CompiladorCalculadora

Aquest repositori conté la part frontal d'un compilador per a una calculadora.

Aprenentatges

- Taula de símbols
- Anàlisi lèxic, sintàctic, semàntic
- Atributs i comprovació de tipus
- Utilització conjunta de Flex, Bison i Symtab

Compilació

1. Clona el repositori:

```
git clone https://github.com/PolPujolSantaella/CompiladorCalculadora.git
cd CompiladorCalculadora
```

2. Compila el projecte

```
make all
```

Execució

Per provar el joc de proves que he realitzat només cal fer un:

```
make eg
```

Aquest et genera un arxiu ex_sortida.out que conté la sortida corresponent.

Per netejar-ho tot menys arxius fonts

```
make clean
```

Descripció del Projecte

Aquest projecte és la part frontal d'un compilador per a una calculadora. Permet analitzar, interpretar i executar expressions aritmètiques introduïdes per l'usuari, incloent operacions de suma, resta, multiplicació, divisió, potència, funcions trigonomètriques (sin, cos, tan) i expressions booleanes.

El compilador processa les expressions a través d'un anàlisi lèxic i sintàctic, convertint l'expressió en una sèrie de tokens, verificant la sintaxi i generant el resultat calculat.

El funcionament del compilador pasa per processar les expressions mitjançant l'anàlisis lèxic i sintàctic. Converteix la expresió en una serie de tokens, verifica la sintaxis i després genera un resultat calculat.

Components del codi

- Anàlisi Lèxic: Descompon l'expressió en tokens individuals, com números i operadors.
- Anàlisi Sintàctic: Avalua l'estructura de l'expressió per garantir que segueixi les regles matemàtiques.
- Anàlisi Semàntic: Verifica que l'expressió tingui sentit matemàtic, comprovant errors com divisió per zero.

Decissions de disseny

- En l'arxiu calculadora.l defineixo les regles lèxiques per el reconeixement de tokens en la calculadora.

  Es defineixen patrons lèxics com parèntesis, operadors (+, - *, /) i operadors d'assignació. També gestiono els comentaris tant de línea com de bloc utilitzant expressions regulars específiques.

  Els tokens estàn organitzats de manera clara, però pot resultar complexe afegir més operadors o estructures.

- En l'arxiu calculadora.y defineixo les regles sintàctiques, es a dir, com s'organitzen els tokens en expressions vàlides.

  La gramàtica proposada utilitza una clara notació per les expressions i un suport adicional per funcions de concatenació i manipulació de cadenas.

- El arxiu symtab.h és una peça clau en el disseny, ja que defineix la interfície i les estructures necesàries per a la taula de símbols. Aquesta permet gestionar i emmagatzemar informació sobre els identificadors utilitzats en les expressions. Per això he modificat el camp sym_value_type per una estructura corresponent a la grmàtica.

Limitacions

Una limitació seria l'agregació de més operadors o tipus de dades, la gramàtica podria ser difícil de gestionar. Un altre limitació que faltaria implementar seria una gestió d'errors més extensa.

També podriem tenir limitacions en afegir altres tipus més complexos o estructures, potser caldria requerir una modificació considerable.

I com a última limitació es la concatenació de cadenas entre expressions aritmètiques i booleanes. Caldria modificar la gramàtica per a que es pugui realitzar aquesta concatenació.

En conclusió, tenim una limitació en la escalabilitat i en algunes funcions de concatenació de cadenes