



# Sistema baseado em conhecimento para definição de restrições para construção de domínios de planejamento para área espacial

CRUZ, C. G. R.<sup>1,2</sup>, FERREIRA, M. G. V.<sup>1</sup>, SILVA, R. R.<sup>3,4</sup>

<sup>1</sup> Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil

<sup>2</sup> Mestrando em Engenharia e Gerenciamento de Sistemas Espaciais - CSE

<sup>3</sup> FATEC Mogi das Cruzes, Faculdade de Tecnologia do Estado de São Paulo, SP, Brasil

<sup>4</sup> CISUC – Centro de Informática e Sistemas da Universidade de Coimbra, Coimbra, Portugal

caio.cruz@inpe.br

---

**Resumo.** O crescente aumento de sistemas automatizados nas missões espaciais tem gerado preocupação com segurança e confiabilidade nas operações realizadas com satélites. Para prevenir a geração de estados inseguros às operações, estratégias foram criadas para validar planos de operação de voo gerados a partir de planejadores automáticos. As pesquisas demonstraram que estados inseguros podem ser gerados. Verificar se é possível remover estados inseguros de um plano a partir de mudanças na definição de domínio de planejamento é o objetivo desta pesquisa. Neste trabalho foi realizado uma análise que comprovou ser possível impedir que um plano contenha um estado inseguro, a partir da modificação de precondições e efeitos das ações do domínio. Assim, após definir a hipótese de que é possível modificar domínios que possibilitem que planejadores automáticos gerem planos sem estados inseguros, foi proposto uma estratégia para modificar domínios de planejamento utilizando-se de uma base de conhecimento de operações de voo.

---

**Palavras-chave:** Planejamento Automático; Definição de Domínios; PDDL.

## 1. Introdução

O Centro de Rastreamento e Controle de Satélites (CRC) no INPE, realiza as operações de controle de satélites através de atividades executadas na maioria dos casos de forma manual [TOMINAGA, 2011]. Encontrar soluções para automatização das operações de voo é um desafio em aberto.

Os trabalhos desenvolvidos entorno de planejadores automáticos limitam-se à criação de Plano de Operação de Voo, porém, sem fazer uso de informações históricas sobre as operações anteriormente realizadas nos satélites, deixando a cargo dos simuladores e geradores de diagnósticos a função de validarem se de fato o plano gerado automaticamente corresponde à realidade que o satélite está sujeito, no qual em muitos casos é rejeitado [SOUZA, 2011]



Os planejadores automáticos são usados para encontrar a sequência de passos que transitam um estado inicial no estado objetivo a partir de ações definidas no domínio de planejamento [KNOBLOCK, 1998]. É importante que a definição de domínio contenha apenas ações que gerem estados válidos, seguros e aplicáveis ao contexto em que o domínio faz parte, portanto, um domínio pode ser falho quando ocasionar a transição por um estado infactível.

Existe grande preocupação com a utilização de sistemas automatizados na operação de satélites, como garantir a segurança e confiabilidade nas operações [TOMINAGA, 2011]. No trabalho de SOUZA, foi proposto uma estratégia para validar planos de operação de voo, que considera um plano rejeitado caso um estado classificado como inseguro à missão compor o plano. Apesar de o planejador encontrar o estado objetivo, um novo plano deverá ser gerado utilizando-se de ações diferentes para obter o estado objetivo com outros passos sem a passagem por estados inseguros.

A validação de planos demonstra a importância da geração de planos válidos para as operações [SOUZA, 2011]. Neste contexto, prevenir que estados inseguros não sejam gerados no plano pode depender da definição de domínios de planejamento mais adequados. Dessa forma, o objetivo desta pesquisa é verificar se alterações feitas no domínio de planejamento podem contribuir para a criação de planos com maior validade, substituindo estados inválidos da sequência de passos planejados automaticamente.

## 2. Metodologia

O planejador automático AI Planning with STRIPS [BECKER, 2015] foi utilizado na execução do planejamento realizado neste trabalho. O problema do mundo dos blocos foi escolhido por ser um problema comumente utilizado em pesquisas entorno de planejamento, que consiste em empilhar blocos sobre uma mesa [KNOBLOCK, 1998].

