

Python

```
if self.strategy.run_check() and ( # v & f
    len(new_states) > 1 and random.uniform(0, 1) <
args.pruning_factor
):
    new_states = [
        state
        for state in new_states
        if state.world_state.constraints.is_possible()
    ]
```

JUMPI é a instrução que gera ramificações (dois caminhos possíveis) quando a condição é verdadeira ou falsa. Ao encontrar um JUMPI, o Laser EVM cria mais de um estado (`new_states`), e então esse bloco é acionado para realizar a “poda” (pruning) desses novos estados.

Em outras palavras, JUMPI gera dois caminhos possíveis (pular ou não pular), e isso produz mais de um novo estado (`new_states`). Quando o número de novos estados é maior que 1, a condição de poda se torna relevante, por isso ela costuma ocorrer após a execução do JUMPI.

Isso significa que o código está filtrando os novos estados para manter apenas aqueles que são considerados possíveis, de acordo com as restrições definidas.