

## **EAO, PHILOSOPHIE ET MICRO-PROLOG**

**Françoise BERNIS**

L'idée de cette association est née d'une conversation avec une enseignante de philosophie de mon établissement. Pratiquant déjà L'EAO en STE avec Micro-Prolog, je souhaitais explorer d'autres domaines. Le thème du syllogisme m'a semblé bien se prêter à un tel travail.

J'ai proposé alors une première maquette de TP à ma collègue philosophe qui m'a suggéré quelques modifications. Ce qui fut dit fut fait. Et la forme définitive du TP a vu Le jour, avec pour but de faire découvrir aux élèves la notion de syllogisme, les trois catégories de syllogismes décrites par Aristote et la notion de sophisme.

Chaque élève a reçu un exemplaire du TP, le guidant pas à pas et lui permettant de travailler sur Les exercices proposés avec une assez grande autonomie. Chaque groupe de deux élèves a travaillé sur un Personna 1600 où Le programme était déjà chargé en début de séance.

Documents joints - Le TP distribué - le Listing du programme - une suite d'exemples d'interrogations du programme.

La classe choisie a été une Terminale H. "H" pour économiser Le temps d'apprentissage du clavier et de découverte d'un travail informatique. Par contre, ce qu'il n'est évidemment pas possible ni souhaitable de supprimer, c'est le temps de préparation du TP (La balle est dans le camp du prof de philo) et le temps de son exploitation ultérieure (...).

Françoise BERNIS  
Lycée Guy Mollet  
ARRAS

## LES SYLLOGISMES

### I - PRÉSENTATION DU TRAVAIL

But : appréhender et manipuler la notion de syllogisme à l'aide d'exercices.

Support utilisé : le langage MICRO-PROLOG.

Un programme, dans le langage MICRO-PROLOG, est un ensemble de clauses (phrases) qui décrivent un univers. Utiliser Le programme, c'est "poser les bonnes questions".

Les 14 formes de syllogismes décrites par Aristote ont été traduites sous une forme acceptable par le langage. Pour ceci, 5 prédicats (prédicat=mot-clé) et 3 variables ont été créés.

Les expressions à traduire sont :

- 1 - Quel que soit l'élément concerné par X, la proposition X implique la proposition Y, ce qui s'écrira : I-QQS(X Y)
- Exemple : "Tous Les chats sont gris" se notera :
  - I-QQS(chat gris)
  - Soit : quel que soit l'élément e,
  - "e est un chat" implique "e est gris".
- 2 - Quel que soit l'élément concerné par X, la proposition X n'implique pas La proposition Y : NIP-QQS(X Y)
- 3 - Il existe des éléments pour Lesquels la proposition X implique La proposition Y : I-IE(X Y)
- 4 - Il existe des éléments pour lesquels la proposition X n'implique pas la proposition Y : NIP-IE(X Y)

Prédicats :

- I-QQS implique, quel que soit
- NIP-QQS : n'implique pas, quel que soit
- I-IE il existe des ... implique
- NIP-IE : il existe des ... n'implique pas
- conclusion

Variables : X, Y, Z

Les quatre prédicats I-QQS, NIP-QQS, I-IEX et NIP-IEX décrivent Les prémisses. Le cinquième exprime La conclusion relative à ces prémisses.

Pour utiliser Le programme, vous devez connaître un minimum d'ordres MICRO-PROLOG:

- Li tout : pour obtenir La liste de toutes les clauses du programme
- Li conclusion : pour obtenir la liste des clauses relatives au prédicat "conclusion"
- Li I-QQS : pour obtenir la Liste des clauses relatives au prédicat "I-QQS"
- Li NIP-QQS : pour Liste "NIP-QQS"
- .
- .
- .
- aj(...): pour ajouter une clause au programme, ce qui nous servira à introduire les prémisses de chaque cas à traiter
- conclusion : pour demander ce que le programme peut déduire des prémisses fournies
- ot : pour ôter une clause du programme, ceci pour "faire le ménage" après obtention d'une conclusion, c'est-à-dire supprimer les prémisses correspondant au cas étudié.

## II - EXEMPLE D'UTILISATION

Soient Les deux prémisses suivantes :

- Tous Les chats comprennent Le français
- Quelques poulets sont des chats

Comment faire chercher une conclusion par le programme ? Conclusion qui n'existe d'ailleurs pas nécessairement.

### A - Etape n°1

Ajouter au programme les deux prémisses en les codant sous une forme simplifiée et en respectant scrupuleusement la syntaxe du langage. Chacun des prédicats exprimant une prémisse est suivi de deux arguments entre parenthèses. Un espace au moins doit séparer les deux

arguments. Par contre, aucun espace n'est autorisé à l'intérieur d'un argument, nous conviendrons d'utiliser à sa place le tiret.

