12 OUTILS VISUELS POUR LA MISE AU POINT DE PROGRAMMES PROLOG IV

(version 1.0béta - Septembre 1998)

ON TROUVERA DANS CE CHAPITRE une description du débogueur visuel de Prolog IV et comment l'utiliser, à l'aide d'exemples.

12.1 CONTEXTE

Le système de développement de Prolog IV dispose d'un débogueur pour suivre les détails de l'exécution d'un programme utilisateur. Le débogueur est basé sur ce qu'il est convenu d'appeler le modèle des boîtes; partie intégrante du noyau de Prolog IV, il distribue les informations aux divers visualiseurs.

Avant cette version, la partie accessible du débogueur par le programmeur était constituée des deux visualiseurs d'exécution tournant de façon synchrone :

- le visualiseur du modèle des boîtes,
- le visualiseur de fichiers sources.

Cette nouvelle version de Prolog IV incorpore un nouveau visualiseur, travaillant en synchronisation avec les autres.

Il est à noter qu'il n'est pas indispensable d'activer tous les visualiseurs pour les faire fonctionner. Celui qui n'est pas nécessaire à un instant donné peut être désactivé, afin d'éviter que trop d'informations ne soient présentées au programmeur, ou que trop de temps soit perdu pour les présenter (une machine Prolog, même en mode “debug”, est bien plus rapide que les entrées/sorties qu'elle produit).

12.2 DEBOGUAGE VISUEL

L'outil principal pour visualiser graphiquement l'exécution d'un programme Prolog IV est le visualiseur d'arbre d'exécution.

Ses buts peuvent être résumés comme suit :

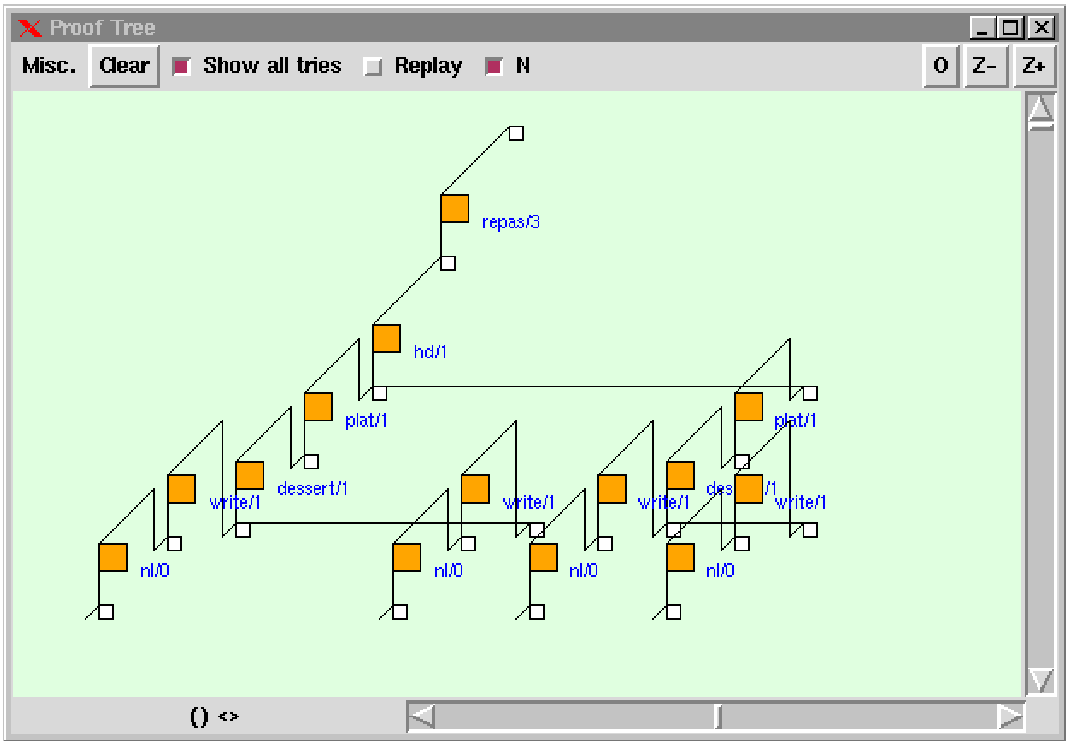
- donner une vue complète d'un arbre d'exécution,
- donner une vue d'une preuve,
- aller rapidement à un endroit donné de l'arbre d'exécution,

Le débogueur visuel de Prolog IV est partiellement financé par le projet ESPRIT « DiSCiPl » numéro 22532

- simplifier le redémarrage de l'exécution jusqu'à ce qu'un point donné de l'arbre soit atteint.

12.2.1 Une vue complète de l'arbre d'exécution

Voici une session de déboguage typique (quoique courte) montrant le visualiseur de l'arbre d'exécution.



