Informe de Problemas y Soluciones
Analizadores LL(1) y SLR(1)

Erick Guerrero, David Ortiz, Anyela Jiménez

Mayo 2025

Detalles de Implementación

- **Sistema Operativo:** Windows 11
- **Lenguaje de Programación:** Python 3.12
- **Librerías utilizadas:** `tabulate` 0.8.9 (para impresión tabular)

Ejecución del Programa

1. Instalar Python 3.6 o superior.
2. Instalar la librería `tabulate`:

   ```
   pip install tabulate
   ```

3. Ejecutar el script `parser.py` desde terminal:

   ```
   python parser.py
   ```

4. Ingresar la gramática y luego las cadenas a analizar como se indica en la interfaz.

Problemas Encontrados y Soluciones

Durante el desarrollo del proyecto se identificaron varios problemas técnicos y conceptuales relacionados con el análisis LL(1) y SLR(1).

1. Recursión Izquierda en la Gramática

Las gramáticas con recursión izquierda directa ($A \to A\alpha$) o indirecta ($A \to B\alpha, B \to A\beta$) causaban bucles infinitos en el analizador LL(1). Estas reglas son problemáticas porque no consumen símbolos del input.

Solución: Se añadió una verificación que detecta recursión izquierda y solicita al usuario reescribir la gramática. Actualmente, el sistema no transforma automáticamente estas reglas, pero avisa claramente cuándo no se puede proceder.

2. Cálculo Incorrecto de FIRST y FOLLOW

Inicialmente, los conjuntos `FIRST` y `FOLLOW` no manejaban adecuadamente la presencia de ε (representada como `e`). Esto provocaba errores al construir la tabla LL(1) y al calcular reducciones en SLR(1).

Solución: Se corrigió la propagación de ε en el cálculo de FIRST. Además, se garantizó que FOLLOW se completara adecuadamente incluso para producciones que derivan únicamente en ε.

3. Conflictos Shift/Reduce en SLR(1)

En el analizador SLR(1), la construcción de la tabla `action` arrojaba conflictos cuando existían situaciones ambiguas, como decisiones múltiples entre desplazamiento y reducción.

Solución: Se identificaron correctamente estos conflictos durante la construcción de la tabla. Si se detecta una ambigüedad, el programa marca la gramática como inválida para SLR(1). Actualmente no se implementan estrategias de resolución como precedencia o asociatividad.

4. Manejo de Tablas

Ocurrieron errores al acceder o modificar celdas inexistentes en las tablas `tabla_ll1`, `action` y `goto`.

Solución: Se inicializaron correctamente las estructuras de datos usando diccionarios anidados. Se añadieron validaciones previas al acceso y se mejoró la depuración con impresión clara de las tablas.

5. Análisis de Cadenas con Gramáticas Inválidas

El análisis de cadenas podía fallar silenciosamente si la gramática no era válida para el tipo de análisis seleccionado.

Solución: El programa verifica primero si la gramática es LL(1), SLR(1), ambas o ninguna. Según el caso, se despliega un menú y se permite seleccionar el tipo de análisis. Si la gramática no es válida para ningún método, se indica con el mensaje:

```
Grammar is neither LL(1) nor SLR(1).
```

6. Ausencia de Resolución de Conflictos en SLR(1)

Aunque se detectan conflictos, el sistema no los resuelve automáticamente. Esto limita el análisis de algunas gramáticas válidas bajo LR(1) pero no SLR(1).

Solución (Parcial): Se documenta claramente el conflicto. La mejora futura sería implementar precedencia y asociatividad para permitir más flexibilidad.

7. Interfaz del Menú de Selección

Aunque el programa distingue si la gramática es LL(1), SLR(1) o ambas, la lógica del menú '"Select a parser (T: for LL(1), B: for SLR(1), Q: quit)« no estaba completamente implementada.

»**Solución:** Se incorporó esta funcionalidad para los casos en que la gramática cumple ambas condiciones, permitiendo al usuario elegir el método de análisis.

»8. Formato de Salida

»El formato de salida no seguía exactamente los ejemplos dados en el documento de asignación (como respuestas `yes`/`no` en una sola línea).

»**Solución:** Se ajustó la salida para cumplir con los requerimientos, asegurando que cada resultado se imprima en una línea, como lo exigen los ejemplos oficiales.

»Conclusión

»El sistema desarrollado logra implementar los analizadores LL(1) y SLR(1) con cálculos correctos de conjuntos FIRST y FOLLOW, construcción de tablas y análisis de cadenas. Los principales problemas fueron resueltos o documentados, y el programa cumple con la especificación de entrada/salida. Quedan pendientes mejoras relacionadas con resolución automática de conflictos, transformaciones gramaticales y depuración más detallada del análisis.