Donc, nos deux prémisses pourront par exemple s'ajouter au programme sous la forme

- aj(I-QQS(chat comprend-le-français))
- aj(I-IEX(poulet chat))

Remarques

- Respectez le code d'écriture des mots du programme (majuscules ou minuscules).
- Fermez autant de parenthèses que vous en ouvrez.
- Vous pouvez à tout moment interrompre un travail en appuyant simultanément sur les touches CRTC et C.

## B - Étape n°2

Demander, la conclusion en tapant : conclusion .

Vous verrez sur l'écran : I-IEX

poulet comprend-le-français

qui se lit : Il existe des poulets qui comprennent le français

## C - Étape n°3

Faire le ménage en enlevant du programme les deux prémisses qui s'y trouvent sous la forme :

I-QQS(chat comprend-le-français)

I-IEX(poulet chat)

Pour les enlever, taper :     ot I-QQS 2

                                  ot I-IEX 2

Pourquoi "2" ? Le programme contient au départ les clauses

- I-QQS()
- NIP-QQS()
- I-IEX()
- NIP-IEX()

Soit au minimum une clause pour chaque prédicat. Ces clauses de départ doivent rester dans le programme. Par contre, les clauses ajoutées et qui correspondent au cas à traiter doivent être ôtées.

Dans l'exemple précédent, le problème était donc d'enlever la deuxième clause de chaque prédicat.

Si dans la première étape, les deux prémisses introduites avaient concerné un seul prédicat, par exemple I-QQS :

- aj(I-QQS(bébé illogique))
- aj(I-QQS(illogique méprisé))

il aurait alors fallu ôter les clauses 2 et 3 de ce prédicat

- ot I-QQS 3
- ot I-QQS 2

vérification éventuelle : Li I-QQS

### III. EXPLOITATION DU PROGRAMME

#### A- Séparer les clauses en trois catégories

Critère de séparation : rôle joué par le moyen terme

Première figure -> (Moyen terme ... )  
( ... Moyen terme)

Deuxième figure -> ( ... Moyen terme)  
( ... Moyen terme)

Troisième figure -> (Moyen terme ... )  
(Moyen terme ... )

Pour lister les clauses intéressantes : Li conclusion

Pour interrompre le défilement : appui simultané sur ctrl et C

Noter les numéros des clauses entrant dans la première figure.  
deuxième figure.  
troisième figure.

On peut remarquer que dans la première figure, la prémisse de la forme (Moyen terme ...) est toujours universelle et que celle de la forme (... Moyen terme) est toujours affirmative.

Quelle remarque pouvez-vous faire sur chaque conclusion :

- de la deuxième figure ?
- de la troisième figure ?

## **B - Rechercher l'éventuelle conclusion de deux prémisses données**

Voir en annexe des exemples de couples de prémisses (pour la plupart extraits de "Logique sans peine" de Lewis Carroll).

Proposition de démarche :

- Interroger le programme en activant le prédicat conclusion
- Lister Le programme et rechercher la ou les clauses qui ont été utilisées

OU BIEN

- Commencer par chercher la conclusion, si elle existe
- Vérifier ensuite que le programme donne bien la même réponse
- Si ce n'est pas le cas, rechercher la cause de la divergence

Certaines prémisses, malgré leur apparence de début de syllogisme, n'entraînent aucune conclusion.

Noter, en suivant La syntaxe de MICRO-PROLOG, les couples de prémisses présents dans les exemples et qui ne débouchent sur rien.

Exemple : Certains légionnaires sentent bon le sable chaud

Certains légionnaires marchent au pas

se notera : I-IEX(légionnaire sent-bon-le-sable-chaud)

I-IEX(légionnaire marche-au-pas)

c'est-à-dire : I-IEX(Z X)

I-IEX(Z Y)

Si une conclusion fautive, selon les règles de la logique, devait quand même être écrite, on obtiendrait un sophisme, par exemple :

Certaines personnes qui sentent le sable chaud marchent au pas.

Imaginer des exemples de couples de prémisses qui ne correspondent à aucune forme de syllogisme.

