PRÉSERVER
SAUVEGARDER
VALORISER

MANUEL DE RÉFÉRENCE : DE PROLOG, DE LA SOCIÉTÉ DELPHIA

FICHE N° 14813

Période de fabrication : 1975-1999
Fabricant : fabricant non renseigné
Domaines : Informatique et Communication
Sous-domaines : Ordinateurs
Organisme : ACONIT
Ville : Nantes (Loire Atlantique)
Modèle :
Matériaux : Carton, Métal, Papier, Plastique

Description

Titre traduit : Prolog manual

Ce Manuel de référence portant le titre synthétique Delphia Prolog existe sous la forme d'un classeur contenant 266 pages. Son titre complet est "Delphia Prolog, système de programmation logique, Manuel de référence", et il a été édité en 1986 par la société Delphia, une société qui avait son siège à Seyssinet, près de Grenoble. Sa structure permet d'aborder les termes et la syntaxe spécifique à ce langage et de prendre en main les fonctions principales de ce langage. Il est accompagné de nombreux schémas.

PROLOG est un langage de programmation basé sur la logique symbolique. Il manipule des « termes », qui sont des structures de données arborescentes. Il cherche à démontrer les propriétés liant certains de ces termes entre eux. Au sein de chaque section, les questions concernent des propriétés qui sont interprétées comme des buts à prouver et qui activent le programme, tendant à prouver l'opérabilité de ces buts.

Le mécanisme de programmation de base appelé « unification » agit sur des structures d'enregistrement générales (qui sont les « termes » de la logique). Dans la version fournie par Delphia, le langage Prolog est compilé, une innovation permettant de gagner un facteur 30 sur les versions classiques interprétées.

Son concept modulaire et son environnement de programmation, de type station de travail multi-fenêtres, a été étudié de façon à faciliter les mises au point, à l'aide de modules utilisés comme outils d'abstraction pouvant s'adapter à la nature dynamique de Prolog, avec des espaces de clauses indépendantes et orientées objets.

Le système Prolog de Delphia offre une « ouverture » qui permet de réutiliser des composants logiciels bâtis à l'aide d'autres langages, par exemple en « C ».

Les objets manipulés par Prolog sont nommés « termes ». Ils sont dits « atomiques » s'ils consistent en des « feuilles » représentés par un symbole (une lettre minuscule suivie d'une séquence ou suite de caractères alphabétiques), et sont dits « composés » si on peut les assimiler à des « arbres » permettant de décrire des données structurées (par exemple un arbre généalogique).

Un terme composé (ou structuré) est constitué d'un atome appelé « foncteur » (ou symbole fonctionnel), et d'une suite de termes appelés « arguments ». L'atome correspondant au foncteur s'écrit généralement suivi de ses arguments.

Les « listes » sont des structures de données ayant un rôle important dans le langage Prolog. Le premier argument de la liste a un rôle équivalent à un « foncteur ». La liste vide est considérée comme un atome.

Une variable peut être soit « libre », si elle n'a pas encore pris la valeur d'un autre terme ; soit « liée » à une autre variable dont elle prend alors la valeur ; soit « instanciée » à un terme dont elle prend la valeur, ce qui implique que les traitements sur l'un s'appliqueront aussi sur l'autre terme.

Les « opérateurs » peuvent être « préfixés » (l'opérateur précède son argument), « postfixés » (l'opérateur suit son argument), ou « infixés » (l'opérateur est entre ses deux arguments). Une « associativité » est définie pour chaque opérateur, pour lever toute ambiguïté dans le but de rendre opérationnels plusieurs opérateurs.

« Les prédicats » permettent de définir le format qui est requis pour un terme et de vérifier si son affectation est bien exacte. Prolog manipule les termes (données structurées arborescentes) : « l'unification » ou « la substitution » consiste à rendre identiques deux « arbres » contigus.

