

Recomendaciones ANTLR

Procesadores del Lenguaje
DLSI - Universidad de Alicante

David Rizo Valero
José Manuel Iñesta Quereda
Francisco Moreno Seco

Componentes

- Lexer
- Parser
- Genera código java

Ejemplo Inicial

- Gramática prueba.g
- Generará
 - pruebaLexer.java
 - pruebaParser.java
 - prueba.tokens

Imports léxico

```
@lexer::header {  
package mipaquete; // no poner en esta práctica  
    import java.util.Vector;  
    import otropaquete.OtraClase;  
}
```

Sobrecargas léxico

```
@lexer::members {  
    public void emitErrorMessage(String msg) {  
        throw new RuntimeException(msg);  
    }  
}
```

Sintáctico

- `@parser::header {...}`
- `@parser::members {...}`

Captura de errores

```
@rulecatch {  
    catch (RecognitionException re) {  
        reportError(re);  
        System.exit(1);  
    }  
}
```

Léxico

- Terminales en mayúscula:

ID : ('a'..'z'|'A'..'Z')('a'..'z'|'A'..'Z'|'0'..'9')*

ENTERO : ('0'..'9')+

NEWLINE : ('\r'? '\n'| '|' '\t')+ {skip();}

CÓMENTARIO
: '/*' .* '*/' {skip();}
;

Sintáctico y trad.

- Noterms en minúscula:

noterm[Tipo param_entrada, int otroParam]

returns [Tipo param_salida, String traduccion, int otroReturn]

@init {

código inicial de la noterm

}

:

otronoterm {

código java

accedemos a los parámetros: **\$param_entrada**

\$otronoterm.value es el valor de retorno de otronoterm

\$ID.text es el lexema del terminal **ID** que es del tipo string

\$traduccion = \$ID.text + "(" + \$otronoterm.value + ");"; // por ejemplo

}

|

otronotermmas

{separamos las subnoterms con el pipe

pasamos parametros con corchetes: otronotermmas[tablasimbolos]

\$traduccion = "loquesea"; // hacemos lo mismo con los otros parámetros de salida

}

;

Sintáctico y trad.

- Cuando hay dos apariciones de la misma noterm:

noterm

:

o1=oteronotermcualquier cosa o2=oteronoterm
{ \$o1.value \$o2.value }

Sintáctico y trad.

- Las noterms se separan con ;

- Para usar clausuras:

e [String param] returns [String value]

@init {

 StringBuffer trad = new StringBuffer();

}

:

(otronoterm[\$param] {trad.append(\$otronoterm.value;})*

{ \$value=trad.toString();}

;

Errores

- Léxicos y sintácticos:
- dejamos los procesados por el propio ANTLR
- Semánticos

```
System.err.println("Error 7 (" + $ID.line + "," + ($ID.pos+1) + "): " +  
$ID.text + " no es una variable");
```

```
System.exit(-1);
```

Errores: utilidades

```
@parser::members {  
    public void emitErrorMessage(String msg) {  
        System.err.println(msg);  
        System.exit(1);  
    }  
  
    protected String filaColumnaLexema(Token token) {  
        return "(" + token.getLine() + "," + (token.getCharPositionInLine()+1) + "): " + token.getText();  
    }  
  
    public void errorSemantico5(Token token) {  
        System.err.println("Error 5 " + filaColumnaLexema(token) + " ya existe en este ambito");  
        System.exit(-1);  
    }  
}
```

--- una noterm -----

```
    ID  
    {  
        Simbolo tsID = ts.busca($ID.text);  
        if (tsID != null) {  
            errorSemantico5($ID);  
        }  
    }
```

Integración Java

- Preferencias > Output path
- Generate -> generate code
- Refrescamos código en IDE. Classpath con jar de antlr
- main en clase principal práctica

P2

- Recursividad izquierda

$A \rightarrow Ax \mid y$

- se puede transformar en

$A \rightarrow yA'$

$A' \rightarrow xA' \mid \epsilon$

- Cambio funcionamiento atributos heredados por eliminación recursividad izquierda

$A \rightarrow y \{A'.h = \text{trad de } y\} A' \{A.\text{sint} = A'.\text{sint}\}$

$A'_1 \rightarrow x \{\text{trad} = \text{trad de } A'_1.h \text{ con } x ; A'_2.h = \text{trad}\} A'_2 \{A'_1.\text{sint} = A'_2.\text{sint}\}$

$A' \rightarrow \epsilon \{A'.\text{sint} = A'.h\}$

- Con EBNF (ANTLR): $A \rightarrow y x^*$

$A \rightarrow y \{\text{trad} = \text{trad de } y\} (x \{\text{acumular trad de } x \text{ a trad}\}^*) \{A.\text{sint} = \text{trad}\}$

P2

- Recordar poner el código {} cuanto antes para evitar problemas de parsing
- Error en desarrollo más complicado de encontrar:
poner un \$var a algo que es variable local (var)
- Añadir el antlr-3.4-complete-no-antlr-v2.jar al build path de eclipse
- Acordarse de copiar las clases java usadas por nosotros