**ANNEXE : EXEMPLES DE COUPLES DE PRÉMISSSES**

- 1 -      Aucun Français n'aime le pudding  
Tous les Anglais aiment le pudding
- 2 -      Aucun vieil avare n'a l'air joyeux  
Quelques vieux avares sont maigres
- 3 -      Tous les soldats savent marcher au pas  
Quelques bébés ne sont pas des soldats
- 4 -      Tous les gens intelligents sont populaires  
Tous les gens serviables sont populaires
- 5 -      Les poules n'ont pas de dents  
Les poules sont gourmandes
- 6 -      Quelques tableaux ne sont pas des coups d'essai  
Aucun coup d'essai n'est vraiment réussi
- 7 -      Ce qui est compréhensible ne m'intrigue jamais  
La logique m'intrigue
- 8 -      Toutes les guêpes sont agressives  
Tous les êtres agressifs sont déplaisants
- 9 -      Tous les canaris bien nourris chantent à plein gosier  
Aucun canari qui chante à plein gosier n'est mélancolique
- 10 -     Le charbon n'est jamais blanc  
Les nègres ne sont jamais blancs
- 11 -     Aucun pont n'est fait en sucre candi  
Quelques ponts sont pittoresques
- 12 -     Aucun quadrupède ne sait siffler  
Quelques chats sont des quadrupèdes
- 13 -     Tout ce qui est d'or est précieux  
Quelques disquettes sont précieuses
- 14 -     Quelques rêves sont épouvantables  
Aucun agneau n'est épouvantable
- 15 -     Aucun homard ne manque de bon sens  
Aucun être doué de bon sens n'espère L'impossible
- 16 -     Toute explication claire est convaincante

Quelques excuses ne sont pas convaincantes

17 - Aucun bébé n'est studieux

Aucun bébé n'est bon violoniste

18 - Personne n'est enthousiaste s'il se dirige vers la gare un jour de grève SNCF

Vous êtes enthousiaste

19 - Tous les lions sont féroces

Quelques lions ne boivent pas de café

20 - Aucun fossile ne peut être malheureux en amour

Une huître peut être malheureuse en amour

21 - Tous les Livres d'informatique sont parfumés à la fraise

Certains livres d'informatique ont été superbement recopiés par des moines du Moyen-Age

22 - Chaque fois que je sors, mon poisson rouge a la migraine

Il arrive qu'un air de musique m'incite à sortir

23 - Tous les canards de ce village appartiennent à Juliette

Tous les canards de ce village portent des cols en dentelle

conclusion (.) si

- I-QQS (X Y) et
- I-QQS (Z X)
- non (EQ (Z Y)) et
- al (I-QQS) et
- faux ()

conclusion (.) si

- NIP-QQS (X Y) et
- I-QQS (Z X) et
- non (EQ (Z Y)) et
- al (NIP-QQS) et
- al (Z Y) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et

- I-IEX (Z X) et
- non (EQ (Z Y)) et
- al (I-IEX) et
- al (Z Y) et
- faux ()

conclusion (.) si

- NIP-QQS (X Y) et
- I-IEX (Z X) et
- non (EQ (Z Y)) et
- al (NIP-IEX) et
- al (Z Y) et
- faux ()

conclusion (.) si

- NIP-QQS (X Y) et
- I-QQS (Z Y) et
- non (EQ (Z X)) et
- al (NIP-QQS) et
- al (Z X) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et
- NIP-QQS (Z Y) et
- non (EQ (Z X)) et
- al (NIP-QQS) et
- al (Z X) et
- faux ()

conclusion (.) si

- NIP-QQS (X Y) et
- I-IEX (Z Y) et
- non (EQ (Z X)) et
- al (Z X) et

- faux ()

conclusion (.) si

- I-QQS (X Y) et
- NIP-IEX (Z Y) et
- non (EQ (Z X)) et
- al (NIP-IEX) et
- al (Z X) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et
- I-QQS (X Z) et
- non (EQ (Z Y)) et
- al (I-IEX) et
- al (Z Y) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et
- I-QQS (X Z) et
- non (EQ (Y Z)) et
- al (I-IEX) et
- al (Y Z) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et
- NIP-QQS (X Z) et
- non (EQ (Y Z)) et
- al (NIP-IEX) et
- al (Y Z) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et

- I-IEX (X Z) et
- non (EQ (Y Z)) et
- al (I-IEX) et
- al (Y Z) et
- faux ()

conclusion (.) si

- I-IEX (X Y) et
- I-QQS (X Z) et
- non (EQ (Y Z)) et
- al (I-IEX) et
- al (Y Z) et
- faux ()

conclusion (.) si

- I-QQS (X Y) et
- NIP-IEX (Y Z) et
- non (EQ (Y Z)) et
- al (NIP-IEX) et
- al (Y Z) et
- faux ()

conclusion (.) si

- NIP-QQS (X Y) et
- I-IEX (X Z) et
- non (EQ (Z Y)) et
- al (NIP-IEX) et
- al (Z Y) et
- faux ()

I-QQS ()

NIP-QQS ()

I-IEX ()

NIP-IEX ()

&.

Li tout

Cet article est déjà paru dans "les cahiers de l'option informatique" de l'Académie de Lille (MAFPEN) n°3 avril 1987.