

Consideraciones de YAPar

Descripción

YAPar (Yet Another Parser) se encuentra basado al estilo de Yacc y toma inspiración sobre la implementación de ocaml yacc, herramienta escrita para OCaml. El objetivo principal de YAPar es la generación de analizadores sintácticos a partir de la especificación de una gramática libre de contexto. La llamada esperada es:

```
yapar parser.yalp -l lexer.yal -o theparser
```

Donde `parser.yalp` es un archivo escrito en Lenguaje YAPar. Esta instrucción genera un archivo `"theparser"` escrito en el lenguaje de su elección, el cual implementa el analizador sintáctico a partir de lo definido en `parser.yalp`. Este archivo se utiliza en conjunto con el lexer previamente generado por YALex, por lo que su llamada debe considerar la información sobre los tokens encontrados según la especificación del archivo `lexer.yal`. Esto puede ser indicado directamente con el argumento `-l`, especificando el archivo con la definición del analizador léxico.

Estructura de archivo yalp

La estructura de un archivo escrito en YAPar es la siguiente:

```
/* Definición de parser */

/* INICIA sección de TOKENS */
%token TOKEN_1
%token TOKEN_2
%token TOKEN_3 TOKEN_4
%token WS
IGNORE WS
/* FINALIZA sección de TOKENS */

%%

/* INICIA sección de PRODUCCIONES */
production1:
    production1 TOKEN_2 production2
    | production2
;
production2:
    production2 TOKEN_2 production3
    | production3
;
production3:
    TOKEN_3 production1 TOKEN_4
    | TOKEN_1
;
/* FINALIZA sección de PRODUCCIONES */
```

- Los comentarios son delimitados por /* y */.
- La primera sección del archivo debe ser la sección de TOKENS. Cada token se define precedido por la palabra reservada token (en minúscula), seguida del nombre de token (en mayúscula). Estos deben hacer match con los tokens producidos por YALex.
- Una línea en esta sección puede contener la declaración de múltiples tokens, separados por un espacio en blanco.
- La palabra reservada IGNORE sirve para ignorar TOKENS.
- Para dividir la sección de TOKENS y la sección de PRODUCCIONES, debe usar %%.
- En la sección de PRODUCCIONES, una producción comienza con el nombre asignado para la producción, seguida de dos puntos (:), luego puede colocar diversas reglas, utilizando minúsculas para los nombres de otras producciones (estos son los no terminales de la gramática) y utilizando el identificador de TOKEN en mayúscula para los tokens correspondientes (estos son los terminales de la gramática).
- Para finalizar la declaración de la producción, utilice punto y coma (;).
- Puede tener la cantidad de producciones que desee. Cuando se llegue a la última línea del archivo, YAPar dejará de interpretar producciones.