

# 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)`

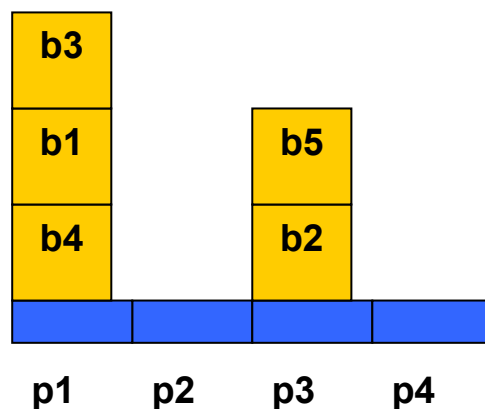
składnik stanu *klocek b7 jest wolny*

jest reprezentowany w postaci struktury `clear(b7)`

#### *Reprezentacja stanu*

--stan jest reprezentowany przez listę struktur reprezentujących składniki stanu

Przykładowy stan:



Reprezentacja w postaci listy składników stanu::

```
[ 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 ukonkretnionego z nałożonymi warunkami**

Cele nie ukonkretnione z nałożonymi warunkami są reprezentowane z wykorzystaniem struktury o funktorze infiksowym / (ukośnik) :

*<zmienna reprezentująca obiekt>/<warunek>*

gdzie *<warunek>* ma postać składniową celu.

### **przykłady:**

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

**clear(X/ on(X, b4))**

Reprezentacja celu **wolne X** z nałożonym warunkiem  
**X leży na klocku Y który leży na klocku b7 :**

**clear(X/on(X, Y/on(Y, b7)))**

Reprezentacja celu **klocek X leży na klocku Y** z nałożonymi warunkami  
**klocek X jest wolny oraz klocek Y leży na obiekcie b4 :**

**on(X/clear(X), Y/on(X, b4))**

\*\*\*\*\*  
**CIĄG DALSZY NA NASTĘPNEJ STRONIE**  
\*\*\*\*\*

### **Reprezentacja akcji--przykład**

Akcja **przenieś b1 z b4 na b3**

jest reprezentowana w postaci struktury

**move(b1, b4, b3)**

### **Reprezentacja akcji nie ukonkretnionej z nałożonymi warunkami**

Obiekty akcji nie ukonkretnione z nałożonymi warunkami są reprezentowane podobnie jak w przypadku celów nie ukonkretnionych: z wykorzystaniem struktury o funktorze infiksowym **/** (ukośnik) :

**<zmienna reprezentująca obiekt>/<warunek>**

gdzie **<warunek>** ma postać składniową celu.

#### **przykłady:**

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

**move(b4, Y/on(b4, Y), b2))**

Reprezentacja akcji **przenieś b1 z b4 na Z** z nałożonym warunkiem **obiekt Z jest wolny** :

**move(b1, b4, Z/clear(Z))**

Reprezentacja akcji **przenieś 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**  
\*\*\*\*\*

## **ZAŁOŻENIA DOTYCZĄCE UKONKRETNIANIA AKCJI**

### **Przypadek 1 -- cel typu on**

Założmy, że celem jest zależność przestrzenna

**on(b1, b5)**

Akcja osiągającą ten cel będzie zapisana w postaci

**move(b1, Y/on(b1, Y), b5)**

Zmienna **Y** zostanie ukonkretniona bezpośrednio przed wykonaniem akcji, w stanie, w którym akcja będzie wykonywana (stan pośredni 1).

### **Przypadek 2 -- cel typu clear**

Założmy, że celem jest zależność przestrzenna

**clear(b2)**

Akcja osiągającą ten cel będzie zapisana w postaci

**move(X/on(X, b2), b2, Z/clear(Z))**

Warunkiem dla tej akcji, przekazywanym jako cel do następnego kroku budowania preplanu, będzie zależność przestrzenna

**clear(X/on(X, b2))**

Zmienna **X** zostanie ukonkretniona w ostatnim kroku rekurencji budowania preplanu.

Natomiast zmienna **Z** zostanie ukonkretniona bezpośrednio przed wykonaniem akcji, w stanie, w którym akcja będzie wykonywana (stan pośredni 1).

Założmy teraz, że cel **clear** jest nie ukonkretniony, np.

**clear(X1/on(X, b7))**

Akcja osiągającą ten cel będzie zapisana w postaci

**move(X2/on(X2, X1/on(X1, b2)), X1/on(X1, b2), Z/clear(Z))**

Gdy pierwszy argument zostanie ukonkretniony w ostatnim kroku rekurencji budowania preplanu, w tej samej operacji zostanie ukonkretniony drugi argument (ponieważ w drugim argumencie występuje ta sama zmienna **X1**, która występuje w pierwszym argumencie).

\*\*\*\*\*  
**CIĄG DALSZY NA NASTĘPNEJ STRONIE**  
\*\*\*\*\*

## KOD PROCEDURY GŁÓWNEJ ---WERSJA WSTĘPNA

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

**STRUKTURA PROCEDURY GŁÓWNEJ JEST WZOROWANA NA IMPLEMENTACJI PLANERA  
(Z INNYMI ZAŁOŻENIAMI NIŻ W TYM ZADANIU)  
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),  
plan (InitState, CondGoals, PrePlan, State1),  
inst\_action(Action, Goal, 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**

