K) qui indique :

- équipement de communication standard,
- équipements de navigation/approche standards et dispositif de navigation à inertie,
- transpondeur 4096 codes modes A, B et C

(selon les dernières normes OACI, un tel équipement serait maintenant décrit par : SI/C).

- Départ de Londres/Heathrow (EGLL) à 18 h 50

- désire atteindre, en croisière, la vitesse de 496 noeuds à l'altitude de 33 000 pieds

- survole Douvres (DVR) puis prend la route UG1 jusqu'à Nattenheim (NTM) puis prend UB6 jusqu'à Stuttgart (TGØ) puis prend la UB1W jusqu'à Rastadt (RTT) puis ...

- Arrivée prévue à Bahrain 00 h 24 (OBBI) ou, en cas de difficultés, à Dubaï (OMDB).

.../...

I.2. Le problème des erreurs

Initialement, les messages Plan de Vol sont transcrits sur des bordereaux par les pilotes ou par les responsables des Compagnies Aériennes. Ces bordereaux parviennent aux services concernés de l'aérodrome qui les introduisent sur télétype et les transmettent, via un réseau de type télex (RSFTA(6)) à tous les Centres de Contrôle concernés par chaque Plan de Vol.

Ces services existaient avant que l'OACI n'édite des règles de codage des messages (les premières normes OACI datent de 1944) et la plupart des personnels du RSFTA furent formés à travailler sur des messages en format libre ; on constate généralement dans des cas semblables une inertie certaine face aux changements d'habitudes. On le constate encore aujourd'hui dans de nombreux cas d'erreurs de Plans de Vol.

Dans les systèmes ATC semi-automatiques actuels, le traitement des erreurs des Plans de Vol est effectué par le module d'extraction des données ; il se limite à la détection d'erreurs et à l'établissement d'un dialogue correctif avec un opérateur.

Les opérateurs doivent donc souvent corriger les mêmes erreurs dont certaines sont triviales mais fastidieuses à corriger.

La notion d'erreur étant relative, précisons que l'on entend par "erreur", les erreurs par rapport à :

- la syntaxe proposée par l'O.A.C.I.
- la logique et les règles de l'ATC
- les données et pratiques opérationnelles en vigueur
- le bon sens et les connaissances empiriques des opérateurs contrôlant les messages.

Ceci nous permet de proposer une classification des types d'erreurs que l'on peut rencontrer :

- erreurs syntaxiques : messages non conformes à la syntaxe OACI et donc rejetés par un analyseur syntaxique de ce langage

- erreurs sémantiques : signification en désaccord avec la logique ATC ou le bon sens des opérateurs

- erreurs opérationnelles : données en désaccord avec les paramètres opérationnels en vigueur.

Remarquons que la structure syntaxique d'un FPL a pour effet d'isoler des groupes dont le contenu sémantique est homogène.

(6) RSFTA = Réseau du Service Fixe des Télécommunications Aéronautiques (en anglais AFTN).

.../...

Dans ces conditions, il est rare que les erreurs soient franchement d'un type ou de l'autre. Par exemple, une mauvaise structuration des champs peut entraîner la perte ou l'altération du contenu sémantique d'un message.

D'une manière générale, les erreurs ont 3 origines possibles :

- inattention du commandant de bord ou du responsable de la Compagnie Aérienne,

- inattention de l'opérateur RSFTA,

- altération liée à la transmission par le RSFTA.

Dans les 2 premiers cas, on peut tenter de réduire les erreurs syntaxiques en substituant au télétype d'entrée des données, un terminal "intelligent" utilisant une logique simple de vérifications pour imposer des données syntaxiquement correctes. Mais cela ne corrige ni les erreurs de transmissions ni les erreurs sémantiques, quelle que soit leur origine.

Nous donnons ici quelques figures d'erreurs assez fréquentes :

- <u>structures incorrectes</u>

1) (FPL - AF330 - I
-B707 - S/C
-LFPØ1430
-EDDF1510
-0300F270 UR10 LUG R10 9DF
-0)

: champ route (0300F270 UR10 ..9DF) et champ "arrivée" (EDDF1510) permutés ;

2) (FPL - LH433 - IS
-B747 - S/C
-LFPØ1500
-0300F270 UR10 LUG R10 9DF
-EDDF1540 EDDL
-0
-FUEL/0450)

: un champ en trop : le champ "FUEL".

- <u>altération de caractères</u>

1) NTM - 0300F250 au lieu de NTM/0300F250

2) TGØUA9 FFM au lieu de TGØ UA9 FFM

.../...

- <u>erreurs de séquences</u>

1) 0450F270 DF9 DL8 ;

: 2 itinéraires standard de départ consécutifs ;

2) 0450F270 UG1 ALB UR9 DCT ZUL

: "DCT" précédé d'une route aérienne UR9 au lieu d'une balise.

3) 0440F330 LST TB1 DKB T ERL MUN

: "T" suivi d'autres termes ;

- <u>altérations de la sémantique</u>

1) FFM A9 TRA/0400F270

: A9 est une route de l'espace inférieur (niveau ⩽ F250) or le niveau requis sur la balise TRA (F270) est un niveau de l'espace supérieur.

2) FFM Y NTE DCT NOR Y UG1

: le 1er Y indique que sur FFM, on passe du vol aux instruments au vol à vue ;

Le 2ème Y est incohérent avec le 1er ; c'est "Z", indiquant le passage inverse qui était attendu.

3) TGØ UG7 NTM

: UG7 ne relie pas TGØ à NTM ;
(probablement il s'agissait de UG1)

- <u>erreur opérationnelle</u>

1) FFM UW26 LUG

: erreur si le vol a lieu en semaine car UW26 est une route de week-end principalement, et l'utiliser en semaine complique la coordination des contrôleurs civils et militaires.

.../...

1.3. Objectifs de l'étude

Notre objectif était de réaliser un système capable, après un certain délai de fonctionnement, de traiter une grande variété de messages, en corrigeant un éventail d'erreurs comparable à celui que sait traiter un opérateur. Pour cela on songeait à mettre en place deux unités :

- une unité d'analyse et correction des erreurs connues

- une unité d'intégration de mécanismes correctifs nouveaux, déduits de cas d'erreurs nouvelles par un apprentissage guidé par une version correcte du message, fournie par un opérateur.

Le travail que nous avons réalisé est essentiellement centré sur le champ 15 (le champ route) pour plusieurs raisons :

- son contenu sémantique est riche et complexe ; or, la plupart des systèmes d'extraction en ont une approche syntaxique assez rudimentaire (analyse syntaxique suivie de quelques vérifications sémantiques) (*). C'est donc un champ privilégié pour tenter une approche syntaxico-sémantique, avec un apprentissage capable d'intégrer des aspects sémantiques nouveaux.

- par ailleurs, le taux d'erreurs dans ce champ est le plus élevé, ce qui justifie que l'on tente d'y remédier, en priorité.

Dans notre système, les champs autres que le champ 15 subiront une analyse simplifiée avec des tests sémantiques faibles ou nuls. Aucun message présenté à SCAPA ne présentera d'erreurs dans ces champs.

Pour des raisons pratiques liées à Eurocontrol, nous avons expérimenté sur la zone de contrôle de l'Allemagne du Sud ("Rhein Control"). Les données opérationnelles et les échantillons de messages sont donc issus des données de l'espace Rhein Control UAC (7) et des espaces limitrophes.

Nous avons seulement voulu construire un système expérimental en vue d'une étude de faisabilité d'une méthode d'apprentissage, relativement au domaine spécifique des Plans de Vol, et ce, avec un minimum de réalisme. Cependant, il conviendrait de réétudier ce système pour en faire une version opérationnelle car, de ce point de vue, il comporte des insuffisances ; nous les signalerons dans IV.3.

.../...

*) Pour des descriptions plus détaillées des systèmes actuels, on peut se référer à la bibliographie sur le système KARLDAP (19), le système CAUTRA (18) ou le système SHANDAP (20) qui sont représentatifs des systèmes modernes européens.

7) UAC : Upper Airspace Control

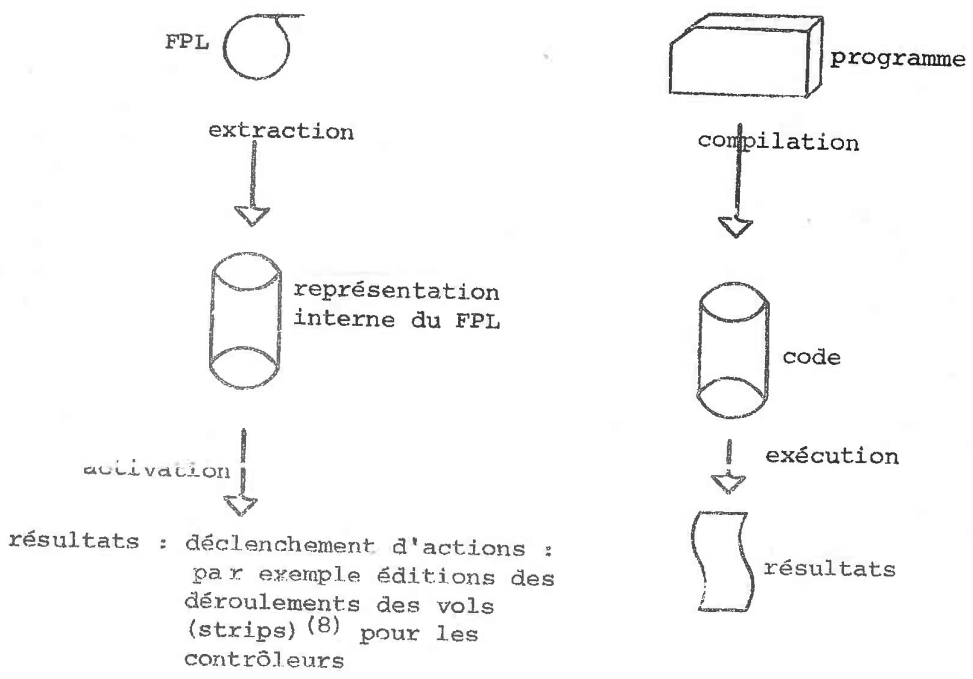
II. LES APPROCHES POSSIBLES

Les systèmes actuels d'extraction des données Plans de Vol proposent une détection d'erreurs, essentiellement syntaxique, bien que des procédures "ad hoc" diagnostiquent certaines erreurs sémantiques prévues. Un dialogue correctif est ensuite établi avec un opérateur humain chargé d'effectuer les corrections.

Cette stratégie a de nombreux points faibles :

- le caractère limité et figé des traitements sémantiques ;
- la faiblesse de l'aide fournie aux opérateurs pour des corrections faisant intervenir la sémantique (même si l'erreur est, à l'origine, syntaxique) ;
- l'absence de prise de décision du système, même en cas d'erreur syntaxique élémentaire. (erreur lexicale prévue, par exemple). Cette insuffisance oblige les opérateurs à effectuer de manière répétitive des corrections simples mais fastidieuses.

Le problème de la correction des erreurs dans les Plans de Vol est voisin de celui de la correction des erreurs dans les programmes qui est abordé en Théorie de la Programmation. On peut schématiser ainsi le parallèle existant entre ces deux problèmes :



(8) Strips : bandes de papier utilisées par les controleurs , où, pour chaque vol, les heures de survol des balises sont calculées.

La tendance générale de ces dix dernières années, pour la correction automatique des erreurs, en compilation, semble être basée sur la mise en oeuvre combinatoire d'opérateurs syntaxiques élémentaires de corrections. Nous abordons le problème sous cet angle purement syntaxique au § 1. Pourtant, pour effectuer des corrections pertinentes, la syntaxe ne suffit pas ; nous développons, au § 2, une approche incorporant la sémantique sous forme de procédures spécialisées (procédures "ad hoc" de correction-interprétation) intégrées dans un analyseur syntaxico-sémantique.
Cette approche conserve cependant beaucoup des points faibles reprochés aux systèmes actuels d'extraction, notamment leur caractère limité. L'approche présentée en § 3 se propose de remédier à ce point en joignant à un analyseur de type décrit au § 2, des mécanismes d'apprentissage lui permettant d'intégrer automatiquement de nouveaux traitements "ad hoc" induits par comparaison entre la version initiale et une version corrigée correcte du même message.

Ces différentes approches ne sont pas exclusives et, d'ailleurs, notre système intègre des aspects de chacune : il part d'un modèle d'analyse de type (§ 2) et le complète par un mécanisme de type (§ 3) mettant en oeuvre des opérateurs généraux de corrections de type (§ 1).

.../...

II.1. Une approche syntaxique générale

On envisage ici la correction d'erreurs par des mécanismes généraux fixes, intégrés initialement dans un système d'analyse. Les idées dérivées de cette approche sont assez largement développées dans les articles orientés vers la compilation.

Le plus souvent, ces mécanismes agissent directement sur les chaînes erronées ; dans [3] cependant, ces corrections sont intégrées initialement au système, sous forme de productions d'erreurs.

En 1963, IRØNS ([9]) proposait, pour les grammaires context-free, des corrections de 3 types : insertions, destructions et substitutions, qui étaient tentées de manière semi-aveugle sur les chaînes erronées, jusqu'à ce que l'analyse se termine. Cette approche a comme inconvénient majeur de ne pas contrôler la minimalité des solutions proposées et de ne pas accepter l'échec, même lorsque des solutions tortueuses sont envisagées. Cette stratégie du "tout pour le tout" est à coup sûr trop risquée pour un sous-système ATC.

A titre d'exemple, citons un résultat tiré de [9] :

texte source : ";(RE = -V2I)(*XM;+X=A+F;X-+HT.(R)*ST;"
correction proposée :

"RE = V+(I)*XM;X=A+F;X=HT+(R)*ST;"

On mesure l'audace des corrections proposées, vu le grand nombre d'interprétations possibles du texte source.

AHØ & PETERSON ([3]) reprennent en 72 l'idée selon laquelle les erreurs de base, en compilation, résultent essentiellement de 3 types d'agents : - substitution d'un symbole à un autre ; - insertion d'un symbole étranger ; - destruction d'un symbole. Toutefois, l'approche proposée est différente : La grammaire de base est étendue à l'aide de "productions d'erreurs" (règles) générant ces types d'erreurs. L'analyseur de cette grammaire étendue, doit ensuite minimiser l'emploi des "productions d'erreur". La version corrigée est le résultat des corrections liées aux "productions d'erreur" utilisées dans la solution minimale proposée. On peut émettre la réserve suivante : dans quelle mesure le critère du nombre minimal des "productions d'erreur" employées, correspond t il à la solution la plus proche de l'idée originelle du programmeur ? Par ailleurs, si une erreur finit par être corrigée, après un traitement long, les occurences ultérieures de cette erreur redemanderont le même délai de traitement ; dans le cas des plans de vols, la contrainte de durée de traitement est impérieuse et semble incompatible avec cet aspect de [3].

.../...

AKNIN, DANAN et MATHIEU ([4]) ont tenté de résoudre le problème spécifique de la correction des erreurs Plans de Vol en adaptant l'algorithme de AHO et PETERSON à un analyseur utilisant des automates à configuration, mais sans pour autant remédier aux faiblesses de cet algorithme.

WAGNER aussi, en 74 ([16]), utilise les 3 mêmes opérations élémentaires mais la stratégie est différente : l'algorithme de correction d'erreur précède l'analyse : il est comparable à un préprocesseur traduisant a priori la chaîne source en une chaîne garantie correcte syntaxiquement. C'est ensuite seulement que le code est fabriqué.
Un critère de minimalité des opérations élémentaires appliquées (- changement - insertion - destruction), a la propriété qu'une chaîne correcte restera inchangée par application de l'algorithme. Cette approche, limitée aux langages réguliers, a l'avantage d'être proportionnelle en temps, à la longueur de la chaîne. Cependant, le critère de distance fait l'objet des mêmes réserves que pour [3] ;

Plus originale, l'approche de GRAHAM & RHODES, décrite dans [6] a lieu en deux temps :

- analyser le contexte de part et d'autre de l'erreur :
  - un "mouvement arrière" tente des réductions sur la pile d'analyse,
  - un "mouvement avant" tente la poursuite de l'analyse au-delà de l'erreur ; ce mouvement se termine soit à la rencontre d'une deuxième erreur, soit au moment de tenter une réduction s'étendant en-deçà de l'erreur.
    C'est la phase de "condensation".
- créer un lien entre le contexte gauche et le contexte droit : la partie problématique entre ces deux contextes est comparée avec les parties droites des productions existantes.
  On en déduit des transformations élémentaires qui conviennent à la fois localement et avec les contextes.

Une évaluation basée sur une distance minimum pondérée, détermine l'ensemble de transformations, retenu comme correction. C'est la phase de correction. Au-delà d'un nombre limité de transformations, on recourre au "panic mode" (on se contente de sauter le texte jusqu'à certains symboles privilégiés, d'où l'on peut continuer l'analyse).

.../...

MICKUNAS & MODRY ([13]), en 1978, étendent cette méthode aux analyseurs LR en remplaçant l'aspect "pattern-matching" de la phase de correction par une recherche directe de liens entre les contextes.

Toutes ces approches syntaxiques s'avèrent néanmoins insuffisantes, si l'on considère que la correction effective d'erreurs purement syntaxiques peut nécessiter une interprétation sémantique, du contexte, par exemple. L'approche qui va suivre étudie les moyens de représenter les traitements sémantiques, à la fois dans l'analyse et dans la correction.

.../...

II.2. Une approche sémantique "ad hoc"

Cette approche s'inspire initialement des systèmes d'extraction existants (à Karlsruhe, entre autres), où sont mises en oeuvre des procédures exceptionnelles de détection d'erreurs sémantiques dans le cadre d'une analyse syntaxique classique.

Une approche de ce type est proposée par AHO & JOHNSON ; dans [2], ils exposent brièvement comment, pour un analyseur LR, les tables (tables d'action ou tables GØTØ) peuvent orienter vers des procédures exceptionnelles de récupération, au lieu de conduire à l'état "erreur". Cependant, ces procédures doivent effectuer des tests sémantiques et rechercher le contexte nécessaire à des corrections pertinentes ; ceci ne s'intègre pas très naturellement au fonctionnement des analyseurs LR (ou d'autres analyseurs syntaxiques) car il y a manque d'homogénéité entre les actions syntaxiques des tables et les actions sémantiques "ad hoc" des procédures.

Il semble par contre plus intéressant d'assurer une détection à la fois syntaxique et sémantique des erreurs, et d'opérer de même pour la correction. La grammaire, utilisée par l'analyseur, recouvre alors outre la description OACI, les cas d'erreurs connues, les corrections correspondantes étant décrites dans les actions sémantiques liées aux règles. De ce point de vue, l'approche de C. L. HEDRICK ([8]), concernant un analyseur syntaxico-sémantique, est intéressante car elle permet de prendre en compte directement et efficacement la sémantique dans l'analyse (et, par ailleurs, elle est conçue explicitement pour supporter un mécanisme d'apprentissage ; cf II.3.).

Les productions de HEDRICK se composent de :

- la structure (partie droite d'une règle)
- un ensemble de conditions d'applications à satisfaire simultanément, décrites sous forme de triplets
- l'identification du résultat (partie gauche)
- des triplets résultats conservant les résultats sémantiques acquis lors de l'application d'une règle.

Un algorithme simple d'application de ces productions construit une analyse de bas en haut.

Cette organisation des productions permet de conférer à certaines la qualité de "procédures exceptionnelles" en y incluant des tests sémantiques de détection d'erreurs et de recherche de solutions correctives, ainsi que des actions effectuant ces corrections.

.../...

Nous avons donc aménagé le modèle de HEDRICK afin :

- d'augmenter l'efficacité des tests, en traduisant leur imbrication par une arborescence au lieu d'une séquence,

- d'introduire la description des actions à effectuer pour corriger la chaîne d'entrée (en cas d'erreur).

Dans [13], McDERMOTT, NEWELL & MOORE décrivent des algorithmes destinés à alléger l'implantation des systèmes à productions du genre de celui de HEDRICK, pour les faire passer d'un stade expérimental à un stade opérationnel. Il s'agit d'algorithmes basés sur l'utilisation de filtres destinés à optimiser la phase de "pattern-matching" déterminant l'application des productions.

L'ensemble des productions élaborées pour analyser les FPL et corriger certaines erreurs par traitements "ad hoc", ne dépassant pas la taille de 50 productions, il nous a semblé inutile d'utiliser ces méthodes (dans [13], il s'agit de systèmes dépassant 300 productions). Nos productions figurent à l'Annexe A2.2).

.../...

II.3. Intégration de processus correctifs par apprentissage

Il s'agit d'assurer à un mécanisme d'analyse-correction pré-existant la possibilité d'accroître progressivement ses moyens correctifs et de corriger ainsi un nombre croissant d'erreurs.

Supposons qu'une première erreur d'un certain type, avec un contexte donné, donne lieu à l'intégration d'une nouvelle action corrective dans le système. Notre propos est d'assurer que d'autres occurrences de ce type d'erreur, avec un contexte analogue, utilisent directement cette nouvelle action corrective. Autrement dit, ces actions doivent être intégrées au sein même des productions, où elles seront utilisées ultérieurement comme de simples tests-actions.

Dans cette approche également, nous nous sommes inspirés de l'étude de C. L. HEDRICK ([8]). Il s'agit, pour lui, d'analyser des exemples, constitués de couples objet initial, objet final , et d'en déduire des productions capables de dériver l'un en l'autre. Utilisé dans notre problème, il s'agit d'analyser des exemples constitués de couples FPL incorrect (même après les tentatives de corrections d'erreurs connues), FPL corrigé (par un opérateur humain) , et d'en déduire les modifications des productions existantes, permettant de prendre en compte le type et le contexte de l'erreur rencontrée et d'effectuer les corrections correspondantes, lors d'occurrences ultérieures semblables. HEDRICK proposait de découvrir ces modifications en scrutant les interstices résiduels entre une analyse ascendante du message source et une analyse descendante du message corrigé. Il nous a semblé lourd d'avoir à construire 2 analyseurs et nous avons choisi une méthode différente : comparer les 2 analyses ascendantes des messages sources et corrigés. La comparaison porte d'ailleurs plus sur les chaînes résultats que sur les graphes eux-mêmes. On en déduit un ensemble d'opérations superficielles élémentaires rapprochant ces 2 chaînes. Ces opérations devront être intégrées aux productions adéquates.

Le système LAURA ([10]) de LAURENT & ADAMS, conçu pour corriger les programmes d'étudiants, en comparaison avec un programme rédigé par le professeur, utilise une approche assez similaire à la nôtre :

Deux types d'actions sont prévus :

- des actions "ad hoc", pouvant être menées à priori, sans grande influence du contexte (par exemple, pour remédier à : l'utilisation d'une variable non définie ; la définition d'une variable jamais utilisée ; l'utilisation d'une variable dans un test de sortie de boucle, alors qu'aucune instruction à l'intérieur de cette boucle ne redéfinit cette variable, etc...),

.../...

- des actions de transformation du programme source, à l'aide d'opérateurs élémentaires, visant à le rapprocher du programme corrigé correspondant.

Comme pour SCAPA, ces transformations sont le résultat d'une comparaison, non pas entre objet initial et objet final, (ici programme "étudiant" et programme "professeur") mais entre des représentations internes de ces deux objets. (Dans SCAPA, cette représentation interne est constituée du graphe d'analyse et de la chaîne fabriquée).

Dans un système comme celui de HEDRICK, où les transformations comprennent des actions de destruction de portions de productions, un des problèmes relevés dans [15] est celui du contrôle pendant l'exécution. En effet, détruire des tests, dans une production, généralise celle-ci du même coup. Le degré de généralisation admis est un paramètre critique du système.

Nous nous sommes interdits la destruction d'éléments dans les productions. Leur ensemble est donc condamné à s'agrandir constamment, à moins de faire fonctionner régulièrement et off-line une enquête sur l'utilisation des séquences de code insérées, pour éventuellement, les détruire ou les regrouper.

.../...

III. SCAPA - DESCRIPTION FONCTIONNELLE

III.1. Aperçu général du fonctionnement de SCAPA

SCAPA, système d'analyse-correction avec apprentissage, fonctionne en deux étapes : une première étape est une analyse lexico-syntaxique de type "shift-reduce", ascendante, de gauche à droite ; des actions sémantiques et certaines corrections d'erreurs connues sont intégrées dans les productions ; c'est une analyse où le déterminisme est assuré par un back-tracking limité.

Si le système rencontre une erreur grave, qu'il ne peut pas corriger, les reprises peuvent cependant aboutir à une analyse satisfaisante ; sinon, le back-tracking se poursuit jusqu'à la limite (fixée arbitrairement à 3 reprises).

Si, la limite étant atteinte, aucune analyse satisfaisante n'a été effectuée, SCAPA fait une ultime reprise jusqu'à la dernière analyse jugée la plus satisfaisante puis déclenche la phase d'apprentissage :

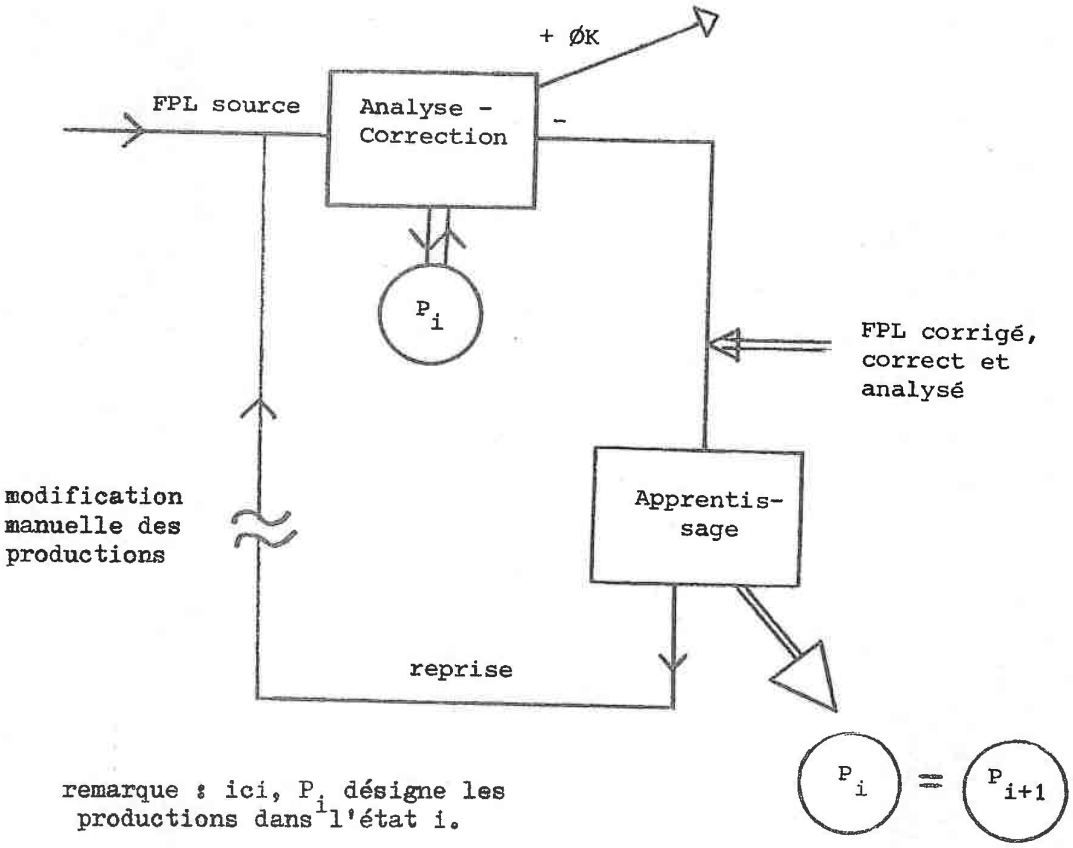
SCAPA requiert une version corrigée du message et procède à son analyse (retour à la 1ère phase). En cas d'erreur irrécupérable, dans cette analyse, SCAPA requiert une autre version corrigée. Lorsque la version corrigée est correcte, SCAPA procède à la comparaison des deux analyses : source et corrigée.
Il fait un recensement des "points d'attaque" : divergences caractérisées entre les deux chaînes résultat des analyses, et choisit parmi une liste d'opérateurs, ceux qui pourraient diminuer ou annuler ces divergences. Une évaluation conduit à choisir un point d'attaque prioritaire et un opérateur privilégié. En scrutant l'analyse du message source, SCAPA détermine sur quelle production on pourrait greffer cet opérateur, en lui adjoignant quel contexte, ou alors, quelle production il faudrait créer avec cet opérateur, et avec quel contexte. Les greffes ou les nouvelles productions sont fabriquées avec leur contexte.

L'ensemble des productions est temporairement modifié et l'analyse du message source est reprise juste en amont de l'erreur, avec les nouvelles productions.

Si l'analyse réussit, ces modifications deviennent permanentes ; sinon (c'est-à-dire si, après 3 reprises, on n'a toujours pas pu fabriquer une analyse correcte), on réitère le processus d'apprentissage. (cette fois on connaît déjà la version corrigée et son analyse).

.../...

Un schéma simplifié illustrant le processus global est :



remarque : ici, P_i désigne les productions dans l'état i.

On fait implicitement l'hypothèse que le pseudo-déterminisme de l'analyse suffira à traiter les cas corrects et que les "erreurs" trouvées correspondent effectivement à des erreurs et non à des back-tracks insuffisants, ce qui lancerait un apprentissage alors qu'il n'y a pas d'erreur et donnerait sur les productions des résultats imprévisibles.

Si l'apprentissage est incapable de fabriquer des greffes ou de nouvelles productions correctives, il y a "erreur de l'apprentissage" et le processus se termine.

Dans les évaluations de la phase apprentissage, nous avons dû introduire des méthodes heuristiques de calcul, ce qui laisse prévoir une période de réglage du système .

.../...

III.2. Description de la phase analyse-correction

On peut examiner le fonctionnement de la partie analyseur de SCAPA sous trois aspects complémentaires, d'ailleurs réunis dans le système réel : aspects purement syntaxiques, aspects sémantiques et aspects correctifs.

III.2. a) Traitements syntaxiques

L'approche syntaxique est la suivante :

- décrire le langage des plans de vols réels (c'est-à-dire avec les erreurs classiques), par une grammaire dans laquelle, par conséquent, certaines productions interprètent et corrigent ces erreurs.

- construire un analyseur utilisant cette grammaire, capable de reconnaître les FPL "corrects", de corriger les cas d'erreurs connus et d'effectuer des reprises afin de récupérer certaines erreurs structurelles.

Le résultat de cette analyse est - soit une analyse correcte (ayant éventuellement effectué des corrections par productions et/ou back-tracking) - soit le diagnostic "erreur irrécupérable".

α) La grammaire de base (débarrassée des auxiliaires ajoutés par la sémantique) est la suivante :

FPL ← "(-"<chp3>"-"...<chp15>....<chp18>")"
nous ne détaillerons que chp15

G4: <chp15> ← <crois> <SGT><eof>

<crois> ← vitesse altitude

<eof> ← ">"

<rpnt> ← t3|t4|t6

<SGT> ← t2t2|t2<rpnt>|<rpnt>t2|<rpnt><rpnt>

<SGT> ← t1<SGT>

<SGT> ← <SGT> t2

<SGT> ← <SGT><rpnt>

<SGT> ← <SGT> t7

<SGT> ← <SGT><SGT>

Remarque :

vitesse, altitude, t1, t2, ..., t7, sont les résultats de l'analyse "lexicale", pour laquelle la description OACI figure à l'Annexe A1.3.

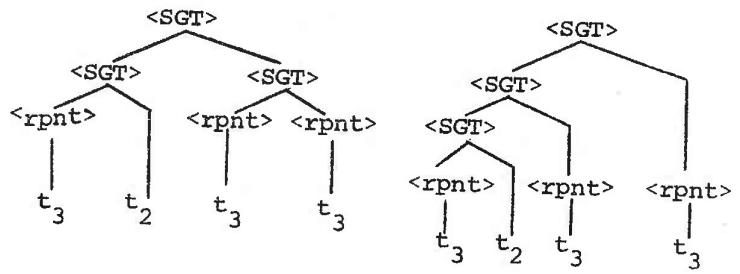
.../...

t1 est le terminal de G4 associé à : itinéraire standard de départ
t2 " " : route cataloguée
t3 " " : point de report
t4 " " : groupement
t5 " " : indicateur
t6 " " : croisière ascendante
t7 " " : itinéraire standard d'arrivée

<SGT> décrit les "segments" ou portions de route analysées
<rpnt> les points de report
<crois> les paramètres de croisière.

Remarque : t2 est aussi noté <arw> pour "airway".

La grammaire G4 est une grammaire "Context-free" ambiguë (il y a par exemple deux analyses possibles pour t3t2t3t3



Ce double choix pour G4, mérite d'être justifié car un examen rapide des spécifications de l'OACI montre que le langage des messages FPL est régulier.

- D'une part, le type d'analyseur dont nous nous sommes inspirés (C.L. HEDRICK, [8]), intégrait très naturellement des productions context-free ; nous avons utilisé cette propriété pour définir dans le champ 15, la notion de segment (classe syntaxique =<SGT>) ; le segment est une portion de route analysée, dont on a extrait les informations intéressantes (conservées dans des triplets appelés triplets résultats ou TR). L'analyse du champ 15 consiste à fusionner les segments jusqu'à l'obtention d'un segment unique. Entre autres, la règle context-free <SGT><<SGT><SGT> est utilisée pour ces fusions.

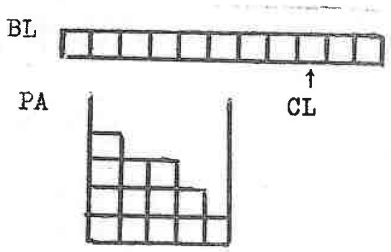
- D'autre part, nous avons voulu disposer, en cas d'erreur finalement jugée irrécupérable, de la faculté de reprendre l'analyse, en appliquant différemment les productions, ceci afin de pouvoir, en dernier lieu, choisir l'analyse, de part et d'autre des erreurs, la plus fiable. D'où la redondance de certaines productions et l'ambiguïté de la grammaire. Néanmoins, le back-tracking limité détaillé plus loin rend l'analyseur déterministe.

β) L'analyseur du champ 15 effectue l'analyse lexicale des chaînes de caractères séparées par au moins un blanc, puis effectue l'analyse syntaxique en utilisant l'algorithme d'application des productions que nous allons maintenant décrire : - symbolisons une production par

pgc ← pdr

(partie gauche) (partie droite)

Le fonctionnement de l'algorithme se représente simplement en utilisant une pile d'analyse (PA) et une bande de lecture (BL) sur laquelle se déplace de gauche à droite un curseur de lecture (CL).



Chaque terme de (PA) est une séquence de symboles.

. Initialement PA est vide, BL contient la séquence des terminaux de G4, résultats de l'analyse lexicale. CL est à gauche.

<u>Algorithme</u>

(1) Chercher dans BL la séquence de symboles à partir de CL, la plus longue, et qui soit la pdr d'une production.

