

Seminário de Compiladores 2

Grupo: Luazinha

Componentes:

Breno Silveira, 551481

Douglas Barbino, 551511

Gabriela Mattos, 551570

Thiago Casaes, 551880



Apresentação Geral da Linguagem

- Linguagem de programação que terá como base a linguagem LA, adicionando as funcionalidades para uso dos seguintes dispositivos:
 - LED;
 - Buzzer;
 - LCD;
 - Botão;
 - Sensor de toque;
 - Potenciômetro;
- A geração de código será feita para gerar um algoritmo em Arduino - sketch.

Apresentação da gramática final

```
programa      : declaracoes 'comando_setup' comandosSetup 'fim_comando_setup'
               'comando_loop' comandos 'fim_comando_loop'
               ;
```

```
cmd           : 'leia' '(' identificador mais_ident ')'
               [...]
               | comandoLoop
               | 'retorne' expressao
               ;
```

```
comandosSetup      : (comandoSetup)+
                    ;
```

```
comandoSetup       : 'ativar' '(' dispositivo ',' pino ')'
                    | declaracao_local
```

```
comandoLoop        : (cmdLoop)+
                    ;
```

Descrição da gramática final

```
cmdLoop : ('ligar' | 'desligar') '(' dispositivoSaida ',' pino (',' volt)? ')'
        | IDENT '<-' 'ler' '(' dispositivoEntrada ',' pino ')'
        | 'esperar' '(' tempo ')'
        | comandoLCD
        ;
```

```
dispositivo : dispositivoSaida
            | dispositivoEntrada
            | lcd
            ;
```

```
dispositivoSaida: 'luz'
                | 'led'
                | 'som'
                ;
```

Descrição da gramática final

```
dispositivoEntrada: 'botao'  
                  | 'sensortoque'  
                  | 'potenciometro'  
                  ;
```

```
comandoLCD        : 'definirCor' '(' lcd ',' pino ',' cor ')'  
                  | 'escrever' '(' lcd ',' pino ',' (CADEIA | IDENT) ')'  
                  ;
```

```
lcd               : 'LCD'  
                  | 'lcd'  
                  | 'tela'  
                  ;
```

Descrição da gramática final

```
pino      : NUM_INT  
          | IDENT  
          ;
```

```
tempo     : NUM_INT  
          | IDENT  
          ;
```

```
volt      : NUM_INT  
          | IDENT  
          ;
```

```
cor       : '(' NUM_INT ',' NUM_INT ',' NUM_INT ')'  
          ;
```

Descrição da análise semântica

- Utilizado o Parse Tree Listener do Antlr
- Regras semânticas:
 - 1 - Erros verificados no ativar:
 - Porta utilizada não ser analógica/digital, depende do dispositivo;
 - 0 a 10 são portas digitais (led, som, luz, botão)
 - 3, 5, 3, 6 são portas PWM (led, som, luz, botão)
 - 10 a 20 portas analógicas (potenciômetro, sensor de toque)
 - 21 porta I2C (usada para o LCD)
 - Portas 'ativar' repetidas;
 - 2 - Verificação de volt: entre 0 e 255
 - definição de volt só pode acontecer em portas analógicas
 - 3 - Verificação das cores: entre 0 e 255
 - 4 - Verificação se a porta utilizada nos comandos loop é a mesma declarada no setup com o mesmo dispositivo
 - 5 - Verificação de uso de portas repetidas

Descrição da análise semântica

- Características:
 - Entrada da tabela de símbolos com novos campos para representar a porta (pino)
 - Valor (número da porta), Dispositivo (associado à essa porta declarada).
 - Obs: Valor também utilizado para constantes
 - Todos os elementos do loop devem possuir equivalente na Tabela de Símbolos do Setup

Descrição do gerador de código:

- Realizado através de sobrescrever os métodos encontrados no BaseListener;
- É dividida em três principais pontos:
 - Comandos antes da função setup(), ou seja, declarações de constantes e determinadas verificações que afetam o código gerado (utilização do dispositivo LCD e comandos de leitura e escrita por meio da porta serial);
 - Comandos da função setup(), que consiste em ativações de portas e inicializações do LCD e da porta serial, caso necessário;
 - Comandos da função loop(), que consiste em vários comandos do Arduino, como por exemplo entradas e saídas por meio dos dispositivos, ler e escrever por meio da porta serial, comandos do LCD, dentre outros.
- Vale destacar que para realizar as verificações necessárias antes da função setup(), é necessário verificar o código à frente em busca de determinados comandos (ativação do LCD na função setup() e algum comando leia ou escreva dentro da regra sintática *comandos*)

Exemplo 1:

```
constante    pinSom:    inteiro    =    5

constante BREATH_DELAY: inteiro = 5

comando_setup

    ativar(led, pinSom)

fim_comando_setup

comando_loop

    para i de 0 ate 256 faca
        ligar(led, pinSom, i)
        esperar(BREATH_DELAY)
    fimpara
    esperar(100)

fim_comando_loop
```

```
const int pinSom = 5;
const int BREATH_DELAY = 5;
void setup()
{
    pinMode(pinSom, OUTPUT);
}
void loop()
{
    for(int i=0; i<256; i++)
    {
        analogWrite(pinSom, i);
        delay(BREATH_DELAY);
    }
    delay(100);
}
```

Exemplo 2:

```
[...]  
comando_setup  
    ativar(potenciometro, pinoPot)  
    ativar(lcd, pinoLCD)  
fim_comando_setup  
comando_loop  
    y <- ler(sensortoque, pinoSensor)  
    definirCor(lcd, pinoLCD, (10, 10, 300))  
    imprimeLCD()  
    esperar(tempo)  
fim_comando_loop  
[...]
```

Linha 17: cor definida incorretamente

Dica: a cor deve ser definida da seguinte forma (a, b, c), onde a corresponde ao R (de 0 a 255), o b ao B (de 0 a 255), e o c ao G (de 0 a 255) do sistema de cores RGB

Fim da compilacao

Exemplo 3:

- Abrir arquivo: exemplo5.faz
- `java -jar trabalho3.jar [caminho] casosDeTeste/semErros/entrada/exemplo5.faz [caminho]/resultados/exemplo5.ino`
- Abrir Arduino;
- Copiar conteúdo do arquivo exemplo5.ino e colar na IDE do Arduino;
- Fazer upload na placa.

Exemplo 4:

- Abrir arquivo: exemplo6.faz
- `java -jar trabalho3.jar [caminho] casosDeTeste/semErros/entrada/exemplo6.faz [caminho]/resultados/exemplo6.ino`

- Abrir Arduino;
- Copiar conteúdo do arquivo exemplo6.ino e colar na IDE do Arduino;
- Fazer upload na placa e abrir monitor serial.

Dificuldades encontradas:

- Como definir uma linguagem de “fácil” entendimento;
- Quais dispositivos são mais relevantes;
- Geração de código com separação de setup e loop;
- Definição de funções;
- Mensagens de aviso.

Referências

- Introdução às tecnologias para desenvolvimento de aplicações em plataformas para IoT. Disponível em: <<https://bytebucket.org/dc-ufscar/iot/raw/master/doc/apostila/iot.pdf>>
- Linguagem LA. Disponível em: Construção de Algoritmos. Autor: Jander Moreira.