A Listagem 1 mostra a definição do problema do mundo dos blocos no formato da Linguagem para Definição de Domínios de Planejamento (PDDL). O problema é composto de um estado inicial contendo três blocos empilhados sobre a mesa *t1* e outras duas mesas *t2* e *t3* respectivamente. O estado objetivo é a disposição da pilha de blocos em cima da mesa *t3*, seguindo a ordem do bloco *a* sobre o bloco *b*, bloco *b* sobre o bloco *c* e bloco *c* sobre a mesa *t3*.

```
1 (define (problem stack-blocks-stacked-cba-from-table1-to-stacked-abc-table3-onepilepertable)
2   (:domain blocksworld)
3   (:init (and (block a) (block b) (block c) (table t1) (table t2) (table t3)
4             (on a t1) (on b a) (on c b) (clear c) (clear t2) (clear t3)))
5   (:goal (and (on a b) (on b c) (on c t3)))
6 )
```

Listagem 1 - Definição do problema do mundo dos blocos

A Listagem 2 apresenta a definição de domínio com as ações possíveis para gerar o plano. As ações definidas no domínio são: *move* – move um bloco de uma mesa para outra, *stack2* – empilha dois blocos sobre uma mesa, *stack3* – empilha três blocos sobre uma mesa, *unstack2* – desempilha dois blocos e *unstack3* – desempilha três blocos.



```
1 (define (domain blocksworld)
2   (:requirements :strips)
3   (:action move
4     :parameters (?b ?x ?y)
5     :precondition (and (block ?b) (table ?x) (table ?y) (on ?b ?x) (clear ?b)
6       (clear ?y))
7     :effect (not (on ?b ?x) (on ?b ?y) (clear ?x) not (clear ?y))
8   )
9   (:action stack2
10    :parameters (?a ?x ?b ?y)
11    :precondition (and (block ?a) (block ?b) (table ?x) (table ?y) (clear ?a)
12      (clear ?b) (on ?a ?x) (on ?b ?y))
13    :effect (and (on ?a ?b) not (on ?a ?x) not (clear ?b) (clear ?x))
14  )
15  (:action stack3
16    :parameters (?a ?x ?b ?c ?y)
17    :precondition (and (block ?a) (block ?b) (block ?c) (table ?x) (table ?y)
18      (clear ?a) (clear ?b) (on ?a ?x) (on ?b ?c) (on ?c ?y))
19    :effect (and (on ?a ?b) not (on ?a ?x) not (clear ?b) (clear ?x))
20  )
21  (:action unstack2
22    :parameters (?a ?b ?x ?y)
23    :precondition (and (block ?a) (block ?b) (table ?x) (table ?y) (on ?b ?x)
24      (on ?a ?b) (clear ?a) (clear ?y))
25    :effect (and (on ?a ?y) not (on ?a ?b) (clear ?b) (clear ?a) not
26      (clear ?y))
27  )
28  (:action unstack3
29    :parameters (?a ?b ?c ?x ?y)
30    :precondition (and (block ?a) (block ?b) (block ?c) (table ?x) (table ?y)
31      (on ?c ?x) (on ?b ?c) (on ?a ?b) (clear ?a) (clear ?y))
32    :effect (and (on ?a ?y) not (on ?a ?b) (clear ?b) (clear ?a) not
33      (clear ?y))
34  )
35 )
```

### Listagem 2 - Domínio do mundo dos blocos

Após realizar o planejamento do problema com a definição de domínio do mundo dos blocos foi escolhido um estado entre os estados de transição do plano gerado para representar um estado inseguro. Desse modo, o estado em que o bloco *a* está sobre o bloco *b* e o bloco *b* está sobre a mesa *t2*, deve ser removido dos estados de transição. A ação que gera este estado é a *stack2*, onde o efeito define que dois blocos serão empilhados.

Afim de alterar a definição de domínio para impedir que o estado inválido faça parte do plano foi modificada a ação *stack2*. Os parâmetros, condições e efeitos foram alterados de modo que o bloco da primeira mesa não seja empilhado sobre o bloco da segunda mesa.