. <u>si</u> aucune pdr ne correspond
. <u>alors</u> : "muter" (shift) l'élément pointé par CL :

soit x_o cet élément

- décaler CL d'un rang à droite
- concaténer x_o à droite de la chaîne au sommet de PA ;
- aller en (2),

. <u>sinon</u> : soit $A_k \leftarrow x_i \overset{n}{\text{----------}} x_j$ la production trouvée ; on fait une "réduction" :

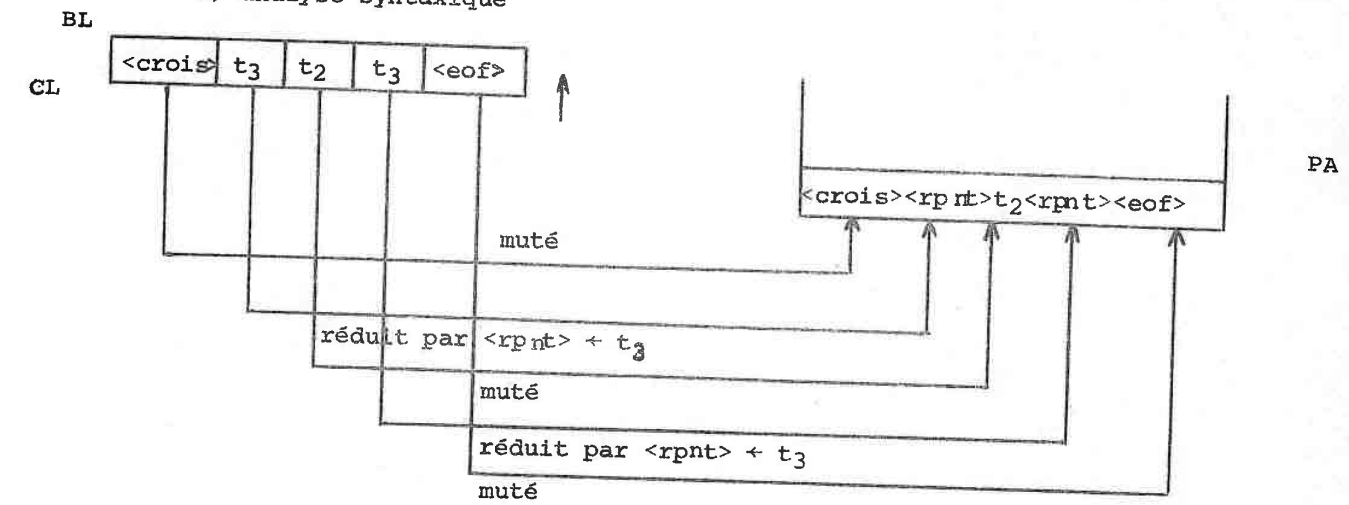
- décaler CL de n rangs à droite
- concaténer A_k à droite de la chaîne au sommet de PA
- aller en (2).

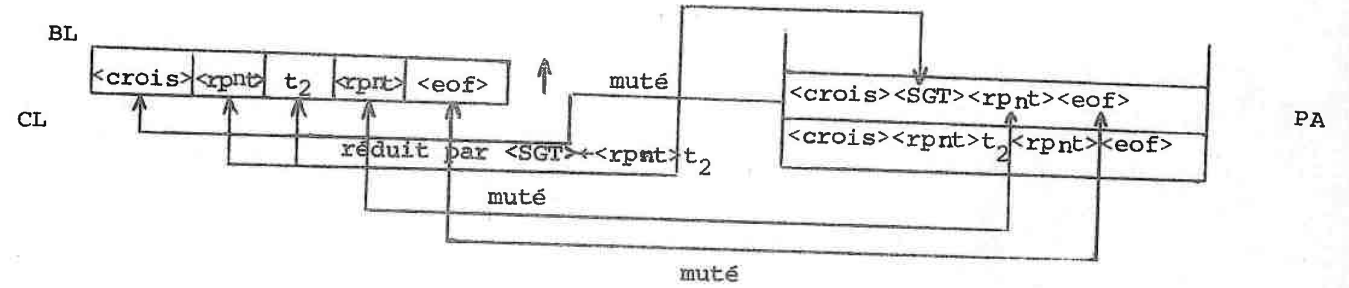
(2) <u>si</u> CL ne pointe sur aucun symbole (chaîne lue en totalité)
<u>alors</u> : <u>si</u> la chaîne sommet de PA n'a qu'un seul symbole
<u>alors</u> analyse terminée ;
<u>sinon</u> : - effacer BL ;
- recopier le sommet PA sur BL - empiler Λ sur PA
- repositionner CL à gauche
- aller en (1)
<u>sinon</u> : aller en (1).

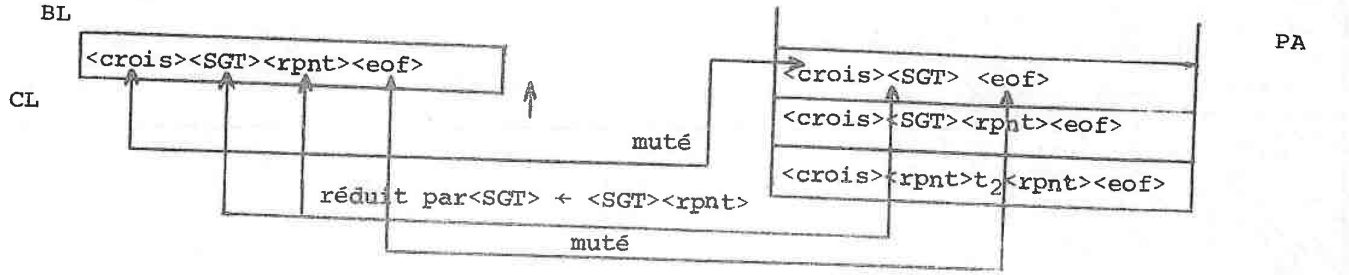
<u>exemple</u> : analyse de 0400F300 LST TB1 DKB >

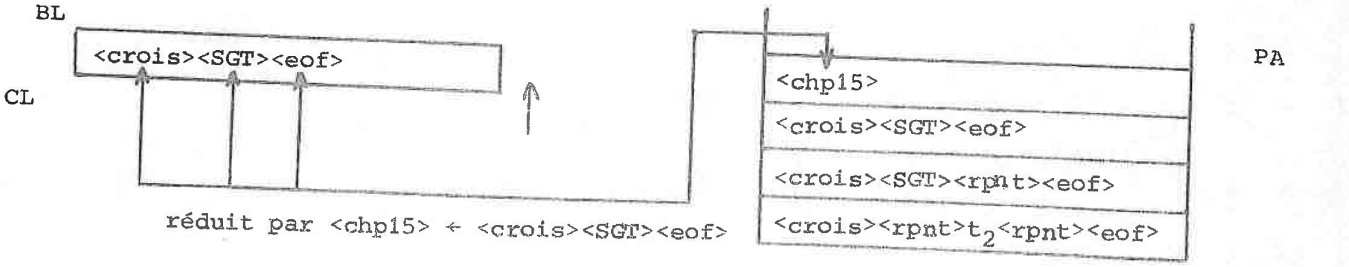
1) les analyses lexicales des groupes séparés par un blanc donnent la séquence <crois>t_3 t_2 t_3 <eof>

2) analyse syntaxique



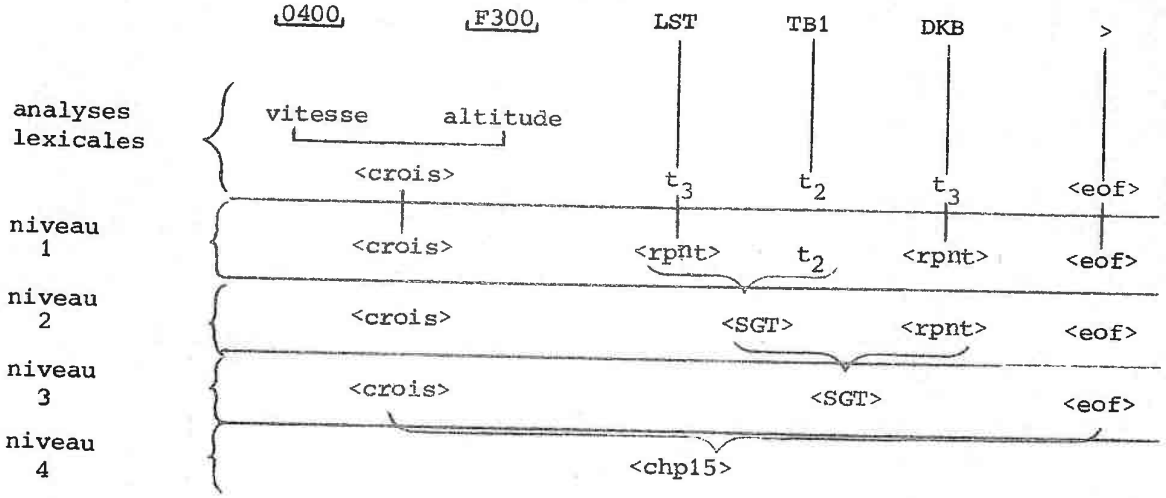






analyse terminée

Une représentation pratique des analyses est un arbre inversé :



* Introduction du back-tracking ;

On déclare qu'il y a erreur si, au pas (2) de l'algorithme, le taux d'éléments mutés pendant l'analyse à ce niveau est supérieur à une certaine limite (ici 85 %); alors la reprise s'effectue à la dernière réduction effectuée. S'il y avait eu un choix de production, on reprend avec le 2ème choix ; sinon, on se contente de muter le premier élément qui avait été lu et on poursuit l'analyse ; on trouvera un exmple de reprises développé au III.2.d).

Remarques :

lorsqu'au pas (2) le taux d'éléments mutés est inférieur à une certaine limite (50 %), on conserve un pointeur sur cette analyse qui sera l'analyse partielle conservée en cas d'arrêt de l'analyse au bout de la 3ème reprise. Cette analyse partielle servira à effectuer l'apprentissage.

Dans les analyses lexicales, l'algorithme s'applique successivement à toutes les chaînes isolées par des blancs. Si l'analyse d'un élément ne se termine pas, même après trois reprises, la meilleure analyse est conservée comme résultat (c'est une suite de résultats partiels) pour l'analyse syntaxique.

.../...

Remarque : le pas (2) avec back-tracking s'écrit :

si CL ne pointe sur aucun symbole (chaîne BL lue)
alors : si la chaîne sommet de PA n'a qu'un seul symbole
alors analyse terminée
sinon : calculer le pourcentage d'éléments mutés (soit %P)
si %P ≥ 85 % alors retour à la dernière réduction
sinon (analyse au niveau suivant)

- effacer BL
- recopier le sommet de PA sur BL
- empiler ∧ sur PA.
- repositionner CL à gauche .
- aller en (1).

L'implantation de cet algorithme utilise de fait une pile où pour chaque étape sont conservées :

- la hauteur (pour retours arrière)
- le n° de la production employée
- le résultat (chaîne)
- le niveau de l'analyse
- la liste des autres choix de productions qui auraient pu convenir à ce niveau (pour les reprises)
- et d'autres pointeurs.

Les symboles sont représentés dans le système au moyen de 2 références et une table :

La première référence est la classe syntaxique assignée par l'analyse lexicale ;

la deuxième est un numéro d'ordre dans cette classe ; ainsi un symbole est référencé par le couple (clss, i) .(noté clss!i).
Une table d'associations lie ce couple à la chaîne correspondante.
Nous avons donc défini un ensemble de classes devant couvrir tous les auxiliaires qu'utilise la grammaire.
Anticipons un peu sur la description des traitements sémantiques pour dire que certaines classes syntaxiques (comme balise ou airway) ont du être subdivisées en sous-classes (p.ex. balise interne, balise frontière, balise externe, etc.).
Notre ensemble de classes est donc muni de l'inclusion qui, prise en compte dans l'algorithme d'analyse, permet par exemple d'appliquer la production générale
<SGT> ← <SGT><rpnt> à la séquence SGT! BI!j puisque la classe BI (balise interne) est incluse dans la classe Rpnt (point de report). On trouvera la liste des classes et leur inclusion en Annexe A2.1.

.../...

III.2. b) Traitement de la sémantique

Le modèle de traitement sémantique de SCAPA est inspiré de celui proposé par HEDRICK dans [8] . Nous avons préféré cependant conférer aux tests, plus nombreux et plus imbriqués que ceux traités par HEDRICK (cf. II.2.), une structure arborescente :

En fait, nos productions se composent de :

α) la structure : c'est la partie droite

β) un arbre de tests (TP) ou triplets tests de production : c'est la description d'un algorithme de tests par un arbre où chaque noeud est une conjonction ou disjonction de triplets indiquant des tests sur les symboles correspondant à la pdr ; de chaque noeud partent 2 branches (séquence satisfaite (+) ou non satisfaite (-)).

Exemple de noeuds de test :

(x,=,t)v
("u"// x,=,t)v
(x,=,"u"// t)v
(y,SR,t)

Il y a deux sortes de triplets, ceux à valeur conditionnelle et ceux à valeur de calcul, cela dépend de la relation qui est le 2ème élément du triplet.

Exemple de triplets calcul :

(v2,SUB,v4) provoque le calcul v4=v2-v4
(*1,ØRD,z) provoque le calcul : ordonner la liste z par distance croissante à *1 ; le résultat est appelé z.

Remarque :

La notation *i, dans les triplets, désigne le i-ème élément de la structure.
La liste des relations figure en Annexe A2.1.

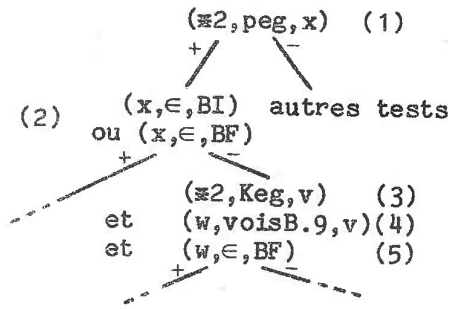
.../...

Les <u>opérandes</u> d'un triplet TP sont les 1er et 3ème éléments ; un opérande peut être :

- le ième élément de la pdr : noté ⌘i
- une classe
- une constante chaîne
- une variable.

Une variable est inconnue à la première référence dans un triplet. Ce triplet provoque une recherche qui, suivant la relation, conduit à lui assigner une valeur ou alors une liste de valeurs. A la rencontre de cette variable dans un test ultérieur, la première valeur de la liste pour laquelle ce dernier test est vérifié est la valeur unique assignée à la variable.

<u>Exemple</u>



(1) recherche <u>une</u> valeur de x ; x est assigné soit $\tilde{\tilde{x}}$
(2) tests sur $\tilde{\tilde{x}}$
(3) recherche <u>d'une</u> valeur de v soit $\tilde{\tilde{v}}$
(4) recherche d'une <u>liste</u> de valeurs de w soit $\tilde{\tilde{w}}(20)$
(5) recherche dans $\tilde{\tilde{w}}(20)$ de la première valeur satisfaisant le triplet, soit $\tilde{\tilde{w}}(io)$ si (w(io),∈,BF).

V) <u>les résultats de production</u> :

Cela comprend : description de la chaîne résultat et description des informations obtenues, à conserver : triplets résultats.

⌘) le <u>résultat</u> est une concaténation d'atomes de la pdr avec éventuellement des variables calculées par les TP. Sa classe doit être précisée et la table d'associations complétée.

On notera par exemple : R = ‖⌘1,⌘2 (R,∈,SGT) ou R = ‖1,2 (R,∈,SGT)

pour indiquer que le résultat est un segment formé de la concaténation des éléments 1 et 2 de la pdr :

SGT!i='__ __'
⌘1 ⌘2

⌘⌘) les <u>triplets résultats</u> (TR) sont destinés à véhiculer, tout le long de l'analyse, des informations acquises pendant l'exécution de certaines productions.

.../...

Ils sont analogues au TP sinon que :

- ils ne conduisent ni à des recherches ni à des assignations d'inconnues
- ils ne comportent pas de triplets de calcul.

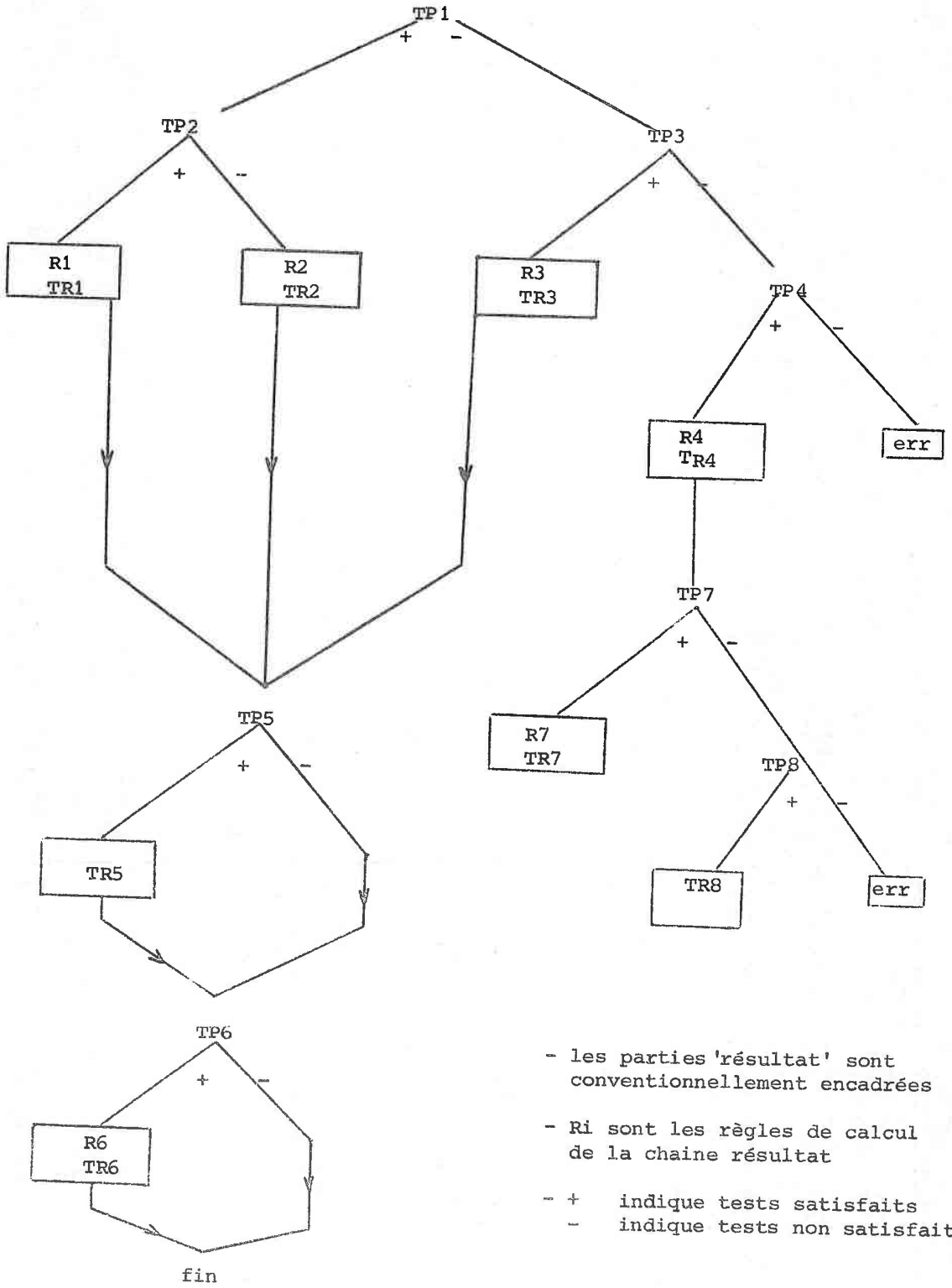
L'exécution d'une production conduit à parcourir un certain chemin dans le graphe des tests et des résultats.
Si ce chemin conduit à "erreur", cet échec est mémorisé dans une table (TABERR) utilisée par l'apprentissage ; la production ne s'applique pas, l'élément *1 est muté et l'analyse reprend. Sinon le graphe débouche sur un noeud "résultat" : la chaîne résultat est calculée, les TR sont fabriqués puis conservés dans une table qui est une extension temporaire de la banque de données statiques où figurent les connaissances fixes de SCAPA (appelé aussi "réseau" bien que son architecture soit pour l'instant, très simple). Eventuellement, les graphes peuvent se prolonger au-delà d'un "résultat" et d'autres TP peuvent modifier le premier résultat et/ou rajouter des triplets résultats.

Ci-après un schéma-type de production :
(Les schémas de productions de SCAPA figurent en Annexe A2.2.).

.../...

Un schéma - type de production :

(structure)



- les parties 'résultat' sont conventionnellement encadrées
- Ri sont les règles de calcul de la chaine résultat
- + indique tests satisfaits
  - indique tests non satisfaits

III.2. c) Les traitements correctifs

Ils sont intégrés dans l'analyse car ils ont lieu par l'intermédiaire des productions :

α) si la structure de la production est plus vaste que ce que prescrit l'OACI,

par exemple la structure (rpnt , sep , crois) tolère les groupements NTM-0400F280 ou NTM)0400F280

alors que l'OACI prescrit NTM/0400F280.

En fait, il s'agit bien d'une simple tolérance car cela n'empêche pas le système d'extraire les informations (TR) : (R,vitk,0400)
(R,altit,280)

par ailleurs, la chaîne résultat est réduite à NTM et le résultat est (R,$\in$, BI).

β) cas des séquences interdites par l'OACI ;
En principe (t2,t2) est interdit (2 routes consécutives) ; cependant, si les TP recherchent une balise intermédiaire qui soit à la fois sur les 2 routes et connue du système, le résultat est une séquence correcte : 't2,balise,t2' qui correspond à une correction connue possible ; il faut toutefois être prudent car certaines insertions sont ambiguës quand le contexte de la production est trop restreint (ici par exemple, il y a le risque que les 2 routes aient 2 intersections dont le système n'en connaisse qu'une et la propose comme correction, alors qu'en fait, il est incapable de lever l'ambiguité).

γ) correction de certaines erreurs sémantiques

exemple 1

Cas de la structure de production (rpnt ,t2)
c'est-à-dire
(rpnt , arw) :

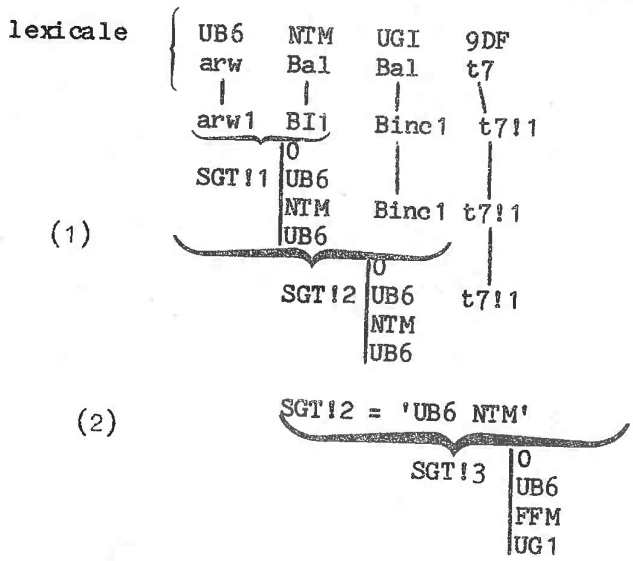
Si la balise ≠1 n'est pas située sur l'airway ≠2, les TP recherchent une balise X proche de ≠1 et située sur ≠2 ; si X existe, on insère cette balise dans le résultat (R=‖≠1,X,≠2) ; sinon, il y a erreur.

.../...

exemple 2

UB6 NTM UGI 9DF

erreur = UGI au lieu de UG1
(UGI est pris pour une balise)



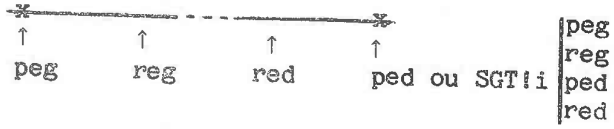
SGT!3 = UB6 NTM UG1 9DF

Remarque

* 4 informations sont associées à chaque segment (SGT) :

le point et l'airway extrémité gauche (peg et reg)
le point et l'airway " droite (ped et red)

on peut représenter un segment par :



* ici :
(1) supprime la balise inconnue dans (SGT, Binc)
(2) calcule que 9DF est un itinéraire d'arrivée pour FFM,
cherche une route entre NTM et FFM, trouve UG1 et accepte la séquence, en rajoutant UG1.

Ainsi les productions détruisent l'information erronée puis calculent l'information manquante.

.../...

III.2. d) Exemples d'analyses

Exposons les analyses de 2 messages :

- un message source comportant des erreurs dont l'analyse va s'arrêter après 3 reprises,
- une version corrigée de ce message dont l'analyse va se terminer normalement.

Ces deux messages sont :

source :

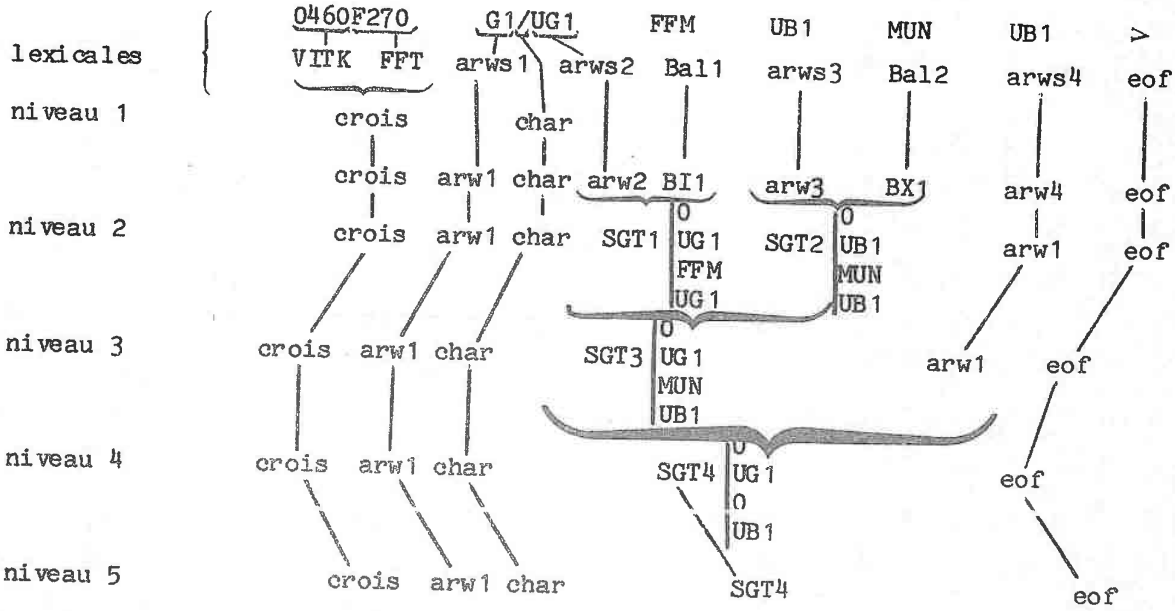
0460F270 G1/UG1 FFM UB1 MUN UB1

corrigé :

0460F270 UG1 FFM UB1

- la cause d'erreur principale est le "/" en trop.

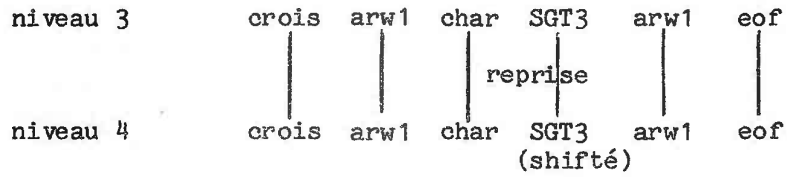
analyse du message source



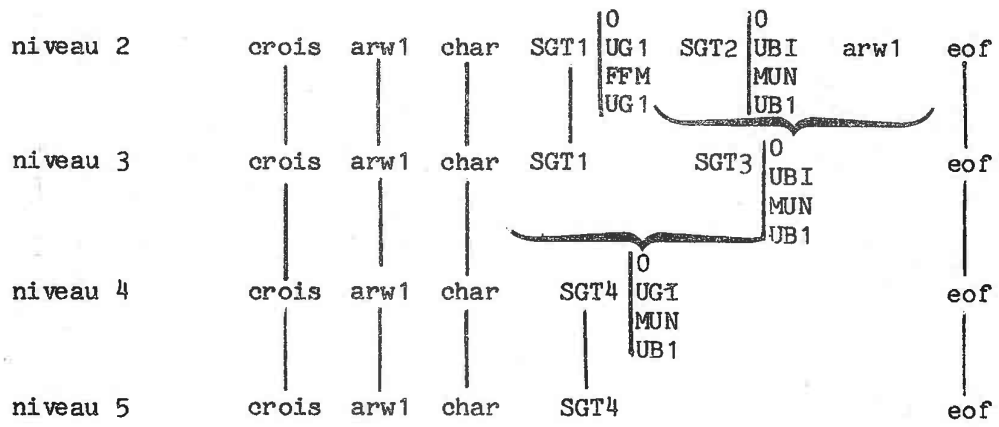
à ce niveau, tous les éléments ont été mutés, il y a donc reprise ; (% P = 100 % !).

On revient à la dernière réduction effectuée et l'on mute le 1er atome ; ici retour à : SGT4 ← SGT3 arw1
et mutation de SGT3 :

.../...



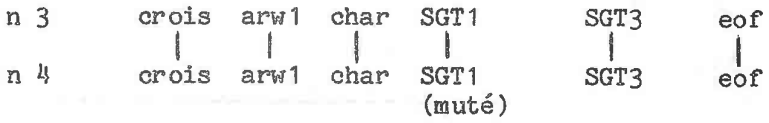
ici à nouveau % P = 100 : reprise N)=2 : reprise de la réduction SGT3 ← SGT1 SGT2



à nouveau %P = 100 : 3ème reprise

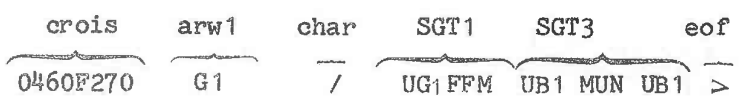
remise en cause de la réduction SGT4 ← SGT1 SGT3

reprise :



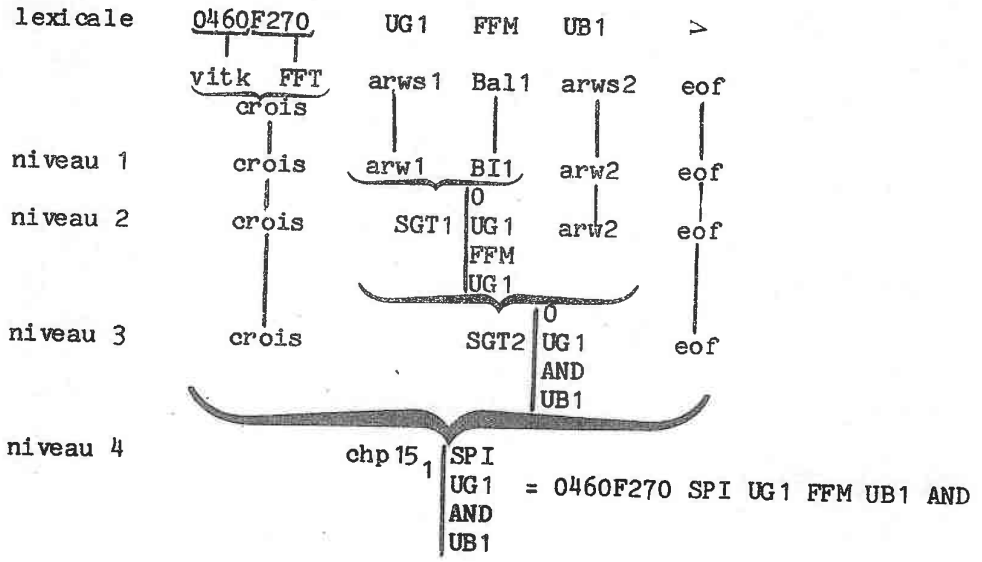
à nouveau %P = 100

cette fois, on ne fait plus de reprise, l'analyse est partielle et termine sur le niveau 3 ;
les sommets sont ici : crois, arw1, char, SGT1, SGT3 et eof qui correspondent aux chaînes :



.../...

analyse du message corrigé



L'analyse se termine ici au niveau 4.
(la dernière réduction rajoute les balises frontière SPI et AND).

La chaîne de référence sera donc

'0460F270 SPI UG1 FFM UB1 AND'.

.../...

III.3. Description de l'apprentissage

III.3. a) Philosophie de l'apprentissage

L'objectif de l'apprentissage est de proposer une, ou plusieurs modifications des productions, telles que, si l'on recommence l'analyse du message erroné, les nouvelles productions puissent fabriquer une analyse correcte du message, avec correction des erreurs. Ceci étant réalisé, le système pourra corriger seul toutes les erreurs "analogues" qui pourraient se présenter ultérieurement.

L'apprentissage vise donc à augmenter le pouvoir correctif des productions :

on utilise comme référence, une version corrigée fournie par un opérateur humain, ou plus exactement, l'analyse de ce message corrigé, ceci pour des raisons d'homogénéité des éléments comparés : d'un côté, l'analyse partielle du message source (après 3 reprises) fournit les auxiliaires du dernier niveau conservé de l'analyse (ceux qui restaient au sommet de la pile d'analyse PA) ; de l'autre, l'analyse complète du message corrigé fournit un résultat unique (car le système exige que la version corrigée fournie soit "correcte" au sens de SCAPA) ; la chaîne correspondant à ce résultat est l'élément de référence pour la comparaison.
On la compare avec l'ensemble des chaînes correspondant aux sommets de l'analyse partielle du message source.

Ainsi pour l'exemple développé au III.2. d), les sommets du dernier niveau de l'analyse source sont :

crois	arw1	char	SGT1	SGT3	eof
0460F270	G1	/	UG1 FFM	UB1 MUN UB1	>

La chaîne résultat d'analyse du message corrigé est :

0460F270 SPI UG1 FFM UB1 AND

et c'est sur ces 2 séquences que porte la comparaison.

Les détails, plutôt techniques, des mécanismes de comparaison, figurent en III.3. d).

.../...

L'apprentissage proprement dit, en tant que modification des productions, ne peut évidemment pas "inventer" ces modifications. Nous avons anticipé un certain nombre d'actions élémentaires sur les chaînes appelées "opérateurs d'apprentissage", auxquels sont associés des modifications-type des productions appelées schémas de greffe.
L'apprentissage consiste à combiner les opérateurs et le contexte au moment de l'apprentissage, puis à insérer le greffon ainsi obtenu dans les productions convenables. Des critères de minimalité permettent d'obtenir les modifications globalement et localement les plus faibles.

Ces opérateurs correspondent aux 3 actions correctives élémentaires que l'on peut entreprendre sur la chaîne source (sommets de l'analyse partielle) afin de minimiser les écarts avec la chaîne corrigée : - suppression d'un terme, - modification d'un terme, - rajout d'un terme.
Ces opérateurs semblent a priori de nature syntaxique mais ils ont par ailleurs des implications sémantiques : les actions correspondantes sont insérées dans les productions, en aval du graphe de tests sémantiques d'application. Cet aspect apparaîtra clairement en c) lorsque l'on détaillera le rôle et le fonctionnement des opérateurs.

Le module d'apprentissage propose, quel que soit le nombre des erreurs, une correction (un opérateur et une production). Certaines erreurs, néanmoins, nécessitent plusieurs actions correctives élémentaires, éventuellement, se recouvrant l'une l'autre. L'algorithme global de la phase d'apprentissage comprend donc une suite récurrente d'appels du module apprentissage, jusqu'à l'obtention d'un résultat satisfaisant. Cette approche permet parfois de ne faire qu'un apprentissage alors qu'il y avait plusieurs points d'attaques (rappelons qu'un point d'attaque est une différence caractérisée entre les chaînes résultat d'analyse, source et corrigée) ; en outre, le premier effectué est le plus simple.

.../...

III.3. b) Les opérateurs d'apprentissage

Nous avons dit qu'il y avait trois types d'actions élémentaires :

- suppression, modification, insertion

Cependant, pour augmenter leur efficacité, nous avons défini des opérateurs plus spécialisés, qui nous semblaient, dans leur ensemble couvrir un large éventail d'erreurs.

Ces opérateurs sont :

suppression

Ø1 : suppression d'un caractère isolé (sommet égal à 1 caractère)
Ø3 : suppression d'un bloc du message source

modifications

Ø4 : substitution spécifique I ⇔ 1
Ø5 : " " Ø ⇔ 0
Ø6 : " " 9 ⇔ G
Ø7 : substitutions :

(3⇔8 ; 3⇔B ; 8⇔B ; U⇔V ; 4⇔A ; 5⇔S ; 7⇔1 ; M⇔H ; R⇔P ; E⇔F ; C⇔G ; I⇔E).
Ø8 : substitution de la dernière lettre X par une autre lettre
Ø9 : substitution de la dernière lettre par une autre lettre
Ø10 : " " " " " par un caractère ou un numérique
Ø11 : substitution d'une lettre par une autre lettre
Ø12 : substitution d'un nombre par un autre nombre (dans la partie numérique d'une chaîne)

insertion

Ø13 : insertion d'un élément.

