

## Relatório de Execução – Autômato de Pilha – ‘aabbaaaa’

Afim de verificar a computação mostrada no relatório (computação\_manual\_PDA.pdf), foi adicionado no algoritmo PDA.py o AFP que reconhece a linguagem  $\mathbb{L}_4 = \{a^n b^m a^{n+m}; n, m \geq 0\}$ , e adicionada também a palavra a ser testada  $w = \text{'aabbaaaa'}$  como pode ser visto a seguir e no algoritmo PDA.py

```
print("PDA for {a^n b^m a^(n+m) }")
Sigma = {"a", "b"}
Q = {"q0", "q1", "q2", "qf"}
q0 = "q0"
F = {"qf"}
V = {"X"}
delta2 = {"q0": [("a", "epsilon", "X", "q0"),
                 ("epsilon", "epsilon", "epsilon", "q1")],
          "q1": [("b", "epsilon", "X", "q1"),
                 ("epsilon", "epsilon", "epsilon", "q2")],
          "q2": [("a", "X", "epsilon", "q2"),
                 ("?", "?", "epsilon", "qf")]}
M = (Sigma, Q, delta2, q0, F, V)
w = "aabbaaaa"
pp.pprint(lifted_delta_clos([(w, "q0", [])], delta2))
```

a execução deste trecho resultou na saída mostrada a seguir

```
[('aabbaaaa', 'q0', []),
 ('abbaaaa', 'q0', ['X']), ('aabbaaaa', 'q1', []),
 ('bbaaaa', 'q0', ['X', 'X']), ('abbaaaa', 'q1', ['X']), ('aabbaaaa', 'q2', []),
 ('bbaaaa', 'q1', ['X', 'X']), ('abbaaaa', 'q2', ['X']),
 ('baaaa', 'q1', ['X', 'X', 'X']), ('bbaaaa', 'q2', ['X', 'X']), ('bbaaaa', 'q2', []),
 ('aaaa', 'q1', ['X', 'X', 'X', 'X']), ('baaaa', 'q2', ['X', 'X', 'X']),
 ('aaaa', 'q2', ['X', 'X', 'X', 'X']),
 ('aaa', 'q2', ['X', 'X', 'X']),
 ('aa', 'q2', ['X', 'X']),
 ('a', 'q2', ['X']),
 ('', 'q2', []),
 ('', 'qf', [])]
```

Process finished with exit code 0

Esta saída que é idêntica a árvore de computações mostrada no relatório  
computação\_manual\_PDA.pdf