

gbroloyalo feat: intermediate representation README

2eed05b · 3 months ago

33 lines (25 loc) · 3.25 KB

Preview Code Blame

Raw

Fase de Compilación: Generación de Código Intermedio

Descripción General

La generación de código intermedio (CI) es la siguiente fase del diseño de nuestro compilador de Compiscript. Luego de haber realizado el análisis semántico (análisis de tipos), utilizarán sus estructuras de datos (árboles sintácticos, tablas de símbolos) para generar una representación intermedia del código de alto nivel. Esta representación intermedia les será de utilidad al momento de la generación de código assembler (u objeto).

- Lea atentamente el README.md en este directorio, en dónde encontrará las generalidades del lenguaje.
- En el directorio `program` encontrará la gramática de este lenguaje en ANTLR y en BNF. Se le otorga un playground similar a los laboratorios para que usted pueda experimentar inicialmente.
- **Modalidad: Grupos de 3 integrantes.**

Requerimientos

- Agregar acciones semánticas necesarias sobre el árbol sintáctico construido, con el objetivo de generar código intermedio. La sintáxis del código intermedio a utilizar es a discreción del diseñador (pueden utilizar la sintáxis de alguna bibliografía conocida o la vista en clase).
- Complementar la información de la tabla de símbolos con datos necesarios para la generación de código assembler u objeto (direcciones de memoria, etiquetas temporales, etc.)
- Implementar un algoritmo para asignación y reciclaje de variables temporales durante la transformación de expresiones aritméticas.
- **Para los puntos anteriores, deberá de escribir una batería de tests para validar casos exitosos y casos fallidos en cada una de los casos que crea convenientes.**
- Al momento de presentar su trabajo, esta batería de tests debe estar presente y será tomada en cuenta para validar el funcionamiento de su compilador.

- Editar la implementación de la **tabla de símbolos** que interactue con cada fase de la compilación, para soportar los ambientes y entornos en tiempo de ejecución, utilizando registros de activación.
- Deberá **desarrollar un IDE** que permita a los usuarios escribir su propio código y compilarlo.
- Deberá crear **documentación asociada a la arquitectura de su implementación y documentación de las generalidades de cómo ejecutar su compilador**.
- **Documentación que explique a detalle del Lenguaje Intermedio diseñado**, junto a ejemplos y supuestos considerados durante la traducción. Esto servirá a los calificadores para comprender sus decisiones de diseño y analizar la veracidad de su implementación.
- Entregar su repositorio de GitHub.
 - Se validan los commits y contribuciones de cada integrante, no se permite "compartir" commits en conjunto, debe notarse claramente qué porción de código implementó cada integrante.



Ponderación

Componente	Puntos
Diseño de CI	25 puntos
Generación de código intermedio desde Compiscript a TAC	65 puntos
Tabla de símbolos con nuevas adiciones	10 puntos
Total	100 puntos