

Situação S0 até Sf1

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
Espaço	lugar(X)	mesa(X)	VAR X : 1..6	Conjunto de coordenadas que os blocos podem preencher.
Bloco	bloco(N)	bloco(N, W)	VAR nome = {a,b,c,d} tamanho = 1..3 DEFINE ba := nome=a & tamanho=1 bb := nome=b & tamanho=1 bc := nome=c & tamanho=2 bd := nome=d & tamanho=3	Codifica o nome e tamanho de um bloco.
Posição do bloco	on(B, X) ou on(B1, B2)	pos(B, mesa(X)) ou pos(B1, on(B2))	INIT pos(ba, on(bc)) pos(bb, 6) pos(bc, 1) pos(bd, 3)	Codifica a posição do bloco.
Espaço livre	livre(X)	livre(X) e livre(B)	INIT livre(ba) livre(bb) livre(bc) livre(bd)	Codifica espaços não ocupados na mesa e acima de blocos.
Movimento	move(B, Pi, Pj)	move(B, mesa(X)) e move(B1, on(B2))	TRANS s1 -> livre(bc) & move(bd,	Codifica os movimentos realizados entre estados.

			on(bc)) s2 -> livre(bb) & move(ba, on(bb)) s3 -> livre(3) & livre(4) & livre(5) & move(bd, 3) s4 -> livre(bc) & move(ba, on(bc)) s5 -> livre(bc) & move(bb, on(bc)) s6 -> livre(6) & move(bd, 4) s7 -> livre(bd) & move(bb, on(bd)) s8 -> livre(bd) & move(ba, on(bd)) s9 -> livre(ba) & livre(bb) & move(bc, 5)	
--	--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Situação S0 até Sf2

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
Espaço	lugar(X)	mesa(X)	VAR X : 1..6	Conjunto de coordenadas que os blocos podem preencher.
Bloco	bloco(N)	bloco(N, W)	VAR	Codifica o

			nome = {a,b,c,d} tamanho = 1..3 DEFINE ba := nome=a & tamanho=1 bb := nome=b & tamanho=1 bc := nome=c & tamanho=2 bd := nome=d & tamanho=3	nome e tamanho de um bloco.
Posição do bloco	on(B, X) ou on(B1, B2)	pos(B, mesa(X)) ou pos(B1, on(B2))	INIT pos(ba, on(bc)) pos(bb, 6) pos(bc, 1) pos(bd, 3)	Codifica a posição do bloco.
Espaço livre	livre(X)	livre(X) e livre(B)	INIT livre(ba) livre(bb) livre(bc) livre(bd)	Codifica espaços não ocupados na mesa e acima de blocos.
Movimento	move(B, Pi, Pj)	move(B, mesa(X)) e move(B1, on(B2))	TRANS s1 -> livre(bc) & move(bd, on(bc)) s2 -> livre(bb) & move(ba, on(bb)) s3 -> livre(3) & livre(4) & livre(5) & move(bd, 3) s4 -> livre(bc) & move(ba, on(bc)) s5 ->	Codifica os movimentos realizados entre estados.

			livre(bc) & move(bb, on(bc)) s6 -> livre(6) & move(bd, 4) s7 -> livre(3) & move(bb, 3) s8 -> livre(bb) & move(ba, on(bb)) s9 -> livre(bd) & move(bc, on(bd)) s10 -> livre(bc) & move(ba, on(bc)) s11 -> livre(bc) & move(bb, on(bc))	
--	--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Situação S0 até Sf3

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
Espaço	lugar(X)	mesa(X)	VAR X : 1..6	Conjunto de coordenadas que os blocos podem preencher.
Bloco	bloco(N)	bloco(N, W)	VAR nome = {a,b,c,d} tamanho = 1..3 DEFINE ba := nome=a & tamanho=1	Codifica o nome e tamanho de um bloco.

			bb := nome=b & tamanho=1 bc := nome=c & tamanho=2 bd := nome=d & tamanho=3	
Posição do bloco	on(B, X) ou on(B1, B2)	pos(B, mesa(X)) ou pos(B1, on(B2))	INIT pos(ba, on(bc)) pos(bb, 6) pos(bc, 1) pos(bd, 3)	Codifica a posição do bloco.
Espaço livre	livre(X)	livre(X) e livre(B)	INIT livre(ba) livre(bb) livre(bc) livre(bd)	Codifica espaços não ocupados na mesa e acima de blocos.
Movimento	move(B, Pi, Pj)	move(B, mesa(X)) e move(B1, on(B2))	TRANS s1 -> livre(bc) & move(bd, on(bc)) s2 -> livre(bb) & move(ba, on(bb)) s3 -> livre(3) & livre(4) & livre(5) & move(bd, 3) s4 -> livre(bc) & move(ba, on(bc)) s5 -> livre(bc) & move(bb, on(bc)) s6 -> livre(6) & move(bd, 6) s7 -> livre(3) &	Codifica os movimentos realizados entre estados.

			move(ba, 3) s8 -> livre(ba) & livre(bc) & move(bd, on(bc)) s9 -> livre(6) & move(bc, 6)	
--	--	--	-----------------------------------------------------------------------------------------------------------------	--

