

Prolog és a kereső algoritmusok

1) Nem működő

Prolog-lik
Keresés

2) Működő

Backtrack
Keresőgráfal keresés

Keresési probléma

A, k, V, O

$k \in A$

$V \subseteq A$

$\sigma \in O$

$\sigma: A \rightarrow A$

$A = \text{állapotok}$

$A = \{a \mid \text{állapot } e(a)\}$

$V = \{a \mid \text{végállapot } e(a)\}$

$a \in A$

$\sigma(a) = b$

$a, b \in A$

esetleg
a predikátumok
teljes

pre- $\sigma(a)$
post- $\sigma(e)$

$op(a, \sigma(a)) = i$

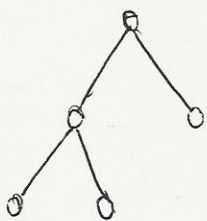
$op(a, k) = i$

$creat_k: NK \rightarrow A$

eredőállapot elváltása

$creat_k(k) = i$

esetleg predikátum



$ose(x, y) \neq \text{művelet}(x, y)$

$ose(x, y) = - \text{művelet}(z, y) + ose(x, z)$

A kereső probléma megoldása a keresés felvétel
egy új a start csúcsból valamely terminális
csúcsba.

A keresőja két operátorok, a csúcsok állapotok.

Az állapotok meg kell látni a rekurzió esetét

keres(Akt, Mem) :- vegallapot(Akt), kiir(Mem).

keres(Akt, Mem) :- szop(Akt, Uj), not(ismem(Uj, Mem)),
Mem2 = [Uj | Mem], keres(Uj, Mem2).

Goal

keres(u, []).

szop(Akt, Uj) :- op1(Akt, Uj); op2(Akt, Uj); op3(Akt, Uj);
op4(Akt, Uj) ...

op1(Akt, Uj) :- pre-op1(Akt),

⋮

} Uj előállítás

allapot(Uj).

Domains

elem = integer

lista = lista *

lista2 = lista *

Predicates

keres(lista, lista2)

vegallapot(lista)

kiir(lista2)

szop(lista, lista)

by(lista, elem, elem, lista)

jb(— " —)

ismem(lista, lista2)

pre-by(lista, elem, elem)

pre-jb(— " —)

allapot(lista)

clauses

vepallapost $(c_0, 0, 1, 3, 3)$.

kuir (T) .

kuir $(x | x_s)$:- write (x) , kuir (x_s) .

rupop (A_{lit}, U_j) :-

lj $(alt, 0, 1, U_j)$;

;

jb $(alt, 0, 1, U_j)$;

;

jb $(alt, 2, 0, U_j)$.

lj $([k_b, s_z, c_s, k_j, s_z])$ as A_{lit}, k, s_z, U_j :-

pre_lj (A_{lit}, k, s_z) ,

$U_j = [k_b - k, s_z - s_z, c_s * (-1), k_j + k, s_z + s_z]$;

allapost (U_j) .

pre_lj $([k_b, s_z, c_s, k_j, s_z], k, s_z)$:-

$c_s = -1, k_b >= k, s_z >= s_z$.