Cette fenêtre principale affiche l'arbre d'exécution d'un programme.

Il s'agit d'un arbre ET/OU, avec une information visuelle sur le niveau de chaque noeud ET.

Cette arbre est destiné à être vu comme étant l'affichage d'un arbre à trois dimensions, en perspective. Nous serions situés “au-dessus” du plan principal de l'arbre.

Différentes sortes de boîtes apparaissent, avec les noms de prédicats associés. Ces boîtes sont reliées par des connexions.

12.2.2 Que représentent toutes ces petites boîtes ?

On supposera le lecteur familiarisé avec le déboguer Prolog IV ou tout au moins avec les bases du déboguer “modèle des boîtes”.

Une grande boîte de couleur est une boîte “appel” (il y en a une par appel à un paquet de règles). Actuellement, seules les boîtes-appels peuvent avoir un nom accolé (pour éviter trop de texte dans la fenêtre).

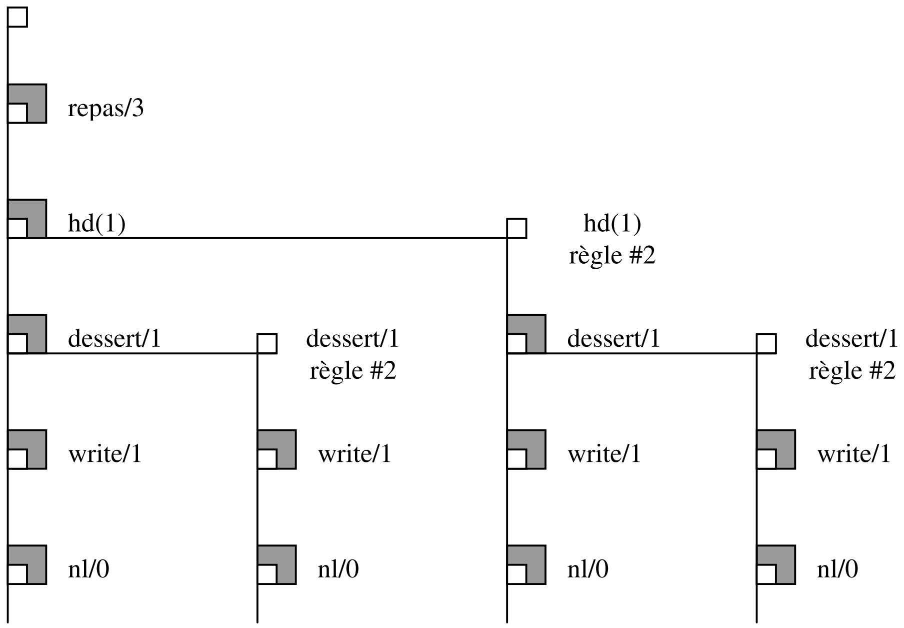
Une petite boîte blanche est une boîte “règle” et montre une tentative d'unification avec la tête d'une règle. Seuls les essais réussis sont affichés si le bouton “Show all tries” est coché.

12.2.3 Comment les interpréter ?

Voici une vue “d'en haut” (pour explication seulement) de l'arbre d'exécution précédemment montré. On suppose également que les boîtes sont mises à plat.

Les alternatives (noeuds OU) vont de la gauche vers la droite.

Les noeuds ET vont du haut vers le bas du dessin.



Pour fournir davantage d’explications sur cette vue, il nous faut introduire le petit programme prolog dont l'exécution est affiché.

```
to_begin :- repas(H,P,D), write([H,P,D]), nl.

repas(H,P,D) :-
   hd(H),
   plat(P),
   dessert(D).

hd(pate).
hd(salade).

plat(viande).

dessert(glace).
dessert(fruit).
```

Avec les explications et conventions données plus haut, il est facile de voir comment l'exécution du but “to_begin” donne l'arbre d'exécution donné.

Alors que nous savons que les appels s'effectuant avec succès s'affichent du haut vers le bas, il n'est pas possible, dans la visualisation plane de l'arbre, de voir que les appels à `write/1` et `nl/0` appartiennent au même corps de règle, et que ce n'est pas le cas pour `write/1` et `dessert/1` .

C'est le but de la vue non “plane” de l'arbre d'exécution que de donner ces indications sur le niveau de chaque appel.

12.3 FENÊTRES ET DESCRIPTION

12.3.1 Description des commandes graphiques

12.3.1.1 Menu “Divers.”

Il contient les items suivants :

- Charger un arbre d'exécution ...
- Montrer les noms
- Mode Replay
- Mode Replay-left
- Noms >

12.3.1.1.1 Item “ Charger un arbre d'exécution...”

Cet item propose au programmeur de choisir un fichier contenant la description d'un arbre d'exécution préalablement enregistré, et le charge dans la fenêtre.

