Programmation Logique

Applications d'IA en Prolog

Enseignant: NGUYEN Thi Minh Tuyen

Eliza

- Est un exemple classique des débuts de l'IA.
- N'utilise absolument pas d'IA.
- Simule une conversation avec un psychologue.

Eliza : un exemple de conversation

Hello. I am ELIZA. How can I help you?

> I am unhappy.

I am sorry to hear you are unhappy.

> But I am really sad.

Do you think coming here will help you not to be sad?

Can you help me?

You believe I can help you, do you not?

> Yes, I need your help.

You seem quite positive.

> You remind me of my sister.

Tell me more about your family.

> I have a big family.

You say you have a big family

> Yes, I have 4 brothers and 4 sisters.

You are sure?

> Certainly.

I see.

> ....

Eliza : principe du programme

- Reconnaître une configuration de mots
- Répondre en utilisant une configuration de réponse correspondante
- On définit donc des paires stimulus/réponse. Par exemple :
  - I am <statement>.
  - How long have you been <statement>?
- On cherche des mots-clés (mother, father, brother, sister, etc.) et on y associe une réponse.

Eliza : algorithme

- Lire une phrase
- Choisir une paire (stimulus, réponse)
- Apparier la phrase et le stimulus
- Écrire la réponse associée
- Recommencer

- A essayer: https://swish.swi-prolog.org/example/eliza.pl

Résolution d'un problème

- Pas de méthode de résolution adaptée
- Description formelle du problème
- Procédure pour tester une solution proposée
- On peut engendrer et énumérer les solutions potentielles

A partir d'un énoncé

- Deux étapes:
  - choix d'une représentation et spécification formelle de l'énoncé
  - résoudre l'énoncé formalisé et construire la solution.

Type d'énoncés

- Énoncés de type combinatoire
- Énoncés de type états/opérateurs de changement d'état
- Énoncés de type buts/opérateurs de décomposition.

Énoncés de type combinatoire

- Ils sont constitués
  - d'un ensemble de solutions potentielles (X)
  - d'un ensemble de contraintes (K)
- Il faut trouver une solution en respectant les contraintes.
- Exemples :
  - problème des huit reines,
  - une addition du type SEND+MORE=MONEY
  - une grille de sudoku.

Énoncés de type états/opérateurs de changement d'état

- Ils contiennent
  - un état initial,
  - des états finaux à atteindre ou leurs caractérisation et
  - des opérateurs <conditions, effets>.
- Il s'agit alors de découvrir un enchaînement d'opérateurs permettant de passer de l'état initial à un état final.
- Exemple:
  - Le jeu de taquin

