

Multi-Level Intermediate Representation Overview

Clavel Vergara Kimbberly Nicole 171080181

González Rivera Emmanuel Alejandro 171080089

Diez Bonilla Guerrero Dilan Eduardo 171080167

López Tello Rodríguez Mario Iván 171080102

Resumen General

El proyecto de "Compilador intermedio de multinivel" es una representación intermedia que se encarga de unir la infraestructura necesaria para poder ejecutar modelos de aprendizaje automático estos son archivos que se realizan para reconocer determinados tipos de patrones para ello debemos proporcionarle algún algoritmo que pueda usar para desbrozar los datos también aprenderlos.

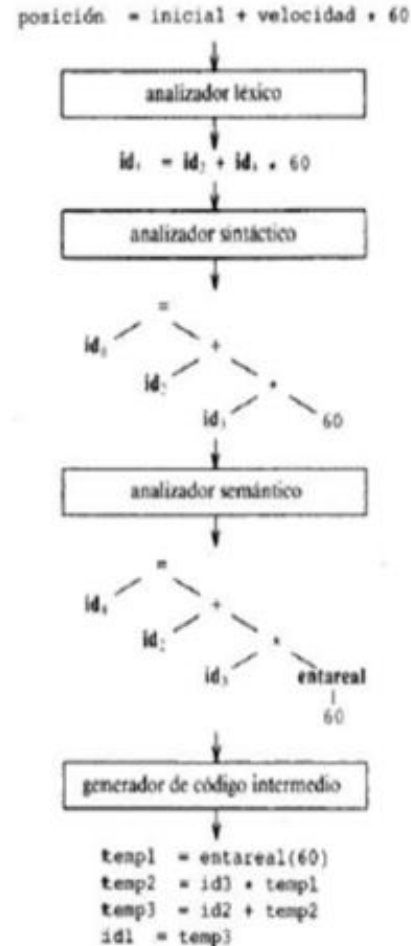
su objetivo principal es reducir el costo en incorporar hardware nuevo así cómo mejorar su usabilidad de la misma forma entre sus principales características podemos encontrar que el sistema operativo que utilicemos puede almacenar los datos mucho más rápido así cómo mejorar el rendimiento de los dispositivos al añadir procesadores diversos.

Ahí podremos realizar operaciones específicas, ya sea lógicas o aritméticas.

Además, a través de la realización de esta proyecto, nosotros podremos dar definiciones claras y concisas acerca del proceso tanto de elaboración, como de ejecución así como de los resultados obtenidos de dicho Compilador.

Funcionamiento

MLIR proporciona la columna vertebral para representar cualquier lenguaje de programación e integrarlo en el ecosistema.



Infraestructura del Compilador

El MLIR consta de una especificación para representaciones intermedias (IR) y un juego de herramientas de código para realizar transformaciones en esa representación. (En el lenguaje del compilador, a medida que pasa de representaciones de nivel superior a representaciones de nivel inferior, estas transformaciones se pueden llamar "reducciones").

Justificación

Para la elaboración de este proyecto el equipo se planteó el propósito de integrar a este documento la mayor cantidad de información de tal manera en abordar a detalle todos los factores que intervienen o conforman la estructura del compilador intermedio Multi-Level.

Para la realización de este proyecto tiene como fin el poder descubrir todo sobre cómo interpretar a detalle el correcto funcionamiento del compilador intermedio Multi-Level, dado que una de sus principales funciones es brindar un tipo de ayuda para los compiladores ya que este compilador se encarga de realizar la traducción de un lenguaje de programación a un lenguaje máquina.

A grandes rasgos, lo que queremos lograr con todo lo anteriormente mencionado, es permitir exploraciones novedosas para así poder optimizar el diseño y la implementación del compilador, respaldados por componentes de calidad de producción, buscaremos abordar las múltiples implementaciones que tiene tanto de compilación como de representación, mismas que ayudarán a mejorar de manera significativa el rendimiento del compilador.

Planteamiento

Conocer el Compilador intermedio multinivel es muy interesante con respecto al funcionamiento y las partes que conforman al compilador, ya que es aquí en donde se realiza la traducción de un lenguaje de programación de alto nivel a un lenguaje de bajo nivel o lenguaje máquina.

Explicaremos paso a paso la estructura que lo compone, para también conocer de manera detallada su funcionamiento dentro del compilador.

Es necesario realizar este proyecto para darnos cuenta de la necesidad de diseñar generadores de código, optimizadores y traductores ya que los marcos de aprendizaje automático de hoy en día tienen diferentes tiempos de ejecución, compiladores y tecnologías de gráficos.

En términos generales, son gran cantidad de puntos los que se tienen y se pretenden abarcar, puesto que ninguno es menos importante que otro y al estudiarlos y comprenderlos podremos sacar nuestras propias conclusiones de manera clara y concisa de tal forma que se determine la ayuda o el apoyo que brindan principalmente a las tareas para las cuales fue desarrollada esta parte dentro de los compiladores.

