



Paradigmas de Programación (EIF-400) Tarea Gramáticas Compilación

CARLOS LORÍA-SÁENZ

OCTUBRE 2015

EI/UNA



Objetivos

2

- ▶ Aplicar nociones de gramáticas en el procesamiento de lenguajes computacionales
- ▶ Usar OOP-FP para su procesamiento
- ▶ Usar patrones de diseño apropiados
- ▶ Asimilar las nociones básicas de un compilador pequeño

Problema

3

- ▶ Ud. recibe un proyecto base de ANTLR y un modelo de objetos (AST con visitor) en Java llamado CH (Lenguaje Chifrijo)
- ▶ Ud. debe extender dicho proyecto de forma que maneje más elementos de lenguaje (se indican más adelante)
- ▶ Implica extender la gramática dada, el modelo de datos y el compilador. Asimismo desarrollar una VM pa el código generado.
- ▶ Ud. deberá usar las herramientas que acá se le indican para lograr el objetivo.
- ▶ Ud. deberá usar técnicas de FP en java8 en su implementación

Material

- ▶ `gramaticasCompilación.rar`
- ▶ Nota: pudieran haber nuevas versiones mejoradas de esta versión según se requiera

Requerimientos de herramientas

- ▶ Usar el proyecto base con su estructura actual
- ▶ Usar ANTLR4 para la gramática y el parser
- ▶ Usar Java8 y FP lo más posible

Fases

- ▶ El problema se divide en dos fases
 - ▶ Fase de Compilación (extender CH)
 - ▶ Fase de ejecución (desarrollar un VM para CH)
- ▶ Se dividen en sprints cada uno con sus features

Requerimientos de Lenguaje CH

- ▶ Sprint relacional
 - ▶ Manejar operadores relaciones `<=`, `!`, `==`, `!=`
 - ▶ Permitir los valores booleanos `true` y `false`
- ▶ Sprint condicional
 - ▶ Permitir operador “Elvis” `() ? :`
- ▶ Sprint listas
 - ▶ Permitir operadores listas `[...]`
 - ▶ Permitir operadores `first`, `rest`,
- ▶ Sprint lambda
 - ▶ Permitir lambdas unarias `x -> e`
 - ▶ `map` y `reduce` (prefijos)

Requerimiento del compilador

- ▶ Extender el compilador para que maneje los nuevos elementos de lenguaje sobre la base del modelo inicial
- ▶ Usar FP en la medida de lo posible

Requerimientos de testing fase Compilación

9

- ▶ Desarrollar al menos 3 casos de prueba distintos que prueben cada sprint del lenguaje

Requerimiento de ejecución (CHVM)

- ▶ Desarrollar un parser en ANTLR4 que parsee código compilado de CH
- ▶ Desarrollar en Java un intérprete (máquina virtual para CH, la CHVM) que ejecute dicho código generado por el compilador
- ▶ Dicha máquina es de pila y será discutida en clase al presentar este proyecto
- ▶ Se desarrolla siguiendo los mismos sprints del compilador

Requerimientos de testing

Fase Ejecución

11

- ▶ Usando los casos de prueba (test cases) de la fase de compilación probar que los resultados esperados funcionan

Evaluación y Revisión

12

- ▶ Demo en clase: 70% (unos 10 minutos por grupo)
- ▶ La demo consiste en correr los casos de prueba y mostrar uno por uno el resultado.
- ▶ Puede traerlo en su propia laptop por si acaso haya problemas en el lab.
- ▶ Revisión de código 30%.
- ▶ Se puede otorgar hasta un 10% extra sobre la nota si se añaden más funcionalidades (previamente acordadas con el profesor)
- ▶ Se entregará de previo a la entrega una guía de revisión por parte del profesor

Valor

13

- ▶ Cada fase vale un 30% del rubro de proyectos de la carta al estudiante (15% proyectos)

Entregable

14

- ▶ El proyecto en un `.rar` llamado `GramaticasCompilacion_AAA.rar` donde AAA son los autores. Se enviará por correo al profesor. El Subject siendo "GramaticasCompilacion_AAA" (AAA como antes) y en el correo de nuevo los autores y grupos
- ▶ Dentro del `rar` viene un directorio (carpeta) CH (con el proyecto) y un `README.txt` con los nombres de los autores, su horario.
- ▶ Debe mantener la estructura de proyecto original
- ▶ Incumplimiento de estos requisitos anula la revisión que hará el profesor (pierden su 30%). Sin derecho a reclamo.

Fecha y hora de entrega

15

- ▶ El día de entrega se dará a conocer oportunamente en clase y por los medios usuales en el curso.
- ▶ Ese día de entrega (de la demo) se envía el entregable por correo en los primeros 5 minutos de la clase
- ▶ Entregas previas o posteriores e injustificadas ese día podrán recibir pérdida de puntos de hasta 5 puntos por minuto de anticipo o atraso hasta llegar a cero.