ANÁLISE DE ALGORITMOS

EXECUTOR SIMBÓLICO

Componentes

Rosialdo Vicente e
Venicius Jacob

Executor Simbólico

Definição: Técnica de teste e verificação de programas onde, em vez de fornecer entradas reais, são usados símbolos que representam valores arbitrários.

Funcionamento: As variáveis mantêm fórmulas simbólicas durante a execução, ao invés de valores concretos.

Ex:

```
int soma(int a, int b) {
   return a + b;
}
```

Teste tradicional: a = 3, b = 5, resultado = 8.

Execução simbólica: a = A, b = B, resultado = A + B.

EXECUÇÃO SIMBÓLICA

Instruções condicionais: Quando um if depende de variáveis simbólicas, a execução simbólica deve explorar ambos os caminhos (verdadeiro falso).

Ex:

```
if (a > b) {
    return a;
} else {
    return b;
}
```

Teste tradicional: a = 3, b = 5, resultado = 5.

Execução simbólica: a = A, b = B.

Forking (bifurcação):

1º caminho: Assume-se que A > B, resultado = A.

2º caminho: Assume-se que A ≤ B, resultado = B.

O programa bifurca em múltiplos ramos, gerando uma árvore de execução, com complexidade crescente conforme o número de ramificações.

Solver Z3

Z3: Um solver SMT baseado em SAT.

SAT: Resolve expressões booleanas (usando and, or, not) e determina se uma fórmula proposicional é satisfatível ou não.

SMT: Extensão do SAT que resolve problemas mais complexos, incluindo restrições aritméticas e expressões não booleanas.

Relação Execução simbólica x Biblioteca Z3

Execução simbólica: Atribui valores simbólicos às variáveis de entrada, gerando fórmulas.

Ex:
```
def soma(a, b):
    if a + b > 10:
        return True
    else:
        return False
```

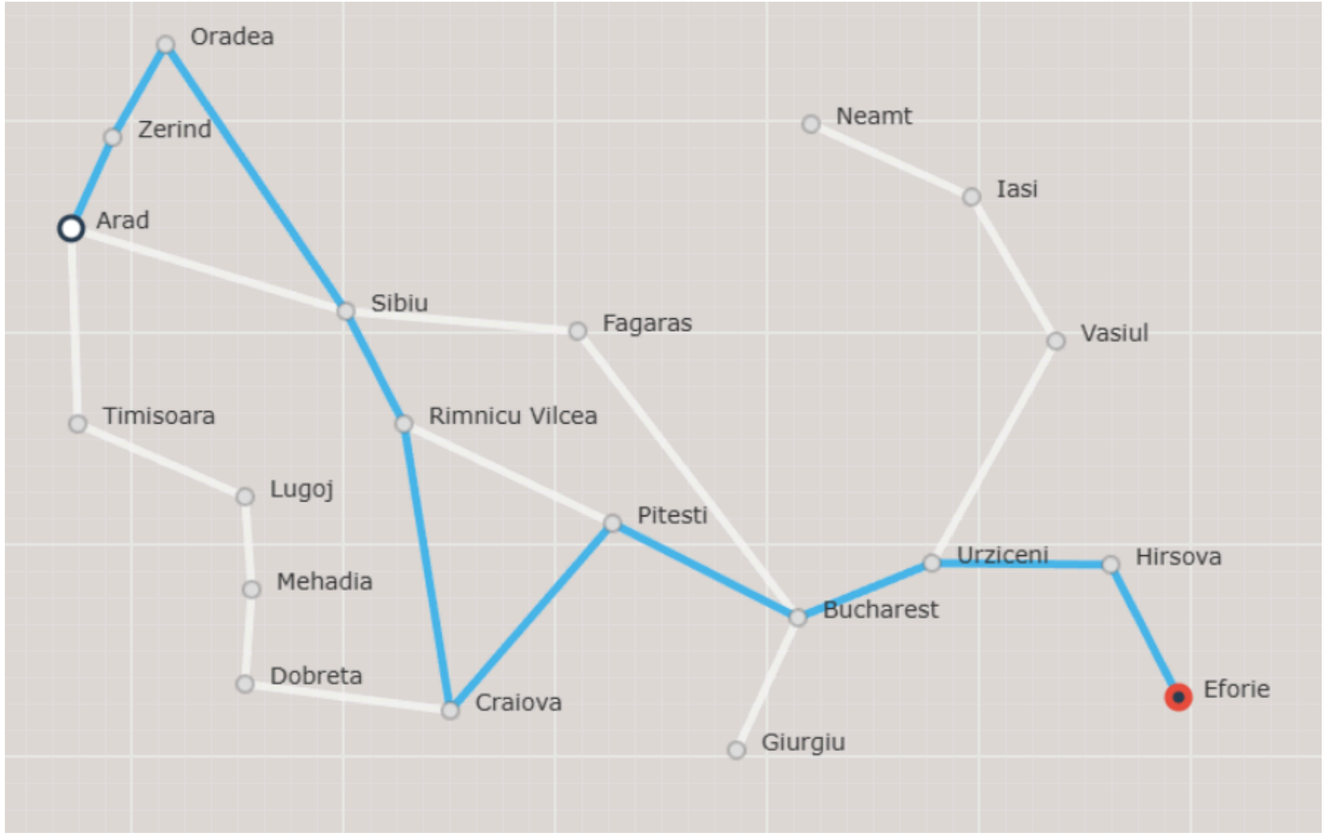
- Caminho 1 (if a + b > 10): fórmula simbólica A + B > 10.
- Caminho 2 (else): fórmula simbólica A + B ≤ 10.