Énoncés de type buts/opérateurs de décomposition

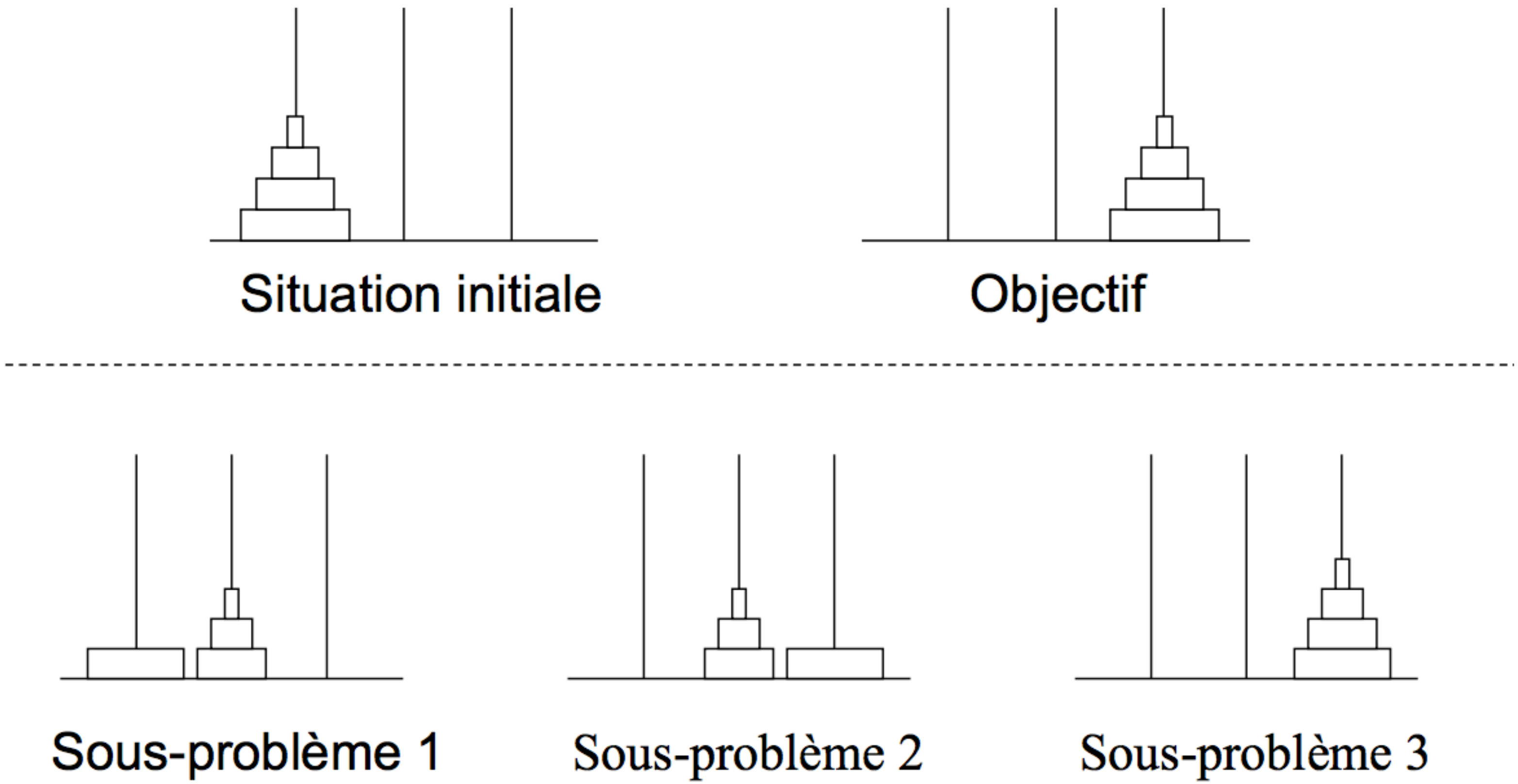
- Nous disposons
  - d'un problème à résoudre (un but),
  - une liste de problèmes que l'on sait résoudre (problèmes primitifs)
  - des opérateurs de décomposition de problèmes en sous-problèmes.
- Il faut trouver une décomposition du problème initial en problèmes primitifs.
- Exemple :
  - l'analyse grammaticale d'une phrase

Représenter le problème

Il faut définir :

- Les états du problème (en particulier l'état initial)
- L'objectif à atteindre (l'état final)
- Les opérateurs de transformation
  - Par une représentation par graphe d'états
  - Par une représentation par graphe de sous-problèmes

Tours de Hanoi : Représenter le problème



Tours de Hanoi : Représenter le problème

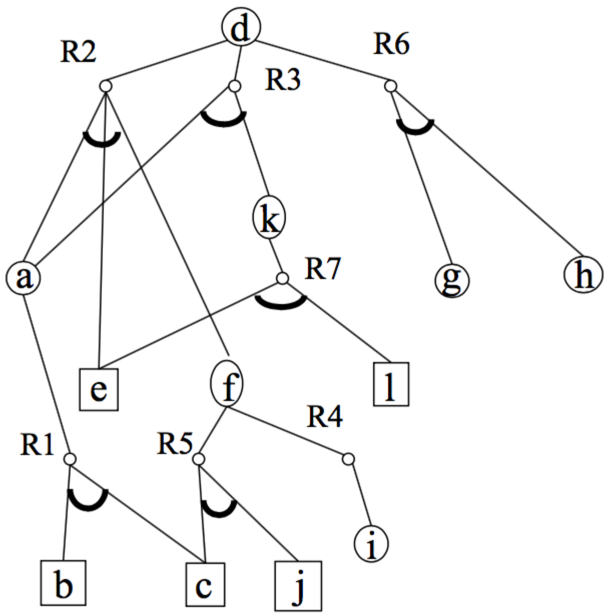
- 3 mats (A, B, C)
- L'état initial: une pyramide des N disques sur le mat A
- L'opérateur : déplacer un disque
- L'état final : la même pyramide de disques sur le mat C
- Il faut résoudre les sous-problèmes
  - Placer N-1 disques de A vers B
  - Placer le N ième disque sur C
  - Placer N-1 disques de B vers C

Représentation par graphes de sous-problèmes

- État initial ou courant : problème à résoudre
- Opérateur : décomposition en plusieurs sous-problèmes
- États finaux : problèmes triviaux
- Représentation par un graphe ET/OU sans circuit :
  - nœuds OU associés aux problèmes
  - nœuds ET de décomposition