Situação S0 até Sf4

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
Espaço	lugar(X)	mesa(X)	VAR X : 1..6	Conjunto de coordenadas que os blocos podem preencher.
Bloco	bloco(N)	bloco(N, W)	VAR nome = {a,b,c,d} tamanho = 1..3 DEFINE ba := nome=a & tamanho=1 bb := nome=b & tamanho=1 bc := nome=c & tamanho=2 bd := nome=d & tamanho=3	Codifica o nome e tamanho de um bloco.
Posição do bloco	on(B, X) ou on(B1, B2)	pos(B, mesa(X)) ou pos(B1, on(B2))	INIT pos(ba, 4) pos(bb, 6) pos(bc, 1) pos(bd, on(ba)) &	Codifica a posição do bloco.

			pos(bd,on(bb))	
Espaço livre	livre(X)	livre(B)	INIT livre(3) livre(5) livre(bc) livre(bd)	Codifica espaços não ocupados na mesa e acima de blocos.
Movimento	move(B, Pi, Pj)	move(B, mesa(X)) e move(B1, on(B2))	TRANS s1 -> livre(bc) & move(bd, on(bc)) s2 -> livre(bb) & move(ba, on(bb)) s3 -> livre(3) & livre(4) & livre(5) & move(bd, 3) s4 -> livre(bc) & move(ba, on(bc))	Codifica os movimentos realizados entre estados.

Situação S0 até S5

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
Espaço	lugar(X)	mesa(X)	VAR X : 1..6	Conjunto de coordenadas que os blocos podem preencher.
Bloco	bloco(N)	bloco(N, W)	VAR nome = {a,b,c,d} tamanho = 1..3 DEFINE ba := nome=a &	Codifica o nome e tamanho de um bloco.

			tamanho=1 bb := nome=b & tamanho=1 bc := nome=c & tamanho=2 bd := nome=d & tamanho=3	
Posição do bloco	on(B, X) ou on(B1, B2)	pos(B, mesa(X)) ou pos(B1, on(B2))	INIT pos(ba, on(bc)) pos(bb, on(bc)) pos(bc, 1) pos(bd, 4)	Codifica a posição do bloco.
Espaço livre	livre(X)	livre(X) e livre(B)	INIT livre(3) livre(bd)	Codifica espaços não ocupados na mesa e acima de blocos.
Movimento	move(B, Pi, Pj)	move(B, mesa(X)) e move(B1, on(B2))	TRANS s1 -> livre(3) & move(bb, 3) s2 -> livre(bb) & move(ba, on(bb)) s3 -> livre(bd) & move(bc, on(bd)) s4 -> livre(bc) & move(ba, on(bc)) s5 -> livre(bc) & move(bb, on(bc))	Codifica os movimentos realizados entre estados.

Situação S0 até S7

Conceito	STRIPS	Prolog estendido	Proposta de modelo NuSMV	Justificativa para projeto NuSMV
Espaço	lugar(X)	mesa(X)	VAR X : 1..6	Conjunto de coordenadas que os blocos podem preencher.
Bloco	bloco(N)	bloco(N, W)	VAR nome = {a,b,c,d} tamanho = 1..3 DEFINE ba := nome=a & tamanho=1 bb := nome=b & tamanho=1 bc := nome=c & tamanho=2 bd := nome=d & tamanho=3	Codifica o nome e tamanho de um bloco.
Posição do bloco	on(B, X) ou on(B1, B2)	pos(B, mesa(X)) ou pos(B1, on(B2))	INIT pos(ba, 4) pos(bb, 6) pos(bc, 1) pos(bd, on(ba)) & pos(bd,on(bb))	Codifica a posição do bloco.
Espaço livre	livre(X)	livre(X) e livre(B)	INIT livre(3) livre(5) livre(bc) livre(bd)	Codifica espaços não ocupados na mesa e acima de blocos.
Movimento	move(B, Pi, Pj)	move(B, mesa(X)) e move(B1, on(B2))	TRANS s1 -> livre(bc) & move(bd, on(bc))	Codifica os movimentos realizados entre estados.

			s2 -> livre(bb) & move(ba, on(bb)) s3 -> livre(3) & livre(4) & livre(5) & move(bd, 3) s4 -> livre(bc) & move(ba, on(bc)) s5 -> livre(bc) & move(bb, on(bc)) s7 -> livre(6) & move(bd, 6)	
--	--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Restrições

Tipo de restrição	Destino	Regra em linguagem natural	Implementação o NuSMV (Exemplo: move(C,A))	Implementação o NuSMV (Exemplo: move(C,table(2)))
Mobilidade	N/D	Um bloco só pode ser movido se não houver nada sobre ele.	livre(X)	livre(2)
Acessibilidade do alvo	on(B1,B2)	O bloco alvo deve ter uma superfície livre para receber o novo bloco.	livre(B2)	N/I
Estabilidade	on(B1,B2)	O novo bloco não pode ser mais largo que o bloco alvo.	tamanho_b1 <= tamanho_b2	N/I

Ocupação espacial	mesa(X)	Deve haver um bloco contíguo de slots livres na mesa, largo o suficiente para acomodar o bloco.	livre(X) .. livre(X+taman ho_novo)	livre(2) .. livre(2+taman ho_novo)
Validade lógica	on(B2)	Um bloco não pode ser colocado sobre si mesmo.	B1 != B2	N/I