Z3 (solver SMT): Recebe essas fórmulas e tenta determinar se existem valores concretos para A e B que as satisfaçam.

- Se satisfatível: Z3 retorna os valores.
- Se insatisfatível: Z3 indica que nenhum valor pode satisfazer a fórmula.

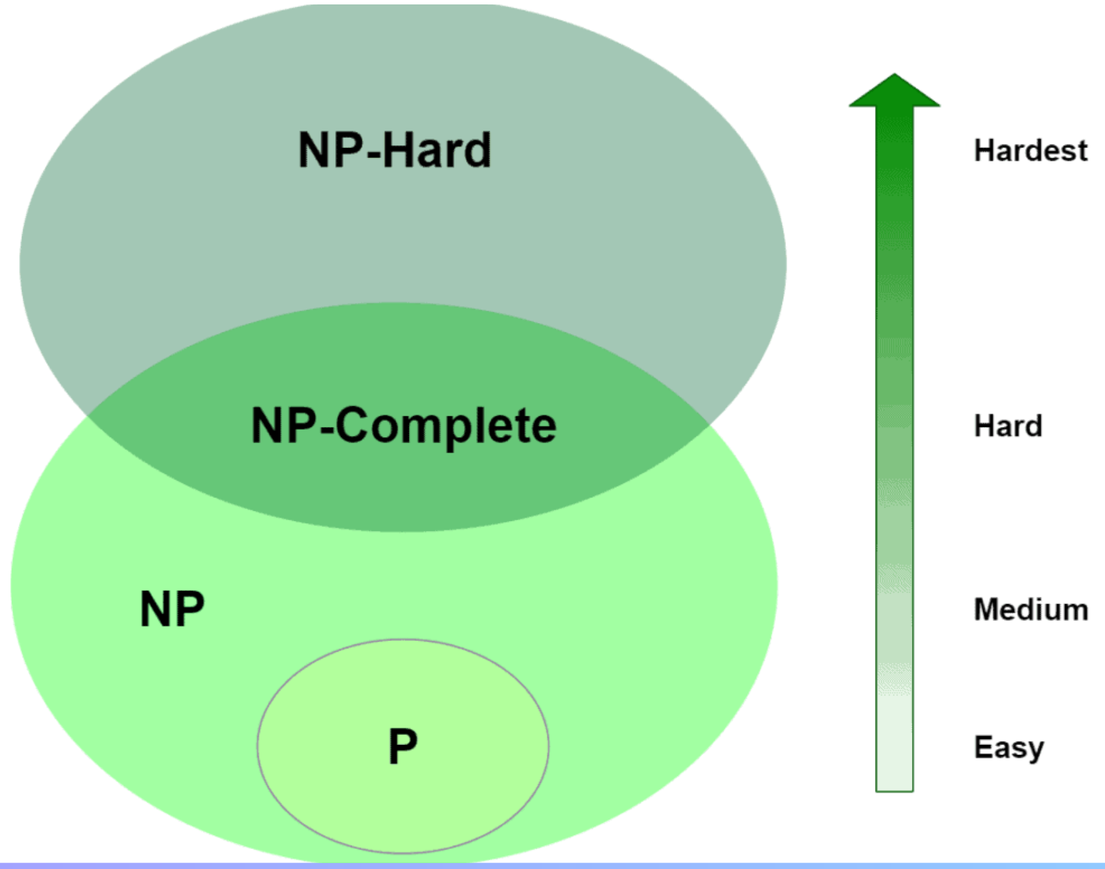
PCV (Problema do Caixeiro Viajante)

É um desafio clássico da ciência da computação onde o objetivo é encontrar o menor caminho possível que permita a um caixeiro visitar todas as cidades exatamente uma vez e retornar à cidade de origem. Formalmente, dado um conjunto de cidades e as distâncias entre cada par de cidades, o problema busca encontrar o ciclo hamiltoniano de menor custo.