12.3.1.1.2 Item “Montrer les noms”

Affiche les noms de prédicats à côté de chacune des boîtes-appels. Ce n'est guère utile que si le facteur zoom actuel est positionné à une valeur raisonnable. Autrement, les noms ne sont plus affichés; (puisque la taille de la police de caractère n'est pas mise à l'échelle lorsque le zoom est activé, ceci empêche que les noms n'en arrivent à être trop près les uns des autres. Réduire la taille de la police rendrait les noms illisibles et serait donc inutile).

12.3.1.1.3 Item “Mode Replay”

Ceci positionne le visualiseur dans un mode où :

- le contenu de la fenêtre n'est pas détruit lorsque la requête suivante est lancée,
- l'arbre peut être utilisé pour choisir un point d'arrêt temporaire et la branche qui le contient,

- le programmeur peut lancer un but qui va n'exécuter que la branche sélectionnée, et ce jusqu'à ce que le point d'arrêt soit atteint.

Note : aucun dessin ne s'effectue pendant cette exécution.

12.3.1.1.4 Item "Mode Replay-left"

Si cet item est positionné (et si on est en mode "replay",) l'exécution de la prochaine requête se fera à travers l'ensemble de la partie gauche de l'arbre, par rapport à la branche choisie.

Sinon, la seule branche choisie sera exécutée. Ceci peut être fait si le programme est en pur prolog, c.à.d. il n'y a pas de prédicats avec effets de bord (comme les E/S, assertions de règles ou affectations globales) dans la partie de l'arbre à gauche de la branche. Si ce n'est pas le cas, il vaut mieux laisser cette option non cochée si on a quelques doutes sur l'innocuité de ces effets de bord.

Il est important de savoir quelles portions de l'exécution sont en pur prolog, puisque exécuter une seule branche est normalement plus rapide que de réaliser la totalité de l'arbre jusqu'à cette branche comprise, l'exécution perdant son temps dans des impasses connues et donnant des solutions qu'on ne souhaite pas voir (puisque on débogue après).

12.3.1.1.5 Item "Noms"

Cet item de menu est un sous-menu contenant les noms de prédicats qui sont apparus pendant l'exécution. Le programmeur peut choisir une couleur à associer à chacun d'eux, avec pour effet visuel qu'une puce de cette couleur soit affichée autour de chaque boîte de ce nom, dans tout l'arbre d'exécution actuellement affiché.

12.3.1.2 Bouton "Effacer"

Pour effacer l'arbre de la fenêtre et tout oublier de l'exécution courante. Ceci est automatiquement appelé lorsqu'un nouveau but est lancé depuis la console prolog, à moins que la boîte "Mode Replay" ne soit cochée.

12.3.1.3 Boîte à cocher "Montrer tous les essais"

A pour but de positionner un booléen indiquant si l'on souhaite afficher chaque tentative d'unification avec une tête de règle ou pas.

Si le booléen est vrai, les unifications manquées de têtes pour un appel donné sont également affichées ; sinon, les succès seuls sont affichés.

Mettre le booléen à faux peut réduire de façon significative la taille de l'arbre d'exécution de la prochaine requête à lancer. (il n'y a pas d'influence sur ce qui est déjà affiché).

12.3.1.4 Boîte à cocher "Replay"

Raccourci pour l'item de menu "Mode Replay". (*voir plus haut*)

12.3.1.5 Boîte à cocher “Noms”

Raccourci pour l'item de menu “Montrer les noms ”. (*voir plus haut*)

12.3.1.6 Boutons “O”, “Z-” ,“Z+” et ascenseurs

Ces contrôles agissent sur l'affichage de ce qui est présenté dans la fenêtre.

Le bouton “O” (pour Origine) agit de sorte que l'origine de l'arbre d'exécution soit visible dans le coin en haut à gauche de la fenêtre.

Les ascenseurs peuvent être utilisés pour se déplacer dans la fenêtre afin de voir les parties cachées de l'arbre d'exécution.

Le bouton du milieu d'une souris à trois boutons fait de même.

Le bouton “Z+” agrandit (avec un facteur deux); il met à l'échelle tous les objets et distances dans la fenêtre. Les objets sont plus grands et les détails de l'arbre d'exécution sont mieux visibles; en contrepartie, une plus petite partie de cet arbre est visible à la fois.

Le bouton “Z-” réduit le grossissement (d'un facteur deux); il met à l'échelle tous les objets et distances de la fenêtre. Les objets graphiques sont plus petits, mais on peut voir une plus grande partie de l'arbre d'exécution.

Les zooms sont cumulatifs. On peut donc avoir de grands facteurs de zoom (en tant que puissances de deux).

