

Progetto di Laboratorio di compilatori A.A. 2023-2024

(V. Cozza)

1. Linguaggio per costruire polilinee

Creare l'analizzatore sintattico di un linguaggio che permette di creare polilinee, utilizzando Lex e Yacc.

Con "polilinea" viene designata una sequenza di segmenti lineari contigui. Se il punto iniziale e quello finale della polilinea si uniscono, la polilinea viene definita 'chiusa', altrimenti 'aperta'.

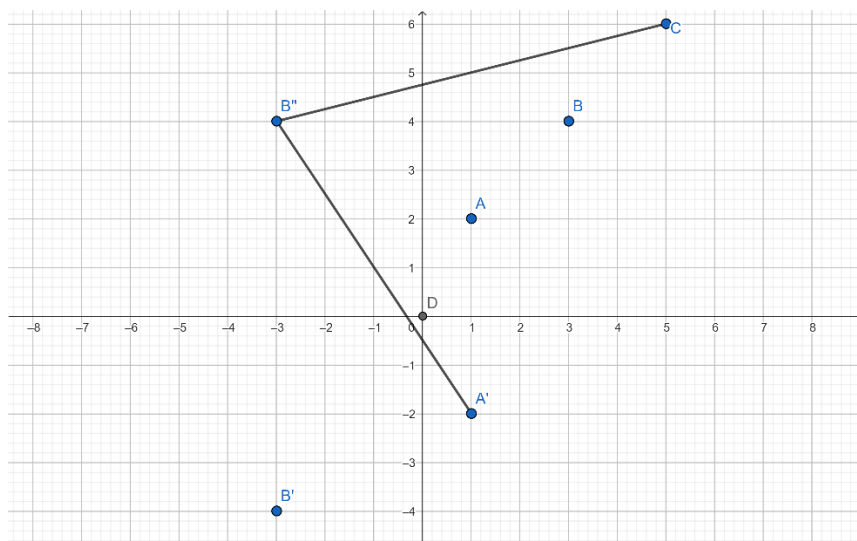


Figura 1: Polilinea P1

Il linguaggio deve permettere di definire una polilinea indicando un numero arbitrario di punti, e una volta definita ne stampa la lunghezza. Una istruzione valida quindi è:

1 -2 -3 4 5 6;

questa rappresenta la polilinea in Figura 1 e restituisce la lunghezza della polilinea pari a 15.457cm.

Un punto oltre ad essere l'insieme di 2 coordinate, è anche ottenibile tramite simmetria di un altro punto, in particolare simmetria rispetto all'origine degli assi indicata con **sim0**, all'asse x indicata con **simx** o all'asse y indicata con **simy**. Si possono usare anche altri simboli a scelta per indicare le simmetrie.

Una espressione valida che usa le simmetrie è:

simx (1 2 sim0(3 4)) 5 6

anche questa rappresenta la polilinea in Figura 1 e restituisce la lunghezza della polilinea pari a 15.457cm.

Sono inoltre possibili due operazioni su una polilinea:

- isOpen che restituisce vero se la polilinea e' aperta, falso altrimenti.

Esempio:

`isOpen simx (1 2 sim0(3 4)) 5 6`

Falso

Si noti che un punto è una polilinea chiusa.

- Close che chiude la polilinea restituendo la lunghezza ottenuta.

Esempio:

`Close simx (1 2 sim0(3 4)) 5 6`

questa istruzione chiude la polilinea in Figura 1 producendo quella in Figura 2. Il questo caso restituisce la lunghezza della polilinea chiusa ovvero il perimetro del poligono ottenuto pari a 24.401 cm. Ovviamente questa istruzione ha effetto solo se la polilinea è aperta.

