

Programmation logique

PROLOG

Prolog (PROGrammer en LOGique, 1970, Robert A. Kowalski (Edinburgh) & Alain Colmerauer (Marseille)) est né du besoin de pouvoir traiter la langue naturelle par ordinateur et, en particulier, la grammaire.



Télécharger et installer [SWI Prolog](#)

Faits

Créer un nouveau fichier (tests.pl) qui va constituer votre base de faits.

Les faits : « Jean aime Marie » ou « Anne aime Jean » sont traduits en Prolog par :

```
aime(jean, marie). % car Jean aime Marie
aime(paul, marie). % Paul est amoureux de Marie
aime(marie, paul). % et Marie aime Paul
```

Avec

1. le nom de la relation ou prédicat (débutant par une minuscule),
2. le/les arguments (ici « jean » et « marie ») séparés par une virgule, dans un ordre qui possède un sens (qui est le sujet et le complément de l'action décrite),
3. le fait se termine par un point « . ».

Les espaces ne jouent pas de rôle et le % indique le début du commentaire. (Le commentaire peut aussi être placé entre /* ... */). Le nom de la relation (ici « aime ») dépend du programmeur / analyste.

Questions

Les questions permettent d'interroger la base de fait. Elles se posent dans la fenêtre d'exécution de Prolog.

The screenshot shows the SWI-Prolog IDE interface. At the top, it says "SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)". Below that is a menu bar with File, Edit, Settings, Run, Debug, Help. A message box displays the welcome text: "Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4) SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software. Please run ?- license. for legal details." Below the message, there's a link to online help: "For online help and background, visit https://www.swi-prolog.org". A command line window shows the input "?- [mytests].", the response "true.", and a prompt "?- █".

Charger votre base de connaissances avec le nom du fichier entre crochets :

```
[tests].
```

En Prolog, les questions débutent par le point d'interrogation suivi du nom de la relation et du/des argument(s) (objet). On parle également de but pour désigner une question. Par exemple, on peut se poser la question de savoir si « Est-ce que Jean aime Marie ? ». Cette question se traduit par :

```
?- aime(jean, marie).
```

Pour y répondre, l'interprète Prolog va essayer d'unifier la question posée avec un des faits de sa base de connaissance. S'il réussit, il répond « Yes » et « No » dans le cas contraire.

Variables

Une variable commence par une Majuscule.

Les variables permettent d'introduire des inconnues, et sont utilisables dans les questions :

```
?- aime(jean, X).  
X = marie  
Yes
```

Il est possible d'obtenir les réponses suivantes, en utilisant le ; :

```
?- aime(jean, X).  
X = marie ;  
X = peche ;  
No
```

Dans un fait, une variable représente tout (quelque chose de quelconque) :

anne aime tout s'écrira :

```
aime(anne, _) % _ est une variable quelconque
```

Conjonctions

Soit la base de connaissances suivante :

```
aime(jean, peche).  
aime(jean, marie).  
aime(jean, paris).  
aime(marie, paul).  
aime(marie, paris).  
aime(paul, marie).  
aime(paul, biere).  
aime(anne, bretagne).  
roi(tintin, belgique).
```

Il est possible de vérifier la véracité simultanée de plusieurs faits avec un ET : , :

```
?- aime(jean, marie) , aime(marie, jean).  
No
```

« Existe-t-il quelque chose que Jean et Paul aiment ? ».

```
?- aime(jean, X), aime(paul, X)
```

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/cnam/utc503/declarative/prolog?rev=1701017059>

Last update: **2023/11/26 17:44**