A Listagem 3 apresenta uma nova definição de domínio contendo a ação *stack2* modificada, a nova condição para aplicá-la é o estado contendo três mesas, três blocos, um bloco sobre cada uma das mesas e o novo efeito gerado é de empilhar o bloco da mesa central sobre o bloco da terceira mesa.



```
1 (define (domain blocksworld)
2   (:requirements :strips)
3   (:action move
4     :parameters (?b ?x ?y)
5     :precondition (and (block ?b) (table ?x) (table ?y) (on ?b ?x) (clear ?b)
6       (clear ?y))
7     :effect (not (on ?b ?x) (on ?b ?y) (clear ?x) not (clear ?y))
8   )
9   (:action stack2
10    :parameters (?a ?x ?b ?y ?c ?z)
11    :precondition (and (block ?a) (block ?b) (block ?c) (table ?x) (table ?y)
12      (table ?z) (clear ?a) (clear ?b) (clear ?c) (on ?a ?x) (on ?b ?y) (on ?c ?z))
13    :effect (and (on ?b ?c) not (on ?b ?y) not (clear ?c) (clear ?y))
14  )
15  (:action stack3
16    :parameters (?a ?x ?b ?c ?y)
17    :precondition (and (block ?a) (block ?b) (block ?c) (table ?x) (table ?y)
18      (clear ?a) (clear ?b) (on ?a ?x) (on ?b ?c) (on ?c ?y))
19    :effect (and (on ?a ?b) not (on ?a ?x) not (clear ?b) (clear ?x))
20  )
21  (:action unstack2
22    :parameters (?a ?b ?x ?y)
23    :precondition (and (block ?a) (block ?b) (table ?x) (table ?y) (on ?b ?x)
24      (on ?a ?b) (clear ?a) (clear ?y))
25    :effect (and (on ?a ?y) not (on ?a ?b) (clear ?b) (clear ?a) not (clear ?y))
26  )
27  (:action unstack3
28    :parameters (?a ?b ?c ?x ?y)
29    :precondition (and (block ?a) (block ?b) (block ?c) (table ?x) (table ?y)
30      (on ?c ?x) (on ?b ?c) (on ?a ?b) (clear ?a) (clear ?y))
31    :effect (and (on ?a ?y) not (on ?a ?b) (clear ?b) (clear ?a) not (clear ?y))
32  )
33 )
```

Listagem 3 - Domínio do mundo dos blocos modificado

A alteração feita na definição da ação *stack2* gerou um novo domínio no qual foi replanejado. Os resultados do planejamento entre o domínio original e o domínio modificado foram comparados e discutidos.

### 3. Resultados e Discussão

A seguir é apresentado o resultado da execução do planejamento para o domínio do mundo dos blocos. A execução do planejamento com a definição do domínio original, encontrou o estado objetivo a partir de um plano gerado com quinze ações. Os estados gerados através de cada ação do plano foram representados graficamente na Figura 1.

