

IMPLEMENTACJA PLANERA

---założenia

Zrozumienie założeń do implementacji wymaga znajomości algorytmu
--w zakresie omówionym na zajęciach wprowadzających oraz w dokumencie

2-Wprowadzenie_do_algorytmu

STRUKTURY DANYCH

Reprezentacja składników stanu--na przykładach

zależność przestrzenna **klocek b5 leży na klocku b2**

jest reprezentowana w postaci struktury **on(b5, b2)**

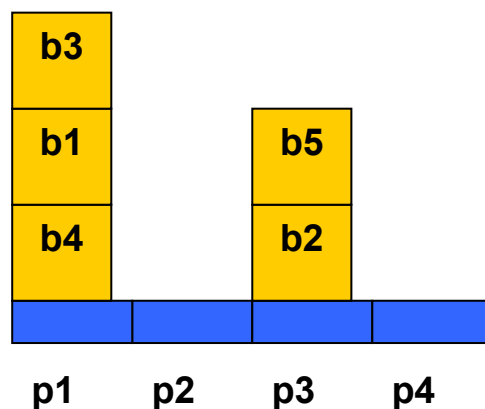
składnik stanu **klocek b7 jest wolny**

jest reprezentowana w postaci struktury **clear(b7)**

Reprezentacja stanu

--lista struktur reprezentujących składniki stanu

Przykładowy stan:



[on (b4, p1), on (b1, b4), on (b3, b1), on (b2, p3),
clear (b3), clear (b2), clear (p2), clear (p4)]

Reprezentacja celów

Cele mają postać składników stanu, np.

on(b1, b3)

Reprezentacja celu nie w pełni ukonkretnionego z nałożonymi więzami --przykład

Reprezentacja celu **wolne/swobodne X** z nałożonym warunkiem
X leży na obiekcie b4 :

clear(X/ on(X, b4))

Reprezentacja akcji--przykład

Akcja **przenies b1 z b4 na b3**

jest reprezentowana w postaci struktury

move(b1, b4, b3)

Reprezentacja akcji nie w pełni ukonkretnionej z nałożonymi więzami --przykłady

Reprezentacja akcji **przenies b4 z obiektu Y na b2** z nałożonym warunkiem
b4 leży na obiekcie Y :

move(b4, Y/ [on(b4, Y)], b2))

Reprezentacja akcji **przenies b1 z b4 na Z** z nałożonymi warunkami
obiekt Z jest wolny oraz **obiekt Z jest różny od b1 :**

move(b1 , b4, Z/[clear(Z), diff(Z, b1)])

Reprezentacja akcji **przenies obiekt X z b4 na b2** z nałożonym warunkiem
obiekt X leży na b4 :

move(X/ [on(X,b4)], b4, b2))

CIĄG DALSZY NA NASTĘPNEJ STRONIE

WSTĘPNA WERSJA KODU PROCEDURY GŁÓWNEJ

**JEST TO WERSJA WSTĘPNA: DO SKORYGOWANIA I UZUPEŁNIENIA
W TRAKCIE OPRACOWANIA ZADANIA**

**STRUKTURA PROCEDURY GŁÓWNEJ WZOROWANA NA IMPLEMENTACJI PLANERA
PRZEDSTAWIONEJ W PODRĘCZNIKU**

**Bratko Prolog Programming for Artificial Intelligence
 ed. 3 Pearson Education / Addison-Wesley 2001**

plan (State, Goals, [], State) : -

goals_achieved (Goals, State) .

plan (InitState, Goals, Plan, FinalState) : -

**choose_goal (Goal, Goals, RestGoals, InitState),
achieves (Goal, Action),
requires (Action, CondGoals, Conditions),
plan (InitState, CondGoals, PrePlan, State1),
inst_action(Action, Conditions, State1, InstAction),
perform_action (State1, InstAction, State2),
plan (State2, RestGoals, PostPlan, FinalState),
conc (PrePlan, [InstAction | PostPlan], Plan) .**

ARGUMENTY W WYRAŻENIACH PREDYKATOWYCH

---ZAMIERZONE ZNACZENIE ODNIESIONE DO DZIEDZINY PROBLEMU

InitState	stan początkowy
Goals	lista celów
Plan	skonstruowany plan
FinalState	stan końcowy
Goal	cel wybrany z listy celów
RestGoals	pozostałe cele
Action	akcja osiągnięta zadany cel
CondGoals	warunki dla akcji, które stają się nowymi celami
Conditions	warunki dla akcji do sprawdzenia w stanie, w którym akcja będzie wykonywana
PrePlan	skonstruowany preplan
State1	stan pośredni 1, osiągnięty po wykonaniu preplanu
InstAction	akcja ukonkretniona przed wykonaniem
State2	stan pośredni 2, osiągnięty po wykonaniu akcji w stanie pośrednim 1
PostPlan	skonstruowany postplan

CIĄG DALSZY NA NASTĘPNEJ STRONIE

ZALECANA KOLEJNOŚĆ PRACY NAD PROCEDURAMI:

goals_achieved

choose_goal

achieves

requires

inst_action

perform_action

UZUPEŁNIENIA W PROCEDURZE GŁÓWNEJ:

---zaimplementowanie ograniczenia długości planu ze zwiększaniem limitu w razie potrzeby

--zabezpieczenie przed niszczeniem celów już osiągniętych w trakcie planowania

---zaimplementowanie testowego trybu wykonania program, umożliwiającego użytkownikowi sterowanie decyzjami algorytmu w poszczególnych krokach.

Szczegółowe założenia do tych uzupełnień funkcjonalności programu będą podane na zajęciach na spotkaniach etapowych

CIĄG DALSZY NA NASTĘPNEJ STRONIE

PRZYDATNE PROCEDURY WBUDOWANE (PREDYKATY SYSTEMOWE)

not *LUB* \+ - *negacja*

Przykład użycia:

not(member(X, Lista)) *LUB* (preferowane) \+ member(X, Lista)

var *i* **nonvar** - *sprawdzenie, czy zmienna jest ukonkretniona*

var(X) - *przetworzenie wywołania kończy się powodzeniem
wtedy i tylko wtedy, gdy w chwili wywołania X jest zmienną nie ukonkretnioną.*

nonvar(X) - *przetworzenie wywołania kończy się powodzeniem
wtedy i tylko wtedy, gdy w chwili wywołania zmienna X jest ukonkretniona.*

=.. - *wyodrębnienie ze struktury funktora i listy argumentów
lub zbudowanie struktury dla zadanego funktora i listy argumentów*

Przykłady użycia:

arc(a, b) =.. L

*po przetworzeniu wywołania zmienna L
będzie związana z listą [arc, a, b]*

X =.. [arc, a, b]

*po przetworzeniu wywołania zmienna X
będzie związana ze strukturą arc(a, b)*