

Trabalho 3 - Especificação Formal de Software

Entregua: 07/Dezembro

Cada grupo pode ter quantos membros quiser, mas nenhum membro do grupo atual pode ter feito um dos trabalhos anteriores juntos.

for distinct students s_1, s_2 , $\text{grp}(s_1, t_1) == \text{grp}(s_2, t_1) \parallel \text{grp}(s_1, t_2) == \text{grp}(s_2, t_2) \rightarrow \text{grp}(s_1, t_3) \neq \text{grp}(s_2, t_3)$, where $\text{grp}(s, t)$ is the group of s in task t .

Faça um programa, em qualquer linguagem, que receba como entrada uma MEF no formato abaixo e um conjunto de sequencias de entrada, e informe se o conjunto é n-completo ou não. Entregue todos os fontes.

mef.txt

```
1 -- a/0 -> 2
1 -- b/0 -> 3
2 -- a/0 -> 2
2 -- b/0 -> 3
3 -- a/0 -> 3
3 -- b/0 -> 1
```

seqs1.txt

```
aaa
aab
abaab
bbabbbb
```

seqs2.txt

```
aaaaab
abaab
baab
bbabbbb
```

```
> check-n-complete mef.txt seqs1.txt
the set is n-complete
```

```
> check-n-complete mef.txt seqs2.txt
the set is not n-complete
```

Referência

Adenilso Simao, Alexandre Petrenko: Checking Completeness of Tests for Finite State Machines. IEEE Trans. Computers 59(8): 1023-1032 (2010)