« La résolution » d'un problème consiste en une « question » pour laquelle Prolog doit trouver un « but » qui sera « vrai » ou « faux ». S'il est « vrai » et qu'il y a arborescence, un « point de choix » permet de poser une autre question.

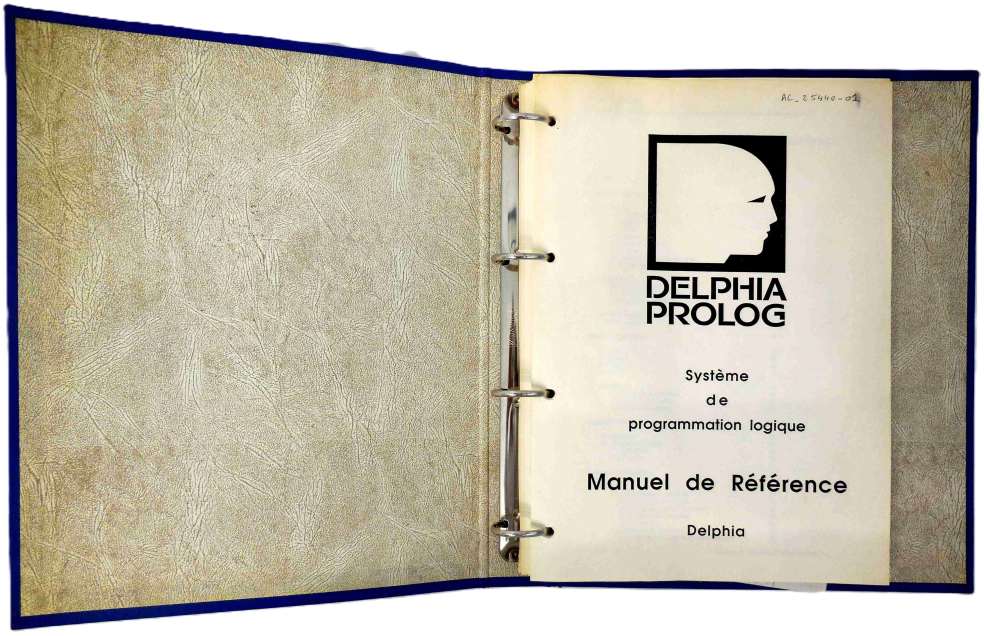
Ce classeur est à considérer en lien avec un second classeur ayant été édité par la même société, dont le titre est "Mylog, Générateur de systèmes experts, Manuel de référence"

(voir AC_25440-02).

Utilisation

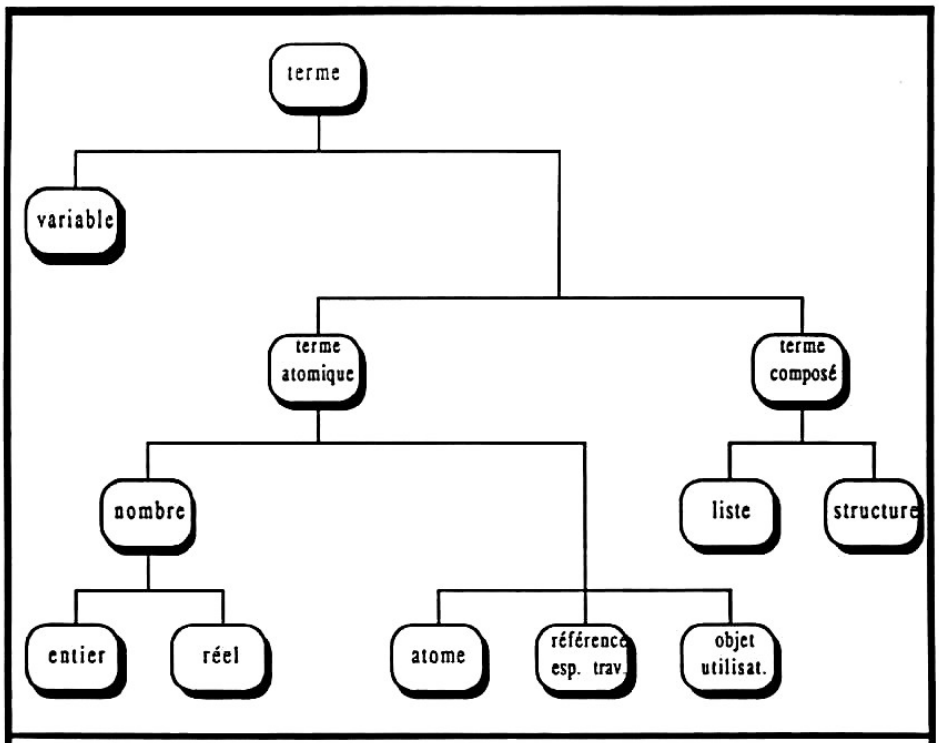
Autour des années 1985, le volume croissant d'applications dédiées à l'Intelligence Artificielle demande au langage Prolog de bien prendre en compte l'environnement de programmation, les qualités de modularité et d'ouverture.