<b>InitState</b>	<b>stan początkowy</b>
<b>Goals</b>	<b>lista celów</b>
<b>Plan</b>	<b>skonstruowany plan</b>
<b>FinalState</b>	<b>stan końcowy</b>
<b>Goal</b>	<b>cel wybrany z listy celów</b>
<b>RestGoals</b>	<b>pozostałe cele</b>
<b>Action</b>	<b>akcja osiagająca zadany cel</b>
<b>CondGoals</b>	<b>warunki dla akcji, które stają się nowymi celami</b>
<b>PrePlan</b>	<b>skonstruowany preplan</b>
<b>State1</b>	<b>stan pośredni 1, osiagany po wykonaniu preplanu</b>
<b>InstAction</b>	<b>akcja ukonkretniona przed wykonaniem</b>
<b>State2</b>	<b>stan pośredni 2, osiagany po wykonaniu akcji</b>
<b>PostPlan</b>	<b>w stanie pośrednim 1</b>
	<b>skonstruowany postplan</b>

\*\*\*\*\*  
**CIĄG DALSZY NA NASTĘPNEJ STRONIE**  
\*\*\*\*\*

**KRÓTKI OPIS PROCEDUR**  
**--POWTÓRZENIE OMÓWIENIA**  
**NA ZAJĘCIACH WPROWADZAJĄCYCH**

**goals\_achieved** --- sprawdza, czy wszystkie cele z podanej listy są spełnione w aktualnym stanie. Cele mogą być zarówno ukonkretnione, jak i nie ukonkretnione—z argumentami w postaci zmiennych z nałożonymi warunkami.

W zależności od wyniku sprawdzenia wykonanie procedury kończy się powodzeniem albo niepowodzeniem.

\*\*\*\*\*

**choose\_goal** --- z podanej listy celów wybiera do przetwarzania cel, który nie jest spełniony w aktualnym stanie.

Wyniki:

---wybrany cel

---lista pozostałych celów

\*\*\*\*\*

**achieves** --- określa akcję osiągającą podany cel. Cel może być zarówno ukonkretniony, jak i nie ukonkretniony--z argumentami w postaci zmiennych z nałożonymi warunkami. W konsekwencji również struktura reprezentująca akcję--wynik wykonania procedury--może zawierać zarówno symbole, reprezentujące konkretne obiekty, jak i zmienne z nałożonymi warunkami.

\*\*\*\*\*

**requires** --- określa warunki wykonania podanej akcji, które stają się celami dla następnego kroku algorytmu.

\*\*\*\*\*

\*\*\*\*\*

**CIĄG DALSZY NA NASTĘPNEJ STRONIE**

\*\*\*\*\*

**inst\_action** --- ukonkretnia akcję przed wykonaniem . W zależności od rodzaju celu, do którego akcja była dobrana przez procedurę **achieves**, niektóre/wszystkie zmienne występujące w argumentach akcji nie ukonkretnionej mogą mieć w chwili wywołania **inst\_action** przypisane wartości (przypisanie wartości nastąpiło w ramach konstruowania preplanu).

Przebieg ukonkretnienia akcji:

---w przypadku nie ukonkretnionego argumentu struktury **move** zawierającego zmienną z nałożonym warunkiem, której to zmiennej została już przypisana wartość (patrz wyżej), do struktury reprezentującej akcję ukonkretnioną zostaje wpisana ta zmienna (o przypisanej wartości).

**przykład:** akcja nie ukonkretniona: **move(X2/on(X2, X1/on(X1, b7)) , .... , .... )**  
zmienna **X** ma przypisaną wartość, uzyskaną w ramach konstruowania preplanu  
akcja ukonkretniona: **move(X, ... , ...)**

---w przypadku argumentu zawierającego zmienną z nałożonym warunkiem, która to zmienna nie została ukonkretniona w trakcie konstruowania preplanu, procedura **inst\_action** określa wartość tej zmiennej przez przetworzenie struktury **<zmienna>/<warunek>** w kontekście reprezentacji aktualnego stanu

**przykład:** akcja nie ukonkretniona: **move(... , Y/on(Y, b2) , ... )**  
aktualny stan zawiera składnik: **on(b5, b2)**  
akcja ukonkretniona: **move(... , b5 , .... )**

To, czy zmienna z nałożonym warunkiem uzyskała wartość w trakcie konstruowania preplanu, można stwierdzić sprawdzając typ celu, do którego została dobrana akcja. Stan ukonkretnienia zmiennej można też sprawdzić używając wbudowanej procedury **var**.

\*\*\*\*\*

**perform\_action** --- określa stan osiągnięty ze stanu podanego przez wykonanie podanej akcji

\*\*\*\*\*

## ZAŁOŻENIE DOTYCZĄCE POCZĄTKOWEJ LISTY CELÓW

***W początkowej liście celów mogą występować zarówno cele ukonkretnione, jak i cele nie ukonkretnione, w których obiekty są reprezentowane przez zmienne z nałożonymi warunkami.***