

Taller 2: Analizador Sintáctico

Dado un programa en lenguaje SR, su tarea consiste en realizar el análisis sintáctico. Nos vamos a enfocar en los errores sintácticos generados. A continuación se muestra la manera correcta de generar las salidas correspondientes.

En caso de que el programa esté bien formado de acuerdo a las reglas de la gramática de SR, se debe mostrar el mensaje:

"El analisis sintactico ha finalizado exitosamente."

En caso contrario, es decir, si se encontró algún error sintáctico, se debe abortar el análisis y reportar únicamente el primer error sintáctico detectado.

Consideraciones gramaticales

- Se deben considerar todas las construcciones sintácticas definidas en los documentos adjuntados.

Errores sintácticos

- En caso que el código fuente no contenga un programa principal, debe mostrar el siguiente error:
Error sintactico: falta funcion_principal
- Las subrutinas van antes o después del programa principal, y a su vez pueden tener variables, constantes y tipos de datos locales. En caso que no sea así se debe informar en el error sintáctico correspondiente, indicando la localización (fila, columna) del error.

En el caso de cualquier otro error sintáctico, se debe informar al programador usando el siguiente formato:

<linea,col> Error sintactico: se encontro: lexema del token encontrado; se esperaba: lista de símbolos/tokens esperados separados por comas.

Donde:

- **línea y col** son los números de línea y columna donde se detectó el error.
- **lexema del token encontrado:** corresponde al lexema encontrado que no se esperaba encerrado entre comillas dobles (OJO: el lexema, no el token).
- **lista de símbolos/tokens esperados separados por comas:** corresponde a lista de tokens esperados separados por comas y encerrados entre comillas dobles. Por ejemplo: “;”, “)”, “,”, identificador, ...

Por ejemplo, para el código de entrada siguiente:

```
resource main()  
    process ex  
        write(vmnum
```

Se debe mostrar el siguiente error:

```
<3,11> Error sintactico: se encontro: "vmnum"; se esperaba: ";", ")", ",".
```

Nótese que debe manejar un lenguaje que sea fácilmente comprensible para el programador de lenguaje SR. Los símbolos “vmnum”, “,”, y “]” pueden haber tenido otros nombres internamente, por ejemplo: id, tk_coma, tk_par_der. No obstante, estos detalles no deberían ser conocidos por el programador de SL.

Entrada

Para probar el analizador sintáctico se evaluarán distintos casos de prueba. Cada caso de prueba será pasado a su programa por la entrada estándar. Cada entrada consiste en un programa escrito en el lenguaje SR.

Salida

Por cada archivo de entrada se debe mostrar la salida (por consola) según lo especificado anteriormente.