Décomposition d'un problème en sous- problèmes

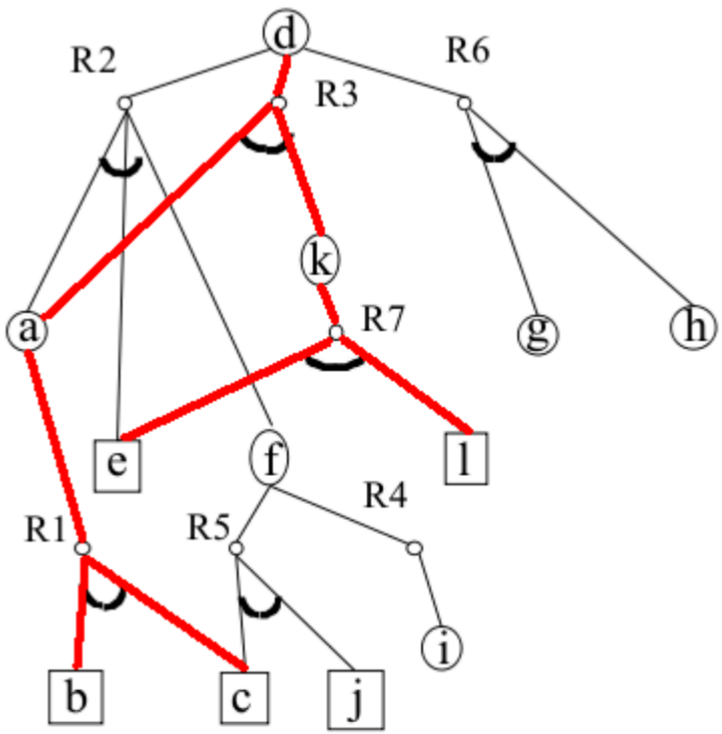
- Règles de décomposition :
  - R1 : a → b, c
  - R2 : d → a, e, f
  - R3 : d → a, k
  - R4 : f → i
  - R5 : f → c, j
  - R6 : d → g, h
  - R7 : k → e, l
- Problèmes terminaux : b, c, e, j, l
- Problème à résoudre : d

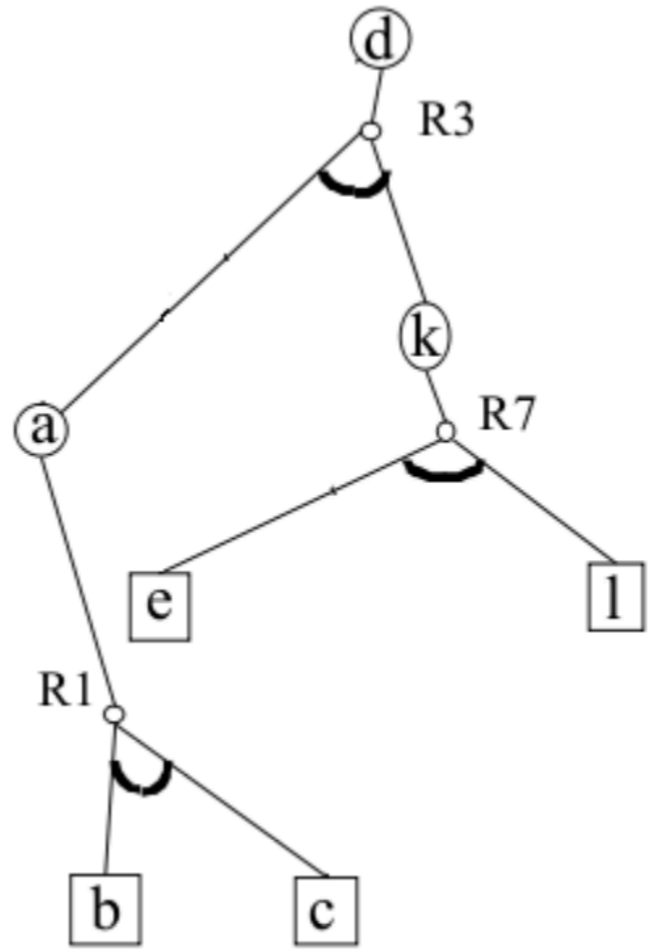


Recherche d'une solution dans un graphe ET/OU

- Recherche d'une solution = recherche d'un sous-graphe contenant
  - le nœud racine
  - pour tout nœud non final : un seul connecteur `et` les nœuds auxquels il mène

Exemple de solution





Recherche de solution dans un graphe ET/OU

On définit

- des états RÉSOLUS :
  - les états terminaux
  - les états dont un connecteur a tous ses fils résolus
- des états INSOLUBLES :
  - les états non terminaux sans successeur
  - les états dont tous les connecteurs ont au moins un successeur insoluble
- On développe en profondeur un sous-graphe représentant une solution partielle

Problèmes de satisfaction de contraintes (CSP)

