

LINGPON 2.0 COM SUPORTE A VETORES

Filipe Lautert, (A. F. Ronszcka), [J. A. Fabro, J. M. Simão]

Requisitos

- Suportar vetores de Atributos
- Suportar vetores de FBEs
- Vetores com valores de índices dinâmicos (vetor[i]) e estáticos (vetor[3])
- Nova estrutura: formation_rule
- Validação de índices dos vetores (impedir uso de índices inexistentes)

Implementação transparente

- Cada elemento do vetor é convertido em um atributo/instância:
bla__vector_0, bla__vector_1... bla_vector_9
- O objetivo de criar como atributos é ser transparente para os geradores de código existentes
- Acessos diretos aos atributos/instâncias deve ser convertida na hora de adicionar ao grafo: bla[1] vira bla__vector_1
- Cada formation rule é convertida em uma nova rule para cada combinação de valores dos índices -> e todos os seus “filhos” também (action, condition, etc)

Declaração de vetores

- `public Alarm[2] sensorVector`
- `public Alarm[5] alarmVector`
- `public boolean[2] xAlarm`
- `public boolean[2] jAlarm = {true, false}` #Implementação sendo finalizada

Utilização de vetores em um method

```
private method mtSendSms
  attribution
  this.jAlarm[1] = true
  end_attribution
end_method
```

Utilização de vetores em uma rule

```
rule rlNormal
  condition
    premise prThird
      this.jAlarm[1] == true
    end_premise
  or
    premise prOne
      sensorVector[0].atStatus == alarmVector[0].atStatus
    end_premise
```

Estrutura gerada na rule

```
# Rule rInNormal
```

```
# Condition cdUnnamed3 [OR]:
```

```
# Premise: prOne
```

```
ELEMENT (sensorVector__vector_0.atStatus) EQUAL ELEMENT (alarmVector__vector_0.atStatus)
```

```
# Premise: prThird
```

```
ELEMENT (this.jAlarm__vector_1) EQUAL BOOLEAN (1)
```

```
# Action fora [PARALLEL]:
```

```
# Instigation inUnnamed4 [PARALLEL]:
```

```
# Call [this.mtSendSms]:
```

Formation rule

```
formation_rule rIFireAlarm
  index i from 0 to 1
  index s from 0 to 1
  condition
    premise prOne
      sensorVector[i].atStatus == alarmVector[s].atStatus
    end_premise
  or
    premise prSecondFRule
      this.jAlarm[i] == true
    end_premise
  end_condition
```


Estrutura gerada na formation_rule

```
# Rule rIFireAlarm_i_0_s_0
# Condition rIFireAlarm_i_0_s_0_cdUnnamed1 [OR]:
# Premise: rIFireAlarm_i_0_s_0_cdUnnamed1_prOne

ELEMENT (sensorVector__vector_0.atStatus) EQUAL ELEMENT (alarmVector__vector_0.atStatus)
# Premise: rIFireAlarm_i_0_s_0_cdUnnamed1_prSecondFRule
ELEMENT (this.jAlarm__vector_0) EQUAL BOOLEAN (1)

# Action rIFireAlarm_i_0_s_0_dentro [PARALLEL]:
# Instigation rIFireAlarm_i_0_s_0 [PARALLEL]:
# Call [this.mtSendSms]:
```

Estrutura gerada na formation_rule

```
# Rule rFireAlarm_i_0_s_1
# Condition rFireAlarm_i_0_s_1_cdUnnamed1 [OR]:
# Premise: rFireAlarm_i_0_s_1_cdUnnamed1_prOne

ELEMENT (sensorVector__vector_0.atStatus) EQUAL ELEMENT (alarmVector__vector_1.atStatus)
# Premise: rFireAlarm_i_0_s_1_cdUnnamed1_prSecondFRule
ELEMENT (this.jAlarm__vector_0) EQUAL BOOLEAN (1)

# Action rFireAlarm_i_0_s_1_dentro [PARALLEL]:
# Instigation rFireAlarm_i_0_s_1 [PARALLEL]:
# Call [this.mtSendSms]:
```

Alterações necessárias

- Alterações no byson.y e lex.l (formation_rule, [,], etc)
- Novas estruturas de dados: VectorElementFactor, VectorDto utilizadas na estrutura interna para geração do Graph Pon
- Mais de 250 linhas até o momento na NOPGraph.cpp (original de ~1050)
- Alterações necessárias (novos métodos) específicos para array por causa do uso de this->current....

Alterações necessárias

```
Action* NOPGraph::createAction(std::string name, Execution *execution) {  
  
    if (name == "") {  
        name = this->generateEntityName("ac");  
    }  
  
    this->currentAction = new Action(name, execution, this->instigations);  
  
    this->instigations = new std::map<std::string, Instigation*>();  
  
    return this->currentAction;  
  
}
```

Alterações necessárias

```
Action* NOPGraph::createVectorAction(std::string name, Execution *execution, std::map<std::string, Instigation*> *instigations) {  
  
    if (name == "") {  
        name = this->generateEntityName("ac");  
    }  
  
    Action* action = new Action(name, execution, instigations);  
  
    return action;  
  
}
```

Dúvidas