12.3.2 La fenêtre de zoom

La fenêtre de zoom permet d'avoir simultanément une vue générale de l'arbre d'exécution et une vue détaillée d'un portion quelconque de celui-ci. En fait, la fenêtre de zoom n'utilise aucun facteur de zoom (ou utilise un facteur de un, par rapport à la valeur de départ) et donc les boîtes ont leurs tailles ordinaires, de sorte que la structure locale de l'arbre d'exécution reste lisible et compréhensible.

La fenêtre de zoom peut être appelée en cliquant le bouton gauche de la souris tout en pressant la touche CONTROL, quelque part dans l'arbre d'exécution que l'on souhaite agrandir. Ceci fait apparaître la fenêtre de zoom, ou change son contenu. Ce qui est dans la fenêtre de zoom est repéré dans la fenêtre principale par un carré jaune.

Cette fenêtre a son propre contrôle pour l'affichage (ou pas) des noms de prédicats.

12.4 TRAVAILLER AVEC LE DÉBOGUEUR

Tout d'abord, il est rappelé que les seuls programmes compilés en mode “debug” peuvent être pris en compte par le débogueur. (un programme qui n'est pas compilé avec cette option sera vu à l'appel comme un simple appel - une boîte noire).

Le programmeur est invité à consulter le chapitre *Environnement* de la documentation Prolog IV pour obtenir des informations sur le débogueur Prolog IV actuel et tout ce qui est relatif à son invocation.

Il est nécessaire d'utiliser Prolog IV au sein de son interface graphique de développement pour pourvoir utiliser le débogueur visuel.

12.4.1 Se connecter au débogueur visuel depuis le débogueur Prolog IV

Avant qu'une requête à déboguer ne soit lancée, il faut mettre Prolog IV en mode “debug”. On doit alors, en lançant la requête, obtenir l'invite du débogueur (dans la fenêtre “Console Prolog IV” ou celle du débogueur).

A cette invite, il faut taper le texte en gras suivant pour activer le visualiseur de l'arbre d'exécution.

`(DBG PIV)` **`graph 1`**

La fenêtre principale “Arbre d'exécution” doit apparaître alors.

12.4.2 Mettre un point d'arrêt temporaire

CTRL-SHIFT- bouton gauche de la souris sur une boîte dessinée met un point d'arrêt temporaire. Il affiche également (en vert) la branche issue de la racine de l'arbre d'exécution (la requête) jusqu'à cette boîte. On peut choisir aussi bien une boîte-appel qu'une boîte-règle.

12.4.3 Rejouer l'exécution

Après qu'un point d'arrêt temporaire ait été placé comme indiqué à l'étape précédente, choisir le mode “Replay” dans la fenêtre.

Cocher l'item (si c'est approprié - voir la description de cette commande) “Mode Replay-left” du menu “Divers”.

Si la requête est la même que celle qui est à l'origine du dessin de l'arbre d'exécution actuel, il suffit de la lancer.

Note : si l'exécution n'est pas la même, soit que la requête lancée n'est pas la même ou parce que le débogueur n'a pas été lancé depuis le début de la requête précédente, des affichages parasites apparaîtront dans la fenêtre.

Note : il ne faut rejouer l'exécution qu'une fois la précédente requête terminée (ou arrêtée).

12.5 DIVERS

12.5.1 Informations textuelles dans la fenêtre

Lorsque la souris est déplacée sur une boîte, quelques informations sur la boîte sont affichées en bas à gauche de la fenêtre.

Ces informations sont sous une forme textuelle condensée :

- l'indicateur de prédicat (de la forme nom/arité)
- le numéro de règle (seulement si on est sur une boîte-règle)
- le numéro de la boîte (entre parenthèses, comme (11))
- le niveau de l'appel (entre chevrons, comme <3>)

12.5.2 Quelques commandes textuelles du débogueur

Voici quelques nouvelles commandes du débogueur Prolog IV ayant un rapport avec le débogueur visuel. Ces commandes peuvent être tapées à l'invite du débogueur.

12.5.2.1 graph v

où ***v*** vaut 1 ou 0, installe ou enlève la connexion au débogueur visuel.

12.5.2.2 rule n

étant sur le port “RULE” d'une boîte, change le choix courant (le numéro de règle dans son paquet) au numéro ***n***, entier donné.

Note : pour des raisons internes, la dernière règle d'un paquet doit être noté ici avec un signe moins (-) et donc comme un entier négatif.

12.5.2.3 gfile nom

Ouvre ou crée un fichier nommé ***nom*** lequel contiendra le journal de l'exécution à venir, autorisant plus tard le chargement de l'arbre d'exécution par le débogueur visuel (voir l'item “Charger un arbre d'exécution ...”).