- CSP = Constraint Satisfaction Problem
- Définition :
  - $X = \{ X_1, X_2, ..., X_n\}$ est l'ensemble des variables du problème
  - On associe à chaque variable X_i son domaine $D(X_i)$, l'ensemble des valeurs que peut prendre X_i
  - $C = \{C_1, C_2, ..., C_k\}$ est l'ensemble des contraintes
  - Chaque contrainte C_j est une relation entre certaines variables de X, restreignant les valeurs que peuvent prendre simultanément ces variables
- Exemple : problème du zèbre, problème des reines

Exemple : le problème des reines

- Placer N reines sur un échiquier NxN sans qu'aucune reine ne puisse en prendre une autre.
- Exemple: pour N = 4

	R		
			R
R			
		R	

Représentation du problème

- Première solution : chaque case échiquier(i,j) prend la valeur *vide* ou *reine*
  - grande combinatoire dans la résolution du problème
- Deuxième solution :
  - ligne(i) = j si la ième ligne a une reine en colonne j
  - ligne(i) = 0 si cette ligne est vide

➔ intègre déjà une partie des contraintes, puisqu'on ne peut pas avoir deux reines sur la même ligne

Résoudre un problème de satisfaction de contraintes

- Résoudre = affecter une valeur à toutes les variables sans violer de contraintes
  - On construit une solution puis on vérifie que les contraintes sont satisfaites
  - Dans l'exemple : placer une reine sur chaque ligne puis vérifier qu'elles ne sont pas en prise

Exemple pour les 4 reines

La modélisation du problème

- Variables : $X = \{X_1, X_2, X_3, X_4\}$
- Domaines : $D(X_1) = D(X_2) = D(X_3) = D(X_4) = \{1, 2, 3, 4\}$
- Contraintes:
  - les reines doivent être sur des lignes différentes
    $Clig = \{X_i \neq X_j \mid i,j \in \{1, 2, 3, 4\} \text{ et } i \neq j\}$
  - les reines doivent être sur des diagonales montantes différentes $Cdm = \{X_i+i \neq X_j+j \mid i, j \in \{1, 2, 3, 4\} \text{ et } i \neq j\}$
  - les reines doivent être sur des diagonales descendantes différentes $Cdd = \{X_i-i \neq X_j-j \mid i,j \in \{1, 2, 3, 4\} \text{ et } i \neq j\}$
  - L'ensemble des contraintes est défini par l'union de ces 3 ensembles C = Clig U Cdm U Cdd .
- Une solution: $A = \{(X_1,2), (X_2,4), (X_3,1), (X_4,3)\}$

Représentation par graphe d'états

- On représente l'ensemble des états du problème par un graphe orienté étiqueté.
  - Les sommets sont les états du problème
  - Il existe un arc **(u,v)** si un opérateur transforme l'état **u** en état **v**, **étiquette(u,v) = opérateur**
- Recherche d'une solution = recherche d'un chemin entre le nœud initial et un nœud terminal
- Solution = séquence des opérateurs étiquetant les arcs de ce chemin

Recherche dans un graphe d'état

- On définit pour chaque problème :
  - Un état initial (ou plusieurs)
  - Un état final (ou plusieurs)
  - Des états interdits (éventuellement)
  - Des opérateurs de transition
- On définit un algorithme général de recherche dans le graphe ainsi construit

Choisir une bonne représentation des états

Problème des huit reines :

1. Si 1 état = 1 case
   - 2^{64} états possibles : case(i,j) ∈ { reine, vide}
   - Opérateur : placer_reine(i,j)

     Si case(i,j) = vide alors case(i,j) ← reine
   - Test : échiquier_sans_prise, échiquier complet
2. Si 1 état = 1 ligne
   - 2^8 états possibles : ligne(i) = {1, 2,...,8,vide}
   - Opérateur : placer_reine(i,j)

     Si ligne(i) = vide alors ligne(i) ← j
   - Test : échiquier_sans_prise, échiquier complet

Exercices

- Tours de Hanoi
- Qui a le zèbre ? Qui boit de l'eau?

Le zèbre

- Cinq maisons consécutives, de couleurs différentes, sont habitées par des hommes de différentes nationalités. Ils possèdent tous un animal différent, ont chacun une boisson préférée différente et fument des cigarettes différentes. On sait que:

1. Le norvégien habite la première maison,
2. La maison à coté de celle du norvégien est bleue,
3. L'habitant de la troisième maison boit du lait,
4. L'anglais habite la maison rouge,
5. L'habitant de la maison verte boit du café,
6. L'habitant de la maison jaune fume des kool
7. La maison blanche se trouve juste après la verte,
8. L'espagnol a un chien,
9. Ukrainian boit du thé,
10. Le japonais fume des craven
11. Le fumeur de old gold a un escargot,
12. Le fumeur de gitane boit du vin,
13. Un voisin du fumeur de chesterfield a un renard,
14. Un voisin du fumeur de kool a un cheval.
15. Qui boit de l'eau ? Qui possède un zèbre ?

Question?