Le langage Prolog de la société Delphia a été utilisé par cette société en lien avec son logiciel « expert » Mylog. Grâce à sa qualité « d'ouverture », les utilisateurs de Mylog (voir le Manuel Mylog AC_25440-02, page.1.14) peuvent l'utiliser pour insérer des modules écrits en divers autres langages, comme le langage "C". L'accès aux systèmes de gestion de bases de données relationnelles devient alors possible.



1.1 Termes

Tous les objets manipulés par un programme PROLOG sont des termes. Les termes dits **atomiques** sont des feuilles, et sont différents des **termes composés**, que l'on peut assimiler à des arbres.
Les **variables** permettent de désigner n'importe quel terme. Le schéma suivant montre la décomposition des termes employés dans **Prolog de Delphia.**



Eléments du langage

Figure 1

1.1.5 Termes composés

Les termes composés (ou structures) permettent de décrire les données structurées. Ils sont constitués d'un atome appelé **foncteur** (ou symbole fonctionnel), et d'une suite de termes appelés **arguments**. Le nombre d'arguments est également appelé **arité**. Par extension, un atome peut être considéré comme un terme composé d'arité zéro. L'atome correspondant au foncteur de la structure s'écrit éventuellement suivi de ses arguments entre parenthèses et séparés par des virgules.

Exemple : pere(alain, paul)

est un terme composé dont le **foncteur** est *pere*, et dont les **arguments** sont *alain* et *paul*. Puisque ce terme composé a deux arguments, *pere* est un foncteur d'**arité** 2.

Notation : pour faciliter leur identification, les termes composés sont notés sous la forme : foncteur/arité.

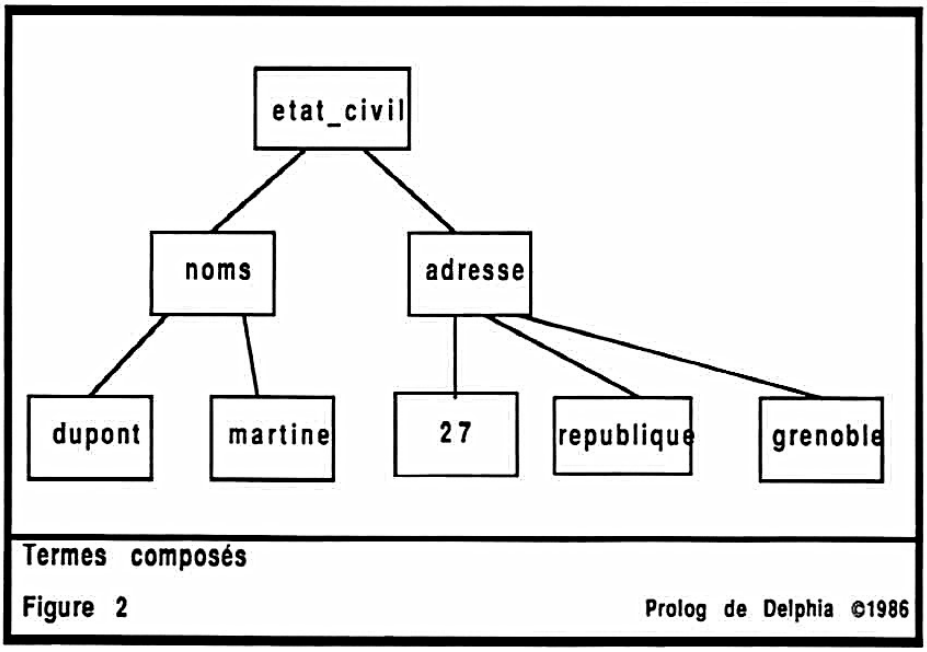
Par exemple, le terme composé *pere* ci-dessus, peut s'écrire : pere/2.