La spécialisation de ces opérateurs correspond en fait aux fréquences des erreurs ou des portions d'erreurs qu'ils prétendent corriger.
Ainsi les erreurs de caractères voisins I⇔1, Ø⇔0 ou 9⇔G nous ont semblé plus fréquentes ou plus probables que les autres substitutions de Ø7. C'est pourquoi nous avons créé Ø4, Ø5, et Ø6. Ainsi, si une greffe est réalisée avec Ø4, la tentative de correction par une substitution de I en 1 ou inversement est plus fiable car moins générale que si l'on avait dû utiliser Ø7.
De même Ø8, Ø9 et Ø10 se justifient relativement à Ø11 parce que l'on a estimé que les erreurs sur la dernière lettre d'un bloc étaient plus fréquentes que les erreurs sur les autres lettres.

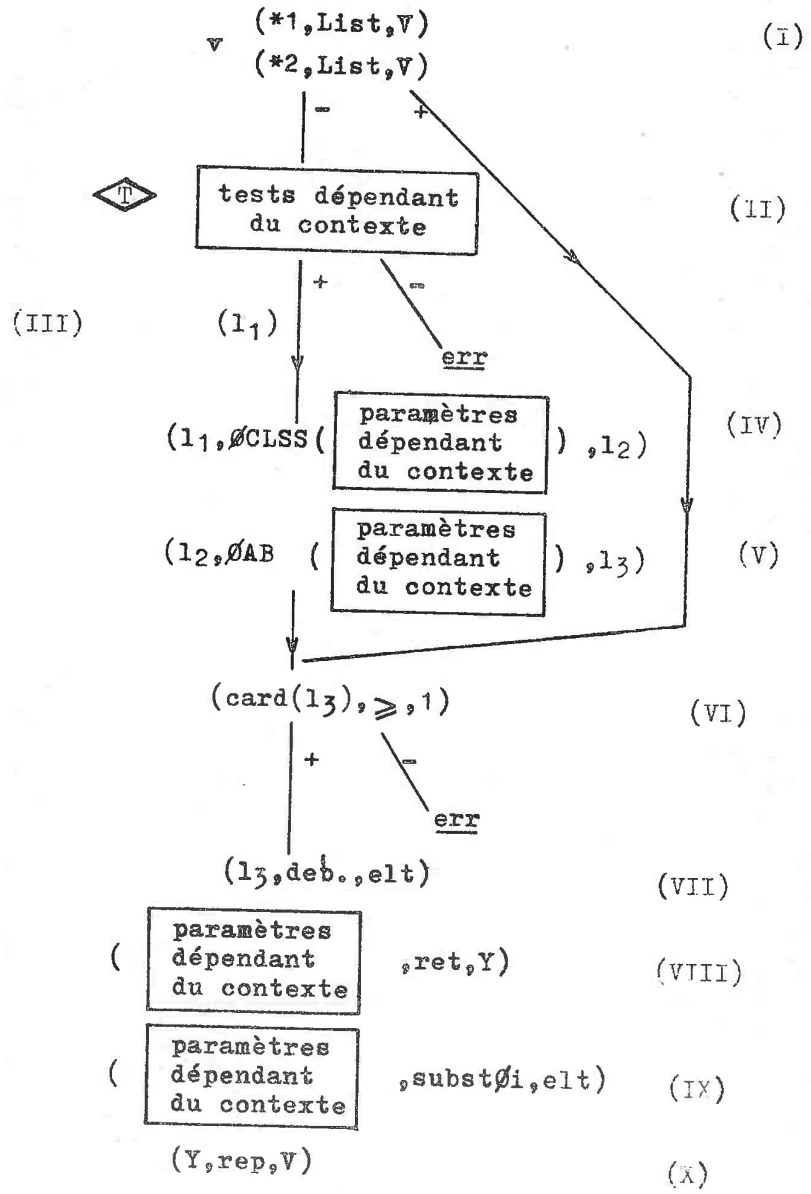
Les fréquences et la classification des opérateurs qui en découlent sont plus des avis personnels, basés sur une observation sur le site (à Karlsruhe), que des résultats d'analyses quantitatives (Ceci explique d'ailleurs comment l'opérateur Ø2 initialement pressenti de suppression des chaînes "trop longues" a été supprimé). Pour ajuster cette classification, de manière plus réaliste, ou en vue de l'adapter à un autre contexte (un autre centre de contrôle par exemple), il suffit de prévoir une pondération des opérateurs que l'on corrigera au cours du temps. La pondération actuellement utilisée est la suivante : (poids proportionnel à la probabilité) :

opérateur/poids	opérateur/poids	opérateur/poids
Ø1 : 90	Ø6 : 70	Ø10 : 55
Ø3 : 85	Ø7 : 65	Ø11 : 50
Ø4 : 70	Ø8 : 60	Ø12 : 50
Ø5 : 70	Ø9 : 55	Ø13 : 20

A chaque opérateur est associé un "schéma de greffe" qui est la partie fixe des insertions faites dans les productions. Au moment de l'apprentissage, le contexte permet de compléter ces schémas de greffe (à chaque opérateur est associé un algorithme de remplissage) ; on obtient le greffon. L'apprentissage aura déterminé ,alors à quel endroit de quelle production ce greffon doit être implanté. Lorsque l'analyse est reprise, si l'on doit appliquer une production modifiée, et si les tests conduisant au greffon sont validés, le greffon est exécuté.

.../...

Soit $\emptyset_i$, un opérateur de substitution;
Son schéma de greffe est :



<u>remarque</u> : les parties à compléter lors d'un apprentissage particulier sont conventionnellement encadrées.

.../...

Si, au cours d'une séquence d'apprentissage, il est fait appel à cet Opérateur Øi, le schéma de greffe est complété : un algorithme spécifique à Øi extrait du contexte certaines informations, les met en forme et complète ainsi les parties encadrées du schéma de greffe. Le greffon est fabriqué.

La logique détaillée du schéma de greffe ci-dessus est la suivante :

(I) : tests devant vérifier que l'on ne tente pas une substitution sur un élément qui est déjà lui-même un substitué en cours de vérification. De tels éléments sont dits "listés" et sont conservés dans une liste liée à la pile d'analyse, au cours de l'étape (IX).

(II) : $\langle T \rangle$ est un ensemble de tests qui sont entièrement fabriqués durant l'apprentissage, et serviront, lors de l'exécution du greffon, à choisir sur quelle chaîne appliquer l'opérateur de substitution et à fabriquer la liste l1 (cf. (III)) des chaînes obtenues après transformations.
L'algorithme de fabrication de ces tests est le suivant :

- soit Eo l'élément auquel on veut appliquer Øi
- trouver les relations liant Eo aux deux premiers éléments de la structure (*1 et *2).
  (en général, on a :
  soit Eo = *1
  soit Eo = *2
  soit $((*1, r_k, Eo))_{k \in K}$
  soit $((*2, r_j, Eo))_{j \in J}$
- fabriquer $\langle T \rangle$ en conséquence :
  (respectivement,
  soit (*1, subst Øi, l1)
  soit (*2, subst Øi, l1)
  soit $((*1, r_k, X))_{k \in K}$
  (X, subst Øi, l1)
  soit $((*2, r_j, X))_{j \in J}$
  (X, subst Øi, l1)

.../...

Prenons le cas où l'on veut effectuer une substitution sur le terme TB7 dans la production (SGT, RPNT), la structure réelle étant, au moment de l'apprentissage :

$$\underbrace{\text{SGT!1}}_{\text{LST TB7}} \qquad \underbrace{\text{RPNT!1}}_{\text{DKB}}$$

L'élément sur lequel on veut appliquer un opérateur de substitution (TB7) n'est ni *1 (= LST TB7) ni *2 (= DKB). Cependant, on a conservé dans les triplets résultats : (SGT!1,red,TB7), (red = route extrémité droite)

Ici le ⟨T⟩ fabriqué est donc : (*1,red,X)
(X,substØ7,l1).

(III) : la liste l1 obtenue dans cet exemple sera (T37,T87,TB1).

(IV) et (V) :
serviront, dans le greffon, à ordonner les listes successives (l1,l2,l3):

(IV) : ordonnera l1 de façon à privilégier les substitutions qui ne modifient pas la classe de la chaîne à transformer. Cette classe est à déterminer pendant l'apprentissage.

(V) : ordonnera l2 de façon à privilégier les chaînes donnant un résultat satisfaisant par application "à blanc" (c'est-à-dire sans effectuer les actions) d'une production déterminée en partie lors de l'apprentissage.

(VI) : teste le cardinal de la liste finale ordonnée, l3.

(VII) : début de l3, "elt" désigne le substitué choisi.

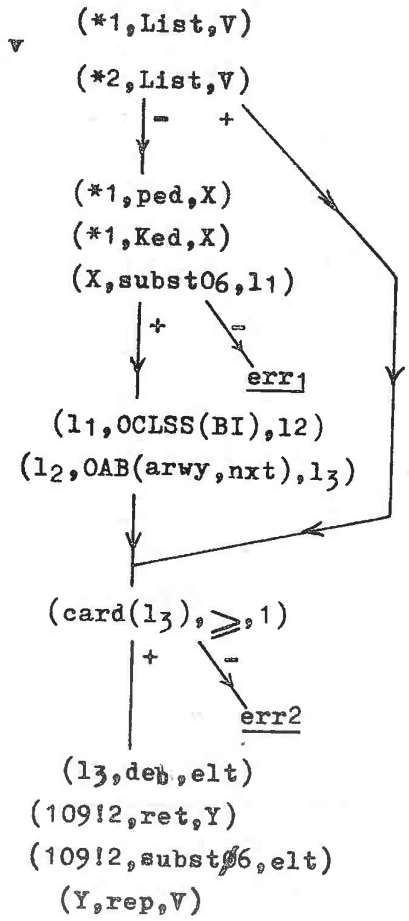
(VIII): est destiné à retrouver le 1er niveau (Y) où l'élément sur lequel on veut faire une substitution est accessible.

(IX) : effectue la substitution à ce niveau.
(si la classe change, la structure à ce niveau doit être modifiée en conséquence).

(X) : déclenche la reprise au niveau Y calculé.
(une valeur n'est effectivement attribuée à Y qu'à l'exécution du greffon, par le triplet VIII).

.../...

Dans un cas particulier d'apprentissage, le greffon obtenu pourrait être le suivant :



Ce greffon sera implanté soit à une sortie sur erreur d'une production déjà existante, soit il servira de noyau à une nouvelle production (cf. NIVB dans III.3. d)).

.../...

III.3. c) Application des opérateurs

Nous allons ici étudier les schémas de greffe et leur remplissage, correspondants aux opérateurs définis plus haut.

Comme dans la description du III.3. b), tout ce qui est à compléter lors d'un apprentissage particulier sera conventionnellement encadré, dans le schéma de greffe.

Tout d'abord, définissons un certain nombre de paramètres généraux utiles à la description de ces schémas.

* paramètres généraux

p1 = numéro disponible pour la création d'une nouvelle production.

p2,p3 = classes des 1er et 2ème termes de la pdr de la production à modifier ou à créer.

p4 = (*1 si p5 = p9 } p4 représente la variable sur
(*2 si p5 = p10 } laquelle on veut appliquer l'opérateur.

p5 = chaîne sur laquelle on désire appliquer l'opérateur.

p6 = classe de p5.

p7 = indice de p5 dans la classe p6.

p8 = (*2 si p4 = *1 } p8 représente le terme inchangé
(*1 si p4 = *2 } de la structure.

p9 = chaîne correspondant à p2

p10 = chaîne correspondant à p3.

p11 = classe de *1 (normalement incluse dans p2)

p12 = classe de *2 (normalement incluse dans p3).

p14 = classe de p8.

p15 = N° de la production à modifier.

p16 = N° du test sur lequel doit être implanté le greffon.

p19 = indice de *1 (dans la classe p11).

p20 = indice de *2 (dans la classe p12).

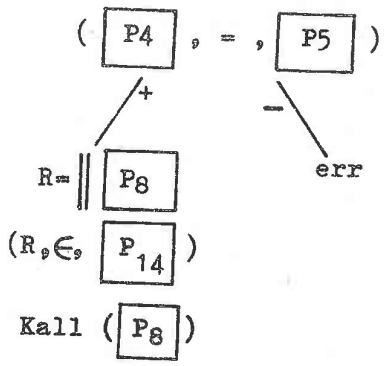
.../...

α) <u>Application de Ø1</u>

Ø1 : "supprimer un bloc du message source, réduit à un caractère (alphanumérique ou séparateur)".

Avec cet opérateur, on peut soit modifier une production existante, soit en créer une nouvelle.

Dans les deux cas, le même schéma de greffe est utilisé :



Il s'agit d'imposer que la suppression ait lieu sur le même caractère que p5, caractère ayant déclenché l'apprentissage. Pour cela, il suffit de calculer le résultat R en ne conservant que p8 (l'autre élément de la structure), et ses propriétés (action notée Kall(p8)).

En cas de création, le schéma ci-dessus constitue le corps de la production dont le numéro est p1 et la structure (p2 , p3).

En cas de modification, la production modifiée est p15 , le test où est implanté le greffon est p16 .

Un apprentissage à effet plus généralisant aurait comme test : (p4 , ∈ , p6) qui impose seulement que l'élément à supprimer ait la même classe (ou une classe incluse) que l'élément p5 ayant déclenché l'apprentissage.

.../...

Exemple

champ 15 source : 0460F270 G1/UG1 FFM UB1 MUN
" corrigé: 0460F270 UG1 FFM UB1
la dernière analyse tentée (après 3 reprises) a comme résultats :

```
   crois    arw  sep      SGT                    eof
0460 F270   G1   /    UG1  FFM  UB1  MUN          >
                 ↑(1)                 ↑(2)
```

L'analyse du champ 15 corrigé aboutit à :

```
0460  F270  UG1 FFM UB1
            ↑(1)     ↑(2)
```

points d'attaque = (1) et (2)

opérateurs applicables :

pour (1) : (Ø3,"G1") supprimer "G1"
(Ø1,"/") supprimer "/"
(Ø3,"/") "

(2) : (Ø3,"MUN") supprimer "MUN".

L'évaluation de ces possibilités conduit à choisir (Ø1,"/").

Pour ce point d'attaque, on recherche alors dans l'analyse source le "niveau de blocage" : niveau où la structure encadrant l'élément "/" reste stationnaire. Ici, dès le niveau 1, la structure (arw1,sep1,arw2) est stationnaire. De plus, le blocage a lieu par absence de production et non pas par erreur dans une production. Il va donc y avoir création d'une nouvelle production.

La production créée est :

production N° 28

structure : (arw, sep)

```
        (≡2,=,"/")
        /+    \-
              err
  R= ▯ 1
(R, ∈ , arw)
Kall(≡1)
```

p1 = 28
p2 = arw
p3 = sep
p5 = "/"
p4 = ≡2
p8 = ≡1
p14= arw
p9 = G1
p10= /

Une fois la production créée, l'analyse du message source est reprise et produit le résultat d'analyse final :

"0460F270 G1 UG1 FFM UB1 MUN"

qui est une analyse satisfaisante.

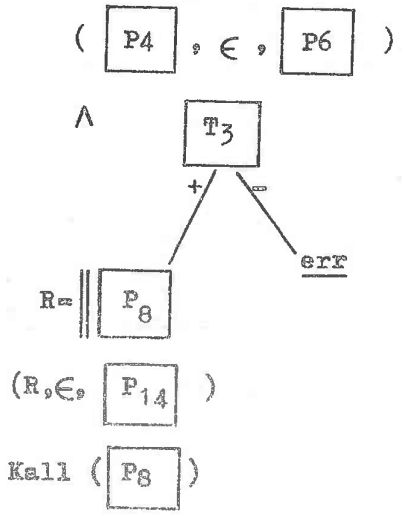
.../...

β) Application de Ø3

Ø3 : "suppression d'un bloc du message source".
Comme pour Ø1, Ø3 peut servir à créer ou modifier une production.

Cependant, il serait trop restrictif ici d'imposer que la chaîne à supprimer soit identique à la chaîne ayant provoqué l'apprentissage. Par exemple, si lors de l'apprentissage, on veut supprimer la route UG1, ce type de greffe ne fonctionnerait ultérieurement que pour supprimer UG1, et, par exemple, pour supprimer UG5, il faudrait une autre greffe. Il serait par contre trop généralisant de n'imposer que l'appartenance à une même classe. Dans l'exemple précédent, la greffe conduirait à supprimer, sans vérifications supplémentaires, toutes les airways. On tente donc d'imposer en plus de cette appartenance, que la chaîne à supprimer ait les mêmes propriétés que la chaîne p5 ayant provoqué l'apprentissage. (par exemple, propriétés de type, espace, règle, vitesse, altitude. Ces tests supplémentaires sont intégralement fabriqués lors de l'apprentissage (ils sont notés T_3).

Le schéma de greffe est le suivant :



- l'algorithme de calcul de T_3 est le suivant :

  ⁑ rechercher les triplets du réseau où figure p5 soit $((p5,ri,ci))_{i\in I}$

  ⁑ fabriquer alors T_3 : $((p4,ri,ci))_{i\in I}$

.../...

par exemple, si l'élément à détruire est UG7, on trouve dans le réseau :

(UG7,esp,'S') espace supérieur
et (UG7,typ.,'C') type civil.

alors si p4 vaut ⋇2, T3 sera :

(⋇2,esp,'S')
(⋇2,typ.,'C')

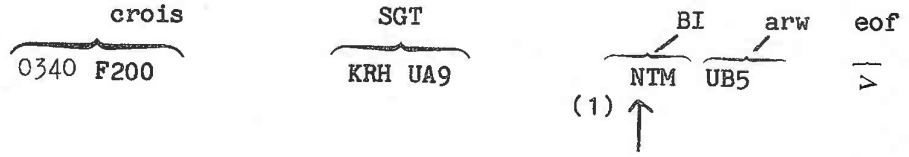
imposant pour la suppression de ⋇2, que cet élément ait les mêmes propriétés qu'avait UG7, lors de la phase d'apprentissage

<u>exemple</u> :

champ 15 source :	0340F200	KRH/0340F310	UA9	NTM	UB5
" corrigé :	0340F200	KRH/0340F310	UA9	UB5	

⋇ <u>remarque</u> : l'erreur est que NTM n'est pas une intersection de UA9 et UB5.

La dernière analyse source conduit à :



L'analyse du message corrigé donne:

0340F200 KRH UA9 EDE UB5
(1)

1 point d'attaque : (1).

.../...

Les possibilités d'application d'opérateur sont :

(Ø3,"NTM") supprimer NTM
(Ø13,"EDE") rajouter EDE

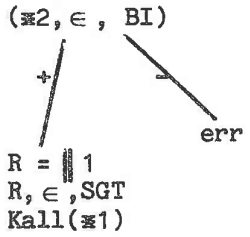
La première solution est retenue.

Au niveau du blocage, on a enregistré (table TABERR) que la production pro24 (SGT,rpnt) avait échoué sur le test N° 39. C'est cette production qui va être modifiée :

Les paramètres, calculés au moment de l'apprentissage, sont :

P4 : ⋇2 T3 est vide
P6 : BI (aucune propriété spécifique de "NTM")
P8 : ⋇1
P14: SGT
P5 : NTM
P9 : KRH UA9
P10: NTM

Le greffon fabriqué est :

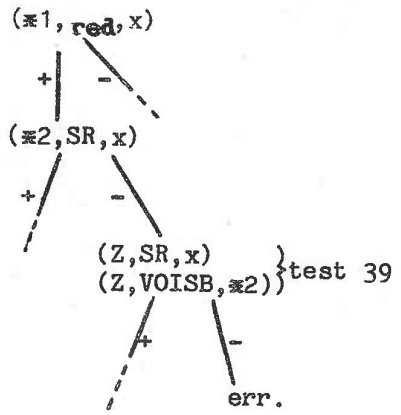


Ce greffon est inséré dans pro24 au niveau du test N° 39 :

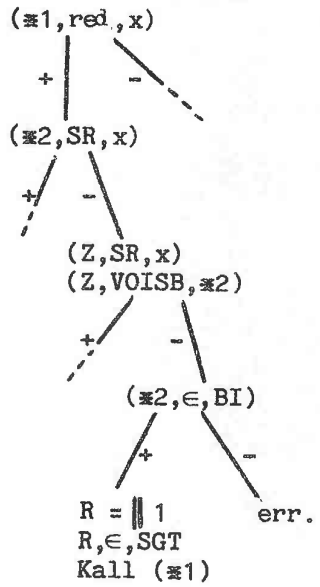
.../...

<u>ancienne pro24</u> (SGT, rpnt)

(sous-cas : $\frac{?}{x}$ x)



<u>nouvelle pro24</u> (SGT, rpnt)



L'analyse du message source est reprise et conduit à l'analyse satisfaisante dont le résultat est :

'0340F200 KRH UA9 EDE UB5'

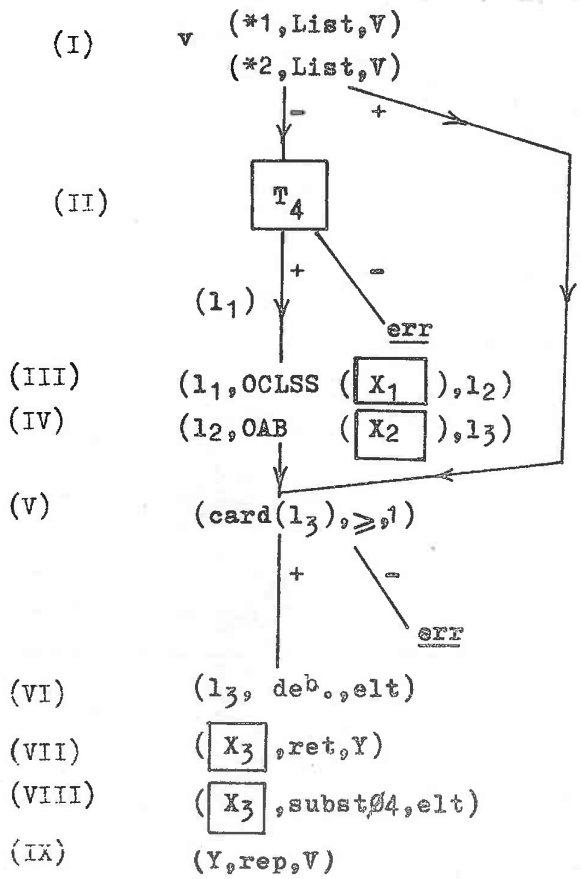
.../...

γ) Application de Ø4

Ø4 : "substitution spécifique I ⇔ 1 dans un bloc du message source".

- La création d'une production ne peut pas être envisagée avec cet opérateur ni aucun des autres opérateurs de substitution (Ø5 à Ø12). En effet, les tests préliminaires ne peuvent pas être inventés ; on devrait immédiatement tenter une substitution ! Si le cas se présente, il provoque une alerte système.

- Modification d'une production



Le schéma ci-contre devra être complété pendant l'apprentissage. A l'exécution, le greffon correspondant aura comme fonction de calculer une liste des substitués possibles (triplets T4), (remarque (II)) puis d'ordonner et de filtrer cette liste (remarques (III) et (IV)).

Selon les résultats du test (V) :

- soit il y a erreur (si la liste finale l3 est vide)

- soit la substitution est réalisée et il y a reprise à un niveau Y calculé. (cf remarques (VI),(VII), (VIII),(IX).

* les paramètres T4, X1, X2, X3 sont calculés au moment de l'apprentissage, en fonction du tableau suivant.

.../...

cas	T4	X1	X2	X3
cas 1 : P5 = P9 l'élément où l'on veut faire une substitution est le 1er terme de la production.	(⌘&,subst∅4,l1)	P11	$\widehat{(ceg(⌘1),P11)}$	P11!P19
cas 2 : P5 = P10 l'élément où l'on veut faire une substitution est le 2ème terme de la production.	(⌘2,subst ∅4,l1)	P12	$\widehat{(P12,CED(⌘2))}$	P12!P20
cas 3 :on a trouvé des relations $(ri)_{i \in I}$ tq (⌘1,ri,P5)	$((⌘1,ri,X))_{i \in I}$ et (X,subst∅4,l1)	CLSS(X)	$\widehat{(CEG(X),CLSS(12)}$	CLSS(X)!IND(X)
cas 4 : on a trouvé des relations $(rj)_{j \in J}$ tq (⌘2,rj,P5)	$((⌘2,rj,X))_{j \in J}$ et (X,subst∅4,l1)	CLSS(X)	$\widehat{(CLSS(12),CED(X))}$	CLSS(X)!IND(X)

Notations :

On utilise les fonctions :

CEG = classe de l'élément à gauche de
CED = " " " à droite de
CLSS = classe de
IND = indice de

$\widehat{(x,y)}$ signifie structure (x,y) généralisée jusqu'à être la pdr d'une production existant.

Remarques

(I) les triplets (*1,LIST.,V) sont destinés à former
(*2,LIST.,V)

un verrou avant application de l'opérateur, de sorte qu'on ne puisse pas redéclencher un apprentissage au moment de l'exécution de la greffe.

En effet, la première fois que l'on passe par le test P16 de la production P15, dans une analyse, on fabrique une liste de substitués possibles ; la substitution la plus favorable (d'après les tests) est choisie et l'on fait une reprise juste avant l'endroit de cette substitution ; cependant, l'élément substitué est dit "listé" et la liste des substitués lui est associée. Si, lors de la reprise, on repasse dans la production P15, test P16, la structure comportant un élément listé, (ce qui signifie que le substitué reprovoque la même erreur), il ne faut pas reconstruire cette liste ; il suffit de prendre l'élément suivant. Le mécanisme de verrouillage est effectué au moyen des triplets (*1,LIST,V) v (*2,LIST,V).

(II) T4 est une séquence de triplets dans lesquels l'élément à substituer a une représentation ne dépendant que de *1 ou *2, c'est-à-dire que cet élément est :

- soit *1
- soit *2
- soit lié à *1 ou *2 par des triplets.

Dans le 3ème cas, ces triplets sont calqués sur ceux existant lors de l'apprentissage pour fabriquer X, l'élément à substituer.
Ensuite, le triplet (*1,SUBST(Ø4),l1)(respectivement (*2,SUBST(Ø4),l1) et (X,SUBST(Ø4),l1)) fabrique la liste des substitués possible de *1 (respectivement, *2 et X) soit l1.

(III)Le triplet (l1,ØCLSS(X1),l2) est destiné à ordonner la liste l1 en fonction de la classe X1 (classe de l'élément à substituer) ; les éléments de l1 de classe plus voisine de X1 (pour la relation ⊂) sont mis en tête de liste ; résultat = liste l2 ; les éléments inconnus dans le réseau sont éliminés.

(IV) Le triplet (l2,ØAB (X2),l3) est destiné à filtrer à nouveau l2 pour favoriser les éléments qui, analysés par la production de structure X2 provoquent une analyse correcte. L'application de cette production est faite "à blanc" (les actions résultats ne sont pas effectuées) ; cette production teste la validité des substitués soit par rapport au contexte gauche soit par rapport au contexte droit.
résultat : liste l3.

.../...

(V) En cas de succès du test (V), les actions décrites par le schéma de greffe sont :

(VI) prendre le début de la liste 13 ; soit "elt" le substitué.

(VII) retrouver le niveau Y le plus haut où l'on peut isoler X3 pour y faire la substitution.

(VIII) faire, à ce niveau, les substitutions de classe nécessaires (si, par exemple, le substitué n'a pas la même classe que l'élément initial, modifier la structure de ce niveau).

(IX) effectuer la reprise de l'analyse au niveau Y.

NB : toutes ces actions ont lieu à l'exécution du greffon seulement, et pas du tout au moment de la fabrication du greffon.

Les opérateurs Ø5, Ø6 et Ø7 fonctionnant presque identiquement à Ø4, nous ne donnerons qu'un seul exemple d'application ; se référer à Ø7.

.../...

δ) <u>Application de Ø5</u> (substitution Ø ⇔ o)

identique à Ø4 : remplacer "Ø4" par "Ø5".

ε) <u>Application de Ø6</u> (substitution 9 ⇔ G)

identique à Ø4 : remplacer "Ø4" par "Ø6".

ζ) <u>Application de Ø7</u> (liste de substitutions).

identique à Ø4 : remplacer "Ø4" par "Ø7".

Ici SUBST Ø7 signifiera : effectuer toutes les substitutions possibles de type Ø7 ; la liste 11 sera simplement plus longue que pour Ø4.

<u>Exemple</u> :

chp15 source : 0520F270 LST TB7 DKB
chp15 corrigé 0520F270 LST TB1 DKB

La dernière analyse source donne :

```
     crois  SGT              BI    eof
  ┌──────┐┌──────────────┐  ┌──┐  ┌─┐
  0520   F270 LST  TB7        DKB  >
                   ↑(1)
```

Le résultat de l'analyse du chp 15 corrigé est :

```
0520   F270 LST  TB1  DKB
            ↑(1)
```

1 point d'attaque : (1)

applications possibles d'opérateurs :

(Ø3,TB7)	supprimer TB7
(Ø7,TB7)	substituer dans TB7
(Ø12,TB7)	changer la partie numérique de TB7
(Ø13,TB1)	rajouter TB1.

L'évaluation détermine (Ø7,TB7).

.../...

Au niveau de blocage, il y avait eu erreur dans la tentative d'application de pro24, test N° 18.

La greffe aura lieu à cet endroit.

Calcul de la greffe :

P5 = TB7
P9 = LST TB7
P10 = DKB

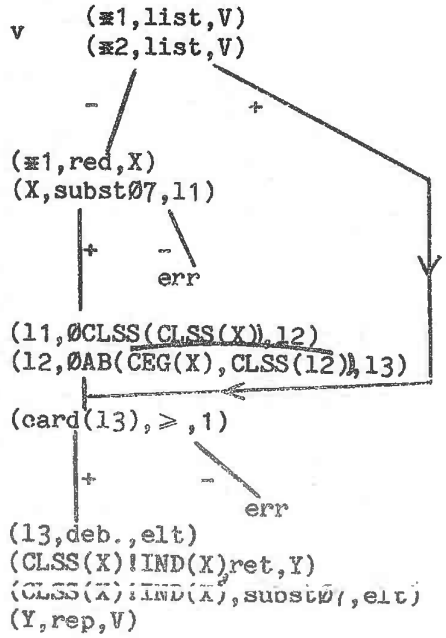
On est dans le cas 3 du tableau, car il existe le triplet :

(SGT!1,$\widetilde{red}$,TB7) ($\widetilde{red}$ = pseudo-route extrémité droite).

Alors T4 est la suite de triplets :

(*1,red,X)
(X,substØ7,11)

Le greffon fabriqué est :



La reprise de l'analyse donne le résultat désiré :

"0520F270 LST TB1 DKB"

.../...

7) Application de Ø8

Ø8 : "substitution de la dernière lettre (si c'est "X") par une autre lettre.

le schéma est identique à celui de Ø4 sauf :

1) remplacer SUBST(Ø4) par subst(Ø8) qui est l'opération substituer la dernière lettre par A puis B puis C puis .... puis Z (sauf X).

2) T4 doit imposer en outre que la dernière lettre de l'élément à substituer soit "X" ; reprenons les 4 cas de calcul de T4 :

cas 1 : T4 = (*1,LST,"X")

+ | - err

(*1,SUBST(Ø8),11)

cas 2 : T4 = (*2,LST,"X")

+ | - err

(*2,SUBST(Ø8),11)

cas 3 : T4 = $((*1,ri,X))_{i \in I}$

+ | - err

(X,LST,"X")

+ | - err

(X,subst(Ø8),11)

et cas 4 :

T4 = $((*2,r_j,X))_{j \in J}$

+ | - err

(X,LST,"X")

+ | - err

(X,SUBST(Ø8),11)

LST est la fonction "Last", dernière lettre d'une chaîne.

.../...

θ) Application de Ø9

Ø9 = "substitution de la dernière lettre par une autre lettre".
Le schéma est identique à Ø4 en remplaçant "Ø4" par "Ø9".

ι) Application de Ø10

Ø10= "substitution de la dernière lettre par un caractère ou un numérique.

Schéma identique à celui de Ø4 ; remplacer "Ø4" par "Ø10".

κ) Application de Ø11

Ø11= "substitution d'une lettre par une autre lettre.

même schéma que Ø4 en remplaçant "Ø4" par "Ø11".

λ) Application de Ø12

Ø12= "substitution d'un nombre par un autre nombre dans la partie numérique d'une chaîne".

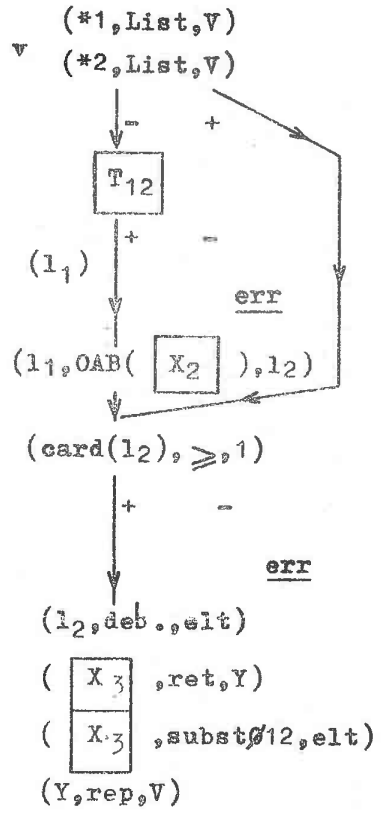
* pour les mêmes raisons que Ø4 on n'envisage pas la création d'une production avec cet opérateur.

* modification

Ø12 est analogue à Ø4, mais ici, vu le grand nombre de cas possibles, on impose aux éléments sur lesquels on fera une substitution d'appartenir à la même classe que l'élément à modifier. La filtration se fait non plus par la fonction ØCLSS qui ne faisait qu'ordonner les substitués mais par RCLSS qui recherche dans les éléments connus de cette classe si certains n'ont pas la même partie littérale que l'élément à modifier.

.../...

Le schéma de greffe est le suivant :



Le calcul de T12 s'effectue différemment suivant les mêmes 4 cas que T4 (dans Ø4).

X1,X2 et X3 : même calcul que pour Ø4

<u>cas 1</u> T12 = (≠1,LITT,Z)

+ : (Z,RCLSS(X1),11)

− : err

<u>cas 2</u> T12 = (≠2,LITT,Z) "

+ : (Z,RCLSS,(X1),11)

− : err

<u>cas 3</u> alors T12 = $((\neq 1, ri, X))_{i \in I}$ "

+ : (X,LITT,Z)

− : err

(X,LITT,Z)

+ : (Z,RCLSS(X1),11)

− : err

<u>cas 4</u> alors T12 = $((\neq 2, rj, X))_{j \in J}$

+ : (X,LITT,Z)

− : err

(X,LITT,Z)

+ : (Z,RCLSS(X1),11)

− : err

LITT est la fonction : " a comme partie litérale".

.../...

μ) Application de Ø13

Ø13 : insertion d'un élément dans le message source.

La stratégie pour déterminer le terme à insérer, qui est inconnu en général, mais connu lors de l'apprentissage, est la suivante : s'inspirer des relations liant ce terme connu avec les éléments de son contexte ; à l'exécution, les mêmes relations doivent permettre de trouver un élément jouant le même rôle.