Complexidade Principais Algoritimos usados no Z3

- SAT solver baseado no DPLL: NP-completo;
- E-Matching: NP-Difícil;
- Complexidade geral do código: O(2^n)



Criação das variáveis simbólicas

```
1   for i in range(n):
2               row = []
3               for j in range(n):
4                   var = Int(f'x[{i}][{j}]')
5                   row.append(var)
6               x.append(row)
7
```

Função objetivo: minimizar a distância total

```
objective_expr = 0
    for i in range(n):
        for j in range(n):
            objective_expr += distance_matrix[i][j] * x[i][j]
    solver.add(objective == objective_expr)
    solver.minimize(objective)
```

Verificando a solução e extraindo o caminho

```python
if solver.check() == sat:
        model = solver.model()
        print("Solução encontrada:")
        caminho = []
        cidade_atual = 0
        caminho.append(cidade_atual)
        for i in range(n - 1):
            for j in range(n):
                if model.evaluate(x[cidade_atual][j]) == 1:
                    caminho.append(j)
                    cidade_atual = j
                    break
        caminho.append(0)
        print(f"Caminho completo: {caminho}")
    else:
        print('Nenhuma solução encontrada.')
```

Análise dos Resultados

Matriz 4x4(Distâncias Variadas)

```
Teste 2:
Número de cidades: 4

Solução encontrada:
Caminho sugerido: [0, 1, 3, 2, 0]
Cálculo da soma das distâncias:
0 → 1: 10
1 → 3: 25
3 → 2: 30
2 → 0: 15
Soma total: 10 + 25 + 30 + 15 = 80
Verificação de outros caminhos possíveis:
[0, 1, 2, 3, 0]: 10 + 35 + 30 + 20 = 95
[0, 1, 3, 2, 0]: 10 + 25 + 30 + 15 = 80
[0, 2, 1, 3, 0]: 15 + 35 + 25 + 20 = 95
[0, 2, 3, 1, 0]: 15 + 30 + 25 + 10 = 80
[0, 3, 1, 2, 0]: 20 + 25 + 35 + 15 = 95
[0, 3, 2, 1, 0]: 20 + 30 + 35 + 10 = 95
O caminho sugerido de custo 80 é de fato o menor. Solução correta.
```

Análise dos Resultados

Matriz 4x4 (Caminhos Longos e Curtos)

```
Teste 4:
Número de cidades: 4

Solução encontrada:
Caminho sugerido: [0, 2, 3, 1, 0]
Cálculo da soma das distâncias:
0 → 2: 150
2 → 3: 90
3 → 1: 80
1 → 0: 100
Soma total: 150 + 90 + 80 + 100 = 420
Verificação de outros caminhos possíveis:
[0, 1, 2, 3, 0]: 100 + 120 + 90 + 200 = 510
[0, 1, 3, 2, 0]: 100 + 80 + 90 + 150 = 420
[0, 2, 1, 3, 0]: 150 + 120 + 80 + 200 = 550
[0, 2, 3, 1, 0]: 150 + 90 + 80 + 100 = 420
[0, 3, 1, 2, 0]: 200 + 80 + 120 + 150 = 550
[0, 3, 2, 1, 0]: 200 + 90 + 120 + 100 = 510
O caminho sugerido de custo 420 é de fato o menor. Solução correta.
```

