

Dicas Trabalho 1

Aula para disciplina de Métodos Formais

Gabriela Moreira

Departamento de Ciência da Computação - DCC
Universidade do Estado de Santa Catarina - UDESC

24 de abril de 2024

Conteúdo



Dicas

Outline



Dicas

[Quint] q::debug

Se você está conseguindo executar (simular) sua especificação, mas não entende **porque** algo está se comportando de determinada maneira, pode ser útil usar o operador auxiliar `q::debug`. Por exemplo, considere a função:

```
pure def f(x) = (x + 1) * 2
```

Se tivermos dúvida se nossa soma está correta, podemos envolvê-la em uma chamada de `q::debug`, onde o primeiro argumento é uma string qualquer e o segundo a expressão a ser avaliada. O resultado de `q::debug` é exatamente igual à expressão em si, mas ele tem um efeito colateral que resulta na impressão daquele valor precedido pela string fornecida:

```
>>> pure def f(x) = q::debug("resultado soma: ", x +  
    1) * 2  
>>> f(1)  
> resultado soma:  2  
4
```

Isso é o mais próximo que temos de `printf` para debugar :)

[TLA+] Print

De forma semelhante, temos o operador `Print` em TLA+, definido no módulo TLC

`EXTENDS` TLC

Usado da mesma forma:

```
otherTeam(team) == Print("otherTeam: ", IF team = "A"  
    THEN "B" ELSE "A")
```

No VSCode, no painel do TLC teremos uma seção “Output” mostrando:

```
"otherTeam: " "A" (6)  
"otherTeam: " "B" (2)  
"otherTeam: " "A" (4)  
"otherTeam: " "B" (4)  
"otherTeam: " "A" (2)  
...
```

Composição

E se, no mesmo turno, um pokemon puder dar dois ataques em vez de somente um?



Composição

E se, no mesmo turno, um pokemon puder dar dois ataques em vez de somente um?

Lembrem-se que não podemos atualizar uma variável duas vezes na mesma ação (transição).

Composição

E se, no mesmo turno, um pokemon puder dar dois ataques em vez de somente um?

Lembrem-se que não podemos atualizar uma variável duas vezes na mesma ação (transição).

Temos duas opções:

- 1 Usar duas ações e só atualizar `next_team` na segunda
- 2 Usar uma unica ação, onde só atualizamos `pokemons` com o resultado do segundo ataque.

Composição

E se, no mesmo turno, um pokemon puder dar dois ataques em vez de somente um?

Lembrem-se que não podemos atualizar uma variável duas vezes na mesma ação (transição).

Temos duas opções:

- ① Usar duas ações e só atualizar `next_team` na segunda
- ② Usar uma unica ação, onde só atualizamos `pokemons` com o resultado do segundo ataque.

A opção (2) é geralmente mais interessante, já que ela funciona melhor se, por exemplo, quisermos adicionar um terceiro ataque.

[Exemplo completo no moodle]

```
type AttackDescription = { attacker: Pokemon, receiver
    : Pokemon, attack: str, damage: int }
var last_attack: Option[List[AttackDescription]]

pure def tackle(attacker: Pokemon, receiver: Pokemon):
    (Pokemon, AttackDescription) = {
        (receiver.damage(10), { attacker: attacker, receiver
```