

XILOG STARTER LE DÉMARREUR PROLOG À BULL...

Jean-Paul ROY

Depuis trois ou quatre ans, on peut assister à une popularisation des idées issues du monde jusqu'alors fermé de l'Intelligence Artificielle. Les langages *symboliques* (*i.e.* adaptés à la programmation non-numérique, typiquement LISP et PROLOG) descendent des grosses unités centrales pour envahir le monde de la micro. C'était fait depuis longtemps pour la branche d'initiation à LISP que représente LOGO, mais plusieurs LISP complets (graphisme, flottants, tableaux, chaînes, compilateur), sont maintenant offerts sur compatibles PC, pas toujours à des prix très abordables (cf. cependant à la fin). Après LISP, c'est au tour de PROLOG, et même un éditeur "grand public" comme Borland y va de son couplet, en mettant sur la scène Turbo-Prolog. Mais nous aurons l'occasion de parler de ce dernier par la suite...

Pour l'instant, nous allons jeter un coup d'œil sur la seconde version du PROLOG sous MS-DOS (? 2.11) développé au sein du CEDIAG (Centre d'Intelligence Artificielle) de Bull ® par l'équipe de Dominique Sciamma. Dénommé XILOG (v2), il fait suite à une première version qui n'a pas eu de distribution notable en-dehors des Universités. Cette version d'essai a fait l'objet de critiques et de bancs d'essais et, bien qu'elle innovait déjà sur de nombreux points (cf. plus bas), son utilisation par des programmeurs expérimentés en PROLOG ont contribué à faire de la version 2 un produit mûr (qui est l'un des premiers à fonctionner déjà sous OS/2!).

Lorsqu'on analyse un dialecte d'un langage de programmation, trois points sont à étudier de près :

- les performances ;
- l'ergonomie ;
- la portabilité.

En quelques mots si vous êtes pressé : bonnes performances, bonne ergonomie, portabilité excellente.

Trois versions sont disponibles : *Starter*, *Développement* et *Professionnel*. Il s'agit donc d'une gamme complète, ce qui n'est pas sans intérêt. Nous retiendrons pour le marché éducatif la version *Starter* à un prix de 380 frs HT, comparable à celui de Turbo-Prolog. Des exemples sur la disquette, et un livre d'accompagnement très bien fait (210 pages), avec des rappels sur la Logique, un tutorial sur PROLOG avec exercices, et un manuel de programmation. Une documentation de très bonne qualité, qui fait oublier le manuel un peu sec de la version 1...

En PROLOG, les performances se mesurent en LIPS (*Logical Inferences Per Second*). Cette unité mesure grosso modo le nombre d'unifications (réussies) par seconde. Des tests comparatifs avec les autres PROLOG (dont Turbo-Prolog) sont disponibles chez le distributeur (cf. plus bas) pour les versions compilées (Xilog Développement). Si je n'ai pas reproduit ces tests, c'est qu'ils s'adressent à un marché professionnel, la version interprétée (*Starter*) conservant une rapidité nettement suffisante pour les applications pédagogiques. L'espace des clauses est limité en *Starter* à 128k, ce qui permet d'écrire plusieurs centaines de clauses, celles-ci étant stockées dans une mémoire dynamique, sous forme de listes. En particulier, il est possible de dépasser la limite des 300 clauses de Turbo-Prolog, qui gère une table fixe de points d'entrée par souci d'efficacité. De plus, il est possible, au moment de l'appel de Xilog, de définir soi-même la taille de l'espace de travail ou des piles.

La version *Starter* correspond à un sur-ensemble de Xilog v.1, amélioré sur certains points. Interprétée (le compilateur arrive avec la version Développement), les primitives sont pensées dans le but d'être le plus près possible de la norme internationale en cours d'élaboration (cf. plus bas), dans un souci de portabilité. Elles comprennent le noyau défini dans l'ouvrage fondamental de Clocksin & Mellish (i), dit "dialecte d'Édimbourg", que Turbo-Prolog prétend revendiquer aussi. Mais Xilog *Starter* ajoute deux améliorations importantes :

- la *gestion de l'environnement* (création de fenêtres, gestion de la souris, code ascii du caractère sous le curseur, musique, couleur, attributs vidéo, etc.). Turbo-Prolog sait aussi faire tout cela. Il est clair qu'en 1987, un langage de programmation digne de ce nom se doit d'offrir tout ceci en standard.
- des nouveautés sur le plan du langage. Trois exemples
 - le calcul fonctionnel, disposant d'un accumulateur, qui permet par exemple d'écrire :
 - ?- p(X) :- 2*X**2 + 3*X - 1.