Análise dos Resultados

Matriz 5x5 (Cidades Muito Próximas)

```
Teste 5:
Número de cidades: 5

Solução encontrada:
Caminho sugerido: [0, 1, 4, 3, 2, 0]
Cálculo da soma das distâncias:
0 → 1: 2
1 → 4: 4
4 → 3: 2
3 → 2: 2
2 → 0: 3
Soma total: 2 + 4 + 2 + 2 + 3 = 13
```

```
Verificação de outros caminhos possíveis:
[0, 1, 2, 3, 4, 0]: 2 + 2 + 2 + 2 + 5 = 13
[0, 1, 2, 4, 3, 0]: 2 + 2 + 3 + 2 + 4 = 13
[0, 1, 3, 2, 4, 0]: 2 + 3 + 2 + 3 + 5 = 15
[0, 1, 3, 4, 2, 0]: 2 + 3 + 2 + 3 + 3 = 13
[0, 1, 4, 2, 3, 0]: 2 + 4 + 3 + 2 + 4 = 15
[0, 1, 4, 3, 2, 0]: 2 + 4 + 2 + 2 + 3 = 13
[0, 2, 1, 3, 4, 0]: 3 + 2 + 3 + 2 + 5 = 15
[0, 2, 1, 4, 3, 0]: 3 + 2 + 4 + 2 + 4 = 15
[0, 2, 3, 1, 4, 0]: 3 + 2 + 3 + 4 + 5 = 17
[0, 2, 3, 4, 1, 0]: 3 + 2 + 2 + 4 + 2 = 13
[0, 2, 4, 1, 3, 0]: 3 + 3 + 4 + 3 + 4 = 17
[0, 2, 4, 3, 1, 0]: 3 + 3 + 2 + 3 + 2 = 13
[0, 3, 1, 2, 4, 0]: 4 + 3 + 2 + 3 + 5 = 17
[0, 3, 1, 4, 2, 0]: 4 + 3 + 4 + 3 + 3 = 17
[0, 3, 2, 1, 4, 0]: 4 + 2 + 2 + 4 + 5 = 17
[0, 3, 2, 4, 1, 0]: 4 + 2 + 3 + 4 + 2 = 15
[0, 3, 4, 1, 2, 0]: 4 + 2 + 4 + 2 + 3 = 15
[0, 3, 4, 2, 1, 0]: 4 + 2 + 3 + 2 + 2 = 13
[0, 4, 1, 2, 3, 0]: 5 + 4 + 2 + 2 + 4 = 17
[0, 4, 1, 3, 2, 0]: 5 + 4 + 3 + 2 + 3 = 17
[0, 4, 2, 1, 3, 0]: 5 + 3 + 2 + 3 + 4 = 17
[0, 4, 2, 3, 1, 0]: 5 + 3 + 2 + 3 + 2 = 15
[0, 4, 3, 1, 2, 0]: 5 + 2 + 3 + 2 + 3 = 15
[0, 4, 3, 2, 1, 0]: 5 + 2 + 2 + 2 + 2 = 13
O caminho sugerido de custo 13 é de fato o menor. Solução correta.
```

Análise dos Resultados

Matriz 5x5 (Grande Desigualdade nas Distâncias)

```
Teste 6:
Número de cidades: 5

Solução encontrada:
Caminho sugerido: [0, 4, 3, 2, 1, 0]
Cálculo da soma das distâncias:
0 → 4: 100
4 → 3: 1
3 → 2: 5
2 → 1: 10
1 → 0: 5
Soma total: 100 + 1 + 5 + 10 + 5 = 121
```

```
Verificação de outros caminhos possíveis:
[0, 1, 2, 3, 4, 0]: 5 + 10 + 5 + 1 + 100 = 121
[0, 1, 2, 4, 3, 0]: 5 + 10 + 5 + 1 + 100 = 121
[0, 1, 3, 2, 4, 0]: 5 + 10 + 5 + 5 + 100 = 125
[0, 1, 3, 4, 2, 0]: 5 + 10 + 1 + 5 + 100 = 121
[0, 1, 4, 2, 3, 0]: 5 + 10 + 5 + 5 + 100 = 125
[0, 1, 4, 3, 2, 0]: 5 + 10 + 1 + 5 + 100 = 121
[0, 2, 1, 3, 4, 0]: 100 + 10 + 10 + 1 + 100 = 221
[0, 2, 1, 4, 3, 0]: 100 + 10 + 10 + 1 + 100 = 221
[0, 2, 3, 1, 4, 0]: 100 + 5 + 10 + 10 + 100 = 225
[0, 2, 3, 4, 1, 0]: 100 + 5 + 1 + 10 + 5 = 121
[0, 2, 4, 1, 3, 0]: 100 + 5 + 10 + 10 + 100 = 225
[0, 2, 4, 3, 1, 0]: 100 + 5 + 1 + 10 + 5 = 121
[0, 3, 1, 2, 4, 0]: 100 + 10 + 10 + 5 + 100 = 225
[0, 3, 1, 4, 2, 0]: 100 + 10 + 10 + 5 + 100 = 225
[0, 3, 2, 1, 4, 0]: 100 + 5 + 10 + 10 + 100 = 225
[0, 3, 2, 4, 1, 0]: 100 + 5 + 5 + 10 + 5 = 125
[0, 3, 4, 1, 2, 0]: 100 + 1 + 10 + 10 + 100 = 221
[0, 3, 4, 2, 1, 0]: 100 + 1 + 5 + 10 + 5 = 121
[0, 4, 1, 2, 3, 0]: 100 + 10 + 10 + 5 + 100 = 225
[0, 4, 1, 3, 2, 0]: 100 + 10 + 10 + 5 + 100 = 225
[0, 4, 2, 1, 3, 0]: 100 + 5 + 10 + 10 + 100 = 225
[0, 4, 2, 3, 1, 0]: 100 + 5 + 5 + 10 + 5 = 125
[0, 4, 3, 1, 2, 0]: 100 + 1 + 10 + 10 + 100 = 221
[0, 4, 3, 2, 1, 0]: 100 + 1 + 5 + 10 + 5 = 121
O caminho sugerido de custo 121 é de fato o menor. Solução correta.
```

OBRIGADO PELA ATENÇÃO