

SOLUÇÃO BASEADA EM IA PARA O PROBLEMA DE PLANEJAMENTO DO MUNDO DOS BLOCOS DE TAMANHO VARIÁVEL

1) Contexto

Este documento integra três referenciais: (1) o mundo dos blocos de tamanho 1 (STRIPS clássico), (2) a formulação em Lógica de Primeira Ordem / Prolog para blocos de tamanho variável e (3) o paradigma *Planning as Model Checking* (NuSMV). Ele demonstra como um ChatBot de IA (como o ChatGPT) pode apoiar a construção, validação e verificação formal de planos para o domínio.

2) Modelo do Domínio

Predicados estáticos

- block(B). size(B, W).
- table_slot(X). table_width(N).

Predicados dinâmicos

- pos(B, table(X)) ou pos(B, on(B2)).
- clear(B).

Regras físicas

- Um bloco só pode ser movido se estiver clear/1.
- Para empilhar: destino clear, diferente de B e $\text{size}(B) \leq \text{size}(B2)$.
- Para mesa: espaço contíguo livre $X..X+W-1$.

3) Núcleo Prolog do domínio

A seguir um esqueleto funcional:

```
% ----- Parte 1: fatos estáticos -----
block(a). block(b). block(c). block(d).
size(a,1). size(b,1). size(c,2). size(d,2).
table_width(7).
table_slot(X) :- integer(X), X >= 0, table_width(W), X < W.

% ----- Parte 2: auxiliares -----
absolute_pos(B, S, X) :- member(pos(B, table(X)), S).
absolute_pos(B, S, X) :- member(pos(B, on(Sup)), S),
                           absolute_pos(Sup, S, XS),
                           X = XS.

busy_slots(B, S, Slots) :-
    absolute_pos(B, S, X),
    size(B, W),
    XE is X + W - 1,
    findall(T, between(X, XE, T), Slots).

slot_busy(Slot, S) :-
    member(pos(B,_), S),
    busy_slots(B, S, Bs),
    member(Slot, Bs).

range_free(X, W, S) :-
    XE is X + W - 1,
    table_width(TW),
    XE < TW,
    forall(between(X, XE, T), \+ slot_busy(T, S)).

% ----- Parte 3: operador move/2 -----
can(move(B, on(TB)), S) :-
    member(clear(B), S),
    B \== TB,
    member(clear(TB), S),
    size(B, W1),
    size(TB, W2),
    W1 =< W2.

can(move(B, table(X)), S) :-
    member(clear(B), S),
    size(B, W),
    range_free(X, W, S).
```

4) Situação de teste (exemplo)

$S_0 = [\text{pos}(d, \text{table}(0)), \text{pos}(b, \text{on}(d)), \text{pos}(a, \text{on}(b)), \text{pos}(c, \text{on}(a)), \text{clear}(c)]$.

$\text{Goal} = [\text{pos}(b, \text{table}(2)), \text{pos}(a, \text{on}(b)), \text{pos}(c, \text{on}(a)), \text{pos}(d, \text{on}(c))]$.

A IA pode gerar um plano mínimo move/2 e validar cada passo via can/2.

5) Modelo NuSMV (Planning as Model Checking)

```
MODULE main
VAR
  xa : 0..6; xb : 0..6; xc : 0..6; xd : 0..6;
  sup_a : {table,b,c,d};
  sup_b : {table,a,c,d};
  sup_c : {table,a,b,d};
  sup_d : {table,a,b,c};

DEFINE
  size_a := 1; size_b := 1; size_c := 2; size_d := 2;
  table_width := 7;
  stable_ab := (sup_a = b) -> (size_a <= size_b);

INIT
  sup_d = table & xd = 0 &
  sup_b = d & sup_a = b & sup_c = a &
  xb = xd & xa = xb & xc = xa;

CTLSPEC ! EF (sup_b = table & xb = 2 & sup_a = b & sup_c = a & sup_d = c)
```

6) Como usar o ChatBot

1. Solicite a geração do domínio Prolog.
2. Peça integração com o planner de regressão.
3. Forneça S0 e Goal; gere o plano mínimo.
4. Gere a modelagem NuSMV e verifique !EF(goal).

7) Benefícios

- Representação unificada ($pos/2 + size/2$).
- Validação lógica e espacial.
- Planejamento validado por *model checking*.

Gerado por: ChatGPT (GPT-5) — Outubro/2025