Par exemple, si lors de l'apprentissage, on trouve qu'il faut insérer UG1 après UG31 FFM et avant NØR :

On recherche les relations liant UG1 et son contexte :

on trouve

(SGT,ped,FFM)	qui établissent un lien entre
(SGT,Ked,FFM)	⋇1 et UG1
(FFM,SR,UG1)	(⋇1 est le 1er élément de (SGT,rpnt))

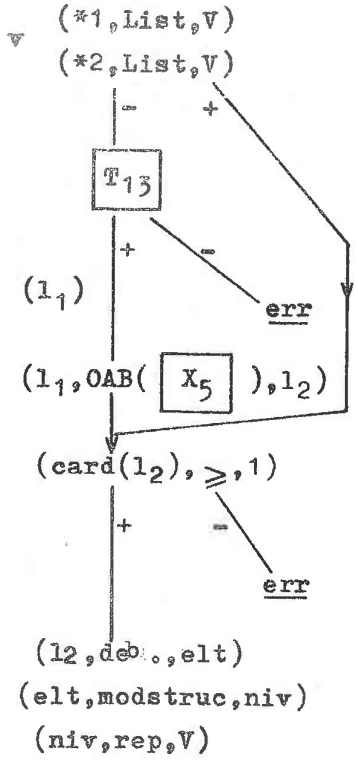
les triplets inferrés seront :

(⋇1,ped,X)	
(⋇1,Ked,X)	
(X,SR,l1)	l1 = liste des éléments possibles à insérer.

.../...

Le schéma de greffe est le suivant :

avec X5 = (CLSS(l1),P12)
(mêmes notations que dans Ø4)



.../...

T13 est une liste de triplets qui devra être fabriquée pendant l'apprentissage en fonction des règles suivantes :

relations trouvées	T13
$((\equiv 1, ri, P5))_{i \in I}$	$((\equiv 1, ri, l1))_{i \in I}$
$((\equiv 2, rj, PS))_{i \in J}$	$((\equiv 2, ri, l1))_{j \in J}$
$((\equiv 1, rk, Xo))_{k \in K}$ et (Xo,rl,P5)	$((\equiv 1, rk, X))_{k \in K}$ et (X,rl,l1)
$((\equiv 2, rm, X1))_{m \in M}$ et (X1,rn,P5)	$((\equiv 2, rm, X))_{m \in M}$ et (X,rn,l1)

<u>Remarque 1</u> : (elt,modstruc,niv) modifie la structure du niveau actuel de l'analyse en y insérant CLSS(elt)!i et en créant le lien CLSS(elt)!i = elt.

<u>Remarque 2</u> : Si le réseau contient des informations relatives à P5, du genre $((P5, rp, Cp))_{p \in P}$, alors les triplets $((l1, rp, Cp))_{p \in P}$ peuvent être adjoints à T13, limitant la liste l1 aux seuls éléments ayant les mêmes propriétés que P5.

.../...

Exemple

Message source : 0440F260 FFM NOR TR10

L'erreur est que FFM et NOR ne sont pas reliées par une route aérienne, sont éloignées, et qu'il n'y a pas de balises significatives entre les deux.

Le message corrigé est :

0440F260 FFM TG1 NTE TR7 NØR TR10

L'analyse du message source produit :

crois	BI	(1)	BF	arw	eof
0440F260	FFM		NOR	TR10	>

celle du message corrigé produit :

0440F260 FFM TG1 NTE TR7 NØR TR10

(1)

1 seul point d'attaque (1)
opérateurs possibles :

(Ø13,TG1)
(Ø13,NTE)
(Ø13,TR7)

L'évaluation favorise le rajoût de l'élément le plus proche du contexte gauche : (Ø13,TG1).

Au niveau du blocage, il y avait erreur dans la production (RPNT,RPNT) : pro23, test N° 16.

Les relations liant TG1 et ж1, ou ж2 sont :

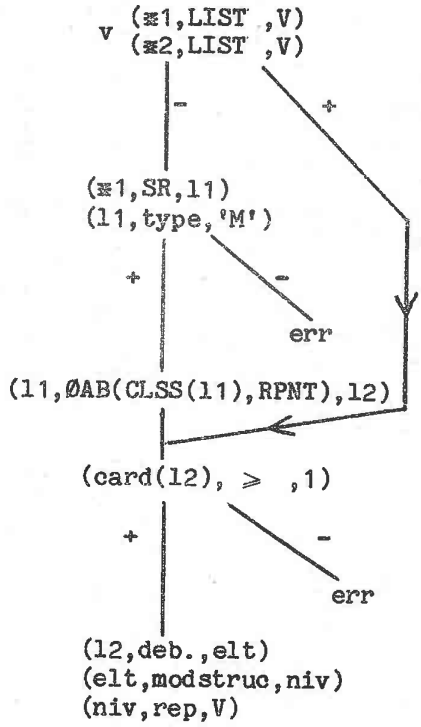
(ж1,SR,TG1), et, par ailleurs, l'analyse a fabriqué :
(TG1,type,'M') (route de type militaire)

T13 sera donc ici :

(ж1,SR,l1)
(l1,type,'M').

.../...

Le schéma de greffe suivant devra être inséré dans la production pro23 , au niveau du test n° 16 :



L'analyse reprise après cette modification produit le résultat satisfaisant :

'0440F260 FFM TG1 NTE NØR TR10'.

.../...

III.3. d) Traitements préliminaires aux greffes

Les premières actions liées à l'apprentissage sont la requête et l'analyse du message corrigé, nous l'avons vu au III.3.a).

Notons C' la chaîne obtenue par analyse du message, corrigée et (S'_i) l'ensemble des chaînes, résultats partiels, de l'analyse du message source.

D'après les restrictions décrites en I.3., nous nous restreignons au chp 15. Un module appelé ISØLC15 recherche donc dans les piles d'analyses les portions relatives au champ 15. ISOLC15 est appliqué 1): à l'analyse donnant (S'_i), 2): à l'analyse donnant C'.
Ensuite le module nommé DELIM délimite le champ 15 pour l'apprentissage, en supprimant l'élément <crois> qui commence normalement ce champ, et l'élément <eof> qui termine (S'_i) (crois est purement opérationnel). Soit C'd et (S'_{id}) les délimités.
Le module MATCH détermine alors les blocs identiques dans C'_d et (S'_{id}).
Le module RPA recherche les points d'attaques, divergences caractérisées entre C'd et (S'_{id}). Il y a 2 catégories de points d'attaque :

- type A :

divergence(s) localisée(s) entre des blocs identiques de C'd et S'_{id}, et dans le même ordre :

ex:

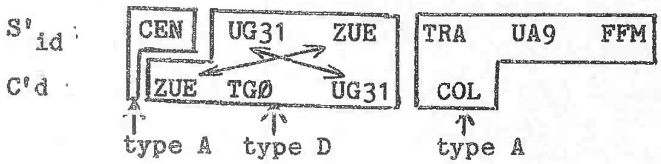
S'_{id} : LST TB7 DKB
C'd : LST TB1 DKB

point d'attaque
type A

- type D :

(désordre) divergence localisée entre des blocs identiques de C'd et S'_{id}, mais dans le désordre

ex:



.../...

Ensuite, une batterie d'heuristiques (module HEURS) recherchent pour chaque point d'attaque, les opérateurs qui, appliqués à un bloc inclus dans ce point d'attaque, permettent de réduire les divergences observées : le résultat est donc une série de triplets (point d'attaque, opérateur, bloc à traiter). Pour l'exemple de PA (point d'attaque) type A ci-dessus cette liste est :

(PA1,Ø3,TB7) (PA1,Ø10,TB7)
(PA1,Ø13,TB1) (PA1,Ø11,TB7)
(PA1,Ø7,TB7) (PA1,Ø12,TB7)

pour l'exemple de PA type D, la liste est :

(PA1,Ø3,CEN)
(PA2,Ø13,TGØ)
(PA3,Ø3,TRA)
(PA3,Ø3,UA9)
(PA3,Ø3,FFM)
(PA3,Ø13,CØL)

Le module EVALMP choisit le point d'attaque qui va être développé en premier. Pour cela, il tient compte :

1) du poids de l'opérateur,

2) de l'importance de la réduction de divergence, entraînée par l'application de l'opérateur,

3) du fait que, si le point d'attaque intervient dans une portion de (S'id) où une analyse partielle avait déjà pratiqué une réduction convenable, il y a lieu de défavoriser légèrement ce point d'attaque (qui remet en cause une analyse partielle fructueuse).

Le module NIVB recherche ensuite le niveau de blocage relativement au bloc à traiter (niveau de l'analyse où l'on décide de modifier ou créer une production). Il scrute TABERR (table des erreurs) et détermine s'il faut modifier une production (et laquelle, à quelle N° d'erreur) ou en créer une nouvelle (et avec quelle structure).

Le module RP recherche tous les paramètres généraux nécessaires au remplissage du schéma de greffe.

le module APPL enfin, remplit les schémas de greffe à l'aide des paramètres généraux, et, pour certains opérateurs, recherche des paramètres particuliers.
A ce niveau, le greffon est prêt ; l'endroit exact de l'insertion (ou la création) sont connues.
La greffe est effectuée et l'analyse est reprise.
On peut suivre ces différentes étapes dans les exemples proposés au III.3.c)

.../...

IV. SCAPA : Expérimentations

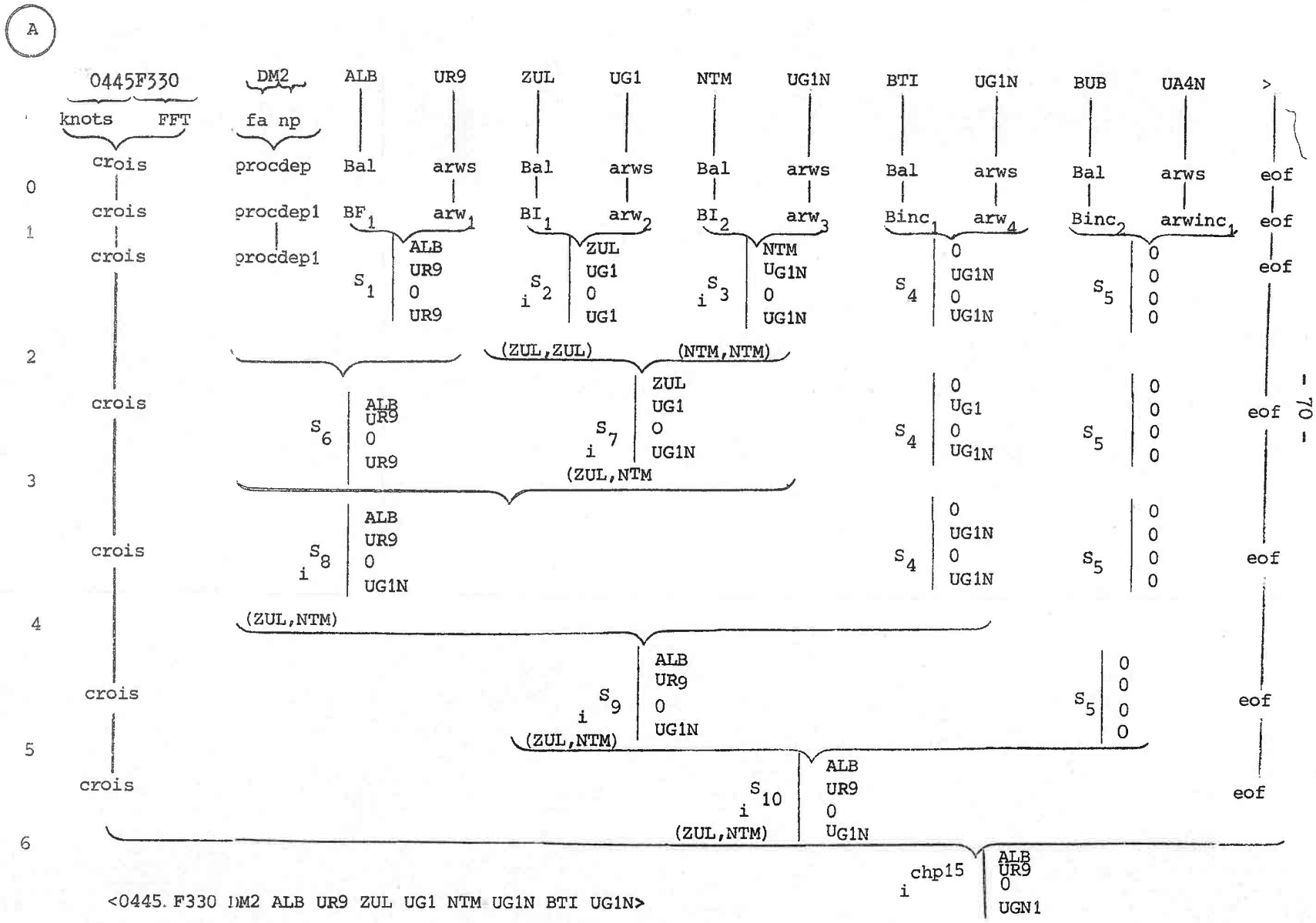
IV.1. Expérimentation de l'analyse - correction

L'expérimentation de l'analyse correction a consisté à faire analyser un jeu d'essai d'une trentaine de messages ne comportant pas d'erreurs graves, donc ne déclenchant pas d'apprentissage. Au cours de cette expérimentation, l'ensemble des productions a été mis au point et affiné pour obtenir des analyses détaillées et efficaces (pour les corrections), dans la mesure du possible. L'expérimentation de l'apprentissage a légèrement interféré avec celui de l'analyse correction en ce que l'on a été conduit à modifier certains algorithmes, relatifs aux reprises et arrêts sur reprises, et qu'il a donc fallu tester à nouveau les 30 messages.

Tout notre travail d'expérimentation et de mise au point a finalement résulté en la mise au point d'un ensemble cohérent de productions. Cela fait, on doit remarquer plusieurs choses, concernant les analyses, remarques étayées par des exemples tirés du jeu d'essai :

1) limite de taille du champ 15, liée à la grandeur des tables du système;

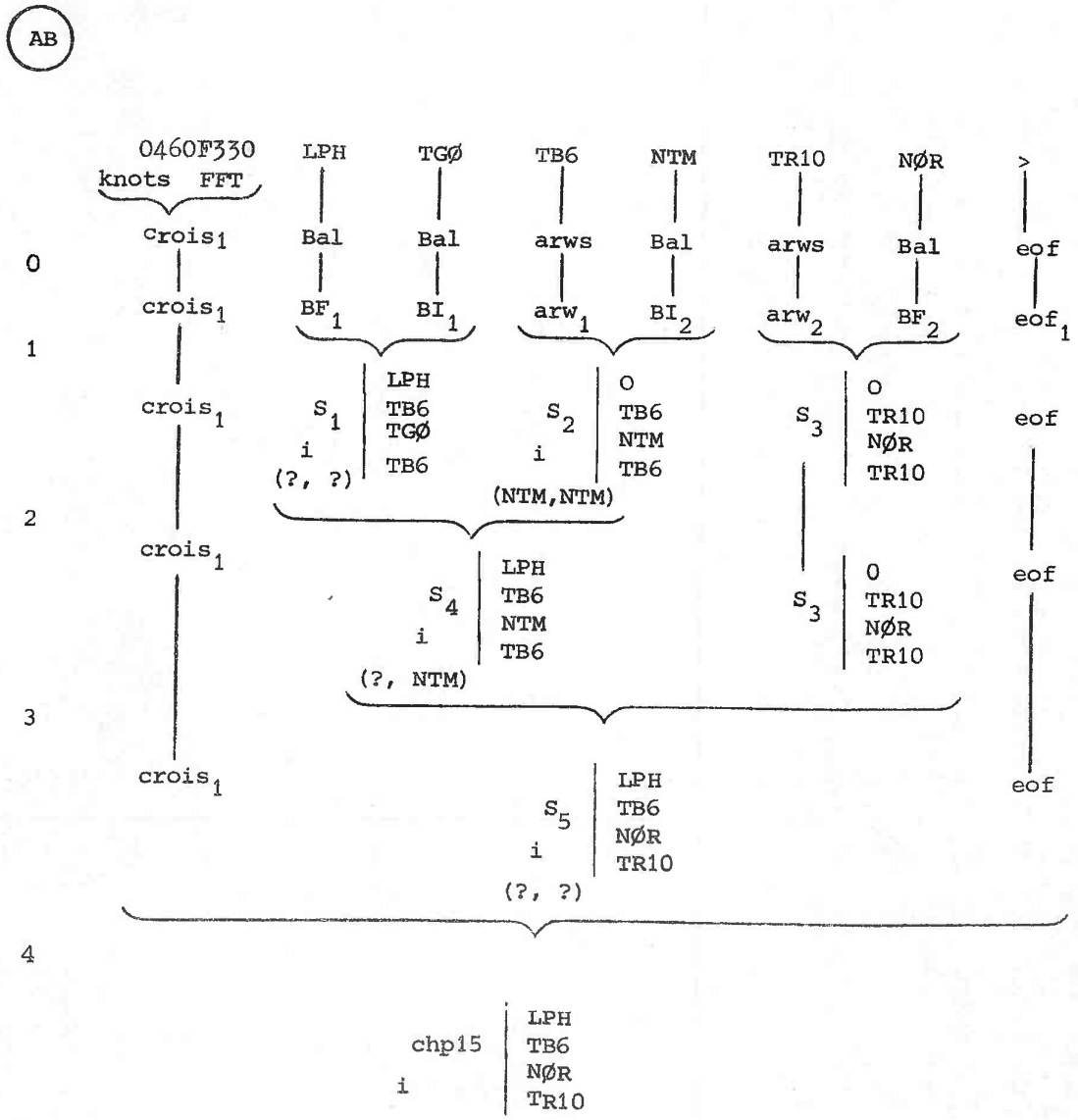
   Le message A ci-après représente la taille limite de champ 15 analysable, avec les tables actuelles de SCAPA. Le listing commenté figure en annexe A5.1/


A
0445F330
knots FFT
DM2
fa np
ALB UR9 ZUL UG1 NTM UG1N BTI UG1N BUB UA4N >
crois procdep Bal arws Bal arws Bal arws Bal arws Bal arws eof
crois procdep1 BF1 arw1 BI1 arw2 BI2 arw3 Binc1 arw4 Binc2 arwinc1 eof
crois procdep1 S1 ALB UR9 0 UR9 Si S2 ZUL UG1 0 UG1 Si S3 NTM UG1N 0 UG1N S4 0 UG1N 0 UG1N S5 0 0 0 0 eof
(ZUL,ZUL) (NTM,NTM)
crois S6 ALB UR9 0 UR9 Si S7 ZUL UG1 0 UG1N S4 0 UG1 0 UG1N S5 0 0 0 0 eof
(ZUL,NTM
crois Si S8 ALB UR9 0 UG1N S4 0 UG1N 0 UG1N S5 0 0 0 0 eof
(ZUL,NTM)
crois Si S9 ALB UR9 0 UG1N S5 0 0 0 0 eof
(ZUL,NTM)
crois Si S10 ALB UR9 0 UG1N eof
(ZUL,NTM)
Ichp15 ALB UR9 0 UGN1
<0445. F330 DM2 ALB UR9 ZUL UG1 NTM UG1N BTI UG1N>

2) <u>Si un message est parfaitement correct</u>, alors le texte du message après analyse est inchangé.

C'est le cas des messages AB, I, L, W, U, J, F, R, Q.

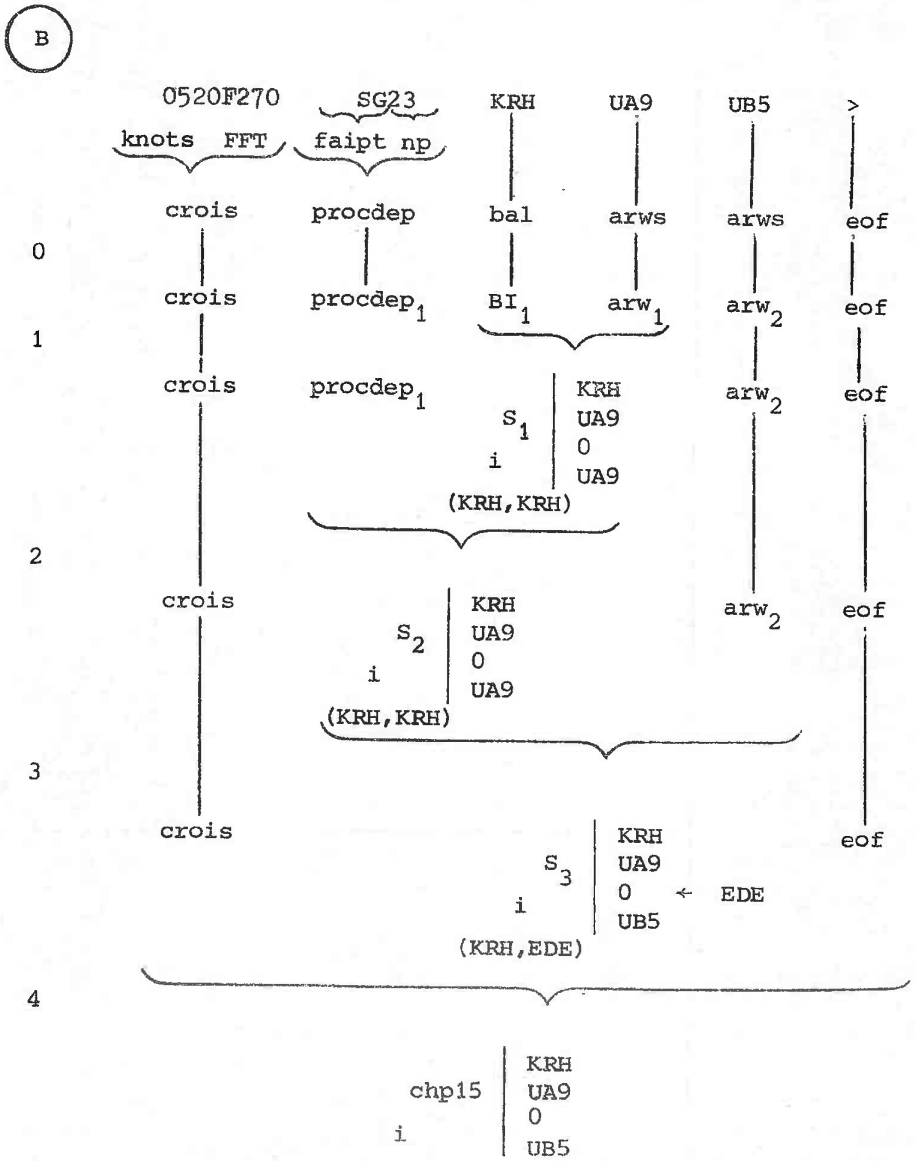
Nous proposons ici l'analyse de AB.



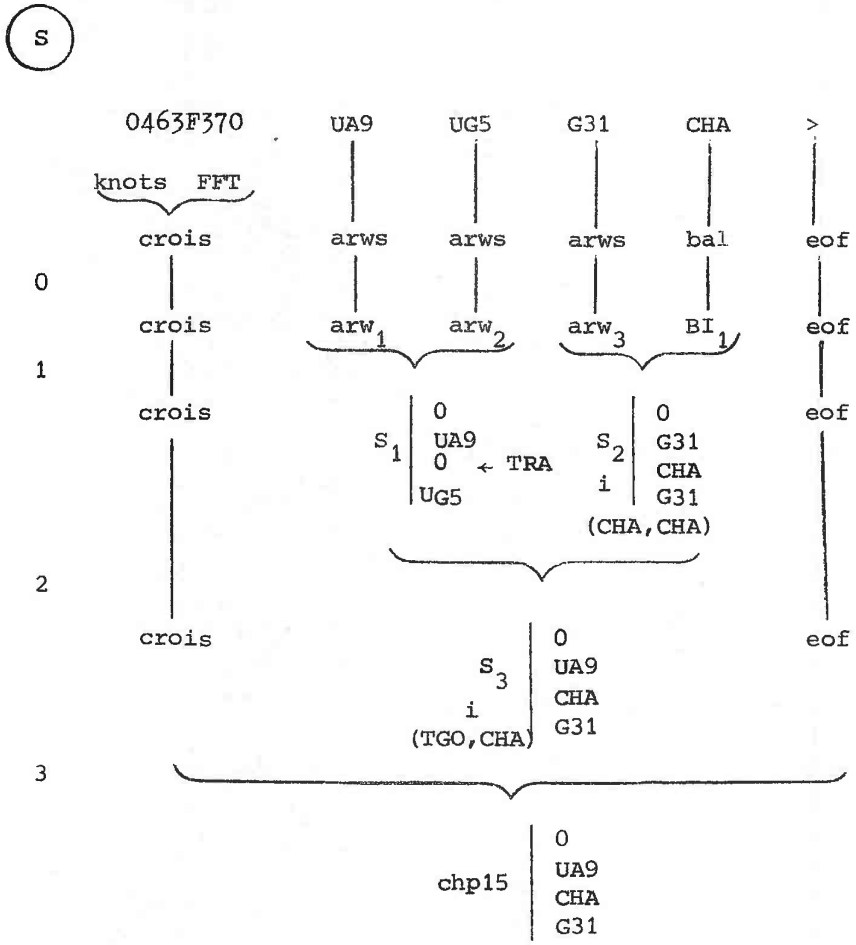
<0460 F330 LPH TGØ TB6 NTM TR10 NØR>

3) <u>correction de la séquence "airway" "airway"</u> par insertion d'une intersection de ces 2 routes aériennes. Cette correction apparait dans les messages B, G, D et S.

Montrons ici l'analyse de B puis celle de S



<0520 F270 SG23 KRH UA9 EDE UB5>

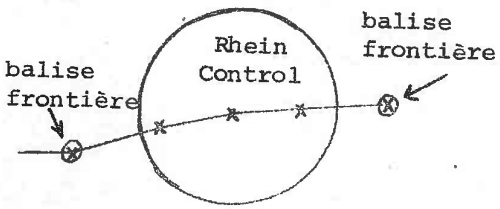


<0463 F370 UA9 TRA UG5 TGØ G31 CHA>

L'analyse de S présente en fait une ambiguité dont l'analyse ne tient pas compte : UA9 et UG5 ont deux intersections dans ou proches de la zone de contrôle "Rhein Control"; une intersection sud (TRA) et une intersection nord (MIC). Le choix peut se faire en connaissant les aéroports de départ ou d'arrivée; mais SCAPA n'effectue pas ces vérifications entre champs.

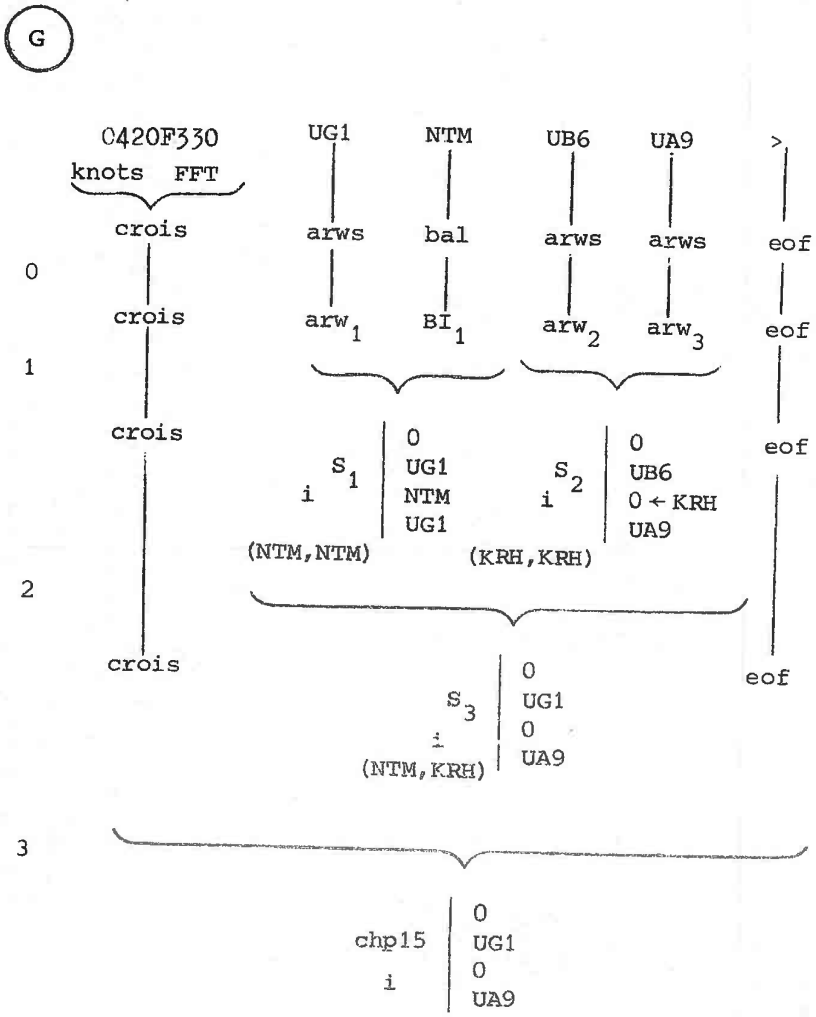
Ici, SCAPA ne connaissant que l'intersection sud, ne relève pas l'ambiguité et corrige en insérant "TRA".

4) Initialement, on pensait limiter tout champ 15 se terminant sur une route, par une balise frontière se trouvant sur cette route. Or le choix de cette balise frontière n'est pas toujours décidable (en général, à chaque route aérienne correspondent deux balises frontières).



Ici, les extrémités du champ 15 qui sont des routes restent inchangées.

Donnons l'analyse du message G.

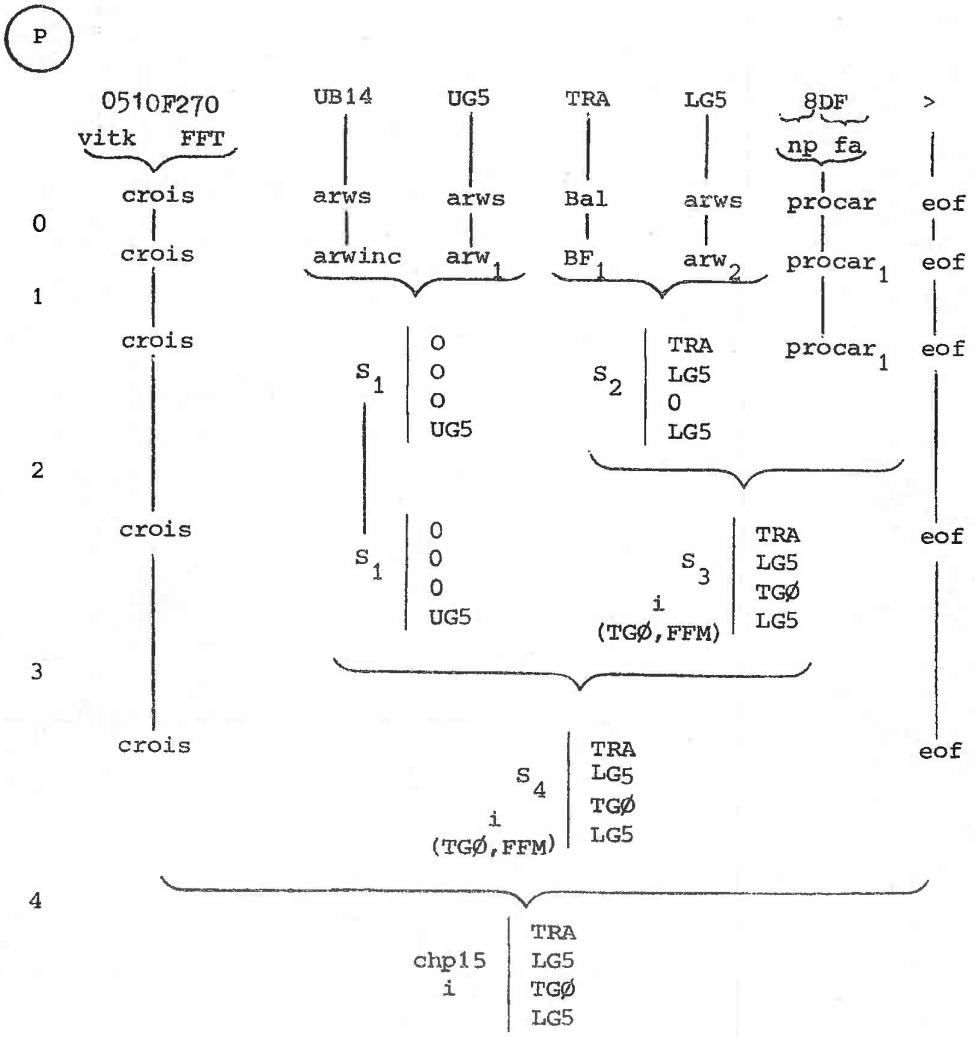


<0420 F330 UG1 NTM UB6 KRH UA9>

5) Les <u>parties entièrement externes</u> à la région de contrôle "Rhein Control" sont supprimées : exemples de messages : T, P, Y, K, E, M.

Nous proposons ici l'analyse du message P où toute la portion UB14 UG5 est supprimée.

Par contre, l'interprétation de 8DF (procédure standard d'arrivée à Francfort) conduit à rajouter TGØ qui est le point initial où s'appliquent les règles de cette procédure d'arrivée.

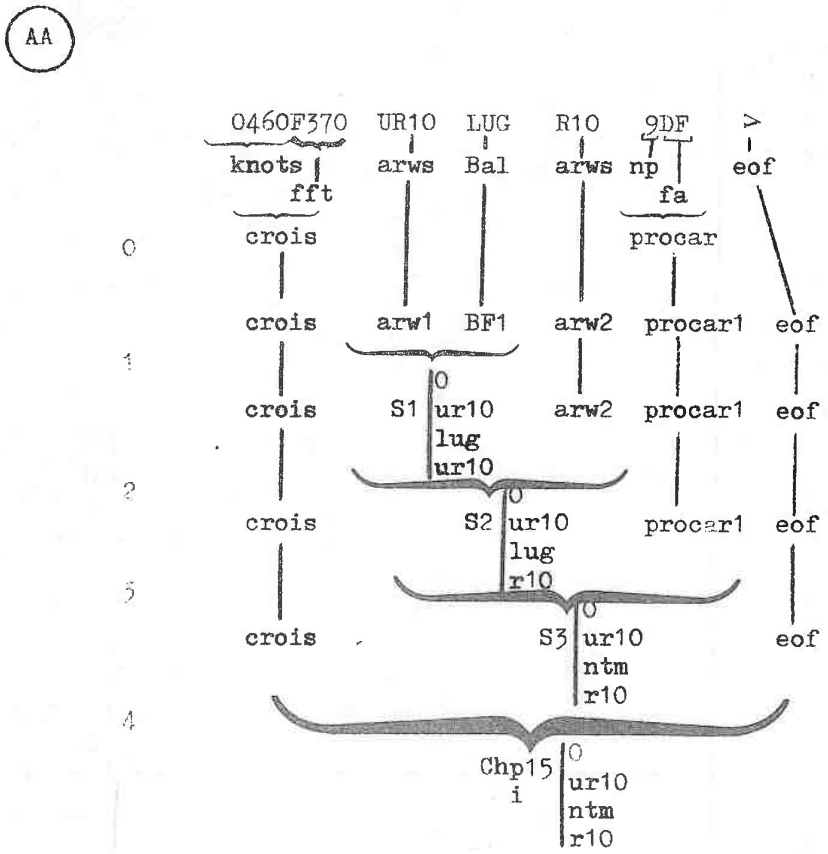


<0510 F270 TRA LG5 TGØ 8DF>

6) <u>Un lien est calculé</u> si possible entre un segment et une procédure de départ ou d'arrivée. C'est une balise correspondant au point initial d'application des règles de cette procédure.

exemples : P (analyse au 5), AA, V, Z, N.