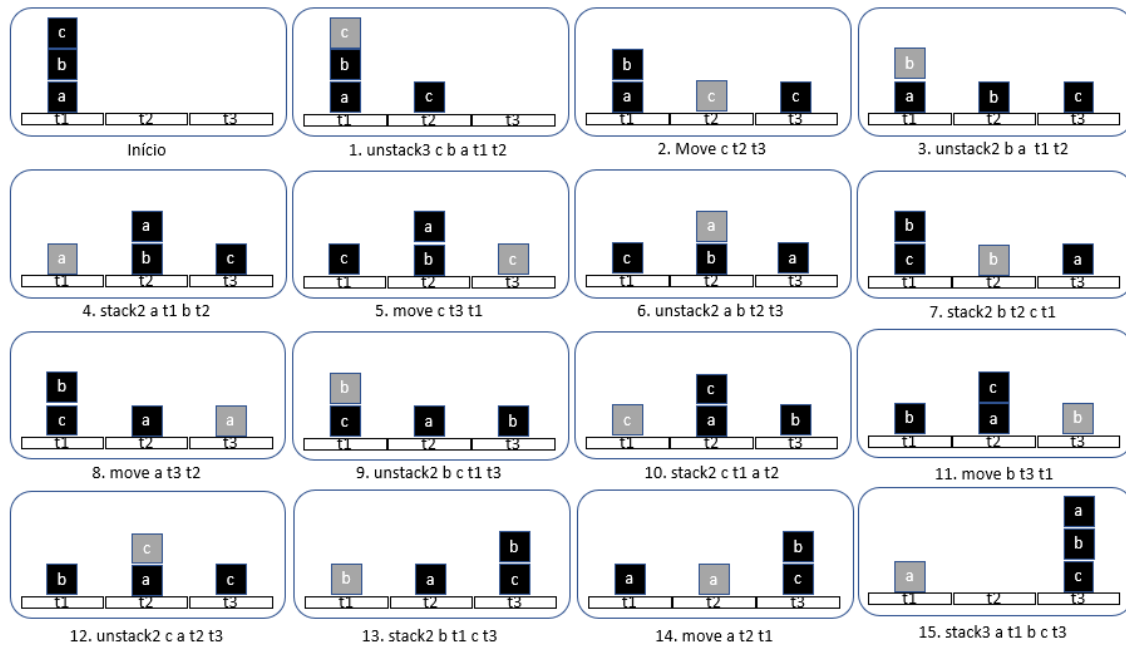


Figura 1 – Passo-a-passo da execução do plano 1

A Figura 2 apresenta o estado gerado a partir da execução da ação *stack2* que foi escolhido ao acaso (apenas como exemplo didático neste artigo) para representar um estado inválido. Assim supondo que para chegar ao estado objetivo é definida como restrição do problema do mundo dos blocos utilizado no trabalho, o estado objetivo do plano seja atingido sem passar por um estado em que o bloco *a* esteja sobre o bloco *b* e o bloco *b* sobre a mesa *t2*.

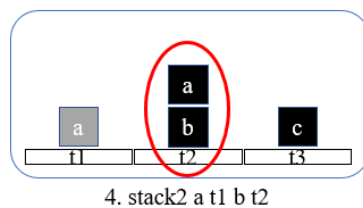


Figura 2 - Estado inválido

A execução do planejamento após a alteração da ação *stack2* (conforme apresentado na Listagem 3) é apresentado graficamente na Figura 3. O estado objetivo foi encontrado a partir de seis ações. As modificações demonstraram que o plano gerado encontrou o estado objetivo sem passar pelo estado definido como inválido.

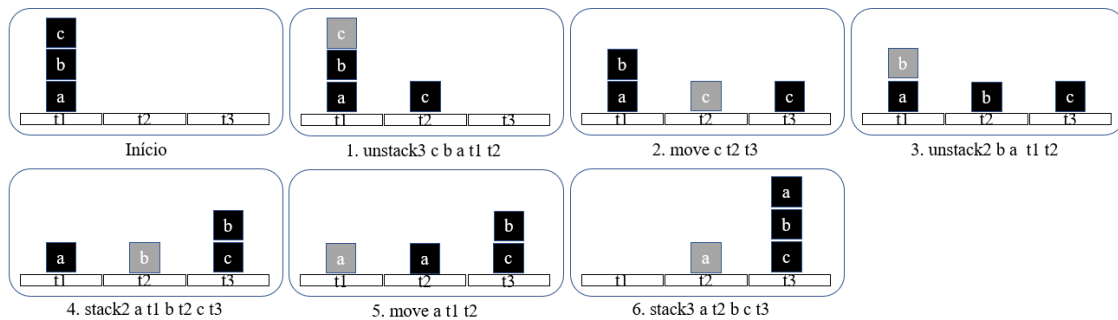


Figura 3 - Passo-a-passo da execução do plano 2

Ainda é possível observar que neste exemplo além de um estado inválido não estar contido no plano foi possível atingir o estado objetivo com um menor número de passos. Porém, acredita-se que este cenário não seja repetido para todos possíveis casos, assim pode haver situações onde um maior número de passos seja necessário para que um estado objetivo seja atingido após a alteração de ações do domínio de planejamento.

A Figura 4 apresenta a comparação entre o estado (B) gerado no primeiro plano e o estado (C) gerado após a alteração da ação *stack2* (Listagem 3). No domínio original a ação *stack2* considerava somente quatro parâmetros, sendo dois blocos e duas mesas, porém, o efeito era empilhar o bloco *a* sobre o bloco *b*, gerando o estado (B). Após a inclusão de novos parâmetros à ação, representando três blocos e três mesas, o efeito gerou o estado (C).

