

PROYECTO # 2

Investigación y Desarrollo

Generador Automático de Casos de Prueba a partir de una Gramática

1. Descripción General

Este proyecto consiste en diseñar y desarrollar una aplicación capaz de generar automáticamente casos de prueba válidos, inválidos y extremos a partir de una Gramática Libre de Contexto (GLC). El objetivo del proyecto es utilizar derivación, análisis formal y mutación sintáctica para generar automáticamente distintos tipos de cadenas —válidas, inválidas y extremas— que permitan comprobar de forma sistemática el comportamiento de software/aplicación, la cual estará orientada exclusivamente al procesamiento de expresiones aritméticas.

A continuación, un diagrama conceptual del sistema,

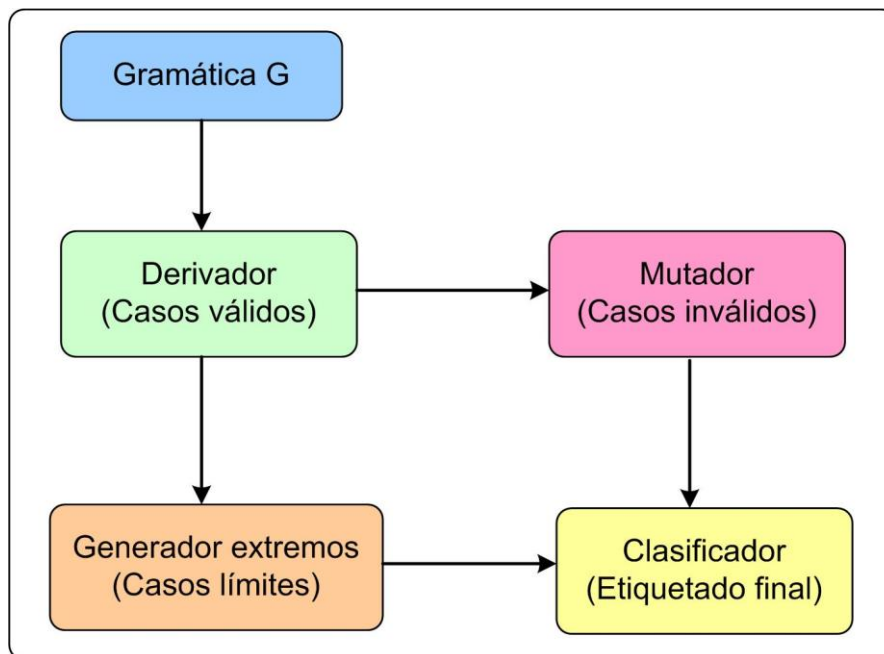


Fig. 1. Diagrama conceptual.

Un generador automático de casos es un sistema que:

- 1) Toma una gramática
- 2) Genera cadenas válidas (derivación)
- 3) Crea cadenas inválidas (mutación)
- 4) Produce cadenas extremas (derivación + restricciones)
- 5) Etiqueta cada una según su categoría
- 6) Reporte estadístico

2. Objetivos

- Aplicar técnicas de derivación a partir de una gramática formal.
- Generar cadenas válidas pertenecientes al lenguaje definido por la gramática.
- Generar cadenas inválidas mediante mutaciones controladas.
- Generar casos extremos aplicando criterios de profundidad, tamaño y complejidad.
- Implementar un sistema (Aplicación) de clasificación automática de los casos generados.
- Exportar los casos generados en formato JSON.
- Integrar una metodología de desarrollo, cascada.

3. Requerimientos

1. La aplicación debe permitir cargar una gramática en archivo .TXT
2. Debe ser capaz de generar automáticamente:
 - Casos válidos derivados de la gramática.
 - Casos inválidos mediante errores sintácticos generados.
 - Casos extremos según configuraciones del usuario.
3. Debe permitir configurar cantidad de casos, profundidad (árbol sintáctico) y longitud (cadena de símbolos terminales).
4. Debe exportar los resultados en JSON.
5. Debe incluir una interfaz básica.
6. Debe entregar un reporte automático con métricas del proceso. (*)

(*) Ejemplos de métricas esperadas:

Cantidad total de cadenas generadas
(válidas, inválidas y extremas)

Distribución porcentual por categoría
(p. ej., 60% válidas, 25% inválidas, 15% extremas)

Longitud promedio de las expresiones
(número de símbolos terminales)

Profundidad máxima del árbol sintáctico
útil para evaluar expresiones anidadas.

Cantidad de operadores generados por tipo (+, -, *, /, %)

Niveles de mutación aplicados
(cuántos errores se generaron y de qué tipo)

Tiempo de ejecución promedio del generador
Para válidas, inválidas y extremas.

7. Explicaciones: descripción técnica del funcionamiento del sistema. Debe explicar cómo y por qué se generaron los resultados.

8. Pruebas: deben mostrar casos concretos de pruebas, acompañados de los resultados obtenidos al ejecutar su aplicación.

9. Evidencia: corresponde a archivos y recursos. Git (enlace en anexo), diagramas, capturas, evidencia visual del flujo del programa/aplicación, manual de usuario. Considera Sección Anexos para incluir las evidencias.

10. Demostrar al Profesor la aplicación/programa.

11. Se implementa en Lenguaje de su preferencia.

Orientación: Revise la carpeta [formatos y rubrica] en la plataforma Blackboard.

Tipo: Grupal (según las reglas del curso)

Plazo: Es indicado en plataforma

Prof. M. Lévano