Diseño y Desarrollo

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <windows.h>
4 main(){
5
6     int x,a,b,c,z;
7     float x1=0,x2=0,temp1=0,temp2=0,temp3=0,temp4=0,temp5=0,temp6=0,temp7=0,temp8=0,temp9=0,temp10=0,temp11=0,temp12=0,temp13=0,temp14=0,res=0;
8
9     system("color f0");
10    do{
11        z=0;
12        system("C:\Users\Chompi\Crazy\Desktop\Nueva carpeta (8)\MLIRO.exe
13        printf("INTEGRANTES:
14        printf("T KIMBERLY NICOLE CLAVEL VERGARA 171080181
15        printf("K DILAN EDUARDO DIEZ BONILLA GUERRERO 171080167
16        printf("K EMMANUEL ALEJANDRO GONZALEZ RIVERA 171080089
17        printf("D MARIO IVAN LOPEZ TELLO RODRIGUEZ 171080102
18        printf("E
19        printf("E ELIGE UNA OPCION:
20        printf("E 1.- PROGRAMA
21        printf("E 2.- SALIR
22        printf("1 OPCION:
23        printf("2
24        printf("0
25        scanf("%i",&z);
26        switch(x){
27            case 1:
28                do{
29                    printf("1. VALORES: a=1 b=6 c=8\n");
30                    printf("FORMULA PARA LA SOLUCION: x1=-b+sqrt(b*b-4*a*c)/(2*a) x2=-b-sqrt(b*b-4*a*c)/(2*a)\n\n");
31                    printf("RESOLUCION DE x1:\n");
32                    temp1=b;
33                    printf("RESOLUCION DE x2:\n");
34                    temp2=b;
35                    printf("RESOLUCION DE x3:\n");
36                    temp3=b;
37                    printf("RESOLUCION DE x4:\n");
38                    temp4=b;
39                    printf("RESOLUCION DE x5:\n");
40                    temp5=b;
41                    printf("RESOLUCION DE x6:\n");
42                    temp6=b;
43                    printf("RESOLUCION DE x7:\n");
44                    temp7=b;
45                    printf("RESOLUCION DE x8:\n");
46                    temp8=b;
47                    printf("RESOLUCION DE x9:\n");
48                    temp9=b;
49                    printf("RESOLUCION DE x10:\n");
50                    temp10=b;
51                    printf("RESOLUCION DE x11:\n");
52                    temp11=b;
53                    printf("RESOLUCION DE x12:\n");
54                    temp12=b;
55                    printf("RESOLUCION DE x13:\n");
56                    temp13=b;
57                    printf("RESOLUCION DE x14:\n");
58                    temp14=b;
59                    printf("RESOLUCION DE res:\n");
60                    res=b;
61                }while(1);
62            case 2:
63                break;
64        }
65    }while(z!=2);
66 }
```

Fuentes de Información

Hugo. (2012). Marco del compilador de infrarrojos multinivel. nov 25, 2020, de MLIR Sitio web: <https://mlir.llvm.org/>

@pypyproject. (2019). Compilador, Generación de código, Lenguaje de programación, C, Java, Programa de computadora. nov 27 2020, de Essentials Sitio web: <https://essentials.news/ai/research/article/multi-level-intermediate-representation-overview-42b8bb3a5e>

joker-eph. (2019). MLIR is now part of LLVM, avoid any mistake by removing the code and a README pointing to the new location. dic 1 2020, de GitHub Sitio web: <https://github.com/tensorflow/mlir>

stellaraccident. (2012). [mlir][CAPI] Add result type inference to the CAPI.. dic 4 2020, de GitHub Sitio web: <https://github.com/llvm/llvm-project/tree/master/mlir/>

Ramón Chávez González. (2013). Teoría de Lenguajes y Compiladores. dic 8 2020, de sites.google.com Sitio web: <https://sites.google.com/site/teoriadelenguajesycompiladores/procesadores-de-lenguaje/generacion-de-codigo>

www.frro.utn.edu.ar/. (2015). SINTAXIS. dic 11 2020, de SSyL Sitio web: https://www.frro.utn.edu.ar/repositorio/catedras/sistemas/2_anio/sintaxis/SSyL-cap1_2015_Introduccion.pdf

Aho, A.V., Sethi, R., Ullman, J.D. (1990), Compiladores: principios, técnicas y herramientas, Tema 8, 9, 10 (pag. 478- 666).

Louden, K.C. (1997), Compiler Construction: Principles and Practice, Tema 8, páginas: 398-481.

Ing. M.A. Sheyla Esquivel. (2020). Compiladores - Código Intermedio. dic 15 2020, de Youtube Sitio web: <https://www.youtube.com/watch?v=Yi0cOCcXR6I>