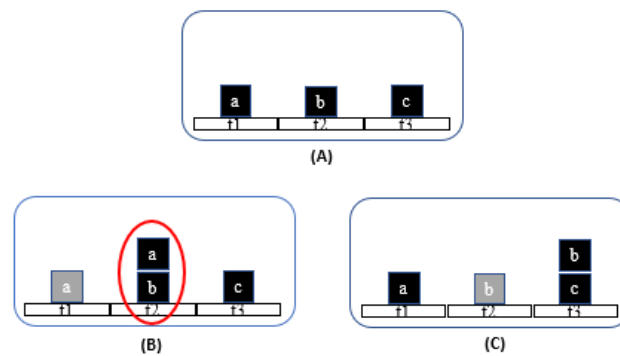


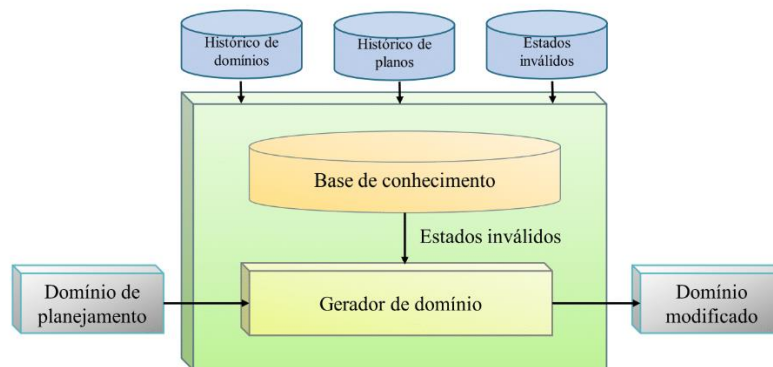
Figura 4 - Comparação dos estados antes e depois

O resultado obtido a partir das alterações feitas no domínio de planejamento, comprovou que modificar ações de um domínio pode impedir que estados inseguros façam parte dos planos gerados automaticamente. Portanto, uma estratégia para construção automática de domínios de planejamento, pode auxiliar na criação de planos com maior validade para o domínio.



A

Figura 5 apresenta a proposta de um sistema baseado em conhecimento para definição de restrições para a construção de domínios de planejamento. O gerador de domínios é o módulo responsável por modificar domínios de planejamento, adicionando parâmetros, condições e efeitos à definição das ações, a partir do conhecimento de estados inválidos obtido através de uma base de conhecimento.



**Figura 5 - Sistema baseado em conhecimento para definição de restrição de domínios de planejamento**

O sistema baseado em conhecimento define uma estratégia para a geração de domínios de planejamento usando predições feitas sobre estados inseguros às operações de satélites. Construir uma base de conhecimento com predição de estados inseguros e converter em restrições de domínio de planejamento é o objetivo desta estratégia.

A base de conhecimento é o módulo responsável por disponibilizar informações importantes sobre os estados do domínio. Os estados inválidos nas operações com satélites são desconhecidos devido à inúmeras quantidades de dados sobre a operação. A estratégia para validação de planos de operação de voo proposta por SOUZA demonstrou que é possível prever estados dos satélites usando técnicas de Mineração de Dados [SOUZA, 2011]. A classificação de estados inválidos pode identificar quando um estado é inseguro. Portanto, uma arquitetura composta por um sistema capaz de encontrar informações desconhecidas sobre domínio permitirá a criação de planos válidos sem que um plano seja rejeitado posteriormente.

Por que esta estratégia seria importante se um especialista já modelou o domínio com as restrições? A área de operação de satélites possui um histórico em que podem haver uma infinidade de estados inválidos desconhecidos que humanamente seria impossível de ser modelado, porém um sistema que saberá ler todo esse volume e traduzir em domínios válidos.

#### 4. Conclusão

A inclusão de novas restrições ao domínio de planejamento permite alterar ações de um plano gerado automaticamente. As mudanças no domínio de planejamento demonstraram que



é possível impedir que um plano contenha ações que gerem um estado inválido, por exemplo, possibilitando que o objetivo do plano seja alcançado sem a passagem por estados inválidos. Os problemas com planejamento automático na área espacial podem ser evitados quando restrições adequadas são incluídas ao domínio de planejamento. Um plano da área espacial é recusado quando ações inseguras à missão tiverem que ser executadas para que o objetivo do plano seja alcançado. Portanto, impedir que ações impróprias sejam executadas depende do conhecimento de restrições sobre o domínio.

A predição de estados dos satélites é importante para validar um plano, assim como, a predição de restrições de domínio pode ser o caminho para a criação de planos mais adequados no contexto da operação de satélites. Conclui-se que para gerar planos sem a passagem por estados inválidos é importante que as ações modeladas sobre o domínio de planejamento contenham precondições e efeitos condizentes apenas com estados válidos.

## **Referências**

- [KNOBLOCK, 1998] Knoblock, C., Barrett, A., Christianson, D., Friedman, M., Kwok, C., Golden, K., ... Weld, D. (1998). PDDL | The Planning Domain Definition Language 1 Introduction. *Most*.
- [SOUZA, 2011] Souza, Primavera Botelho De. (2011) A Classification Model to Generate Prognosis of Satellite.
- [TOMINAGA, 2011] Tominaga, J., Ferreira, M., Silva, J. (2011) A rule-based satellite simulator for use in flight operations planning.
- [BECKER, 2015] Becker, K. AI Planning with STRIPS. Copyright (c) 2015 Kory Becker disponível em <https://github.com/primaryobjects/strips>