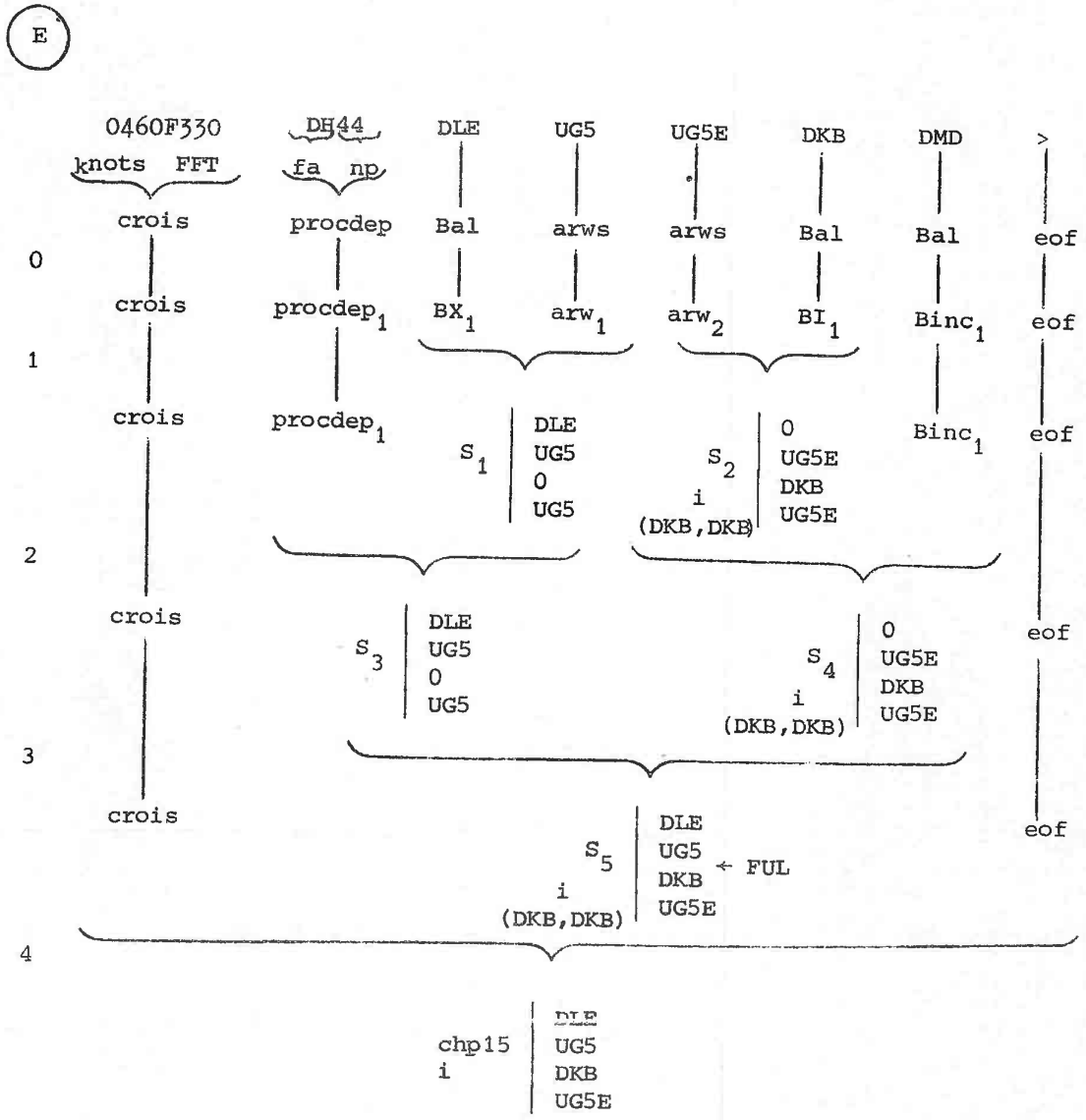
Nous donnons l'analyse du message AA



<0460F370 UR10 LUG R10 NTM 9DF>

si le calcul de ce lien n'est pas possible, la procédure est ignorée et supprimée.

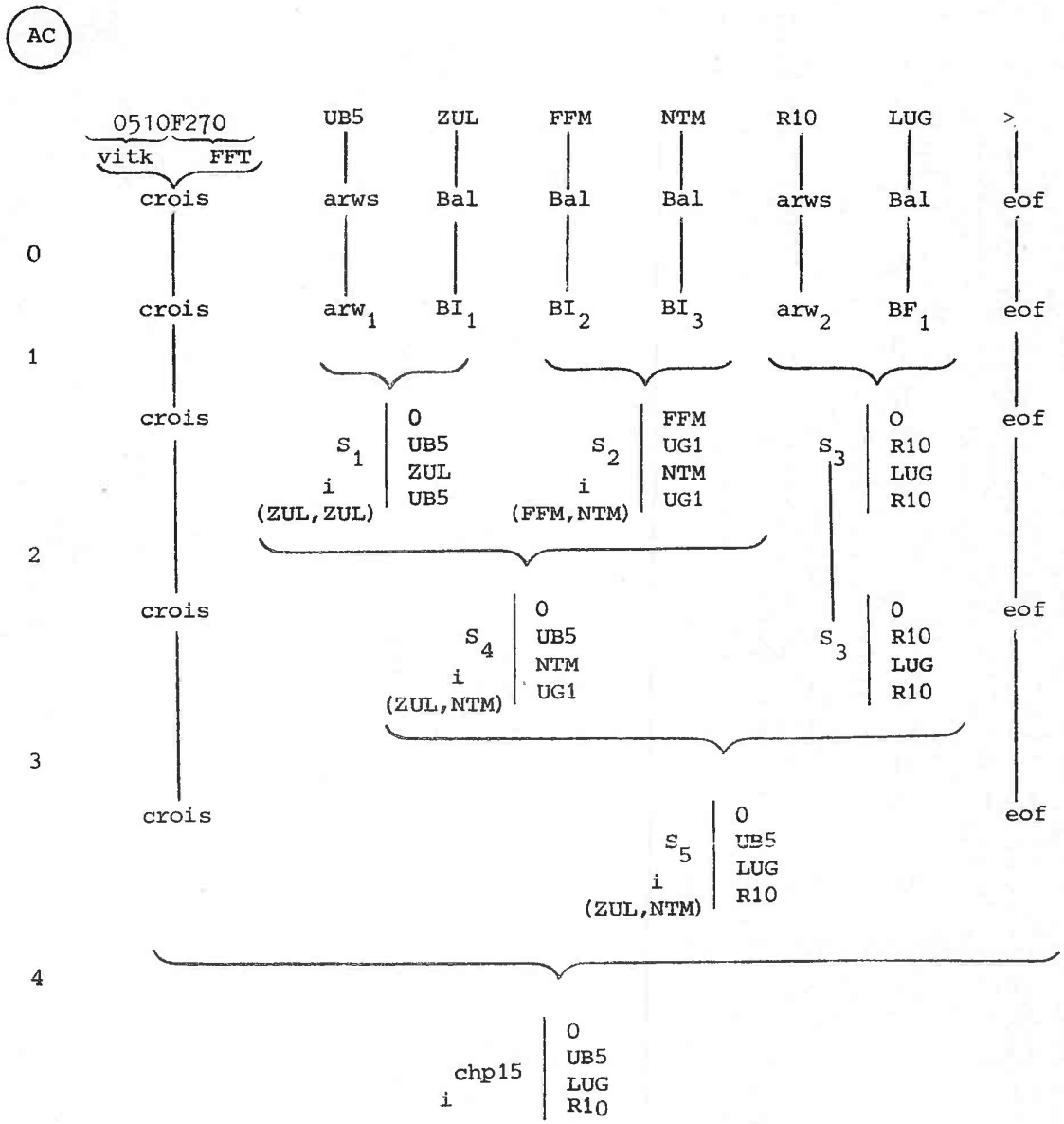
C'est le cas dans l'exemple E ci-dessous



<0460 F330 DLE UG5 FUL UG5E DKB>

7) Dans le cas d'une séquence de 2 balises, une route les joignant est recherchée afin de valider la logique du segment. Mais cette route n'est pas insérée car cela peut modifier la sémantique du message : l'ATC distingue le vol off-route (de point en point) et on-route (empruntant les couloirs aériens). Dans ces conditions, l'insertion d'une route entre 2 balises risque d'être une trahison des intentions de vol.

Exemple AC : entre ZUL et FFM on a trouvé UB5 qui n'est pas rajouté. De même pour UG1 entre FFM et NTM.

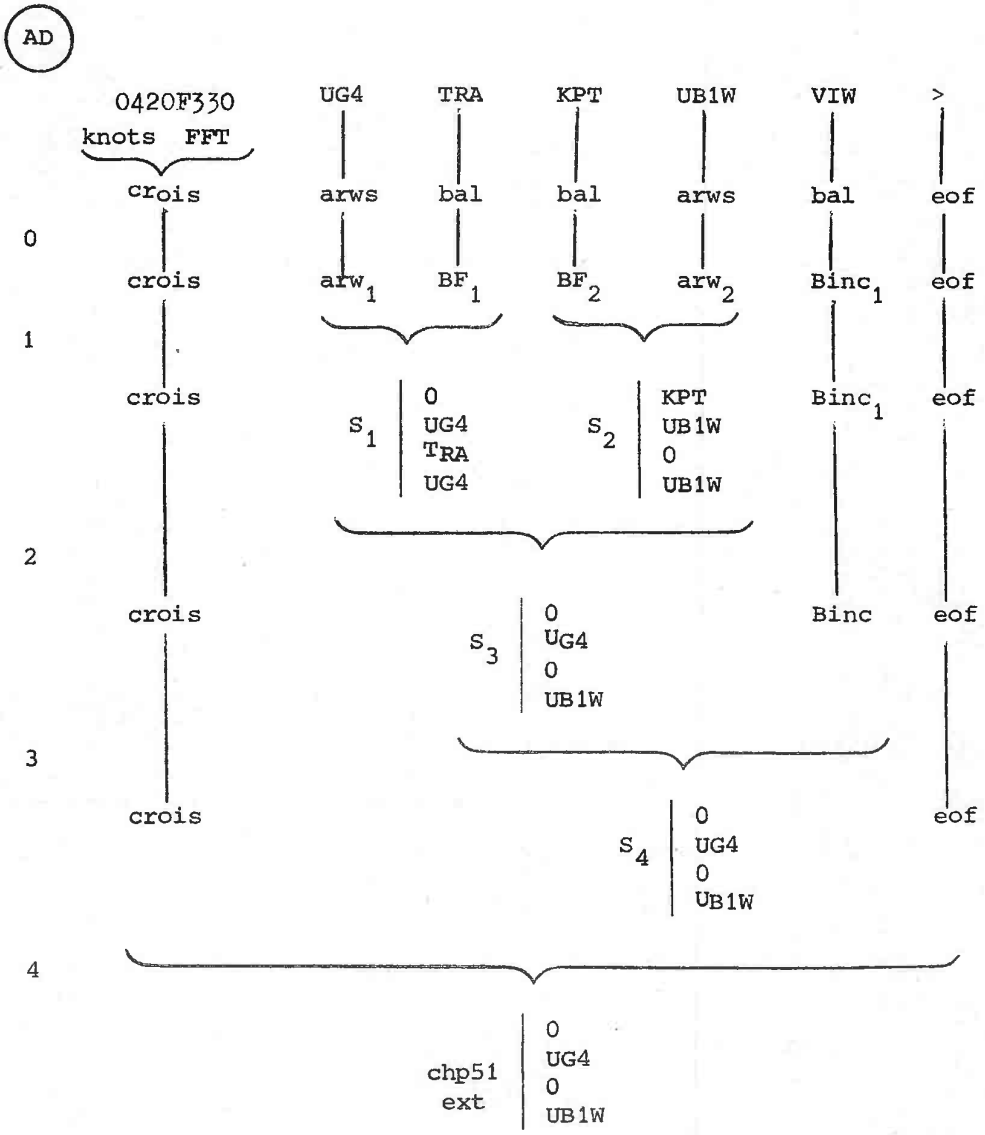


<0510 F270 UB5 ZUL FFM NTM R10 LUG>

8) Champ 15 externe à la zone "Rhein Control"

exemples : messages C, AD

Donnons l'analyse de AD :



<0420 F330 UG4 TRA KPT UB1W>

9) Pour conclure on peut donner approximativement les scores moyens de SCAPA pour analyser un message moyen (sans apprentissage)

- le temps CPU est de l'ordre de 4' sur l'IBM 360/50 du C.E.E.
- le temps d'exécution est de l'ordre de 10'
- l'espace CPU est de l'ordre de 170 Koctets

IV.2. Expérimentation de l'apprentissage

La stratégie d'expérimentation adoptée pour l'apprentissage est de se donner des messages comportant des erreurs (volontairement introduites dans un message correct) et de les soumettre à l'analyse. Dans certains cas, l'erreur est corrigée par le système des productions, sans apprentissage. Sinon, l'apprentissage fabrique un greffon dont il propose soit de faire une nouvelle production, soit de l'implanter à l'intérieur d'une production existante.

A ce stade, la boucle de fonctionnement de SCAPA représentée au III.1. est en fait ouverte, bien que, pour la modification de productions, l'analyse ait été réalisée.
Les productions sont donc créées ou modifiées manuellement, conformément aux indications de l'apprentissage.

Ensuite, le message erroné est resoumis à l'analyse pour contrôler l'efficacité de l'apprentissage réalisé.
Il y a trois grands types d'opérateurs : destruction, modification par substitution de caractère, rajout d'un terme.
Nous allons présenter et commenter trois exemples d'expérimentation du fonctionnement de l'apprentissage.

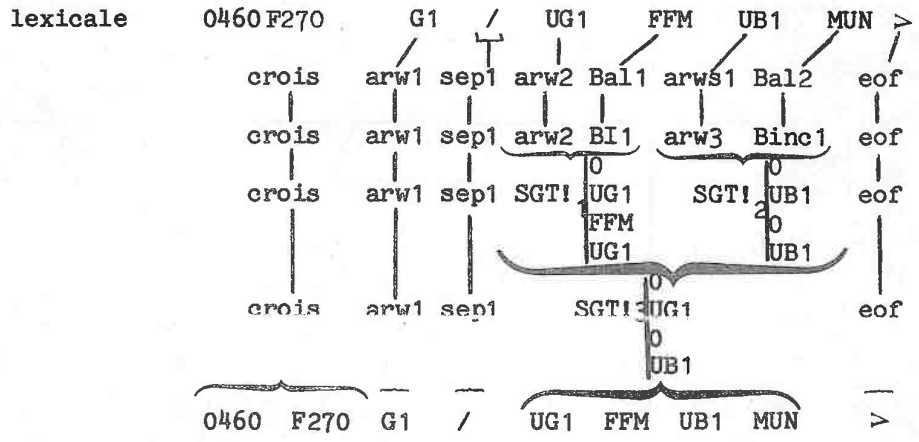
1) Exemple 1

source : 0460F270 G1/UG1 FFM UB1 MUN

corrigé : 0460F270 UG1 FFM UB1

L'erreur introduite est essentiellement le "/".

La dernière analyse tentée, (après 3 reprises), est :



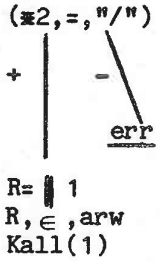
.../...

L'analyse du message corrigé aboutit à :

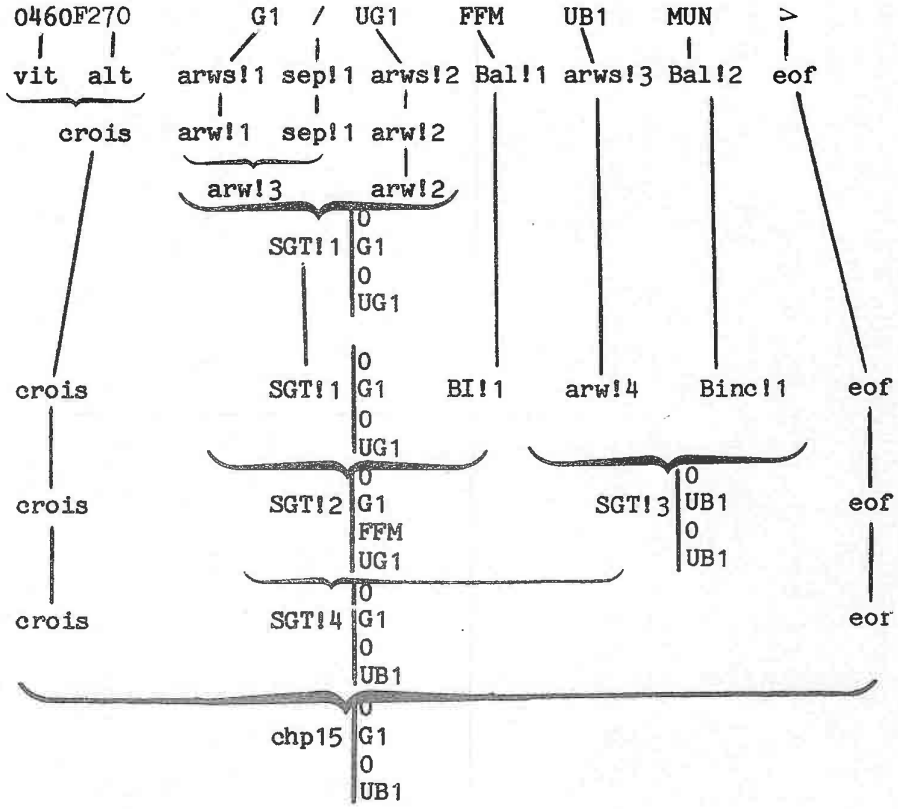
0460F270 UG1 FFM UB1

Nous avons décrit au III.3.c. α) comment l'apprentissage propose finalement de créer la production :

pro 28 : (arw,sep)



Une fois créée cette production, l'analyse du message source est reprise :



"0460F270 G1 UG1 FFM UB1 MUN"

L'analyse est alors satisfaisante.

.../...

Remarque :

Sur le listing, le greffon est décrit conventionnellement par :

```
TYPE :  0
PRODUCTION :  28 TEST :   0
STRUCTURE : < 113    4 >
ELT : /
TEST VERROU : NON
TEST='(*2 , = , / )';
FI(1)=''                FI(2)=''                FI(3)=''
TCAR='';
RES(1)='R = ||*1 '      RES(2)='R E 00000113'   RES(3)='KALL(*1)'
```

Pour les détails de la signification de ces listings se reporter à l'annexe A5.2.

Remarque :

Ici, les lignes TEST VERROU et TCAR n'ont aucune signification.

L'indication type = 0 désigne une création de production

(type = 1 désigne une modification de production).

.../...

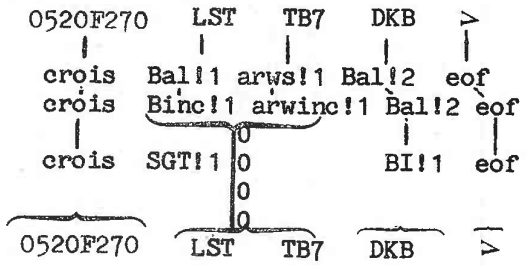
2) <u>Exemple 2</u>

<u>source</u> : 0520F270 LST TB7 DKB

<u>corrigé</u> : 0520F270 LST TB1 DKB

L'erreur est que TB7 n'est pas une route existant dans Rhein Control.

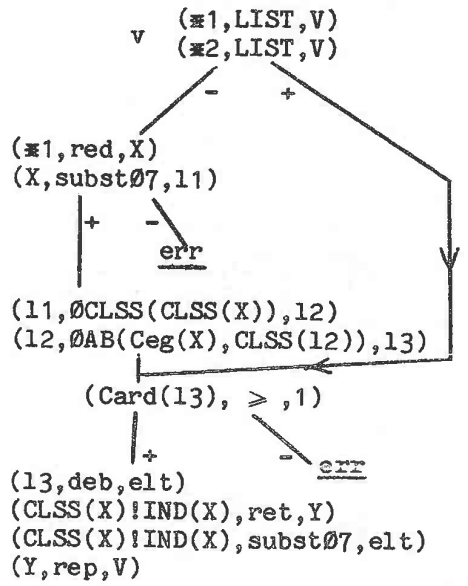
La dernière analyse (après 3 reprises) donnait :



L'analyse du message corrigé donnant :

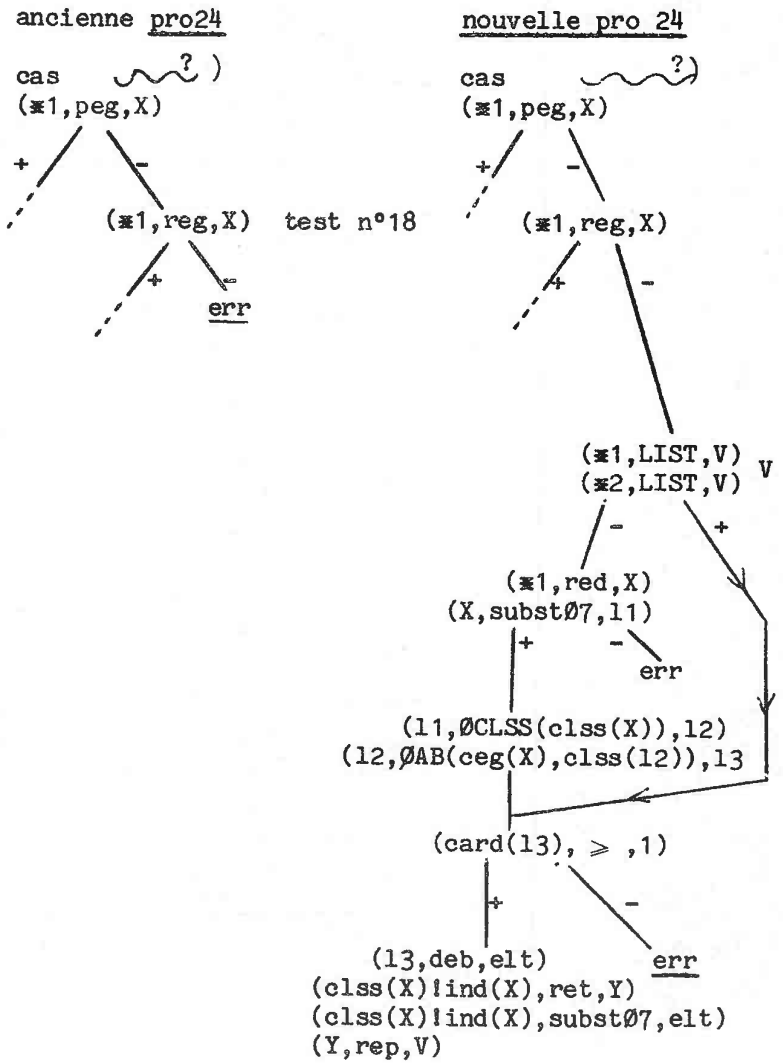
0520F270 LST TB1 DKB

Au III.3.c. ζ) nous avons fabriqué le greffon :



.../...

Ce greffon doit être implanté sur la production de structure (SGT,Rpnt) , c'est-à-dire pro24, test N° 18.



* remarque :

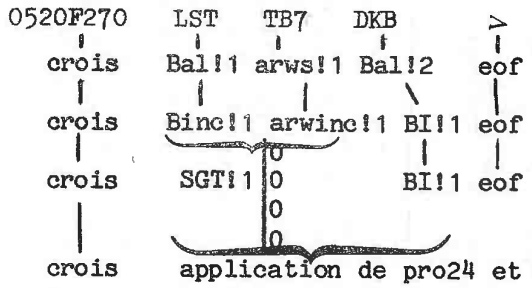
La notation ?) indique que le segment en question n'a pas de route extrémité droite (ou elle est inconnue) non plus que de point extrémité droite (ou ce point est inconnu). De même pour (?, avec les extrémités gauches.

.../...

L'extrait de listing permettant de dessiner ce greffon est :

```
TYPE :  1
PRODUCTION :  24 TEST :  18
STRUCTURE : < 119  112 >
ELT : TB7
TEST VERROU : OUI
TEST='( *1 , REL13 , X )';
FI(1)='( X , SUBSTØ7 , L1 )'                FI(2)='( L1 , ØCLASS<00000113> , L2 )'
FI(3)='( L2 ,ØAB PRO<CLSS(L2),00000010'     FI(4)=''
                                                                  FI(5)='';
TCAR='(L3 , CARDSUP , 1 )';
RES(1)='( L3 , DEPILE , ATOM )'             RES(2)='( X  , RETNIV , Y )'
RES(3)='( X  ,SUBSTØ7NIV Y , ATOM )'        RES(4)='( * , RETOUR NIV , Y )'
```

La reprise de l'analyse donne alors :



application de pro24 et de la greffe :

exécution du greffon

(*1,red,X) ⇒ X = TB7

(X,substØ7,l1) ⇒ l1 = | T37 / T87 / TB1

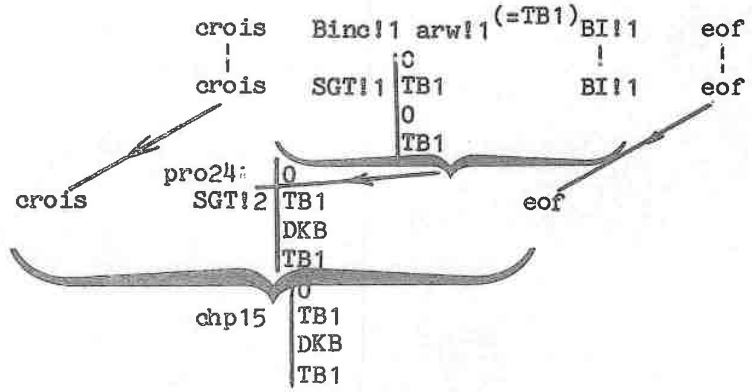
(l1,ØCLSS(arwinc),l2) ⇒ l2 = | TB1 / - / -

(l2,ØAB(rpnt,arwy),l3) ⇒ l3 = | TB1 / - / -

elt = TB1
Y = 2

au niveau 2 la structure est modifiée :

et l'analyse est reprise à ce niveau



"0520F270 LST TB1 DKB"

qui est une analyse satisfaisante.

.../...

3) Exemple 3

source : 0440F260 FFM NOR TR10

corrigé : 0440F260 FFM TG1 NTE TR7 NOR TR10

Le message source ne détaille pas l'itinéraire à prendre entre FFM et NOR .

L'analyse du message source (après 3 reprises) est :

	0440F260	FFM	NOR	TR10	>
1	crois!1	Bal!1	Bal!2	arws!1	eof
2	crois!1	BI!1	BI!2	arws!1	eof
	crois!1	BI!1	BI!2	arw!1	eof
	0440F260	FFM	NOR	TR10	>

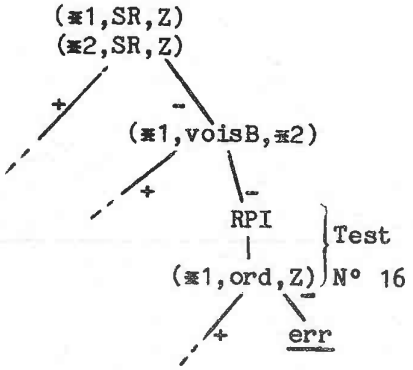
L'analyse du message corrigé fournit :

0440F260 FFM TG1 NTE TR7 NOR TR10

Nous avons calculé au III.3.c. μ) le greffon qui doit être implanté sur pro23 test N° 16 (pro23 = production (Rpnt,Rpnt).

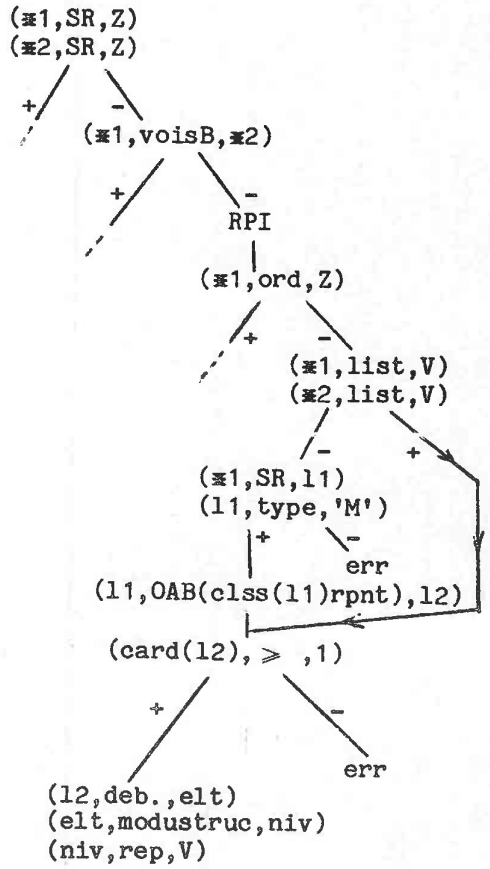
ancienne pro23

cas général

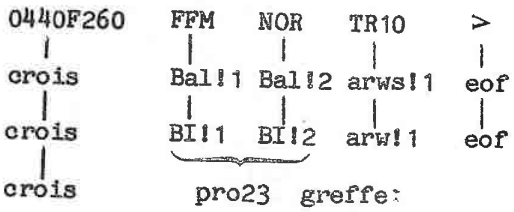


nouvelle pro23

cas général



L'analyse est reprise :



exécution du greffon :

- (*1,SR,l1) provoque le calcul de :

  l1 = {TB1,TG1S,TG1,TL8, UB1,UG1,UG1S,UR10, UA19,....

- (l1,type,'M') restreint l1 à :

  l1 : {TB1,TG1S,TG1,TL8}

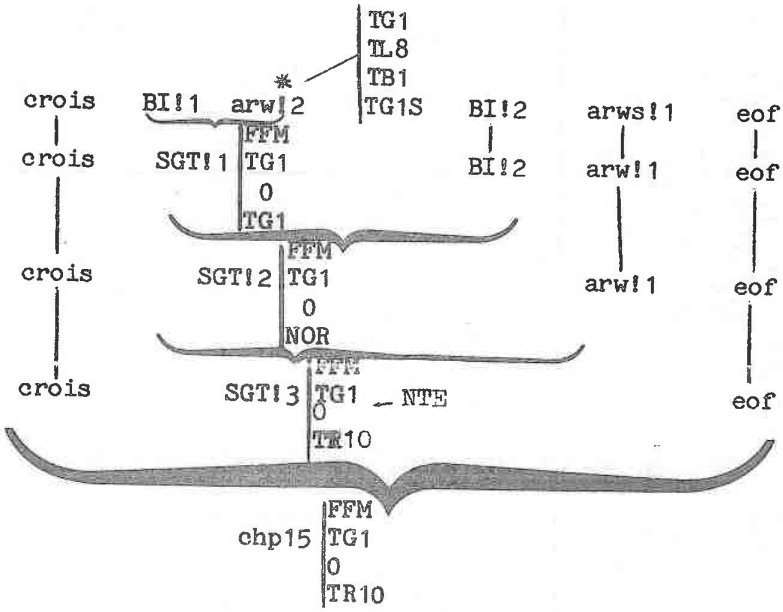
- (l1,ØAB,(arw,rpnt),l2) calcule l2 =

  l1 NOR {TG1,TL8,TB1,TG1S}

  car la production (arw,rpnt) s'applique à TG1 NOR (en effet NTE est sur TG1 et proche de NOR).

- (l2,deb,elt) donne elt = TG1

- (elt,modustruc,niv) (niv,rep,V) imposent que la structure du niveau 2 soit modifiée (arw* rajouté), et l'analyse est reprise :



"0440F260 FFM TG1 NTE NOR TR10"

<u>Remarque</u>

arw!2* indique qu'il s'agit d'un élément listé.

IV.3. Evaluation de SCAPA

SCAPA a été conçu comme un modèle de faisabilité devant démontrer :

1° - la possibilité d'utiliser des mécanismes correctifs syntaxico-sémantiques liés à un analyseur de même type

2° - la possibilité d'accroître le nombre et la spécifité des actions correctives, par apprentissage à partir d'exemples.

a) La stratégie de l'analyse-correction, répondant au premier point, permet en principe de corriger les trois types d'erreurs : syntaxiques sémantiques et opérationnelles, dans la mesure où les erreurs ont été prévues et des stratégies correctives définies.

Nous l'avons fait pour un certain nombre d'erreurs typiques en nous fixant initialement un ensemble de connaissances géographiques et opérationnelles.

Les connaissances limitées que nous avons utilisées permettent de se rendre compte des possibilités offertes par cette stratégie malgré les limites atteintes dans les exemples :

. IV.1. - cas 1) : la taille du champ 15 est limitée à 12 termes à cause de la limite des tables système.

. IV.1. - cas 3) : ambiguïté dans le choix de l'intersection à rajouter pour une séquence <arw> <arw> .

. IV.1. - cas 4) : il n'est pas toujours possible de clore un champ 15 par des balises frontières.

La décomposition du champ 15 en segments constitue une représentation pratique des principales données ou portions de données du champ 15 (à savoir les extrémités -routes ou balises -, les niveaux ou vitesses à ces extrémités, la situation -interne ou externe- ...). Cependant, du point de vue de l'implantation, l'analyseur est lourd et surpuissant : le langage des FPL, régulier, ne nécéssitait pas un analyseur à pile avec back-tracking dont la justification principale est d'autoriser le choix d'une analyse jugée la plus sûre, pour procéder à l'apprentissage. Par ailleurs, nous ne nous sommes pas trop souciés d'optimiser les performances de ce qui ne devait être qu'un modèle d'étude.

.../...

SCAPA peut réaliser certaines corrections à caractère sémantique comme en témoignent les expérimentations du IV.1 - cas 3) (correction d'une séquence <arw> <arw> par adjonction d'une intersection), cas 5 et 6) (rajout du point d'activation d'une procédure standard d'arrivée ou de départ), cas 7) (recherche de routes liant deux balises), etc. .....

Nous n'avons pas inclu dans SCAPA de mécanismes corrigeant les erreurs opérationnelles (problèmes de niveaux, de vitesses, de règles de vol, etc. ...) cependant la forme de ces mécanismes serait la même que pour les autres types d'erreurs (c'est-à-dire tests sur les données et triplets - résultats inclus dans les productions).

b) L'apprentissage, bien qu'il utilise un nombre limité d'opérateurs, est capable de prendre en compte des erreurs variées :

. altération de caractère (cf. IV.2. - exemple 2)
. omission de termes (cf. IV.2. - exemple 3)
. erreurs de séquences
. erreurs opérationnelles

Cependant, le mode d'expérimentation n'a pas permis de confronter l'apprentissage à un nombre suffisant d'erreurs réelles pour en déterminer l'efficacité pratique. Les quelques exemples présentés (cf. III.3.c) ou IV.2) montrent le type de corrections envisageables, mais des corrections plus complexes, résultant de plusieurs corrections élémentaires du type de celles présentées doivent être possibles.

De plus, la boucle d'apprentissage étant ouverte, une intervention manuelle est nécessaire pour réaliser les modifications des productions, ce qui rend le système malaisé à manipuler.

Les greffes dans les productions proposent d'appliquer un opérateur de correction qui, en général, fabrique plusieurs solutions.

Les tests qui suivent l'application de cet opérateur sont fabriqués lors de l'apprentissage pour caractériser la solution à retenir, par rapport au contexte.

Nous allons voir comment, si ce contexte n'est pas reproduit, la greffe ne peut pas s'appliquer, même si l'opérateur, employé seul, semblait pouvoir effectuer une correction.

Imaginons le message faux : TGO TB6 KRH

corrigé en : TGØ TB6 KRH.

.../...

Vraisemblablement une greffe est proposée dans la production (arw, arw), utilisant Ø5 (Ø ↔ O)
avec le contexte caractérisé par : (*1,SubstØ4,L1)
(l1,SR,*2).

En effet, lors de l'apprentissage on avait trouvé la relation

(TGØ,SR,TB6).

Soumettant maintenant le message :

TGO TG5 FUL

le contexte de la greffe précédente ne convient pas, lors de l'application de (arw, arw), bien que TGØ soit une balise existante.(Mais TGØ n'est pas sur. TG5).

Ici, le message corrigé pourrait être TG1 TG5 FUL et la nouvelle greffe à construire utiliserait Ø12 (modification de la partie numérique d'une chaîne), le contexte imposant que la route substituée ait une intersection avec *2 (2ème route de (arw, arw)).

On voit donc que l'opportunité et la pertinence des corrections sont liées à la caractérisation du contexte de l'apprentissage.

