

# Relatório 4<sup>a</sup> fase

César Gasparini, Daniel Nunes e Artur Santos  
10297630, 10297612, 10297734

13 de dezembro de 2017

## Introdução

Na quarta fase do projeto, construímos os novos comandos da linguagem de alto nível que comandará cada robô, assim como corrigimos eventuais bugs de versões anteriores.

## Novos comandos

### Direções

#### Flex

Para implementar as direções no flex, adicionamos as palavras reservadas: **northeast**, **east**, **southeast**, **southwest**, **west**, **northwest** e **current**. Todas elas retornam o token DIRECAO, mas com o valor (yylval.val) correspondendo ao enumerador de sua direção, por exemplo:

```
northeast  yylval.val = NORTHEAST; return DIRECAO;
```

onde **northeast** corresponde à palavra reservada, **NORTHEAST** ao enumerador e **DIRECAO** ao token.

### Bison

Para implementar a direção no bison, adicionamos apenas a linha

```
%token < val > ATRIBUTO
```

Ou seja, apenas o token DIRECAO que possui um valor associado.

## Ações do Robô

- **Flex**

Para implementar as ações do robô no flex, adicionamos as palavras reservadas: **mova**, **ataque**, **colete**, **deposite**. Elas retornam, respectivamente, os tokens, **MOVA**, **ATAQUE**, **COLETE**, **DEPOSITE**.

- **Bison**

Adicionamos os tokens: **MOVA**, **ATAQUE**, **COLETE**, **DEPOSITE**. Cada comando de mover é tratado como uma expressão e deve estar da forma **AÇÃO OPEN DIRECAO CLOSE**, onde ação é um dos tokens a cima. Ou seja, caso se queira verificar se o robô conseguiu se mover para a direita, basta colocar no código:

```
if (mova( east ) == 1)
```

Se quiser guardar em uma variável se a ação foi bem sucedida, basta fazer

```
x = mova( east )
```

Ao compilar esses comandos de ações, o compilador apenas adiciona a instrução correspondente, por exemplo, se ao compilar **move(northeast)**, o compilador adicionará a instrução **MOV 0**.

*Nota: Essas ações não podem ser tratadas apenas como chamadas de função, pois retornam um valor, por isso tratamos elas como expressões.*

## Informação da Célula

- **Flex**

Para implementar informação da célula no flex, adicionamos a palavra reservada: **informacao**.

Ela retorna o token **INFORMACAO**.

- **Bison**

Adicionamos o token: **INFORMACAO**. Seu uso e funcionamento é análogo aos das ações.

## Atributos da Variável

- **Flex**

Para implementar os atributos da variável no flex, adicionamos as pa-

lavras reservadas **terreno**, **vazia**, **nCristais**, **base**, que retornam o token **ATRIBUTO**, mas com valor igual a **0**, **1**, **2**, **3**, respectivamente. Adicionamos também a palavra reservada **”.**”, que retorna o token **PONTO**

- **Bison**

No bison, adicionamos o token **ATRIBUTO** e **PONTO**. Para utilizar o atributo, tratamos **ID PONTO ATRIBUTO** como uma expressão, logo sua utilização é parecida com o das ações. Um exemplo:

```
x = informacao ( east );  
if ( x.nCristais > 0 ) {  
    y = colete ( east );  
}
```

O que esse trecho faz é verificar se a célula à direita possui cristais e, se possuir, coleta 1 cristal.

O seu funcionamento é simples, quando compilado, o compilador adiciona a instrução que coloca na pilha a variável que está sendo requisitada e coloca também a instrução **ATR** com o valor do atributo.

## Implementação do comando *else*:

O **else** foi implementado utilizando as instruções **JIF** e **JMP**, conforme pode ser visto no código. Um **JIF** é direcionado a um **JMP**, e isto é feito por meio do empilhamento dos valores do **ip** em uma pilha auxiliar. O **JMP** redireciona para o **ip**.

Ou seja, abrangimos os seguintes casos:

- o **if** é verdadeiro, ou é falso;
- o **if** é verdadeiro e possui um **else** atrelado
- o **if** é falso e possui um **else** atrelado - o qual o deve executar.

Dentro do **if**, o **JMP** já direciona para fora do laço do **if**, e isso quando nosso **if** é falso e devemos executar o **else**. Agora no caso do **if** ser verdadeiro, o que é feito no programa é que dentro da definição do **else**, mudamos o **ip** no qual o **JMP** deve ser feito, sendo ele direcionado para fora do laço do **else**, fazendo com que o funcionamento esteja correto. Vale lembrar que a sintaxe permite **else if** mas com a seguinte escrita:

```
if ( expr. ) { Bloco } else { if ( expr. ) { Bloco } }
```