

3. Planejamento como Verificação de Modelo

A abordagem de **Planejamento como Verificação de Modelo** (*Planning as Model Checking*) é uma metodologia poderosa que reformula o problema de encontrar um plano de ação como um problema de encontrar um caminho em um grafo de estados. O verificador de modelos, uma ferramenta originalmente desenvolvida para verificar a correção de sistemas de software ou hardware, é adaptado para buscar um "caminho de execução" que leve de um estado inicial a um estado objetivo.

3.1 O Princípio Central

A ideia principal é codificar o problema de planejamento em uma linguagem de descrição de modelo (como a linguagem SMV) e usar um verificador de modelos para provar ou refutar uma propriedade lógica. Se a propriedade for refutada, o verificador de modelos pode gerar um "contra-exemplo", que corresponde ao plano de ações que o planejador está procurando.

O processo de codificação envolve os seguintes passos:

- **Modelagem dos Estados:** Os estados do mundo são representados como um conjunto de variáveis que descrevem a configuração atual dos blocos. Por exemplo, a posição e a clareza de cada bloco.
- **Modelagem das Transições:** As transições entre os estados são definidas pelas ações possíveis (por exemplo, move(X, From, To)). As transições só são possíveis se as pré-condições da ação forem satisfeitas.
- **Especificação do Objetivo:** O objetivo é expresso como uma propriedade lógica na linguagem do verificador de modelos. A propriedade a ser verificada é a negação de que o estado objetivo é alcançável. Por exemplo, em uma linguagem como CTL (Computation Tree Logic), a propriedade seria !EF (Goal), que significa "não é possível que exista um caminho (*Exists a Path*) para o estado objetivo (*Future*)".

3.2 A Busca por um Plano

O verificador de modelos executa uma busca exaustiva no espaço de estados para determinar se a propriedade $!EF(\text{Goal})$ é verdadeira.

- **Se a propriedade for verdadeira**, significa que não há nenhum caminho possível do estado inicial para o estado objetivo. Isso indica que o objetivo é **inalcançável** ou que o modelo está incorreto.
- **Se a propriedade for falsa**, o verificador de modelos gera um **contra-exemplo**. Esse contra-exemplo é uma sequência de estados e transições que leva do estado inicial ao estado objetivo. Essa sequência de transições é, por definição, o plano de ações que resolve o problema.

3.3 Vantagens da Abordagem

O Planejamento como Verificação de Modelo oferece benefícios significativos, especialmente para problemas complexos:

- **Garantia de Solução**: Se um plano existe, o verificador de modelos o encontrará. O método é completo, ou seja, se a propriedade é refutável, um contra-exemplo será gerado.
- **Deteção de Impossibilidade**: Se o objetivo não puder ser alcançado, o verificador de modelos pode provar formalmente que a tarefa é impossível, o que é uma informação valiosa em domínios de planejamento.
- **Tratamento de Restrições**: As complexas restrições físicas e espaciais do mundo dos blocos de tamanho variável podem ser elegantemente codificadas como invariantes no modelo, garantindo que o planejador explore apenas estados e transições válidos.

Em resumo, a abordagem de Planejamento como Verificação de Modelo transforma um problema de busca em um problema de prova de lógica, aproveitando ferramentas poderosas para encontrar soluções e provar a impossibilidade de

objetivos.