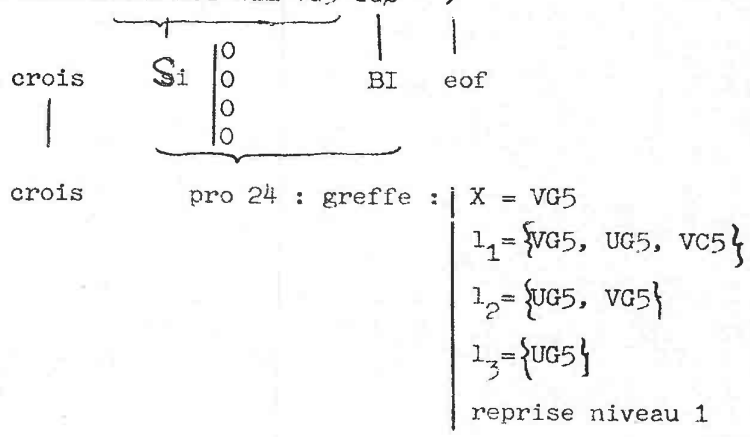
L'exemple 2 du IV.2 ayant provoqué une séquence d'apprentissage, on peut observer comment un message présentant une erreur analogue serait corrigé, et comment un message présentant une erreur insuffisamment analogue demeurerait refusée :

L'exemple 2 était : 0520F270 LST TB7 DKB

Rappelons que la greffe sur la production pro 24 : (SGT, Rpnt) proposait de trouver la red. (route extrémité droite) du segment, d'y tenter des substitutions (type Ø7) et de filtrer la liste obtenue, avant de reprendre l'analyse.

L'analyse du message : 0400F200 XIK DUT WIL VG5 TGØ >

utilise la greffe précedente au stade :



la reprise donnant finalement la version corrigée

0400F200 XIK DUT WIL UG5 TGØ

.../...

La greffe s'applique car la transformation concerne une airway inconnue à l'extrémité droite d'un segment et suivied'une balise. Le contexte était donc le même.

Par contre, le message : 0420F200 LST TB2 DKB

ne serait pas corrigé et provoquerait un autre apprentissage car la greffe, tenant une substitution spécifique (où 1↔2 ne figure pas), échouerait. L'opérateur qui serait mis en jeu ici serait Ø12 : substitution d'un nombre par un autre nombre.

La sensibilité des corrections au contexte, montrée par les exemples précédents, peut être gênante dans des cas où l'opérateur conviendrait effectivement mais où le contexte trop restreint empêche son application ; ou de même, dans des cas où la structure de la production où a été greffée la correction adéquate n'est pas atteinte au cours de l'analyse ; car dans notre système une correction est liée rigidement à une production.

Ce dernier problème pourrait se résoudre en envisagenat des productions s'appelant l'une l'autre : les productions complexes appelant des productions élémentaires, les greffes effectuées de préférence sur les productions élémentaires pouvant alors être utilisées par toutes les productions appelantes.

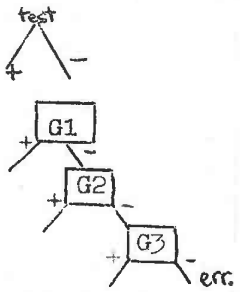
Il se pose par ailleurs, le problème de la stabilité du système des productions face à des versions corrigées successives fournies par des opérateurs humains éventuellement différents.

Les greffes risquent alors de devenir des accumulations de repêchages.

On peut tenter d'aborder le problème de la mesure du degré de généralité des greffes par la voie suivante :

Si les greffes sont trop restrictives (le contexte imposé est trop précis) la structure en cascade des greffes sera apparente.

On pourra alors, en comparant les contextes successifs de ces greffes reposer le problème du degré de finesse des tests sur le contexte à introduire.



Si au contraire les greffes sont trop générales, on aura des corrections intempestives ; l'amélioration des recherches de relations entre éléments du réseau devrait alors augmenter la finesse des greffons. Jusqu'à présent ces recherches sont limitées dans leur transitivité à deux termes.

V. CONCLUSION

L'extraction automatique des données Plan de vol pose problème dans la mesure où ces messages comportent des erreurs :
certaines sont fréquentes et leur correction est connue;
d'autres sont aléatoires (ou moins fréquentes) et non répertoriées: leur correction ne peut être prévue à l'avance .
SCAPA propose une solution double :

- intégrer les corrections "ad hoc " des erreurs anticipées au sein d'un analyseur dont les règles incluent , sur un même plan , les composantes syntaxiques et sémantiques.
- effectuer un apprentissage en déduisant de la comparaison entre le message source et une version corrigée correcte de ce message, un ensemble de modifications des règles de l'analyseur , correspondant à la correction apprise .

a) La première solution : l'analyse-correction ,met en jeu , à la fois un analyseur syntaxique très général et des productions où des corrections très spécifiques ont lieu.
L'analyseur est basé sur une grammaire"context-free" ambigüe et utilise un back-tracking limité. Ceci permet de tenter plusieurs analyses partielles avant de passer à l'apprentissage , de localiser les erreurs , et d'obtenir une analyse plus poussée des portions restantes .
Dans les productions comportant des actions correctives , les connaissances géographiques et opérationnelles jouent un rôle primordial(pour vérifier les tests et évaluer les solutions).
Nous avons validé certaines portions de ces productions et connaissances sur place , au centre de contrôle de Karlsruhe ,sans prétendre cependant traiter le problème dans son ensemble .
L'expérimentation a montré que SCAPA pouvait corriger un certain nombre d'erreurs , mais certains points comme les mécanismes de reprise ou le choix de la dernière analyse avant l'apprentissage pourraient avantageusement être affinés .

.../...

b) L'apprentissage dans SCAPA ne fonctionne pas par adjonction de nouvelles fonctions sémantiques mais par adaptation dans les productions d'un nombre limité de fonctions correctives .
Les greffes des productions sont construites à partir de schémas prédéfinis de greffe , complétés par des paramètres contextuels dont le calcul est l'objet essentiel de l'apprentissage .
Les quelques exemples traités montrent la faisabilité de ce type d'approche .
Il faudrait cependant entreprendre des tests plus systématiques pour mesurer son efficacité , et peut-être , avant cela , compléter certains algorithmes comme ceux d'un interprêteur des productions et d'un générateur des greffes , et, tenter de pondérer des choix actuellement assez abrupts de SCAPA (choix entre la modification d'une production ou la création d'une nouvelle , détermination de la production à corriger , choix de la chaine à modifier , caractérisation du substitué le plus intéressant , ...).
Le problème de la mesure de la généralisation introduite par une greffe dans une production reste par ailleurs posé ; il peut s'avérer important dans le cadre du contrôle de l'évolution globale du système des productions au cours de l'apprentissage .

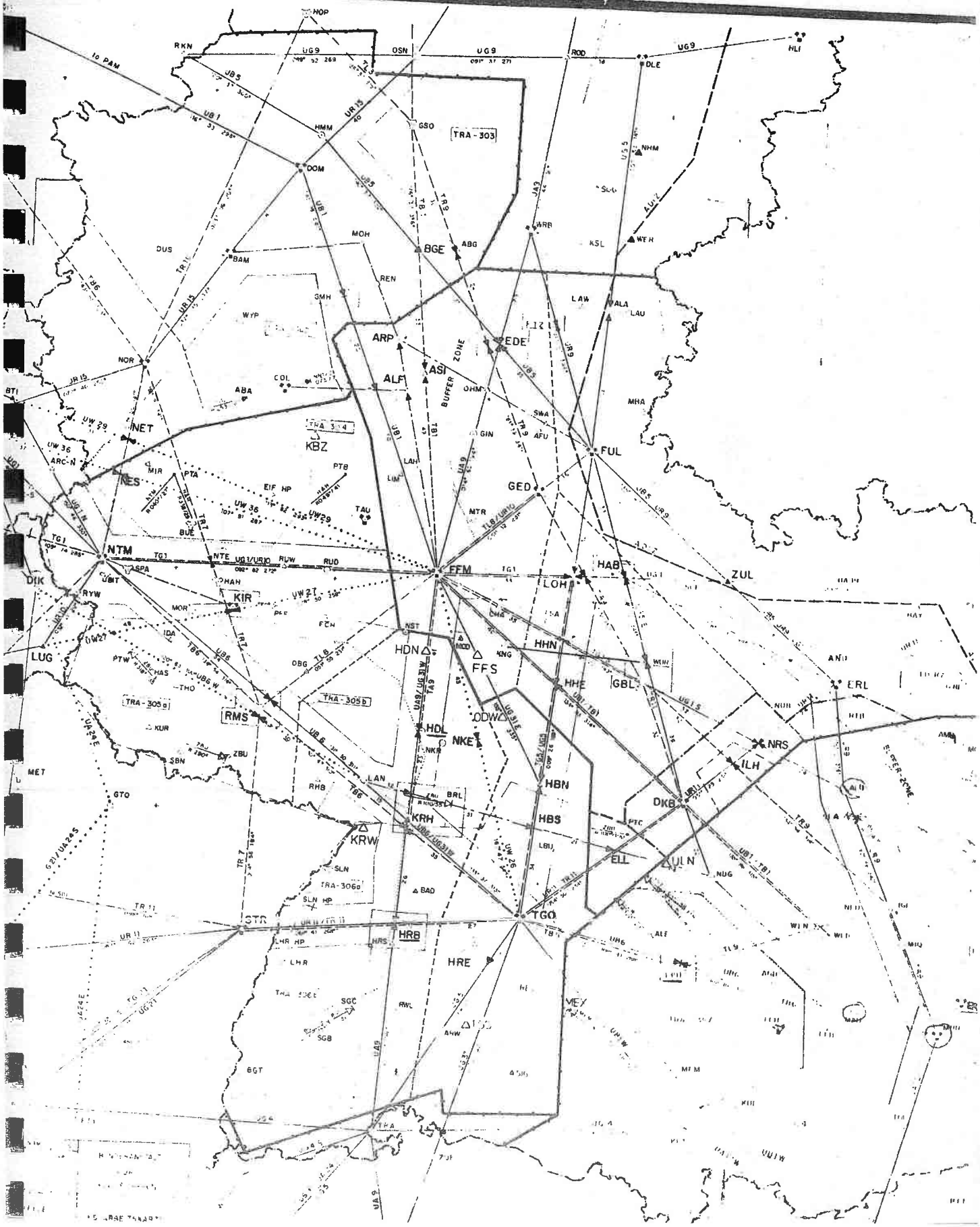
ANNEXE 1

A1. CONTROLE DE LA NAVIGATION AERIENNE .

A1.1/ Carte de la zone de Karlsruhe.

A1.2/ Exemples de messages Plan de Vol .

A1.3/ Description O.A.C.I des champs d'un Plan de Vol .


HOP
RKN
OSN
ROD
DLE
HLI
UG9
UB5
UB1
UR15
HMM
GSO
TRA-303
DOM
NHM
SLG
JA9
HRB
WFH
KSL
DUS
BAM
MOH
BGE
ABG
REN
GMH
WYP
LAW
VALA
LAU
ARP
EDE
ASI
ZONE
BUFFER
ALF
NOR
COL
ABA
OHM
SWA
AFU
MHA
NET
TRA 304
KBZ
GIN
FUL
ARC-N
MIR
PTA
PTB
LAH
LIM
GED
NES
EIF HP
TAU
MTR
BUE
NTM
NTE
RUW
RUD
FFM
HAB
ZUL
DIK
ASPA
HAH
LOH
RYW
KIR
MOR
FCH
NST
HDN
HHN
LUG
IDA
OBG
FFS
HHE
GBL
ERL
PTW
HAS
THO
TRA-305a
TRA-305b
RMS
HDL
ODW
NKE
NRS
ILH
KUR
SBN
ZBU
MET
RHB
LAN
HBN
DKB
GTO
KRH
HBS
KRW
ELL
ULN
SLN
TRA-306a
BAD
STR
TGO
HRB
HRE
SGC
SGB
MEX
EGT
MFM

nnnnzczc dub507 221714
gg edsgzp edduzq lirrzq lirszp limmzq lfmmzq edduyu
221633 liedzp
(arr-pirol51-lied1507)

nnnnzczc dub508 221716
ff edduzq eddyzq edzzzq edjbyv
221713 ehamzp
(cnl-sfpl-kl921-eham/edds)

nnnnzczc dub509 221720
ch
nnnnzczc dub510 221722
ff egttzqfp ebbrzq eburzq eddyzq eddfzq edduzq edjayv eddmzq
221720 eggwzg
(fpl-damat-i
-ba11-r/r/k
-eggw2010
-0450f290 bpk a2 g1 ug1 spi ug1 ntm ub6 lan 1dml
-eddm2130 eddn
-opr/bv5583)

nnnnzczc dub511 221724

ff lecmzr lemdyaeu lecbzq lfffzq lfbbzq eburzq edduzq edarzp legtzt
221720 edahyo
(arr-yakka51-edah1446)
sorry for delay times posted wrong thanks for coor:

nnnnzczc dub512 221725
ff eddxyi edzzzq edduzq eddyzq lszhzq lszhzt
221724 edvvzt
(fpl-bb859-i
-dc8-r/r/k
-edvv1835
-0480f310 vv86 pa9 ug31 zue sha
-lszh1935 lfsb
-0)

nnnnzczc dub513 221726
ff lemdyaeu lecbzq lfffzq lfmmzq lsggzq lszhzq eddyzq edduzq edzzzq
221721 lepazp
ref/089
(fpl-tr356-1
-s210 r/r/k
-lepa1745
-0430f310 vor1a mhn ub16 lima ub46 spr ug5 tra ua9 rod ua9e ham ug5
ges g5 aor dct sna jx
-esmx2050 essj
-reg/ecbrj)

nnnnzczc dub514 221731
ff ebbrzt eburzq eddyzq edzzzq edduzq
221728 ebbrzp
(fpl-snoosdn-i
-b737-r/r/k
-ebbr1805 eddf1820 eddm1845
-0430f270 gaa spi ug1 ntm ub6 1 dml
-eddm1905 eddn
-opr/sn737 sel/elfh est/spi 1815)

nnnnzczc dub515 221732
ff lfffzq edduzq lkprzq lkprzp lfpoxj
221735 lfpozp
(fpl-ok1025-i
-tu34-r/r/k
-lfpo2030 eddu2117 lkpr2145
-0440f330 xmg ur11 str pr
-lkpr2205 lkib
-reg/okcfe)

nnnnzczc dub516 221733
ff hecazq lgatzq lirrzq limmzq lowwzq edduzq ehamzq ebbrzi oejdzq
220003 oejdzq
(fpl-sv691-i
-b707c/rd/r/k
-oejd1155 heca1250 lgat1452 lirr1600 limm1652 loww1722 eddu1727 eha
-0480 f350 a26 w4 a51 w2 w1 a1 ug8 ua18 ua14 ua12 ur9 ieee ur9hv5
-hema1830 ebbr
-reg cj)

nnnnzczc dub517 221734
ff ednpzp eduozp ednnzp lirrzq lfmmzq lfffzq edduyu
221454 liedzp
(dep- gaf 3785-1414)

nnnnzczc dub518 221740
ch
nnnnzczc dub519 221740
ff lfrrzq lfffzq eburzr edduzq edafyo edaryo
ff eggxzo lfrrzq lfffzq eburzr edduzq eguxyf edafyo edaryo
221729 lplayx
(fpl-m40647-i
-c-141-r/r/j
-lpla1855 lecm1940 lfrr2025 lfff2103 ebur2136 eddu2140
-0435f330 alfa radar 2 dept alfa/m074f330 41n020w 43n013w
stg/0430f310 va5 pon ur10n cll 0435f330 ur10 ffm
-edaf 2208 edar
-rmk/opr us ml gat rmk/bz flight us code 12 est/alfa
radar 1822 alfa 1825 41n 020w 1900 pass atd eta ata to
edaryw)

nnnnzczc dub520 221744
jj ednznd edsznb edzznm eddynm edaayo
221655 eddzyo
d 2505 notamc d 1785 frankfurt fir. edop schwaebisch hall.

nnnnzczc dub521 221749
gg eddxyi edzzzq edduzq eddyzq lszhzq limmzq lipwzq limjzt
221747 edvvzp
(cnl-oefdp-edvv1745)

nnnnzczc dub522 221758
dd lszhzt eddmzt gclpzt edzzzq edduzq
221756 lsggyf
(div-lsggzr
-dfdabil
-b727
-gclp
-lszh
-diverting to eddm due company altn/edds
-0210
-eta 1845 at eddm)

nnnnzczc dub539 221901
ff efhkzt efeszq essazq esmmzq ekchzq eddyzq edjayv edjbyv edduzq
221858 efhkzp
(dla-ay1505-2000)

nnnnzczc dub540 221912
ff lfffzq lfpgzp lfpozp edduzq lowwzq lyzazq lybezq ltbazq
(fpl-gf024-i
-l10-r/r/k
-lfpg2040 lyza2145 lbsf2233 ltba2253 ltac2319 orbb0023 cbb10134
-0490f330 xmg rlp ur11 str tgo ub6 mun ub1 viw kft zag ub5
sm1 ua4 dim a4 rad va4n ist va16 vaa vg8 ezs 0480f370 vg8 srt
r21 ba b55 bd
-otbd0234 omdb omad obb1
-reg/a40ty sel/abdl opr/gulf air)

nnnnzczc dub541 221916
ff edzzzq eddxy1 eddyzq edduzq edjayv edjbyv
221912 eddhzp
(fpl-jk626-i
-dc8-r/r/k
-eddh2020
-0460f330 dh44 dle wrb ua9 tra ug5 spr ub16 mtl ua6 pcs pa
-lepa2250 leib
-reg/eccdc)

nnnnzczc dub542 221920
ch
nnnnzczc dub543 221923
ff limmzq lszhzq edduzq edjbyv eddyzq edjayv ekchzq ekchzt
221917 limczp
(fpl-dm206-i
-b72s-r/r/k
-limc2100
-0466f390 rmg mkr ua9 bravo dct zue ug31 tgo ug5 ges dct cda
-ekch2235 eddh
-reg/oyapw sel/hkaf)

nnnnzczc dub533 221837
ff eburzq eddyzq ebbrzq edduzq lowwzq lyzazq lybezq
ff eburzq eddyzq ebbrzq edduzq lowwzq lyzazq lybezq lgatzq
221831 egllzp
(fpl-hbitr-i
-g2-r/r/k
-egll2100
-0480f370 dvr ug1 ub6 tgo ub1w ub1 b1 tgr egn
-lgat0030 lgir heca
-opr/pvt)

nnnnzczc dub534 221840
ch
nnnnzczc dub535 221846
ff eburzq eddyzq edduzq eddmzq lowwzq lyzazq lybezq lbsfzq
ff eburzq eddyzq edduzq eddmzq lowwzq lyzazq orbbia lybezq lbsfzq
221844 egkkzp
(dep-br6019-egkk1836)

nnnnzczc dub536 221855
ff eburzq eddyzq ebbrzq edduzq lowwzq lyzazq lybezq lbsfzq
ff obbizt egxybbim egllbako obbibako
221830 egllzp
(fpl-ba888-i
-b747-r/r1/k
-gll1850 lbsf2102 ltba2117 ltac2149 orbb2247 obb10009
-0496f330 dvr ug1 ntm ub6 tgo ub1w rtt ub1w v1w direct
kft ub5 smi ua4 dim/k905s1005 a4 rad/0496f330 va4n ist vg8
srt vr21 az r21 ba b55
-obb10024 omdb
-sel/baf reg/gawmb)

nnnnzczc dub537 221900
ch
nnnnzczc dub538 221901
ff lfrrzq lfffzq eburzr edduzq edafyo edaryo
ff lpazzq eggxzo lfrrzq lfffzq eburzr edduzq eguxyf edafyo edaryo
221810 lplayx
(dep-m40647-1805)

Field Type 3 - Message Type, Number and Reference Data

FIELD TYPE 3

Format: (*[a] [b] [c]

OPEN BRACKET

This type of field is used in	Next type of field
ALR	5
RCF	7
CPL	7
FPL	7
SPL	7
DEP	7
DLA	7
ARR	7
EST	7
CHG	7
CDN	7
ACP	7
TNR	7
TRA	7
RBI	7
AOC	7
CNL	7
TCX	7
RQS	7
RAR	7

(a) Message Type Designator

3 LETTERS as follows:

ALR	Alerting
RCF	Radio communication failure
CPL	Current flight plan
FPL	Filed flight plan
SPL	Supplementary flight plan
DEP	Departure
DLA	Delay
ARR	Arrival
EST	Boundary estimate
CHG	Modification
CDN	Co-ordination
ACP	Acceptance
TNR	Non-radar transfer of control
TRA	Radar transfer of control
RBI	Radar blip identification
AOC	Assumption of control
CNL	Flight plan cancellation
TCX	Transfer of control cancellation
RQS	Request supplementary flight plan
RAR	Request radar blip identification

* Unless instructed otherwise, this field shall contain only the single element (a). Elements (b) or (b) and (c) are for use when messages are generated by, and/or exchanged between the computer systems of ATS units.

(b) Message Number

1 LETTER identifying the sending ATC unit, followed by

1 LETTER identifying the receiving ATC unit, followed by

3 DECIMAL NUMERICS giving the serial number of this message in the sequence of messages transmitted from this ATC unit.

(c) Reference Data

2 LETTERS followed by 3 DECIMAL NUMERICS, giving the "message number" contained in element (b) of the operational message which began the sequence of messages of which this message is a part.

Examples: (FPL
(CNL
(CHGAB234AB123
(CPLAB002

Field Type 7 - Aircraft Identification and SSR Data

Format: -(a)Max. 7 characters*/b c

SINGLE HYPHEN

(a) Aircraft Identification

NOT MORE THAN 7 CHARACTERS, being the aircraft identification shown in the filed flight plan and composed as specified in Appendix 2, Section 2.

* This field may be terminated here in messages relating to flights operating within areas where SSR is not used, or when the SSR Code information is not known or would not be meaningful to the accepting unit.

OBLIQUE STROKE

(b) SSR Mode

1 LETTER giving the SSR Mode related to (c).

(c) SSR Code

4 NUMERICS giving the SSR Code assigned to the aircraft by ATS and transmitted in the Mode given in (b).

Examples: -BA902
-SK912/A5100
-VHABC/B2021

FIELD TYPE 7

Previous type of field	This type of field is used in	Next type of field
5	ALR	8
3	RCF	21
3	CPL	8
3	FPL	8
3	SPL	17
3	DEP	13
3	DLA	13
3	ARR	17
3	EST	13
3	CHG	13
3	CDN	13
3	ACP	13
3	TNR	23
3	TRA	24
3	RBI	24
3	AOC	)
3	CNL	13
3	TCX	)
3	RQS	13
3	RAR	)

Field Type 8 - Flight Rules and Type of Flight

FIELD TYPE 8		
Previous type of field	This type of field is used in	Next type of field
7	ALR	9
7	CPL	9
7	FPL	9

Format: - [a] [b] *

SINGLE HYPHEN

(a) Flight Rules

1 LETTER as follows:

I if IFR
V if VFR
Y if IFR first
Z if VFR first

Note.- If the letter Y or Z is used, the point at which the change of flight rules is planned is to be shown as indicated in Field Type 15.

* This field shall be terminated here unless indication of the type of flight is required by the appropriate ATS authority.

(b) Type of flight

1 LETTER as follows:

S if scheduled air transport
N if non-scheduled air transport
G if general aviation
M if military
X other flights

Examples: - V
- IS

Field Type 9 - Number and Type of Aircraft and Wake Turbulence Category

Format: -[a][b]/[c]

FIELD TYPE 9		
Previous type of field	This type of field is used in	Next type of field
8	ALR	10
8	CPL	10
8	FPL	10

SINGLE HYPHEN

(a) Number of Aircraft (if more than one)

Note.- This element is included only in the case of formation flights.

1 OR 2 NUMERICS giving the number of aircraft in the flight.

(b) Type of Aircraft

2 to 4 CHARACTERS, being the appropriate designator chosen from ICAO Doc 8643, Aircraft Type Designators, or

ZZZZ if no designator has been assigned or if there is more than one type of aircraft in the flight.

Note.- If the letters ZZZZ are used, the type(s) of aircraft is (are) to be shown in the Other Information Field (see Field Type 18).

OBLIQUE STROKE

(c) Wake Turbulence Category

1 LETTER to indicate maximum certificated take-off weight of the aircraft:

H - Heavy
M - Medium
L - Light

Examples: -DC3
-B707
-2DC4
-ZZZZ
-3ZZZZ
-B747/H

Field Type 10 - Equipment

Format: - [a] * / [b]

SINGLE HYPHEN

FIELD TYPE 10		
Previous type of field	This type of field is used in	Next type of field
9	ALR	13
9	CPL	13
9	FPL	13

(a) Radio Communication, Navigation and Approach Aid Equipment

1 LETTER as follows:

N no COM/NAV/Approach Aid equipment for the route to be flown is carried, or the equipment is unserviceable

OR S Standard COM/NAV/Approach aid equipment for the route to be flown is carried (See Note 1)

AND/OR ONE OR MORE OF THE FOLLOWING LETTERS to indicate the COM/NAV/Approach aid equipment serviceable

A	LORAN A	M	Omega
B	(Not allocated)	O	VOR
C	LORAN C	P	Doppler
D	DME	Q	(Not allocated)
E	Decca	R	RNAV route equipment
F	ADF	T	TACAN
G	(Not allocated)	U	UHF RTF
H	HF RTF	V	VHF RTF
I	Inertial Navigation	W	When prescribed by ATS
J	(Not allocated)	X	When prescribed by ATS
K	(Not allocated)	Y	When prescribed by ATS
L	ILS	Z	Other equipment carried (See Note 2)

Note 1.- Standard equipment is considered to be VHF RTF, ADF, VOR and ILS, unless another combination is prescribed by the appropriate ATS authority.

Note 2.- If the letter Z is used, the equipment carried is to be specified in Item 18, preceded by COM/ and/or NAV/ , as appropriate.

* This field shall be terminated here in messages relating to flights operating within areas where SSR is not used.

Field Type 10 (contd)

(b) SSR Equipment

1 CHARACTER to describe the serviceable SSR equipment carried:

N None

0 Transponder - no coding

2 Transponder (2 digits) - 64 Codes - Mode A

4 Transponder (4 digits) - 4096 Codes - Mode A but no Mode C

C Transponder (4 digits) - 4096 Codes - Modes A and C

Examples:
- N
- S/N
- SCH/C
- SZ/4

Field Type 13 - Aerodrome of Departure and Time, FIR Boundaries and Estimated Times

FIELD TYPE 13		
Previous type of field	This type of field is used in	Next type of field
10	ALR	15
10	CPL	14
10	FPL	15
7	DEP	17
7	DLA	17
7	EST	14
7	CHG	17
7	CDN	17
7	ACP	17
7	CNL	17
7	RQS	17

Format: - a* b** (sp) c d

See Note in margin on page A-39.

SINGLE HYPHEN

(a) Aerodrome of Departure

4 LETTERS, being

the 4-letter ICAO location indicator allocated to the aerodrome of departure, or

ZZZZ if no ICAO location indicator has been allocated (see Note 1), or

AFIL if the flight plan has been filed in the air (see Note 2).

Note 1.- If ZZZZ is used, the name of the aerodrome of departure is to be shown in the Other Information Field (see Field Type 18).

Note 2.- If AFIL is used, the ATS unit from which supplementary flight data can be obtained is to be shown in the Other Information Field (Field Type 18):

* This field shall be terminated here in message types CPL, EST, CHG, CDN, ACP, CNL and RQS.

(b) Time of Departure

4 NUMERICS giving

the estimated time of departure from the aerodrome in (a) in FPL and DLA messages transmitted before departure, or

the actual time of departure from the aerodrome in (a) in ALR and DEP messages, or

the actual time of departure from the first point shown in the Route Field (see Field Type 15) in FPL messages derived from flight plans filed in the air, as shown by the letters AFIL in (a).

Note.- The term 'estimated time of departure' denotes the time at which it is estimated that the aircraft will take off provided no delay is experienced.

** This field is to be terminated here if the flight is within a single FIR or it is known that the ACCs or FICs concerned do not require the information in elements (c) and (d).

Field Type 13 (contd)

SPACE

(c) FIR Boundary

4 LETTERS being the 4-letter ICAO location indicator allocated to the ACC or FIC serving an FIR to be entered.

(d) Estimated Time at FIR Boundary

4 NUMERICS giving the time at which the aircraft is estimated to enter the FIR in (c).

Note.- Further pairs of elements (c) and (d) may be added in continuous sequence, each pair of elements to be preceded by a space.

Examples: - EHAM0730
- EGTT1300 EINN1345 EGGX1500 CYQX1650

Field Type 14 - Boundary Estimate

FIELD TYPE 14		
Previous type of field	This type of field is used in	Next type of field
13	CPL	15
13	EST	17

Format: - a / b c * d e

SINGLE HYPHEN

(a) Boundary Point (see Note 1)

The BOUNDARY POINT expressed either by a designator consisting of 2 to 5 characters, in Geographical Coordinates, in Abbreviated Geographical Coordinates, or by bearing and distance from a designated point (e.g. a VOR).

Note 1.- This point may be an agreed point located close to, rather than on, the FIR boundary.

Note 2.- See 1.6 for data conventions.

OBLIQUE STROKE

(b) Time at Boundary Point

4 NUMERICS giving the estimated time at the Boundary Point.

(c) Cleared Level

F followed by 3 NUMERICS, or
S followed by 4 NUMERICS, or
A followed by 3 NUMERICS, or
M followed by 4 NUMERICS

} See data conventions in 1.6 of this Appendix.

giving the cleared level at which the aircraft will cross the Boundary Point, if in level cruising flight, or the cleared level to which it is proceeding, if climbing or descending at the Boundary Point.

* This field will be terminated here if the aircraft will cross the Boundary Point in level cruising flight.

Field Type 14 (contd)

(d) Supplementary Crossing Data

A LEVEL, expressed as in (c), at or above which or at or below which (see (e)) the aircraft will cross the Boundary Point.

(e) Crossing Condition

1 LETTER as follows:

A if the aircraft will cross the Boundary Point at or above the level in (d), or

B if the aircraft will cross the Boundary Point at or below the level in (d).

Examples:
- LN/1746F160
- CLN/1831F240F180A
- 5420N05000W/0417F290
- LNX/1205F160F200B
- ZD126028/0653F130

Field Type 15 - Route

FIELD TYPE 15		
Previous type of field	This type of field is used in	Next type of field
13	ALR	17
14	CPL	17
13	FPL	17

Format: - [a] [b] (sp) [c]

SINGLE HYPHEN

See Note in margin.

(a) Cruising Speed or Mach Number

4 NUMERICS giving the Cruising True Airspeed in knots, or

where prescribed on the basis of regional air navigation agreement, M, followed by 3 NUMERICS giving the Mach Number in hundredths of unit Mach.

(b) Requested Cruising Level

F followed by 3 NUMERICS, or
S followed by 4 NUMERICS, or
A followed by 3 NUMERICS, or
M followed by 4 NUMERICS, or
VFR.

See data conventions in 1.6 of this Appendix.

SPACE

followed by a string of elements/groups of elements of the following seven types separated by SPACES, in whatever sequence is necessary to describe the route in an unambiguous manner (see Appendix 2, Section 2).

Note.- Further element groups of elements (c) should be added, as necessary, each to be preceded by a space.

(c1) Standard Departure Route

The designator for the Standard Departure Route from the aerodrome of departure to the first significant point on the defined route to be flown.

Note 1.- See data convention in 1.6.3.1 (a) of this Appendix.

Note 2.- Element (c1) may be followed by (c3) or (c4).

Note 3.- Standard Departure Route need be included only where appropriate.

Field Type 15 (contd)

(c2) ATS Route Designator

Note 1.- See data convention in 1.6.3.1 (a) of this Appendix.

Note 2.- Element (c2) may be followed by (c3) or (c4) only.

(c3) Significant Point

Note.- See alternative data conventions in 1.6.3.1 (b), (c), (d) and (e) of this Appendix.

(c4) Significant Point/Cruising Speed and Cruising Level

SIGNIFICANT POINT (as in element (c2))

OBLIQUE STROKE

CRUISING SPEED OR MACH NUMBER (as in element (a))

REQUESTED CRUISING LEVEL (as in element (b)).

(c5) Indicator

VFR if a change to VFR is to be made at the preceding point, or

IFR if a change to IFR is to be made at the preceding point, or

DCT if the flight to the next point will be outside a designated route, unless both points are defined by geographical co-ordinates or by bearing and distance.

T if the route description is truncated at the preceding point and the remainder is to be sought in a previously transmitted FPL or other data.

Note 1.- Element (c5) may follow (c3) or (c4) and (c6) only.

Note 2.- When used, T must conclude the Route Field.

(c6) Cruise Climb

The letter C followed by an oblique stroke; then the point at which cruise climb is planned to start, expressed exactly as in (c3) above, followed by an oblique stroke; then the speed to be maintained during cruise climb expressed exactly as in (a) above followed by the two levels defining the layer to be occupied during cruise climb; each level expressed as in (b) above, or the level above which cruise climb is planned, followed by the letters PLUS, without a space between them.

(c7) Standard Arrival Route

The designator for the Standard Arrival Route from the point of leaving the defined route to the point at which the approach procedure is initiated.

Note.- Standard Arrival Route need only be included where appropriate.

Examples: -0255S1500 A4 CCV R11
-0180A120 BR 614
-0220F120 DF9 KIR VFR
-0360F290 A6 UA6 DEN/0440F290 UA5 RBT UB19 DXM UR10 RB DCT 41N005W DCT CCV/0340F100 UR24 R24 TUR 15A3
-M082F310 UG1 52N015W 52N020W 52N030W 50N040W 49N050W

Field Type 17 - Aerodrome of Destination and Time, Alternate Aerodrome(s)

FIELD TYPE 17		
Previous type of field	This type of field is used in	Next type of field
15	CPL	18
15	FPL	18
7	SPL	18
13	DEP	)
13	DLA	)
7	ARR	)
14	EST	)
13	CHG	22
13	CDN	22
13	ACP	)
13	CNL	)
13	RQS	)

Format: - [a] [b] (sp) [c]

* **

SINGLE HYPHEN

See Note in margin on page A-46.

(a) Aerodrome of Destination

4 LETTERS, being

the 4-letter ICAO location indicator allocated to the aerodrome of destination, or

ZZZZ if no ICAO location indicator has been allocated.

Note.- If ZZZZ is used, the name of the aerodrome of destination is to be shown in the Other Information Field (see Field Type 18).

* This field is to be terminated here in message types CPL, DEP, DLA, EST, CHG, CDN, ACP, CNL and RQS.

(b) Time of Arrival

4 NUMERICS, giving

the estimated time of arrival in ALR, FPL and SPL messages;

the actual time of arrival in ARR messages.

Note 1.- The term 'estimated time of arrival' denotes the time at which it is estimated that the aircraft will land provided no delay is experienced. Calculation of the estimated time of arrival in the case of an IFR flight to an aerodrome served by one or more navigation aids is based on the average time required by the aircraft to complete an instrument approach procedure at the aerodrome.

Note 2.- If and where so prescribed, the estimated time over the appropriate designated navigation aid serving the aerodrome of intended landing is inserted in Field Type 18.

** This field is to be terminated here in ARR messages and may be terminated here in FPL messages when so agreed between the ATS units concerned or prescribed on the basis of regional air navigation agreements.

Field Type 17 (contd)

SPACE

(c) Alternate Aerodrome(s)

4 LETTERS, being

the 4-letter ICAO location indicator allocated to an alternate aerodrome, or

ZZZZ if no ICAO location indicator has been allocated.

Note.- If ZZZZ is used, the name of the alternate aerodrome is to be shown in the Other Information Field (see Field Type 18).

Note.- One further element of (c) should be added, as necessary, preceded by a space.

