

**Tabela 1 — Conceitos do Mundo dos Blocos com tamanhos variáveis**

Conceito	Definição em linguagem natural
Block Properties	Cada bloco tem um tamanho fixo (largura em slots da mesa).
Mobility	Um bloco só pode ser movido se estiver livre (clear).
Target Accessibility	O destino de um movimento (outro bloco ou mesa) precisa estar livre (clear).
Stability	Só é permitido colocar um bloco sobre outro de largura maior ou igual.
Spatial Occupancy	Ao posicionar na mesa, todos os slots contíguos necessários ao tamanho do bloco devem estar livres.
Logical Validity	Nenhum bloco pode ser colocado sobre si mesmo.
Busy Slots	Cada bloco ocupa uma faixa de slots: pos + size - 1.
Absolute Position	O ponto de referência de cada bloco é o slot mais à esquerda da sua base.
Space Check	Verifica se a faixa de slots de um bloco está livre antes de mover.
Size Check	Verifica se a largura do bloco de suporte é suficiente para o bloco que será colocado sobre ele.
Goal Specification	Objetivo definido como configuração final desejada (ex.: torre ordenada).
Plan Extraction	O plano é obtido como sequência de move que transforma o estado inicial no estado meta.

**Tabela 2 — Representação dos conceitos em STRIPS, Prolog estendido e NuSMV (tamanhos variáveis)**

Conceito	STRIPS	Prolog estendido	Proposta em NuSMV	Justificativa
Block Properties	block(X).	size(X, W).	DEFINE size_a := 1; size_b := 2; ...	Tamanhos imutáveis → definidos como constantes.
Mobility	clear(X)	clear(Block)	DEFINE clear_a := !(on_b = a   on_c = a   ...)	Simplifica condição para mover.
Target Accessibility	clear(Target) requisito	clear(Target)	DEFINE target_ok_a := clear_a;	Evita empilhar sobre blocos ocupados.
Stability	não tratado	size(B, W1), size(T, W2), W1 ≤ W2	DEFINE size_check_b_a := (size_b ≤ size_a);	Evita instabilidade física.

Conceito	STRIPS	Prolog estendido	Proposta em NuSMV	Justificativa
Spatial Occupancy	não tratado	busy_slots(Block, State)	DEFINE busy_a(slot) := (on_a=tableX & slot in [X..X+size_a-1])	Representa uso de múltiplos slots.
Busy Slots	não tratado	absolute_pos(Block, S), size(Block, W) → lista [S..S+W-1]	DEFINE busy_a(slot) como acima	Permite verificar colisão de blocos.
Space Check	não tratado	space_check(Block, Dest) verifica se slots estão livres	DEFINE space_ok_c_table2 := is_free(2) & is_free(3)...	Garante que não haja sobreposição na mesa.
Logical Validity	neq(A,B)	restrição explícita Block ≠ Block	TRANS !(move = move_a_a) ou implícito	Evita auto-colocação.
Absolute Position	não tratado	absolute_pos(Block, Pos)	on_a pode armazenar índice da mesa (slot mais à esquerda)	Permite calcular ocupação e derivar busy_slots.
Goal Specification	conjunto de literais on(...)	pos/2 ou on/2 metas	CTLSPEC !EF (on_a=b & on_b=table2 & ...)	Contraexemplo é o plano.
Plan Extraction	sequência de operadores	sequência de move(...)	Trace de NuSMV mostrando valores de move e on_*	Contraexemplo funciona como plano.