La définition d'une structure est récursive. Les arguments d'une structure peuvent consister en des termes PROLOG quelconques. Donc, une structure peut avoir pour arguments des termes atomiques, des variables, ou des termes qui sont eux-mêmes des structures.

Par exemple :

etat_civil(noms(dupont, martine), adresse(27, republique, grenoble))

Cette définition récursive conduit à représenter une structure sous la forme d'un arbre. La structure donnée en exemple peut être représentée sous la forme de l'arbre de la figure suivante :



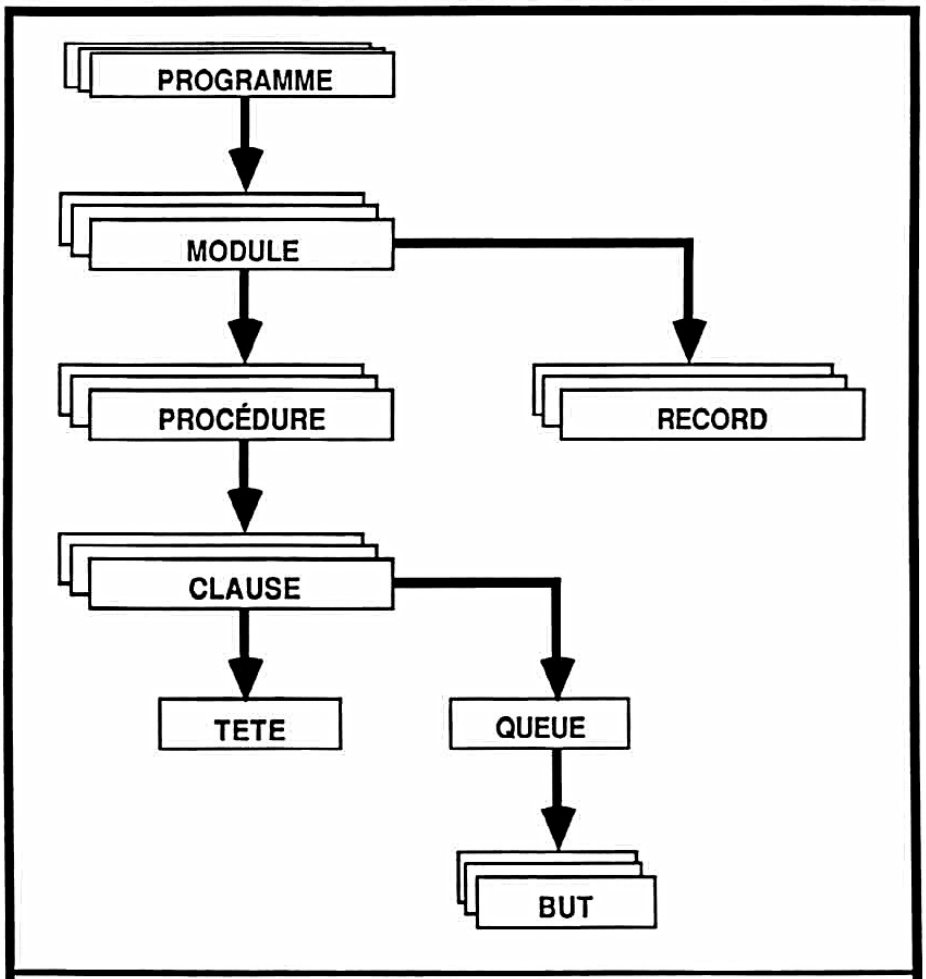
Termes composés

Figure 2 **Prolog de Delphia ©1986**

Chapitre2

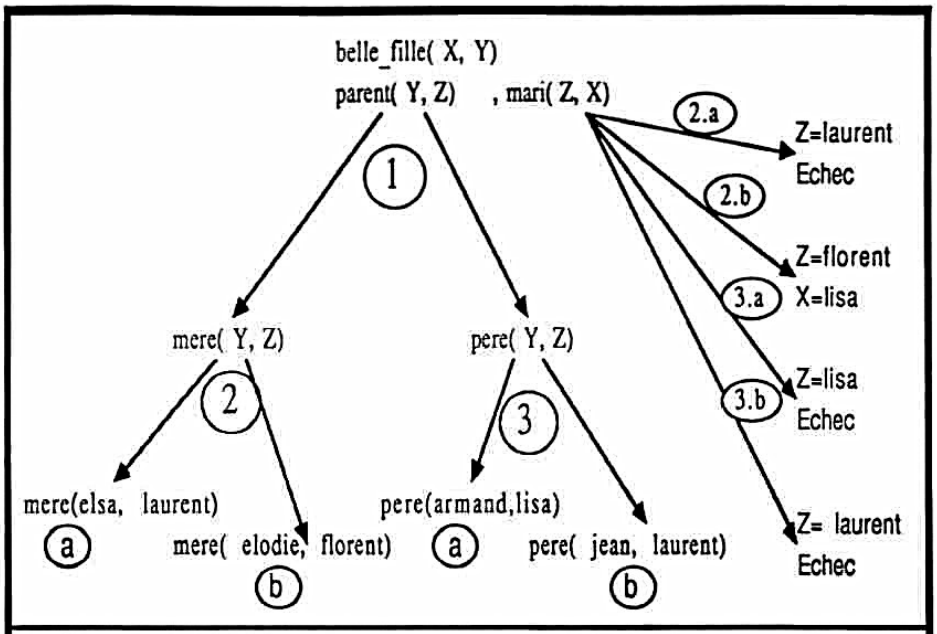
Programmes

La structure des éléments d'une application est résumée dans la figure suivante et détaillée ensuite, en allant des éléments les plus simples aux plus complexes.



Eléments d'une Application

Figure 3 Prolog de Delphia ©1986



Points de choix

Figure 5 **Prolog de Delphia ©1986**

Il y a création de points de choix chaque fois qu'une solution (partielle ou totale) est trouvée, alors qu'il existe d'autres possibilités. Les feuilles ***mere(elsa , laurent)***, ***mere(elodie , florent)***, ***pere(armand , lisa)*** de la figure 5 en sont des exemples, alors que ***pere(jean, laurent)*** n'en est pas un puisque c'est la dernière solution de ***parent(Y, Z)***.

Lors d'un échec, comme lorsqu'on ne trouve personne dont laurent soit le mari (2.a), il y a retour-arrière jusqu'au point de choix précédent, pour tenter de satisfaire le but ***pere***.
Toutefois même après un succès, il y a retour-arrière pour rechercher toutes les solutions du but ***pere***.

Aucun point de choix n'est créé dans la partie d'arbre de racine ***mari(Z, X)***, puisque lorsque la variable Z est instanciée, Z ne peut être mari de plusieurs X.