Examples: -EINN1230
-EINN1230 EGLL
-EINN1230 EGLL LFPO

Field Type 18 - Other Information

FIELD TYPE 18		
Previous type of field	This type of field is used in	Next type of field
17	ALR	19
17	CPL	)
17	FPL	)
17	SPL	19

Format: - [a]

or

- [ ](sp)[ ](sp)*(sp)[ ]

(*additional elements as necessary)

SINGLE HYPHEN

(a) 0 (zero) if no other information

OR

Any other necessary information in the preferred sequence shown hereunder, in the form of the appropriate abbreviation followed by an oblique stroke and the information to be recorded:

REG/ The registration markings of the aircraft, only if necessary and if different from the aircraft identification in Item 7.

SEL/ SELCAL Code, if so prescribed by the appropriate ATS authority.

OPR/ Name of the operator, if not obvious from the aircraft identification in Item 7.

STS/ Reason for special handling by ATS e.g. hospital aircraft, one engine inoperative, e.g. STS/HOSP, STS/ONE ENG INOP.

TYP/ Type(s) of aircraft, preceded if necessary by number(s) of aircraft, if ZZZZ is used in Item 9.

PER/ Aircraft performance data, if so prescribed by the appropriate ATS authority.

COM/ Any additional COM equipment, if the letter Z is inserted in Item 10.

Field Type 18 (contd)

NAV/	Any additional NAV equipment, if the letter Z is inserted in Item 10.
DEP/	Name of aerodrome of departure, if ZZZZ is inserted in Item 13, or the 4-letter ICAO location indicator of the location of the ATS unit from which supplementary flight plan data can be obtained, if AFIL is inserted in Item 13.
EST/	Significant points and estimated times over such points, when so prescribed on the basis of regional air navigation agreements or specified by the appropriate authority. In each group of information, separate the point and the estimated time over the point by an oblique stroke, e.g. EST/CAP/1745 XYZ/1830. *Note.- When the estimated times refer to geographical co-ordinates, use only the latitude or longitude, as appropriate, to identify the points, e.g.* EST/60N/1540 55N/1613 50N/1648 *etc., or* EST/10W/1025 20W/1105 30W/1145 *etc.*
DEST/	Name of aerodrome of destination, if ZZZZ is inserted in Item 17.
ALTN/	Name of alternate aerodrome(s), if ZZZZ is inserted in Item 17.
RMK/	Any other plain language remarks when required by the appropriate ATS authority or deemed necessary by the pilot-in-command for the provision of air traffic services.

Examples:
- 0
- EST/15W/1515 20W/1537 30W/1620 40W/1702
- STS/ONE ENG INOP

A2. DONNEES DE BASE ET NOTATIONS DE SCAPA

A2.1/ Liste des classes, des relations, des productions, des opérateurs.

A2.2/ Graphe des productions

A2.1.a) Liste des classes et leur équivalent numérique

N°	Abbréviation	Plus petite classe contenant cette classe	signification
1	α	3	Caractère alphabètique
2	v	3	Caractère numérique
3	αv	7	Caractère alphanumérique
4	char	6	Caractère non alphanumérique autre que @ ou ␣ ou - ou) ou /
5	sep	6	Séparateur : @ ou ␣ ou - ou) ou /
6	anychar	7	Caractère non alphanumérique
7	any	-	Caractère quelconque
10	chp7	-	Callsign
16	chp15	-	Champ 15
20	chp8	-	Règles et type du vol
30	chp9	-	Nombre et type d'avions et classe de turbulence
40	chp10	-	Equipement radio et radar secondaire
50	chp13	-	Aéroport et heure de départ/arrivée
60	chp18	-	Commentaires
100	LL1	-	Ligne logique 1
101	knots	103	Vitesse en noeuds
102	FFT	104	Niveau de vol en pieds
103	vitk	-	Vitesse
104	haut	-	Hauteur
105	crois	-	Paramètre de croisière
106	Bal	-	Balise syntaxique
107	Arws	-	Airway syntaxique
108	Binc	112	Balise inconnue
109	BI	112	Balise interne
110	BF	112	Balise frontière
111	BX	112	Balise externe
112	RPNT	-	Point de report
113	arw	115	Airway connue
114	arwinc	115	Airway inconnue
115	arwy	-	Airway
117	Noproc (ou NP)	-	Numéro de procédure
118	procdep	-	Procédure de départ

A2.1.a) suite

N°	Abbréviation	Plus petite classe contenant cette classe	signification
119	SGT	115	Segment
120	eof	-	Fin de champ 15
121	FARPT ou FA	-	2 lettres finales d'indicatif d'aéroport
122	AFT	104	Altitude en pieds
123	MACHS	103	Vitesse en nombre de Machs
124	VFR	-	Règles de vol à vue
125	IFR	-	Règles de vol aux instruments
126	DCT	-	Indicateur "direct"
127	INTXN	-	Indicateur "intersection"
128	ARPT	-	Aéroport
129	FM	104	Niveau de vol en mètres
130	AM	104	Altitude en mètres
131	Dblt	-	Doublet
132	procar	-	Procédure d'arrivée
133	chp15X	-	Champ 15 externe
200	LL2	-	Ligne logique 2
300	LL5	-	Ligne logique 5
400	FPL	-	Message FPL

A2.1.b) Liste des relations

Nom de relation	N°	type	signification
vitk	1	RECHERCHES GENERALES	vitesse en noeuds (knots)
hautF	2		hauteur en pieds (feet)
arpt	3		aéroport
Keg	4		point interne le plus à gauche
Ked	5		point interne le plus à droite
type	6		type (civil/militaire/interne)
rule	7		règles (instruments/vue)
esp	8		espace (Supérieur/Inférieur)
peg	10		point extrémité gauche
reg	11		route extrémité gauche
ped	12		point extrémité droite
red	13		route extrémité droite
vitG	22		vitesse à gauche
vitD	23		vitesse à droite
vit	24		vitesse (gauche-droite)
hautG	25		hauteur à gauche
hautD	26		hauteur à droite
haut	27		hauteur (gauche-droite)
ruleG	28		règle de vol (I/V) à gauche
ruleD	29		règles de vol(I/V) à droite
rule	30		règles de vol (changement de règles)
nxthaut	31		a comme hauteur après
ppeg (ou $\widetilde{peg}$)	40		pseudo relations
preg (ou $\widetilde{reg}$)	41		
pped (ou $\widetilde{ped}$)	42		
pred (ou $\widetilde{red}$)	43		
	20	RECHERCHES RESEAU	appartient
alias	9		alias (balise correspondant à un aéroport
entry	14		entrée dans l'UIR
exit			sortie de l'UIR
SR/prch	15		est sur route ou est proche de la route
CX	16		a comme coordonnées en x
CY	17		a comme coordonnées en y
$=, \neq, <, >, \ldots$	3		significations classiques

A2.1.b) suite

nom de relation	N°	type	signification
CDX	18	calcul	déplacement en x correspondant à une radiale distance
CDY	19		déplacement en y correspondant à une radiale distance
ADD			addition
ABS			valeur absolue
SUB			soustraction
MAX,MIN			maximum, minimum
MUL		CALCUL	multiplication
TMK			transformation de machs en noeuds
DIV			division
ORD			ordonner une liste
MOINS			opposé
LIST			élément "listé"
OCLSS			ordonné par rapport à la classe
OAB			ordonné par application "à blanc"
DEB (ou depile)			début de liste
RET (ou retniv)			retrouve le niveau
SUBST$\emptyset_i$			substitution par l'opérateur $\emptyset_i$
REP (ou retour niv.)		APPRENTISSAGE	reprise au niveau
LST			a comme dernier caractère
RCLSS			recherche par rapport à la classe
LITT			a comme partie littérale
MODSTRUC			modifie la structure
CARDSUP			a un cardinal supérieur à

A2.1.c) Liste des productions pour le champ 15

N°	Type Lexical/ Sémantique	Structure	Résultats
1	L	$(v)^4$	knots
2	L	$\alpha(v)^3$	AFT|Machs|FFT
3	L	(vitk,haut)	crois
4	L	$(\alpha)^2$	Farpt
5	L	v	Noproc
6	L	(Farpt,Noproc)	procdep
7	L	$(\alpha)^3$	VFR|IFR|Bal|DCT|INT
8	L	$\alpha(\alpha v)^2$	arws
9	L	char	eof
10	L	$(\alpha)^2(\alpha v)^2$	arws
11	L	arws	arw|arwinc
12	L	Bal	BI|BF|BX|Binc
13	S	(Rpnt,arwy)	SGT
14	S	(procdep,SGT)	SGT
15	S	(SGT,SGT)	SGT|Dblt
16	S	(crois,SGT,eof)	chp15|chp15X
17	L	(Rpnt,char,crois)	Rpnt
18	S	(arwy,arwy)	SGT
19	L	$(v)^2$	Noproc
20	S	(SGT,arwy)	SGT
21	S	(arwy,Rpnt)	SGT
22	L	$\alpha\ \alpha v$	arws
23	S	(Rpnt,Rpnt)	SGT
24	S	(SGT,Rpnt)	SGT
25	L	$(\alpha)^5$	Bal
26	L	(Noproc,Farpt)	procar
27	S	(SGT,procar)	SGT

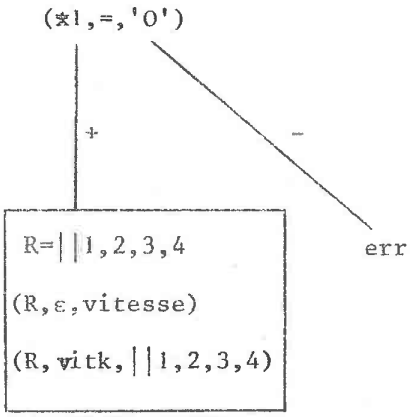
A2.1.d) <u>Liste des Opérateurs</u> (et leur pondération)

Opérateur	Poids	fonction
Ø1	90	suppression d'un caractère isolé (sommet égal à un caractère)
Ø3	85	suppression d'un bloc du message source
Ø4	70	substitution spécifique I ⟷ 1
Ø5	70	substitution spécifique Ø ⟷ 0
Ø6	70	substitution spécifique 9 ⟷ G
Ø7	65	substitutions : (3↔8 ; 3↔B ; 8↔B ; U↔V ; 4↔A ; 5↔S ; 7↔1 ; M↔H ; R↔P ; E↔F ; C↔G ; I↔E)
Ø8	60	substitution de la dernière lettre =X par une autre lettre
Ø9	55	substitution de la dernière lettre par une autre lettre
Ø10	55	substitution de la dernière lettre par un caractère ou un numérique
Ø11	50	substitution d'une lettre par une autre lettre
Ø12	50	substitution d'un nombre par un autre nombre (dans la partie numérique d'une chaine)
Ø13	20	insertion d'un élément

A2.2/ GRAPHES DES PRODUCTIONS.

Note : Le méta-langage des graphes de productions est décrit au III.2.b).

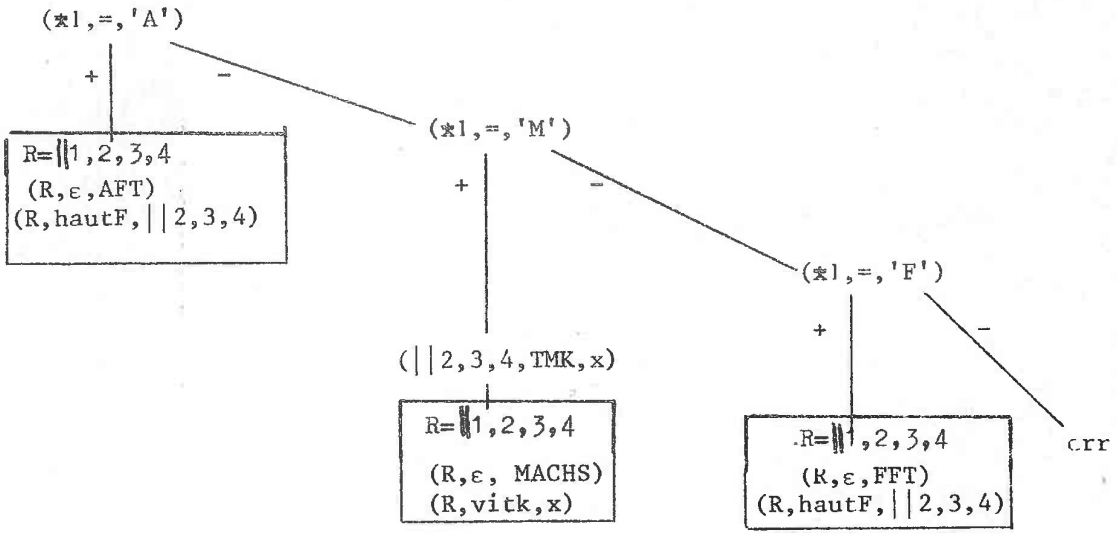
PRO 1 $(<\nu>^4)$



Remarque :

Les vitesses ne doivent pas dépasser 999 noeuds.

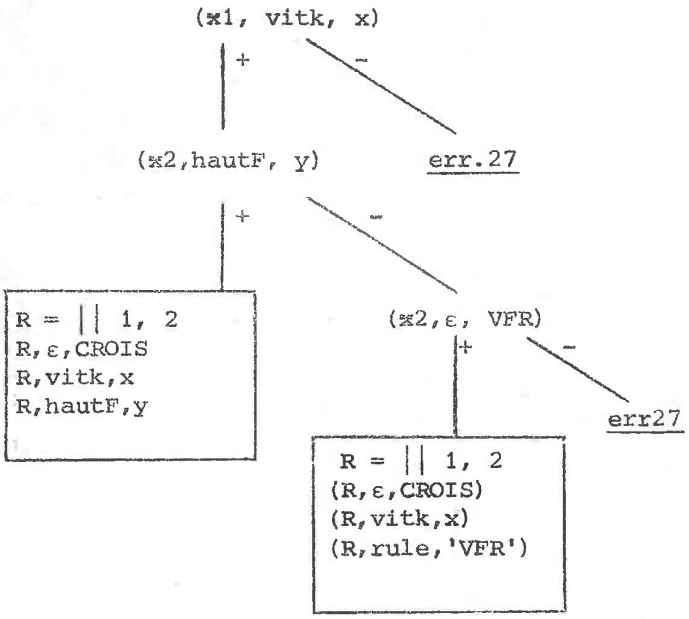
PRO 2 $(<\alpha>,<\nu>^3)$



Remarque :

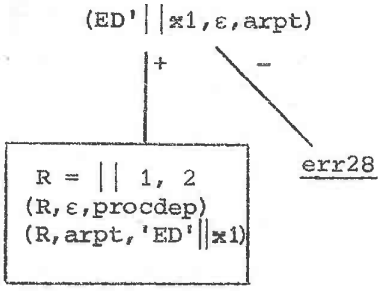
Selon que la 1ère lettre est A, M ou F, le résultat est AFT (altitude in feet), MACHS (vitesse en nombre de Machs) ou FFT (Flight Level in feet). Pour MACHS, le triplet (||2,3,4,TMK,x) transforme la vitesse, de Machs en noeuds et assigne x.

pro3 (vitesse, altitude)



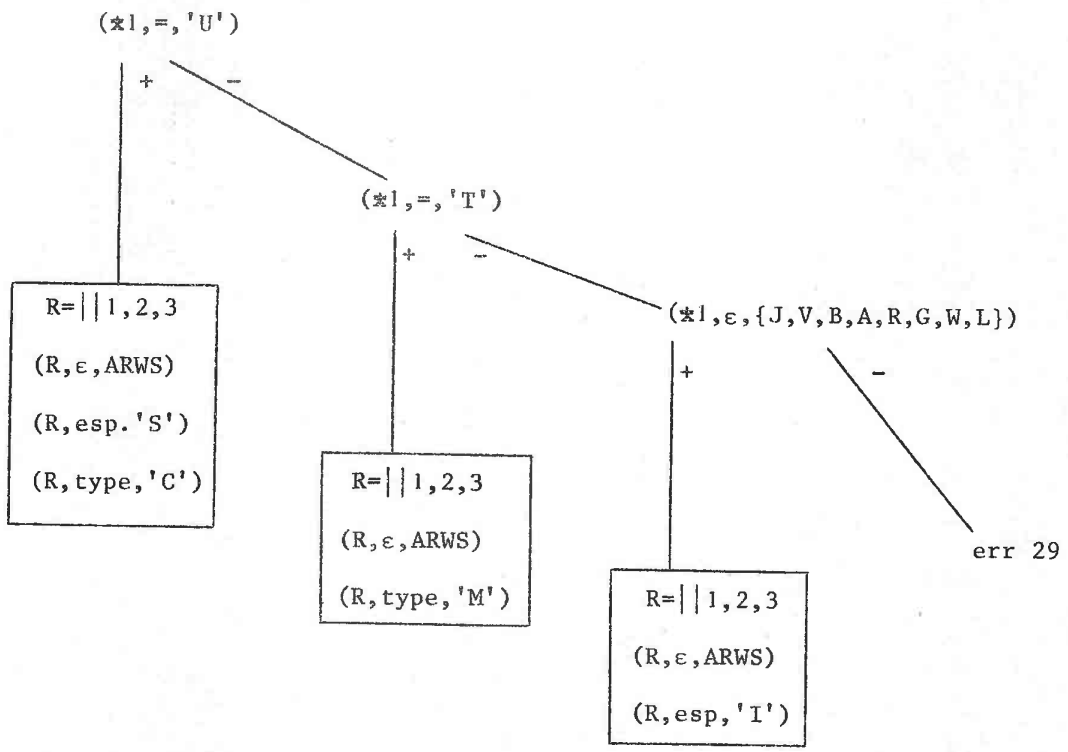
remarque : les tests retrouvent la vitesse en noeuds et la hauteur en centaines de pieds si le 2ème terme est bien une vitesse. Si c'est "VFR", vol à vue, l'information est conservée dans (R,rule,'VFR').

<u>pro6</u> (Farpt, Noproc)



<u>remarque</u> : si l'aéroport relatif à cette procédure de départ est connu, le résultat est fabriqué et le triplet (R,arpt,"ED"||x1) conserve le nom d'aéroport.

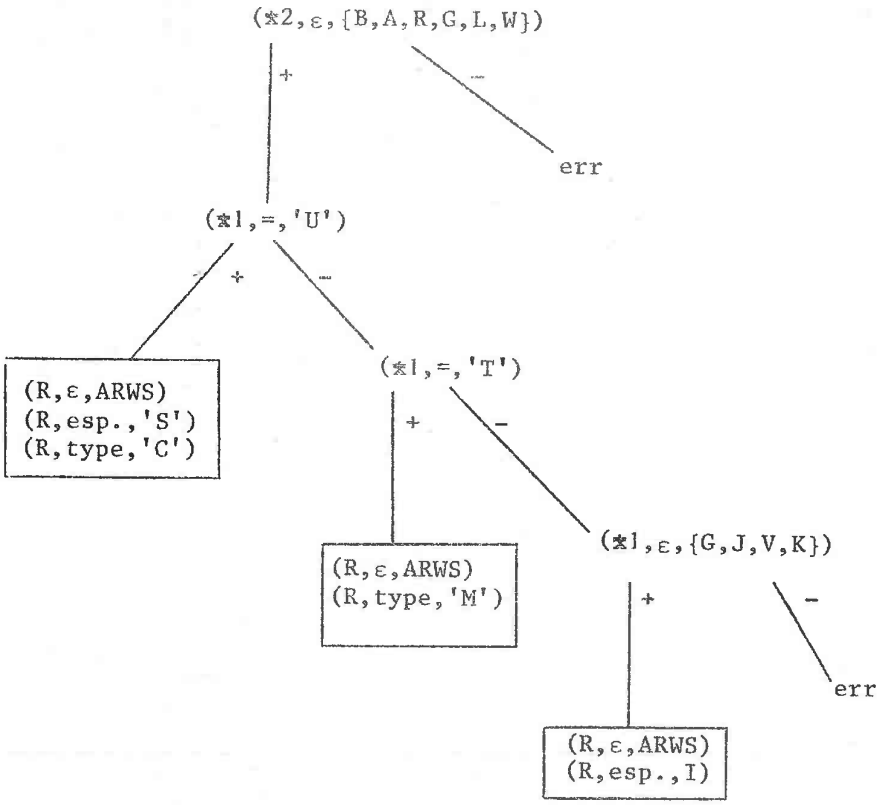
PRO 8 $(<\alpha>,<\alpha\nu>^2)$



Remarque :

- Les routes commençant par U sont dans l'espace Supérieur de type Civil
- celles commençant par 'T' sont de type Militaire
- les autres (commençant par {J,V,B,A,R,G,W,L}) sont dans l'espace Inférieur.

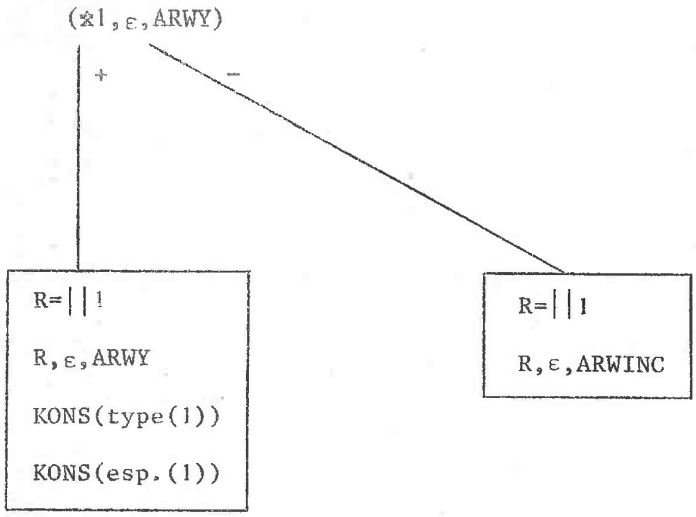
PRO 10 $(<\alpha>^2, <\alpha\nu>^2)$



Remarque :

- Les routes commençant pas 'U', suivies de {B,A,R,G,L,W} sont dans l'espace Supérieur, de type Civil
- celles commençant par 'T' suivies de {B,A,R,G,L,W} sont de type Militaire
- celles commençant par 'J' ou 'G' ou 'V' ou 'K' suivies de {B,A,R,G,L,W} sont dans l'espace Inférieur.

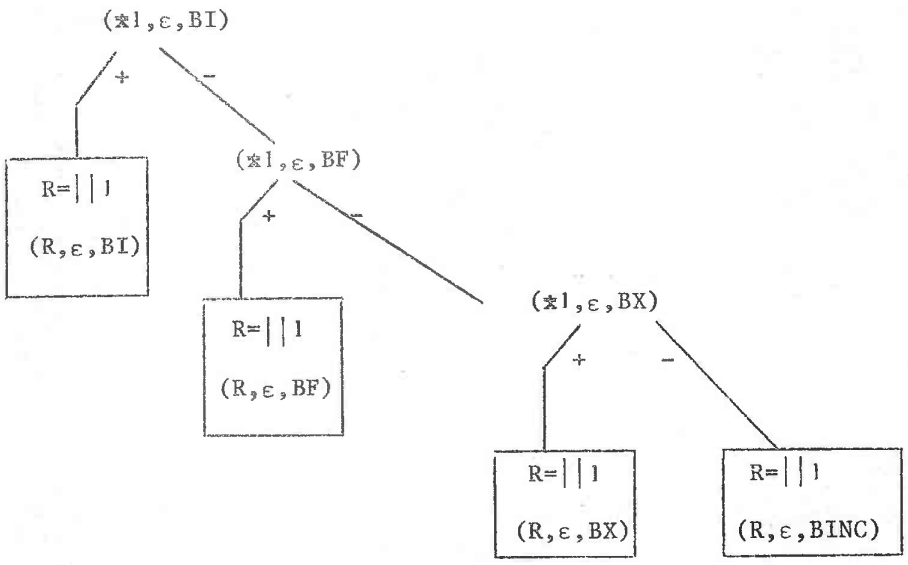
PRO 11 (ARWS)



Remarques :

- recherche si la route est connue dans le réseau
- KONS (rel(i)) est une notation indiquant que si l'élément *i figure dans un triplet du genre (*i,rel,x), alors cette information est conservée pour le résultat de production R, et le triplet (R,rel,x) est fabriqué.
- ici : si la route est connue, on conserve les informations type et espace, présumées dans la phase lexicale (qui produisait ARWS : AiRWay Syntaxique).

PRO 12 (Bal)

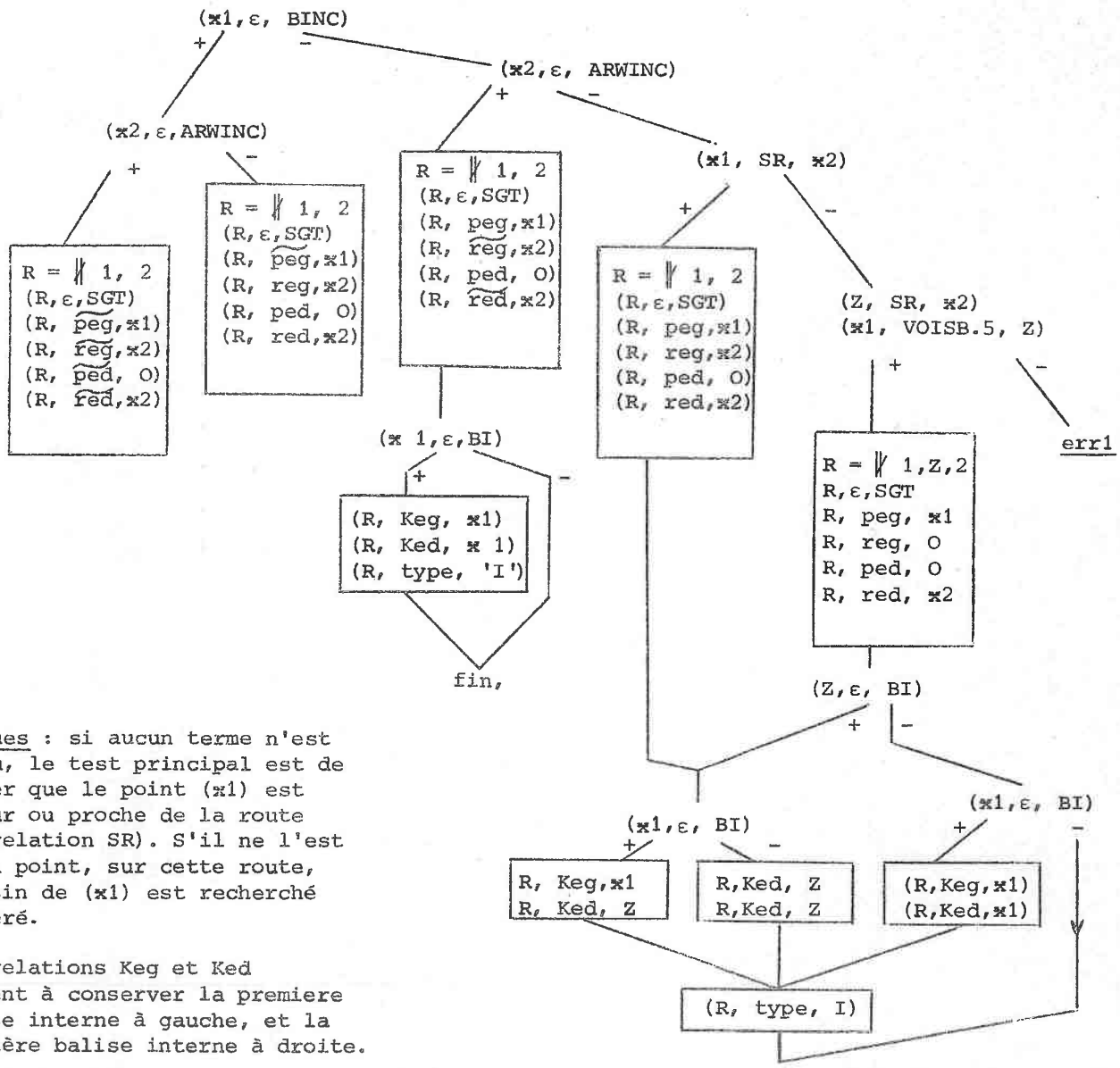


Remarques :

- Bal est la classe des objets présumés balise après l'analyse lexicale

- ici, la distinction BI (balise interne), BF (balise frontière) BX (balise externe), BINC (balise inconnue) est faite en fonction des informations du réseau.

pro13 (RPNT,ARW)



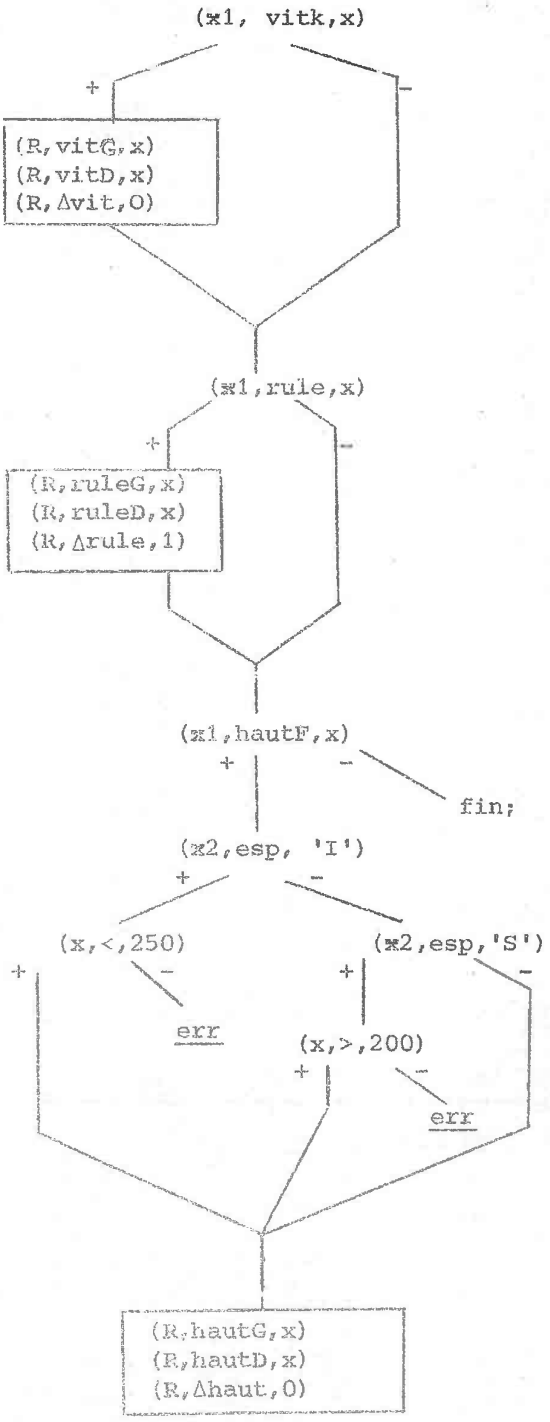
tests de compatibilité
(page suivante)

Remarques : si aucun terme n'est inconnu, le test principal est de vérifier que le point (x1) est bien sur ou proche de la route (x2).(relation SR). S'il ne l'est pas, un point, sur cette route, et voisin de (x1) est recherché et inséré.

- Les relations Keg et Ked servent à conserver la premiere balise interne à gauche, et la dernière balise interne à droite.
  - même si le point ou la route sont inconnus, un segment est crée.
  - les extrémités connues sont conservées dans les triplets
    (R,peg,x) x est le point extrémité gauche
    (R,reg,y) y est la route extrémité gauche
    (R,ped,z) z est le point extrémité droite
    (R,red,t) t est la route extrémité droite
  - si une extrémité est inconnue, on conserve cette "pseudo-extrémité" avec les pseudo-relations correspondantes : peg, reg, ped, red.

pro13 (suite)

Tests de compatibilité



remarque : les tests de compatibilité conservent au segment la vitesse et les règles du point (x1) et confrontent l'altitude éventuelle en ce point avec l'espace associé à la route (espace supérieur : $x > 200$ espace inférieur : $x < 250$)

ANNEXE 4

A4. NOTES TECHNIQUES SUR L'IMPLANTATION DE SCAPA

A4.1/ Les Modules

A4.2/ L'Overlay

A4.3/ Les Fichiers

A4.1. Les Modules

SCAPA est divisé en plusieurs modules :

Il appelle le module d'analyse - correction : ASC
puis les modules de l'apprentissage :

ISØLC15	isole le champ 15
DELIM	délimite la portion soumise à l'apprentissage
MATCH	match les parties identiques
RPA	recherche les points d'attaque
HEURS	recherche les opérateurs applicables
EVALMP	évalue la solution (pt. d'attaque, opérateur) minimale
NIVB	évalue le niveau de blocage et l'action à faire (création/modification)
RP	recherche les paramètres généraux et particuliers
APPL	construit les greffons et les applique ; effectue la reprise

Le module ASC reflète les considérations du III.2 a) β) c'est à dire qu'il utilise un algorithme d'analyse : le module ASCANAL ; il l'applique d'abord à tous les blocs du message (analyse lexicale) puis ensuite à tous les résultats d'une ligne ; puis à toutes les lignes.

exemple pour analyser 0300F250 LST TB1 DKB

ASC procède à :

lexicale
- call ASCANAL (0300F250) ⟶ crois
- call ASCANAL (LST) ⟶ Bal
- call ASCANAL (TB1) ⟶ arws
- call ASCANAL (DKB) ⟶ Bal

syntaxique
- call ASCANAL (crois, Bal , arws , Bal) ⟶ chp 15

Nous allons donner pour tous les modules un certain nombre de renseignements : fichier PL1 source, taille en K octets du load module avec références externes non résolues (NCAL), paramètres d'appels, groupes de déclarations par % include, procédures ou modules appelés.

a) tous les textes PL1 source sont dans le Dataset de travail B5DER. DEV. PLI, sur disque WØRKOi

b)

Nom du module	Taille du Load avec NCAL (K octets)	Paramètres d'appel	DCL sections (% include)	appels de procédures ou modules externes
ASC	4,1	RES	-	DCPCHP, SPBLK, ASCANAL
ASCANAL	6,1	L, B, DEB, FIN	SCAPEX	ASTRUC, ARECH, ATLIST STKL, AAPPL, PRØØØ, APA, CSC2
ISØLC15	4,7	P, T, B, TXT12, RM	-	SØML, RECHAPP,
DELIM		B, T, NT	-	-
MATCH	0,8	A1, TXT1, A2, TXT2	-	-
RPA	3,1	-	-	-
HEURS	13	A1, TXT1, A2, TXT2	DCLAP	-
EVALMP	1,2	-	-	-
NIVB	7,35	A1, pile 1, TS1, Taberr 1 A2	DCLAP	SØML
RP	12,2	A, TXT1,LISTR1, TXTR1,A2,TXT2, CWA,PG,TRPR,SC, TINF	DCLAP	RECHCH RUM
APPL	13	PG,TRPR,TINF, RM	DCLAP	

c) les modules objets correspondants sont dans le dataset B5DER.DEV.ØBJ sur disques WORKi

d) les load modules avec références externes non résolues (NCAL) sont dans B5DER.DEV.LØAD sur disque WØRKi.

e) <u>les autres procédures externes</u>

- Ce sont

  - <u>les productions</u> : PRØ1 à PRØ27 + M150,M151,M152,M153,M154,MM152A, MM152B

  - <u>les procédures spécialisées de ASCANAL</u> :
    (découpage effectué en vue de l'overlay)

    - ASTRUC (calcule la structure)
    - ARECH (recherche les productions)
    - ATLIST (scrute la liste LISP)
    - AAPPL (applique les productions)
    - PRØØØ (applique les productions)
    - APA (après l'application)
    - CSC2 (poursuite ou reprise ?)