- ?- Vis p(1) ?
- V = 4
- YES
- des tableaux, conservant l'aspect non-déterministe de Prolog !
Pour vous convaincre de leur puissance, je vous laisse méditer l'exemple suivant qui joue avec une matrice 2x2 nommée "tab" :
 - array (tab, [2, 2]) initialise tab avec des 0
 - t ab (I, I) <- : p (I) signifie tab(1,1) := 4 et tab(2,2) := 13
 - V is tab(I,2) -> V=0 & I=1 ->
 - V=13 & I=2
- le calcul avec registres, pour faire des affectations et obtenir encore plus d'efficacité, au détriment de l'aspect déclaratif.
- des types optionnels, permettant de renforcer les contraintes au moment de l'unification. Par exemple, si X et Y sont deux variables, l'unification X :int = Y :real échoue même si X et Y ne sont pas encore instanciées.

Turbo-Prolog n'innove pas sur le plan du langage. Il est même en retrait par rapport au standard d'Edimbourg. Le programmeur, pour gagner en rapidité d'exécution, est constamment en train d'aider le compilateur par des déclarations de types étrangères à cette famille de langages, sans avoir la satisfaction pour autant de programmer "par objets". Pour la même raison, il doit se plier à définir une arité fixe pour chaque prédicat, à n'utiliser que des éléments de même type dans une liste, et perdre encore un peu plus de souplesse. Ce qui a fait dire à certains que Turbo-Prolog est plus proche de PASCAL que de PROLOG... C'est probablement faux, mais il reste que Turbo-Prolog est plutôt adapté à un style "déterministe", qui ne gênera sans doute pas le programmeur PASCAL débutant en PROLOG, qui voudra sans doute programmer ... comme en PASCAL. Une utilisation didactique doit, me semble-t-il, jouer la carte de la pureté (je n'utilise personnellement pas le calcul fonctionnel ni les registres ni les types avec les stagiaires du CFAPI, dans le cadre bien sûr d'un atelier limité à 50 heures). Par contre, la réalisation d'une application du type base de données déductive, "fermée", rapide, se fera sans problème avec Turbo-Prolog (ou la version développement de Xilog).

Sur l'environnement de programmation, deux outils de base : un éditeur de clauses pleine page et un pisteur (trace, spy) pour la mise au point. L'éditeur ne permet d'éditer qu'une seule clause à la fois, les

LE BULLETIN DE L'EPI XILOG STARTER

touches "Page Up" et "Page Down" permettant de se promener dans la base de clauses. Il est à noter que Turbo-Prolog dispose, lui, d'un éditeur pleine page sur la base, ce qui est un bon point. Ceci devrait être corrigé dans les versions ultérieures de Xilog, mais n'est pas vraiment gênant. Les outils de trace sont standard (modèle des boîtes).

Enfin, la portabilité. La syntaxe la plus utilisée dans le monde (= 90%) est celle dite *d'Edimbourg*, définie dans ¹. Le Groupe d'Intelligence Artificielle (GIA) de l'université de Marseille-Luminy a défini, quant à lui, un Prolog II basé sur une syntaxe différente, très agréable, mais peu utilisée en-dehors de ². Malgré l'origine marseillaise de PROLOG, le standard en cours de définition sera probablement très proche de l'actuel dialecte d'Edimbourg. La France, en PROLOG comme en LISP, est très active dans les commissions de normalisation (AFNOR, ISO), par le biais d'instituts nationaux comme l'INRIA. Les marseillais auront néanmoins à leur actif le très attendu Prolog III, en partie dans le cadre du programme Esprit, et les adeptes de Turbo-Prolog auront une base livresque sûrement très large (mais de qualité très inégale). Il existe sur le marché américain de très beaux ouvrages sur PROLOG (Édimbourg), mais les traductions se font attendre, les universités et centres de recherche étant jusqu'à présent les seuls demandeurs. E reste à espérer que notre Ministère, après l'étape d'expérimentation logicielle actuelle, validera un produit français compétitif au plan international.

Jean-Paul ROY
CFAPI, Paris

BIBLIOGRAPHIE :

Prix (IBM-PC, MS-DOS >_ 2.11) :

XILOG 2 Starter	380 frs HT
XILOG 2 Développement	2277 frs HT
XILOG 2 Éducation	4975 fis HT (3 Starter + 2 Dév. + 3 doc.)
Le Lisp Starter	380 frs HT

Distribution : ACT-INFORMATIQUE
(demander la " Convention-Éducation") 12, rue de la Montagne Sainte-Geneviève
75005 PARIS [tél. (1) 46.33.72.60]

¹ "Programmer en Prolog", par W.F.Clocksins & C.S.Mellish (Eyrolles) Si l'anglais ne vous gêne pas, nous vous en conseillons cependant la troisième édition révisée et augmentée : "Programming in Prolog", (Springer-Verlag, 1987)

² "Prolog", par F.Giannesini & al. (InterEditions)