5,1

Leandro Ribeiro Ritter

(0,3) 4) a) Que há sofá e estantes nas lojas e não verdes ✗

4) b) Propriedade de segurança: sofá e estante não verdes

Propriedade de vivacidade: há sofá e estante verde ✗

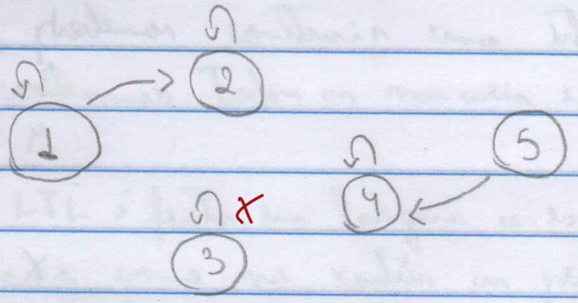
4) c) A propriedade de segurança pode ser escrita como invariante, pois só precisamos observar o estado onde o sofá e a estante não verdes.

A propriedade de vivacidade pode ser escrita como Temporal, pois queremos saber se em momentos distintos há sofá e estantes verdes.

4) d) Que existe sofá e estantes verde nas lojas, ✗ que irão ✓ reservá-los se houver e a tonalidade dos verdes são os mesmos. ✓

6.) Entradas podem ser casos que você quer precisar em um software e o resultado pode ser um exemplo onde um software atingiu um caso e como foi atingido (Passo a Passo).

(2,7) 3) a)



S : ?

→ : ?

I : ?

.5

3) b) Não é determinística, pois $\underbrace{\text{múltiplos}}_{\delta}$ estados podem alcançar mais de 1 estado. .4

3c) Sim, é possível encontrar todas as possíveis execuções .15

3) d) 2, 3 ✗ e 4, pois após chegar neles não é possível sair ✓

3) e) Ser igual a 5, ser menor que 3 ou/e ser menor que 5 ✓

3)f)
```
module BLA {
        ⋮
    action step = any {
            ⋮
        all {
            x == 2,
            nondet a = Set(1,5).oneOf(),
        } x' = a,
        }
}
```
Set(1,2,4,5)

.9

(0,8) 2) Satisfaz | Não Satisfaz

Faltou justificar

a)

X	Y	X	Y
1	6	1	3
2	8	2	1
3	9	-1	7

Seus exemplos foram bons o suficiente (específicos) para demonstrar que sabia as justificativas

b)

X	Y	X	Y
0	5	1	2
1	0	3	0
0	8	4	7

c)

X	Y	X	Y
0	0	2	1
1	3	3	0
7	9	7	5

d)

X	Y	X	Y
1	2	1	7
1	5	3	4
3	1	0	1

2) Como serão feitos milhares de execuções pode ser que conseguimos satisfazer as nossas formulas mas também há casos que podem não satisfazê-las. ?

(1,2) 1) b) jogador1NPerde = dado1 ≥ dado2, não é satisfeita pois é possível em algum momento da execução encontrar um contra-exemplo. ✓ ✗ .3

Justificativa genérica

```
1) a) Status = if dado1 > dado2 Vencem(dado1)        ✓ + .4
       else if dado2 > dado1 Vencem(dado2)
       else Pendente
```

d1 > d2

d2 > d1

1) c) Satisfeita: $\underbrace{\text{VPE}}_{?}$ = (Status == Vencem(dado1)) or (Status == Vencem(dado2)) or (Status == Pendente) ✓

Não satisfeita: jogador2NPerde = dado2 ≥ dado1

1) d) jogador1vence = eventually(Status == Vencem(dado1)) ✓ + .2

não é satisfeita

1) e) Sim, podemos construir uma estrutura de Kripke para ela pois conhecemos todas as variáveis e estados possíveis. ✓ ✗ .2

(0,1) 5) b) V,✗ em LTL é feita em largura a execução de cada ramo da árvore e como no xadrez as peças brancas são sempre as primeiras logo em toda a primeira ramificação da árvore será o movimento da peça branca.

5) d) V,✓ como foi dito as primeiras jogadas é dos peões ou cavalos logo todas as primeiras ramificações da árvore será com peão ou cavalos. ✗ Não há árvores em LTL

5) a) F✓

5) c) F✗