  - <u>les procédures spécialisées de ASC</u> :

    - DCPCHP (découpe en lignes)
    - SPBLK (découpe les lignes en blocs)

  - <u>des utilitaires</u> :

    - VOISB (vérifie le voisinage de balises)
    - KONS (conserve des triplets TR)
    - NSGT (crée les triplets pour un nouveau SGT)
    - TKK (crée les triplets, Keg, Ked, type pour SGT)

    - RECH
    - RECHCH
    - RECHL < MRECHLA / MRECHLB — recherche de triplets dans le réseau ou l'extension
    - RLCH
    - RECHAPP
    - RUM

    - RET (retrouve le dernier indice libre d'une classe)
    - SOMMET (calcule le sommet d'une pile)
    - STK (empile)
    - RMV (dépile)
    - SOML (calcule le sommet d'une pile à 2 dimensions)
    - STKL (empile le sommet d'une pile à 2 dimensions)
    - RMVL (dépile le sommet d'une pile à 2 dimensions)
    - MESIMP (imprime certains messages)
    - ANALT (analyse terminale (caractères))

A4.2. L'overlay

Afin de réduire la place mémoire nécessaire à l'exécution de SCAPA et ainsi d'obtenir un "turn around time" plus court (soit ½ journée), nous avons structuré SCAPA en overlay : au lieu d'avoir un load-module dont toutes les "control-sections" sont dans l'UC pendant toute l'exécution, le load module est structuré de sorte qu'à l'exécution, certaines "control-sections" ne sont chargées que lorsqu'elles sont référencées. Lorsqu'une référence apparait dans une c-sect en cours d'exécution, le système détermine si le code est déjà ou non dans l'UC. S'il n'y est pas, le code est chargé dynamiquement et peut recouvrir une portion non utilisée du load-module en mémoire.

Nous avons dû en outre utiliser un overlay à régions multiples : il s'agit de pratiquer à nouveau un overlay sur les c-sect qui sont dans la racine. On peut utiliser 4 régions ; pour chaque région, les règles d'exclusion de c-sect sont celles de l'overlay à une région. Ainsi un segment a accès à des segments qui ne sont pas sur son chemin, s'ils sont dans une autre région.

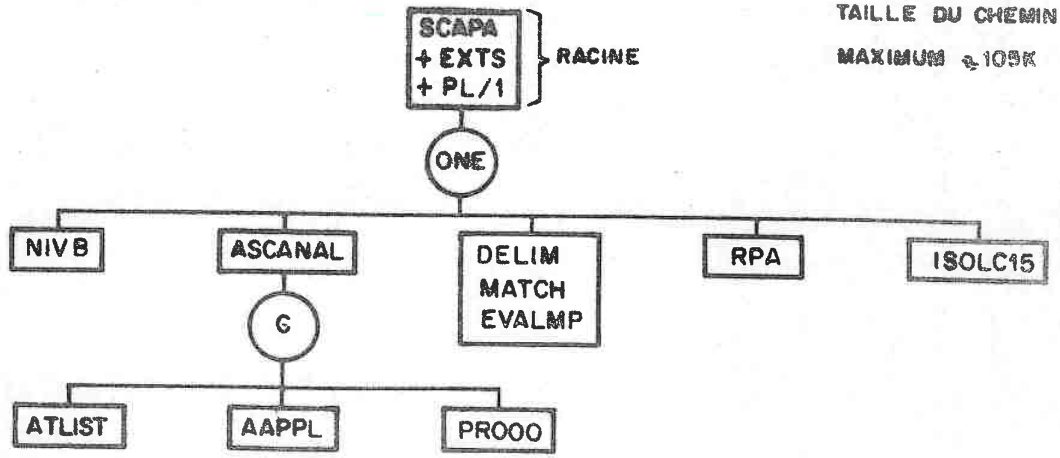
Nous avons utilisé 4 régions.

Les modules initialement de grande taille ont été découpés en procédures, et le nombre des variables externes diminué. Le résultat final est un load module d'environ 140 K octets, exécutable en 180 K octets (40 K octets pour les variables dynamiques et routines PL/1 dynamiques).

ARBRE D' OVERLAY

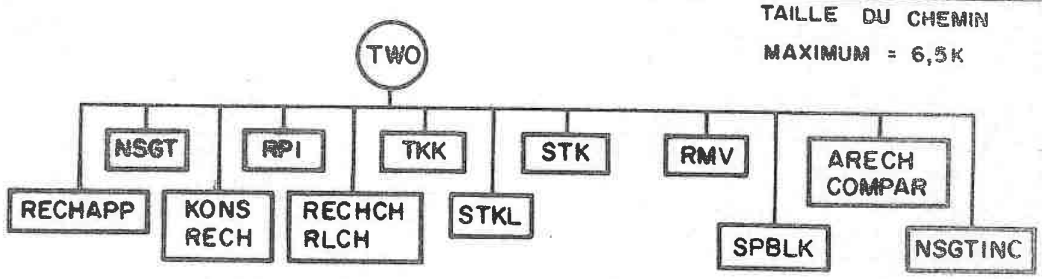
REGION 1

TAILLE DU CHEMIN MAXIMUM ≈ 109K



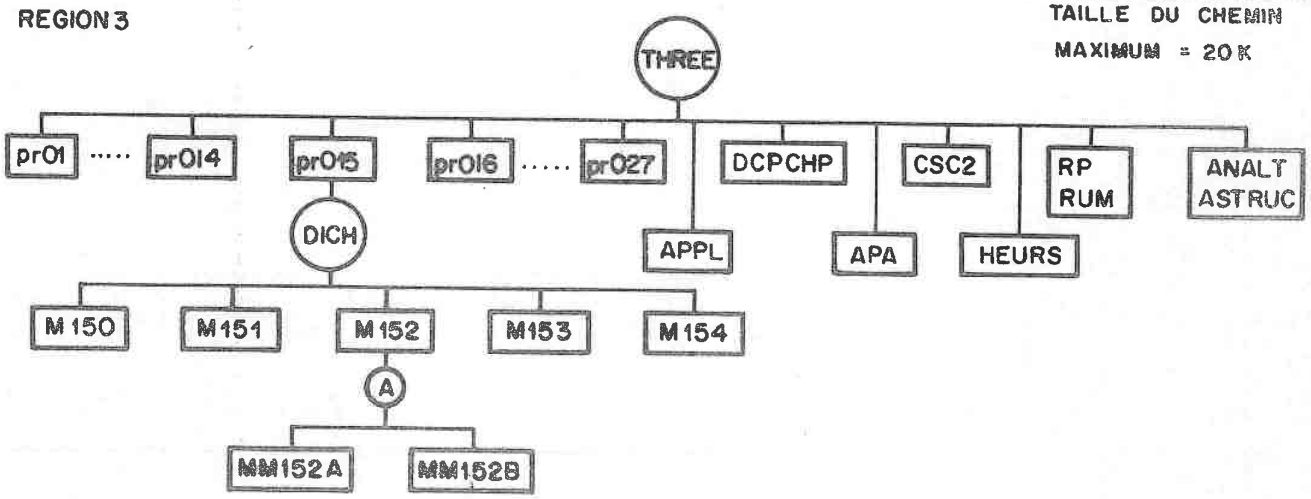
REGION 2

TAILLE DU CHEMIN MAXIMUM = 6,5K



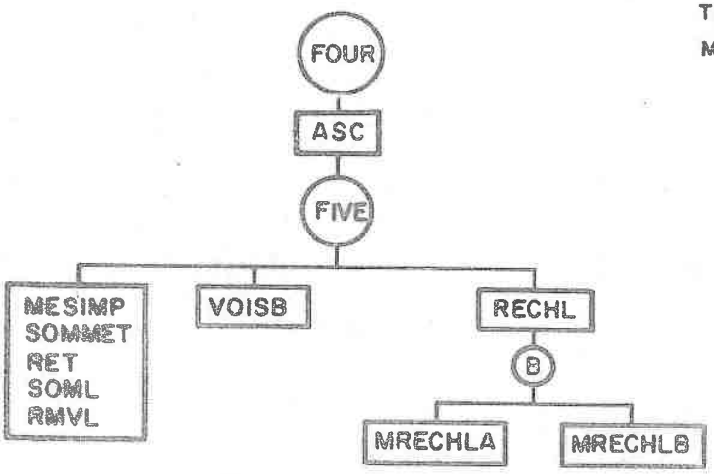
REGION 3

TAILLE DU CHEMIN MAXIMUM = 20K



REGION 4

TAILLE DU CHEMIN MAXIMUM = 8K



LES NOEUDS D'OVERLAY SONT ENCERCLES ; LES SEGMENTS ENCADRES ;
LONGUEUR TOTALE = 139.5K

A4.3. Les fichiers

a) Tous les textessource PL/1 sont dans
B5DER.DEV.PLI

b) Tous les modules objet sont dans
B5DER.DEV.ØBJ

c) Certains load modules (exécutables ou non) sont dans
B5DER.DEV.LOAD

d) Les données :

- La table décrivant les productions (structure, longueur, zone) est dans B5DER.PRØCS.CNTL (prod)
(qui est lue par un get data)

- les fichiers fournissant les couples (FPL source, FPL corrigé) sont : B5DER.procs.cntl (FPLSC)
  : B5DER.procs.cntl (FPLSC2)
  : B5DER.procs.cntl (FPLSC3)

- certaines données du réseau (extraits) sont dans
  : B5DER.procs.cntl (DATAP)
  ou : B5DER.procs.cntl (DAT1) ou : B5DER.procs.cntl (DATU)
  ou : B5DER.procs.cntl (DATY) ou : B5DER.procs.cntl (DATT)
  ou : B5DER.procs.cntl (DATZ)
  ou : B5DER.procs.cntl (DATX)

- toutes les données du réseau sont dans 3 fichiers :

  B5DER.BALT.DATA
  B5DER.ARWY.DATA
  B5DER. SR. DATA

  fichiers d'accès directs, de type régional (1).
  sur disque privé PR1404.

- l'ensemble des messages sources disponibles est dans le dataset KAX.SRCTX.DATA ;

  un ensemble de messages corrigés est dans KAX.CØRTX.DATA ;
  tous 2 d'accès direct sur PR1404.

e) Les statements d'overlay

Ils sont dans B5DER.procs.cntl (OVL7)
et décrivent au loader l'overlay tree du A4.2.

f) Les fichiers output

Afin de réduire le volume du print out, certains résultats sont copiés sur un fichier output (fichier ZZZ dans lesJCL).

Le dataset correspondant est B5DER.SCLIST.DATA, sur disque WØRK.

Ce fichier est utilisé pour le debugging ;

Pour une exécution avec print réduit, la carte

//ZZZ DD DSN = B5DER.SCLIST.DATA, DISP = old

est remplacé par

//ZZZ DD DUMMY

g) Les JCL

Les cartes de commandes des différents jobs sont des membres du Data Set B5DER.procs.c ntl sur disque WØRK.

B5DER.procs.cntl (SCPL)	exécute des compilations de sous-programmes PL/1
B5DER.procs.cntl (SCLK)	effectue l'édition de lien des modules objets et ordonne le load-module suivant l'arbre d'overlay
B5DER.procs.cntl (SCAPA)	effectue en option les opérations précédentes et lance l'exécution.

Des jobs dérivés de SCAPA comportent des échantillons de messages différents :

SCAP1, SCAPX, SCAPY, SCAPZ, SCAPT, SCAPU.

ANNEXE 5

A5 : Jeu d'essai

A5.1/ Autres exemples d'analyse-correction

A5.2/ Autres exemples d'apprentissage

A5.1/ Autres exemples d'analyse-correction

Nous allons ici détailler le listing de l'exemple d'analyse-correction de la page 71 (message AB).

Suivront les listings de quelques autres analyses .

Message AB :

```
(FPL-TESCAPX-Z
-C130-N
-EDNP
-0520F260 LPH TGO TB6 NTM TR10 NOR
-EDSL EDSF
-0)
```

ETAT DE LA PILE

L	B	NIV	DEB	FIN	CLSSR		INDIC
1	1	1	5	11	10	1	1
1	2	1	13	13	20	1	1
1	0	1	1	2	100	1	2
2	1	1	17	20	30	1	1
2	2	1	22	22	40	1	1
2	0	1	1	2	200	1	2
3	1	1	26	29	50	1	1
3	0	1	1	1	50	1	2
4	1	1	33	36	101	1	3
4	1	1	37	40	102	1	3
4	1	2	1	2	105	1	1
4	2	1	42	44	106	1	1
4	3	1	46	48	106	2	1
4	4	1	50	52	107	1	1
4	5	1	54	56	106	3	1
4	6	1	58	61	107	2	1
4	7	1	63	65	106	4	1
4	8	1	71	71	120	1	1
4	0	1	1	1	105	1	3
4	0	1	2	2	110	1	3
4	0	1	3	3	109	1	3
4	0	1	4	4	113	1	3
4	0	1	5	5	109	2	3
4	0	1	6	6	113	2	3
4	0	1	7	7	110	2	3
4	0	1	8	8	120	1	3

classe du résultat

indice dans cette classe

analyses lexicales

analyse du niveau 1

n° de la ligne

n° du bloc

niveau

indicateur du type du résultat

TABLE ASSOCIATIVE

CLS	NO	PTR	LG CHNE
10	1	1	7: TESCAPX
1	2	8	1 Z
20	1	9	1: Z
100	1	10	9 TESCAPX Z
30	1	19	4 C130
1	6	23	1 N
40	1	24	1 N
200	1	25	6 C130 N
50	1	31	4 EDNP
101	1	35	4 0500
102	1	39	4 F260
105	1	43	8 0500F260
106	1	51	3 LPH
106	2	54	3 TGO
107	1	57	3 TB6
106	3	60	3 NTM
107	2	63	4 TR10
106	4	67	3 NOR
4	19	70	1 >
120	1	71	1 >
110	1	72	3 LPH
109	1	75	3 TGO
113	1	78	3 TB6
109	2	81	3 NTM
113	2	84	4 TR10
110	2	88	3 NOR

4	0	2	1	1	105	1	3	niveau 2	119	1	91	7	LPH TGO	
4	0	2	2	3	119	1	3		119	2	98	7	TB6 NTM	
4	0	2	4	5	119	2	3		119	3	105	8	TR10 NOR	
4	0	2	6	7	119	3	3		119	4	113	15	LPH TGO TB6 NTM	
4	0	2	8	8	120	1	3		119	5	128	24	LPH TGO TB6 NTM TR10 NOR	
4	0	3	1	1	105	1	3	niveau 3	chpl5 = 16	1	152	33	0500F260 LPH TGO TB6 NTM TR10 NOR	
4	0	3	2	3	119	4	3		50	2	185	4	EDSL	
4	0	3	4	4	119	3	3		50	3	189	4	EDSF	
4	0	3	5	5	120	1	3		300	1	193	9	EDSL EDSF	
4	0	4	1	1	105	1	3	niveau 4	2	36	202	1	0	
4	0	4	2	3	119	5	3		60	1	203	1	0	
4	0	4	4	4	120	1	3		400	1	204	67	TESCAPX Z 0130 N EDNP 0500F260 LPH TGO	
0														
4	0	5	1	3	16	1	2	niveau 5 :	FPL				TB6 NTM TR10 NOR EDSL EDSF	
5	1	1	74	77	50	2	1							
5	2	1	79	82	50	3	1							
5	0	1	1	2	300	1	2							
6	1	1	88	88	60	1	1							
6	0	1	1	1	60	1	2							
0	0	1	1	6	400	1	4	analyse de toutes les lignes						

classe indice chaine

	opérande 1		n° de relation	opérande 2			
	classe	indice		pointeur sur TXTR	longueur	nombre de triplets de la série	chaine
TRIPLETS RESULTATS							
	101	1	1 (vitk)	1	4	1	0500
	102	1	2 (hauff)	5	3	1	260
	105	1	1	8	4	2	0500
	105	1	2	12	3	1	260
	107	1	6 (type)	15	1	1	M
	107	2	6	16	1	1	M
SGT 1	119	1	10 (peg)	17	3	4	LPH
	119	1	11 (reg)	20	1	3	0
	119	1	12 (ped)	21	3	2	TGG
	119	1	13 (red)	24	1	1	0
	119	2	10	28	1	16	0
	119	2	11	32	3	0	TB6
	119	2	12	29	3	1	NTM
	119	2	13	25	3	1	TB6
	119	2	4 (Keg)	35	3	3	NTM
	119	2	5 (Ked)	38	3	2	NTM
	119	2	6 (type)	41	1	1	I
	119	2	22	42	1	9	0
	119	2	23	43	1	8	0
	119	2	24	44	1	7	0
	119	2	25	45	1	6	0
	119	2	26	46	1	5	0
	119	2	27	47	1	4	0
	119	2	28	48	1	3	0
	119	2	29	49	1	2	0
	119	2	30	50	1	1	0
	119	3	10	55	1	13	0
	119	3	11	59	4	0	TR10
	119	3	12	56	3	1	NOR
	119	3	13	51	4	1	TR10
	119	3	22	63	1	9	0
	119	3	23	64	1	8	0
	119	3	24	65	1	7	0
	119	3	25	66	1	6	0
	119	3	26	67	1	5	0
	119	3	27	68	1	4	0
	119	3	28	69	1	3	0
	119	3	29	70	1	2	0
	119	3	30	71	1	1	0

.../...

Triplets résultats (suite)

119	4	10	72	3	11	LPH
119	4	11	75	1	1	0
119	4	12	76	3	1	NTM
119	4	13	79	3	1	TB6
119	4	4	82	3	1	NTM
119	4	5	85	3	1	NTM
119	4	6	88	1	1	I
119	4	10	89	3	4	TB6
119	4	11	92	3	3	TB6
119	4	12	95	3	2	TB6
119	4	13	98	3	1	TB6
119	5	10	101	3	7	LPH
119	5	11	104	1	1	0
119	5	12	105	3	1	NOR
119	5	13	108	4	1	TR10
119	5	4	112	3	3	NTM
119	5	5	115	0	2	
119	5	6	115	1	1	I
16	1	10	116	3	4	LPH
16	1	11	119	1	1	0
16	1	12	120	3	1	NOR
16	1	13	123	4	1	TR10

chp15

Message G :

(FPL-PIROLAB-Z
-B747-N
-EGLL
-0420F330 UG1 NTM UB6 UA9
-LIMA LIFL
-0)

ETAT DE LA PILE

L	B	NIV	DEB	FIN	CLSSR		INDIC
1	1	1	5	11	10	1	1
1	2	1	13	13	20	1	1
1	0	1	1	2	100	1	2
2	1	1	17	20	30	1	1
2	2	1	22	22	40	1	1
2	0	1	1	2	200	1	2
3	1	1	26	29	50	1	1
3	0	1	1	1	50	1	2
4	1	1	33	36	101	1	3
4	1	1	37	40	102	1	3
4	1	2	1	2	105	1	1
4	2	1	42	44	107	1	1
4	3	1	46	48	106	1	1
4	4	1	50	52	107	2	1
4	5	1	54	56	107	3	1
4	6	1	70	70	120	1	1
4	0	1	1	1	105	1	3
4	0	1	2	2	113	1	3
4	0	1	3	3	109	1	3
4	0	1	4	4	113	2	3
4	0	1	5	5	113	3	3
4	0	1	6	6	120	1	3
4	0	2	1	1	105	1	3
4	0	2	2	3	119	1	3
4	0	2	4	5	119	2	3
4	0	2	6	6	120	1	3
4	0	3	1	1	105	1	3
4	0	3	2	3	119	3	3
4	0	3	4	4	120	1	3
4	0	4	1	3	16	1	2
5	1	1	73	76	50	2	1
5	2	1	79	82	50	3	1
5	0	1	1	2	300	1	2
6	1	1	87	87	60	1	1
6	0	1	1	1	60	1	2
0	0	1	1	6	400	1	4

FPL

TABLE ASSOCIATIVE

CLS	NO	PTR	LG	CHNE
10	1	1	7	PIROLAB
1	2	8	1	Z
20	1	9	1	Z
100	1	10	9	PIROLAB Z
30	1	19	4	B747
1	6	23	1	N
40	1	24	1	N
200	1	25	6	B747 N
50	1	31	4	EGLL
101	1	35	4	0420
102	1	39	4	F330
105	1	43	8	0420F330
107	1	51	3	UG1
106	1	54	3	NTM
107	2	57	3	UB6
107	3	60	3	UA9
4	17	63	1	>
120	1	64	1	>
113	1	65	3	UG1
109	1	68	3	NTM
113	2	71	3	UB6
113	3	74	3	UA9
119	1	77	7	UG1 NTM
119	2	84	11	UB6 KRH UA9
119	3	95	19	UG1 NTM UB6 KRH UA9
16	1	114	28	0420F330 UG1 NTM UB6 KRH
50	2	142	4	LIMA
50	3	146	4	LIFL UA9
300	1	150	9	LIMA LIFL
2	30	159	1	0
60	1	160	1	0
400	1	161	62	PIROLAB Z B747 N EGLL

0420F330 UG1 NTM UB6 KRH UA9 LIMA LIFL 0

KRH rajouté par l'analyse

Message A :

(FPL-DAYTONA-Z
-B707-N
-EDDL
-0445F330 DM2 ALB UR9 ZUL UG1 NTM UG1N BTI UG1N BUB UA4N
-EDDF EDDL
-0)

perturbations liées à la saturation de la table

ETAT DE LA PILE

L	B	NIV	DEB	FIN	CLSSR		INDIC
1	1	1	5	11	10	1	1
1	2	1	13	13	20	1	1
1	0	1	1	2	100	1	2
2	1	1	17	20	30	1	1
2	2	1	22	22	40	1	1
2	0	1	1	2	200	1	2
3	1	1	26	29	50	1	1
3	0	1	1	1	50	1	2
4	1	1	33	36	101	1	3
4	1	1	37	40	102	1	3
4	1	2	1	2	105	1	1
4	2	1	42	43	121	1	3
4	2	1	44	44	117	1	3
4	2	2	1	2	118	1	1
4	3	1	46	48	106	1	1
4	4	1	50	52	107	1	1
4	5	1	54	56	106	2	1
4	6	1	58	60	107	2	1
4	7	1	62	64	106	3	1
4	8	1	66	69	107	3	1
4	9	1	71	73	106	4	1
4	10	1	75	78	107	4	1
4	11	1	80	82	106	5	1
4	12	1	84	87	107	5	1
4	13	1	89	89	120	1	1
4	0	1	1	1	105	1	3

TABLE ASSOCIATIVE

CLS	NO	PTR	LG	CHNE
10	1	1	7	2 CA
1	2	8	1	D
20	1	9	1	A
100	1	10	9	% AA; I B
30	1	19	4	5AS
1	6	23	1	
40	1	24	1	A
200	1	25	6	@ AAT
50	1	31	4	AA&
101	1	35	4	AAX
102	1	39	4	130
105	1	43	8	0445F330
121	1	51	2	DM
2	14	53	1	2
117	1	54	1	2
118	1	55	3	DM2
106	1	58	3	ALB
107	1	61	3	UR9
106	2	64	3	ZUL
107	2	67	3	UG1
106	3	70	3	NTM
107	3	73	4	UG1N
106	4	77	3	BTI
107	4	80	4	UG1N
106	5	84	3	BUB
107	5	87	4	UA4N

.../...

4	0	1	2	2	118	1	3	niveau 1
4	0	1	3	3	110	1	3	
4	0	1	4	4	113	1	3	
4	0	1	5	5	109	1	3	
4	0	1	6	6	113	2	3	
4	0	1	7	7	109	2	3	
4	0	1	8	8	113	3	3	
4	0	1	9	9	108	1	3	
4	0	1	10	10	113	4	3	
4	0	1	11	11	108	2	3	
4	0	1	12	12	114	1	3	
4	0	1	13	13	120	1	3	
4	0	2	1	1	105	1	3	niveau 2
4	0	2	2	2	118	1	3	
4	0	2	3	4	119	1	3	
4	0	2	5	6	119	2	3	
4	0	2	7	8	119	3	3	
4	0	2	9	10	119	4	3	
4	0	2	11	12	119	5	3	
4	0	2	13	13	120	1	3	
4	0	3	1	1	105	1	3	niveau 3
4	0	3	2	3	119	6	3	
4	0	3	4	5	119	7	3	
4	0	3	6	6	119	4	3	
4	0	3	7	7	119	5	3	
4	0	3	8	8	120	1	3	
4	0	4	1	1	105	1	3	niveau 4
4	0	4	2	3	119	8	3	
4	0	4	4	4	119	4	3	
4	0	4	5	5	119	5	3	
4	0	4	6	6	120	1	3	
4	0	5	1	1	105	1	3	niveau 5
4	0	5	2	3	119	9	3	
4	0	5	4	4	119	5	3	
4	0	5	5	5	120	1	3	
4	0	6	1	1	105	1	3	niveau 6
4	0	6	2	3	119	10	3	
4	0	6	4	4	120	1	3	
4	0	7	1	3	16	D	2	niveau 7 : chp15
5	1	1	92	95	50	2	1	
5	2	1	97	100	50	3	1	
5	0	1	1	2	300	1	2	
6	1	1	103	103	60	1	1	
6	0	1	1	1	60	1	2	
0	0	1	1	6	400	1	4	analyse de toutes les lignes : FPL

```
  4 27  91   1 >
120  1  92   1 >
110  1  93   3 ALB
113  1  96   3 UR9
109  1  99   3 ZUL
113  2 102   3 UG1
109  2 105   3 NTM
113  3 108   4 UG1N
108  1 112   3 BTI
113  4 115   4 UG1N
108  2 119   3 BUB
114  1 122   4 UA4N
119  1 126   7 ALB UR9
119  2 133   7 ZUL UG1
119  3 140   8 NTM UG1N
119  4 148   8 BTI UG1N
119  5 156   8 BUB UA4N
119  6 164  11 DM2 ALB UR9
119  7 175  16 ZUL UG1 NTM UG1N
119  8 191  28 DM2 ALB UR9 ZUL UG1 NTM UG1N
119  9 219  38 DM2 ALB UR9 ZUL UG1 NTM UG1N  BTI UG1N
119 10 257  38 DM2 ALB UR9 ZUL UG1 NTM UG1N  BTI UG1N
 16  1 295  47 0445F330 DM2 ALB JR9 ZUL UG1 NTM UG1N  BTI UG1N
 50  2 342   4 EDDF
```

--- association non conservée car table saturée.

A5.2/ Autres exemples d'apprentissage

Exemple 1 (page 81)

Listing commenté :

```
TYPE : 0
PRODUCTION : 28 TEST : 0
STRUCTURE : < 113    4 >
ELT : /
TEST VERROU : NON
TEST='(*2 , = , / )';
FI(1)=''
TCAR='';
RES(1)='R = ||*1 '
```

0 = création
1 = modification

: classes composant la structure

tests complémentaires (contexte)

: tests du cardinal

```
FI(2)=''
RES(2)='R ∈ 00000113'

FI(3)=''
RES(3)='KALL(*1)'
```

fabrication du résultat

classe du résultat

triplet résultat ...

→ voir le greffon fabriqué page 82.

Exemple 2 (page 84)

Listing commenté :

```
TYPE : 1
PRODUCTION : 24 TEST : 18
STRUCTURE : < 119  112 >
ELT : TB7
TEST VERROU : OUI
TEST='( *1 , REL13 , X )';
FI(1)='( X , SUBSTØ7 , L1 )'
FI(3)='( L2 , ØAB PRO<CLSS(L2),0000010'
TCAR='(L3 , CARDSUP , 1 )';
RES(1)='( L3 , DEPILE , ATOM )'
RES(3)='( X  ,SUBSTØ7NIV Y , ATOM )'

FI(2)='( L1 , ØCLASS<00000113> , L2 )'
FI(4)=''
FI(5)='';
RES(2)='( X  , RETNIV , Y )'
RES(4)='( * , RETOUR NIV , Y )'
```

relation n° 13 = ped. ou ped.

= ams

équivaut à (x, ret, Y) du texte)

test du cardinal de l_3

équivaut à : (l_3, deb, elt) du texte.

équivaut à (X, substØ7, elt) du texte

équivaut à (Y, rep, V)

→ voir la production annexée page 85.

Exemple 4 :

FPL source : KRH UA9 NTM UB5

FPL corrigé : KRH UA9 UB5

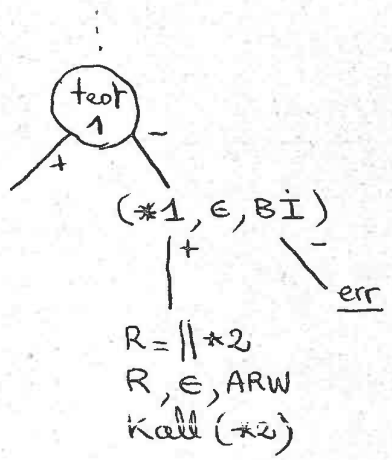
" " après analyse : KRH UA9 EDE UB5

Greffe proposée par l'apprentissage :

```
TYPE : 1          ← modification
PRODUCTION : 13 TEST : 1
STRUCTURE : < 112  115 >
ELT : NTM
TEST VERROU : NON
TEST='(*1 , E , 109 )'   FI(1)=''
FI(5)=''                 TCAR=''
RES(4)=''                RES(5)='';
```

```
FI(2)=''                 FI(3)=''                 FI(4)=''
RES(1)='R = ||*2 '       RES(2)='R E 115'         RES(3)='KALL(*2) '
```

C'est à dire : modifier PR013 : (RPNT, ARW)



Exemple 5 :

FPL source : 0420F330 UG1 X NTM UB6 UA9

FPL corrigé : 0420F330 UG1 NTM UB6 UA9

" " après analyse : 0420F330 UG1 NTM UB6 KRH UA9

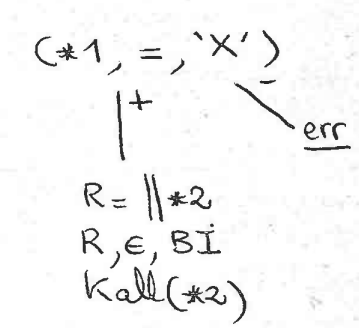
Greffe prpposée :

```
TYPE : 0
PRODUCTION : 28 TEST : 0
STRUCTURE : <  1  109 >
ELT : X
TEST VERROU : NON
TEST='(*1 , = , X )'        FI(1)=''
FI(5)=''                    TCAR=''
RES(4)=''                   RES(5)='';

FI(2)=''                    FI(3)=''
RES(1)='R = ||*2 '          RES(2)='R E 109'

FI(4)=''
RES(3)='KALL(*2) '
```

C'est à dire : créer PR028 :



L E X I Q U E

AFTN	Aeronautical Fixed Telecommunication Network (cf RSFTA)
ATC	Control du trafic aérien (Air Traffic Control)
ATS	Messages "Air Traffic System"
ASC	Module de SCAPA correspondant à l'analyse - correction
Back-tracking	Stratégie de reprises d'une analyse
Callsign	Indicatif d'appel d'un avion
Champ	Portion de plan de vol de contenu sémantique homogène (cfI.1)
Classe	Nom de terminal ou d'auxiliaire de la grammaire servant à l'analyse syntaxique des plans de vols (liste des classes :cf A2.1)
FPL	Plan de vol (Flight Plan) (description :cfI.1)
Greffon	Schéma de greffe complété et prêt à être utilisé dans les productions (cfIII.3.b)
OACI	Organisation de l'Aviation Civile Internationale
Opérateur d'apprentissage	Type d'opération superficielle élémentaire applicable à une chaine de caractères (définition :cfIII.3.a) ; liste des opérateurs :cfA2.1)
Panic-mode	En cas d'erreur d'une analyse, stratégie permettant de passer à la suite de l'analyse
Pdr	Partie droite d'une production
Plan de vol	Message destiné à véhiculer les principales informations statiques relatives à un vol
Pgc	Partie gauche d'une production
Point d'attaque	Endroit de divergences bien délimitées, entre une chaine résultant d'analyse d'un message source et la chaine résultat d'analyse du message corrigé correspondant
Production	Au sens général, règle d'une grammaire. Ici, règle syntaxico-sémantique, y compris les tests d'application de cette règle (liste des productions :cfA2.1)

.../...

RSFTA	Réseau du Service Fixe des Télécommunications aéronautiques
SCAPA	Système de Correction Automatique des Plans de Vol, avec Apprentissage
SGT	Abbréviation de la classe Segment
Schéma de greffe	Schéma traduisant la fonction d'un opérateur, à l'intérieur d'une production (cfIII.3.b)
Shift-reduce	Type d'analyseur procédant par actions de saut (shift) ou de réduction (reduce) et fabriquant une analyse ascendante, de gauche à droite
Strips	Bandes de papier où les heures estimées de survol des balises sont calculées
Structure (d'une production)	Classes des éléments constituant la partie droite de la règle équivalente
Tests de production (TP)	Arbre de Tests (Triplets) vérifiant l'applicabilité d'une production, du point de vue sémantique
Triplets Résultats (TR)	Triplets destinés à véhiculer les résultats sémantiques d'application des productions
UAC	Contrôle de l'espace aérien supérieur (Upper Airspace Control)
KARLDAP	Karlsruhe Display and Data Processing
SHANDAP	Shannon Display and Data processing

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

1) A.V. AHO & J.D. ULLMAN :

"The theory of parsing, translation and compiling"

Englewood Cliffs (N.J.) : Prentice Hall 1972

2) A.V. AHO & S.C. JOHNSON :

"LR parsing"
in Computing Surveys 6, 2, June 1972, p 448-458

3) A.V. AHO & T.G. PETERSON :

"A minimum distance error-correcting parser for Context free languages"
in Siam J. of Computer, vol 1, N° 4, dec. 1972, p 305-312

4) Y. AKNIN & C. DANAN & M.A. MATHIEU :
"Etude d'une méthode d'analyse syntaxique de détection d'erreurs sur quelques domaines d'application"
DEA informatique, Institut de Programmation, Juin 1975

5) J. COHEN & M.S. ROTH :
"Analysis of deterministic parsing algorithms"
in Comm. of the ACM, vol 21,6,1972, p 448-458

6) S.L. GRAHAM & S.P. RHODES :
"Practical syntactic error recovery"
in Comm. of the ACM, 18, nov 1975, p 639-650

7) T.V. GRIFFITHS & S.R. PETRICKS :
"On the relative efficiencies of context free grammars recognizers"
in Comm. of the ACM, vol 8, 5, 1965, p 289-300

8) C.L. HEDRICK :
"Learning production systems from examples"
in Artificial Intellignce, 7, 1976, p 21-49

9) E.T. IRONS
"An error correcting parse algorithm
in Comm. of the ACM, 6, 11, nov 63, p 669-673

10) J.P. LAURENT & A. ADAMS :
"Les connaissances nécessaires à la correction automatique des programmes"
Congrès AFCET "Reconnaissance des Formes & Intelligence Artificielle", Toulouse 79

11) R. LEINIUS :
"Error detection and recovery for syntax directed compiler systems"
Ph.D.Th., Computer Science Dpt, U. of Wisconsin, Madison, Wis., 1970

12) J.P. LEVY :
"Automatic correction of syntax errors in programming languages"
Ph. D. Th., Tech. Rpt. TR71-116, Computer Science Dpt.,
Cornell U., Ithaca, N.Y., 71

13) J. Mc. DERMOTT, A. NEWELL, J. MOORE :
"The efficiency of certain production systems implementations"
publication of the Carnegie Melon U., sept. 76, pp 30

14) M.D. MICKUNAS and J.A. MODRY :
"Automatic error recovery for LR parsers"
in Comm. of the A.C.M., vol. 21, N° 6, 1978, p 459-465

15) M. RYCHENER & C. FORGY & P.W. LANGLEY & J. McDERMOTT
& A. NEWELL & K. RAMAKRISHNA :
"Problems in building an instructable production system"
in 5th I.J.C.A.I., 77, vol. 1, p 337-338

16) R.A. WAGNER :
"Order n correcting for regular languages"
in Comm. of the A.C.M., Vol. 17, N° 5, Mai 74, p 265-268

17) D.S. WILLIAMS :
"Computer program organisation induced by problem examples"
Electrical Engineering Dpt., Carnegie-Melon Un., 1969

18) H. CZECH and P. TENA :
Rapport d'étude sur le système CAUTRA
(Traitement Plan de Vol)
Note CEE n° 9, vol. I & II

19) C. EYMANN :
"The KARLDAP A software implementation"
rapport EUROCONTROL n° 97 - E 103 - Octobre 1976

20) H. CZECH :
"SHANDAP 2A : Flight Data Processing System,
Detailed Functional Specifications, Vol. I and II".