

### Situação 1

Conceito	STRIPS	Prolog extendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
<b>Block Properties</b>	block(a). block(b). block(c). block(d).	tamanho(a,1).t amanho(b,1).ta manho(c,2).ta manho(d,3)	DEFINE size_a := 1;size_b:=1; size_c:= 2; size_d:=3;	Cada bloco tem largura fixa, representada como constante

Tipo de Restrição	Destino	Regra em Linguagem Natural	Implementaç ã o NuSMV (Exemplo: move(C, A))	Implementaç ã o NuSMV (Exemplo: move(C, table(2)))
<b>Mobility</b>	Bloco a	Só pode mover a se não houver outro bloco em cima dele	TRANS move_a_b -> clear_a;	TRANS move_a_table2 -> clear_a;
<b>Target Accessibility</b>	Bloco b ou c	Só é possível colocar a em cima de b ou c se estes estiverem livres	TRANS move_a_b -> clear_b;	—

<b>Stability</b>	Empilhamento	<b>a</b> só pode ser colocado sobre blocos de tamanho $\geq$ tamanho de <b>a</b>	TRANS move_a_b -> size_a <= size_b;	TRANS move_a_c -> size_a <= size_c;
------------------	--------------	--	---	---

<b>Spatial Occupancy</b>	Slots da mesa	Para colocar <b>c</b> na mesa em posição 3, slots 3 e 4 precisam estar livres	—	TRANS move_c_table3 -> free(3) & free(4);
<b>Logical Validity</b>	—	Um bloco não pode ser colocado sobre si mesmo.	TRUE (implicit)	implícito

### Situação 2

Conceito	STRIPS	Prolog extendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
<b>Block Properties</b>	block(a). block(b). block(c). block(d).	tamanho(a,1).t amanho(b,1).ta manho(c,2).ta manho(d,3)	DEFINE size_a := 1;size_b:=1; size_c:= 2; size_d:=3;	Cada bloco tem largura fixa, representada como constante

Tipo de Restrição	Destino	Regra em Linguagem Natural	Implementaç ã o NuSMV (Exemplo: move(C, A))	Implementaç ã o NuSMV (Exemplo: move(C, table(2)))
<b>Mobility</b>	Bloco a, c	Só podem ser movidos se não houver outro bloco em cima	TRANS move_c_d -> clear_c;	TRANS move_a_table1 -> clear_a;
<b>Target Accessibility</b>	Bloco b,d	Só pode empilhar a em b ou c em d se estes estiverem livres	TRANS move_c_d -> size_c <= size_d;	—

<b>Stability</b>	Empilhamento	Só é permitido colocar c sobre d se $\text{size}(c) \leq \text{size}(d)$	TRANS move_c_d -> size_c <= size_d;	
------------------	--------------	--	---	--

<b>Spatial Occupancy</b>	Slots mesa	Para colocar d na mesa, slots necessários devem estar livres	—	TRANS move_d_table3 -> free(3) & free(4);
<b>Logical Validity</b>	—	Não pode mover bloco para cima de si mesmo	Implícito	Implícito

### Situação 3

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
<b>Block Properties</b>	block(a). block(b). block(c). block(d).	tamanho(a,1).t amanho(b,1).ta manho(c,2).ta manho(d,3)	DEFINE size_a := 1;size_b:=1; size_c:= 2; size_d:=3;	Cada bloco tem largura fixa, representada como constante

Tipo de Restrição	Destino	Regra em Linguagem Natural	Implementaç ã o NuSMV (Exemplo: move(C, A))	Implementaç ã o NuSMV (Exemplo: move(B, table(2)))
<b>Mobility</b>	Bloco c	Só o topo da torre (c) pode ser movido inicialmente.	TRANS move_c_a -> clear_c;	TRANS move_b_table2 -> clear_b;
<b>Target Accessibility</b>	Bloco a ou b	Para reconstruir a torre, só é possível empilhar em blocos livres.	TRANS move_c_a -> clear_a;	TRANS move_b_table2 -> clear_b;

<b>Stability</b>	Empilhamento	d só pode ser colocado sobre c se $\text{size}(d) \leq \text{size}(c)$	TRANS move_d_c -> size_d <= size_c;	—
------------------	--------------	--	---	---

<b>Spatial Occupancy</b>	Slots da mesa	Para desmontar a torre, deve haver espaço na mesa	—	TRANS move_b_table2 -> free(2);
<b>Logical Validity</b>	—	Nenhum bloco pode ser colocado sobre si mesmo	TRUE (implicit)	implícito