

# **Progetto di Laboratorio di compilatori**

## **A.A. 2023-2024**

Tommaso Vilotto - VR471487  
Alex Gaiga - VR471343

# **INDICE**

|                                      |          |
|--------------------------------------|----------|
| <b>Descrizione</b>                   | <b>3</b> |
| <b>Requisiti</b>                     | <b>4</b> |
| <b>Compilazione &amp; Esecuzione</b> | <b>4</b> |
| <b>Comandi - esempi Input/Output</b> | <b>5</b> |
| <b>Scelte progettuali</b>            | <b>7</b> |

# Descrizione

Questo progetto utilizza Lex e Yacc per creare un analizzatore sintattico di un linguaggio specifico per la definizione e la manipolazione di polilinee. Una polilnea è una sequenza di segmenti lineari contigui. Se il punto iniziale e quello finale della polilnea coincidono, la polilnea è definita 'chiusa', altrimenti è 'aperta'.

## Funzionalità:

### 1. Definizione delle Polilinee:

- È possibile definire una polilnea specificando un numero arbitrario di punti.

### 2. Simmetrie:

- I punti possono essere ottenuti tramite simmetria rispetto all'origine (sim0), all'asse x (simx) o all'asse y (simy).

### 3. Operazioni sulle Polilinee:

- isOpen: Verifica se la polilnea è aperta. Un punto è considerato una polilnea chiusa.
- Close: Chiude una polilnea aperta e calcola la lunghezza della polilnea chiusa risultante (perimetro del poligono).

### 4. Salvataggio delle Polilinee in Variabili:

- È possibile salvare una polilnea associandola a una variabile.
- Eseguire i comandi isOpen e Close con le polilinee salvate
- Eseguire sommatorie binarie tra polilinee assegnando il risultato ad una variabile

### 5. Abstract Syntax Tree (AST)

Si traduce un testo valido nel linguaggio delle polilinee in un Abstract Syntax Tree, si ottiene una struttura che rappresenta la gerarchia e le relazioni tra i punti e le simmetrie.

Attenzione, questa feature è disponibile compilando un file diverso da quello che contiene tutte le feature sopra citate. Vedasi la sezione [Compilazione & Esecuzione](#) per maggiori informazioni.

## Requisiti

Per poter compilare su Linux i sorgenti ed eseguire il programma è necessario avere installati due pacchetti, flex e bison.

Se fosse necessario installarli basterà utilizzare il seguente comando:

```
sudo apt-get install flex bison
```

## Compilazione & Esecuzione

Per la compilazione è sufficiente posizionarsi all'interno della cartella con i file e eseguire sul terminale ed eseguire il comando `make`

Questo comando genererà due file eseguibili:

1. elaborato
2. elaboratoPlus

Il primo contiene tutte le feature a meno dell'AST, il secondo contiene esclusivamente la feature dell'AST.

Per eseguire il programma basterà lanciare uno dei due eseguibili in questo modo:

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaborato
```

## Comandi - esempi Input/Output

### ESECUZIONE ./elaborato

- Creazione polilinea temporanea e visualizzazione lunghezza della sua lunghezza:

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaborato  
1 -2 -3 4 5 6  
Lughezza: 15.457 cm
```

- Creazione polilinea temporanea e visualizzazione lunghezza della sua lunghezza usando coordinate simmetriche rispetto all'origine (sim0), all'asse x (simx) o all'asse y (simy):

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaborato  
simx (1 2 sim0(3 4) ) 5 6  
Lughezza: 15.457 cm
```

- Controllo se la polilinea temporanea è aperta:

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaborato  
isOpen simx (1 2 sim0(3 4) ) 5 6  
True
```

- Chiusura della polilinea temporanea con output della sua lunghezza:

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaborato  
Close simx (1 2 sim0(3 4) ) 5 6  
Lughezza: 24.402 cm
```

- Salvataggio di una polilinea associando ad essa una variabile e visualizzazione della sua lunghezza richiamando il nome:

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaborato  
P1 = simx (1 2 sim0(3 4) ) 5 6  
P1  
Lughezza: 15.457 cm
```

- Controllo se la polilinea salvata in una variabile è aperta:

```
isOpen P1  
True
```

- Assegnamento ad una variabile la chiusura di una polilinea:

```
P2 = Close P1
isOpen P2
False
```

- Assegnamento ad una variabile la somma ottenuta collegando due polilinee salvate a loro volta in variabili:

```
P3 = P1+P2
```

## ESECUZIONE ./elaboratoPlus

E' possibile visualizzare l'Abstract Syntax Tree di tutti i comandi sopra illustrati, visualizzando a video la profondità degli elementi che ne fanno parte, a seguire un paio di esempi.

Esempio polilinea temporanea con coordinate simmetriche all'asse x e origine:

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaboratoPlus
simx (1 2 sim0(3 4) ) 5 6
simx
-----1.00
-----2.00
-----sim0
-----3.00
-----4.00
5.00
6.00
```

Esempio assegnamento congiunzione di due polilinee

```
tom@tom:~/Scrivania/COMPILATORI/elaborato$ ./elaboratoPlus
P1 = P2 + P3
P1
=
-----+
-----P2
-----P3
```

# **Scelte progettuali**

## Prima parte

- l'input viene accettato se e solo se è esente di punto e virgola, con l'utilizzo dell'end of line è possibile dare in input più comandi in maniera sequenziale.
- abbiamo deciso di trattare i punti sul piano cartesiano come reali, gestendoli come float nel codice per una migliore accuratezza
- oltre alle funzionalità della consegna abbiamo aggiunto anche i seguenti comandi non specificati:
  - string eq string

## Seconda parte

La grammatica usata per la consegna del AST è la medesima degli altri esercizi, è stato utilizzato un albero n-ario per la sua rappresentazione, essendo i punti rappresentabili infiniti.