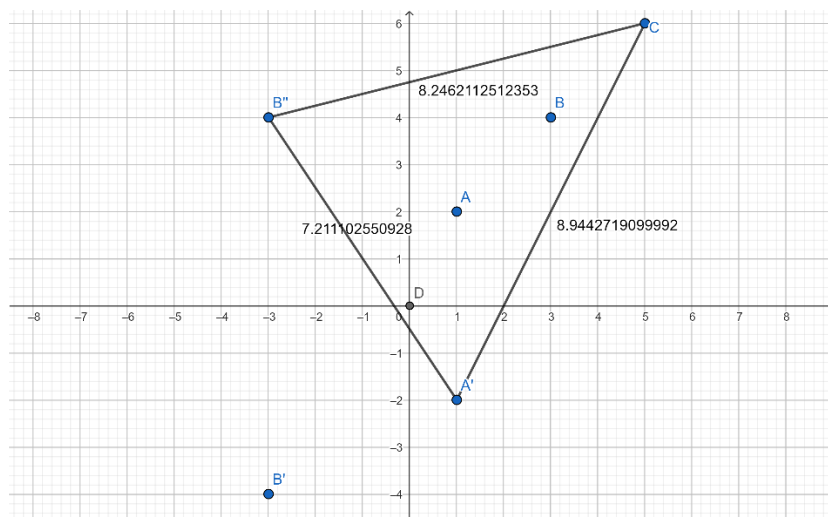


Figura 2: Close Polilinea P1

Opzionale:

Salvare la polilinea associando ad essa una variabile:

`P1 = simx (1 2 sim0(3 4)) 5 6`

E quindi prevedere anche operazioni quali:

`P1` che restituisce la lunghezza

`isOpen P1` che restituisce vero se la polilinea è aperta, falso altrimenti

`P2 = Close P1` che memorizza in P2 la polilinea chiusa

`P3 = P1+P2` che restituisce la polilinea somma ottenuta collegando le due polilinee P1 e P2 con un segmento, questo segmento ha come estremi il punto finale di P1 con il punto iniziale di P2.

Esempio

$P1 = 1 \ -2 \ -3 \ 4 \ 5 \ 6$

$P2 = 3 \ 3 \ 6 \ 1 \ 5 \ -2$

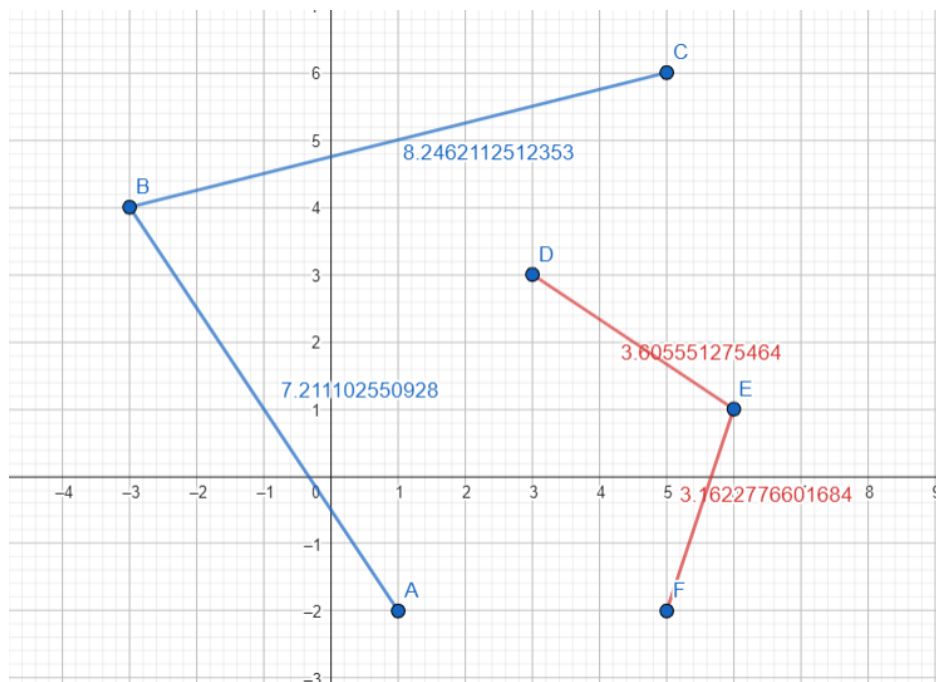


Figura 3: Rappresentazione di $P1$ e $P2$

$P3 = 1 \ -2 \ -3 \ 4 \ 5 \ 6 \ 3 \ 3 \ 6 \ 1 \ 5 \ -2$

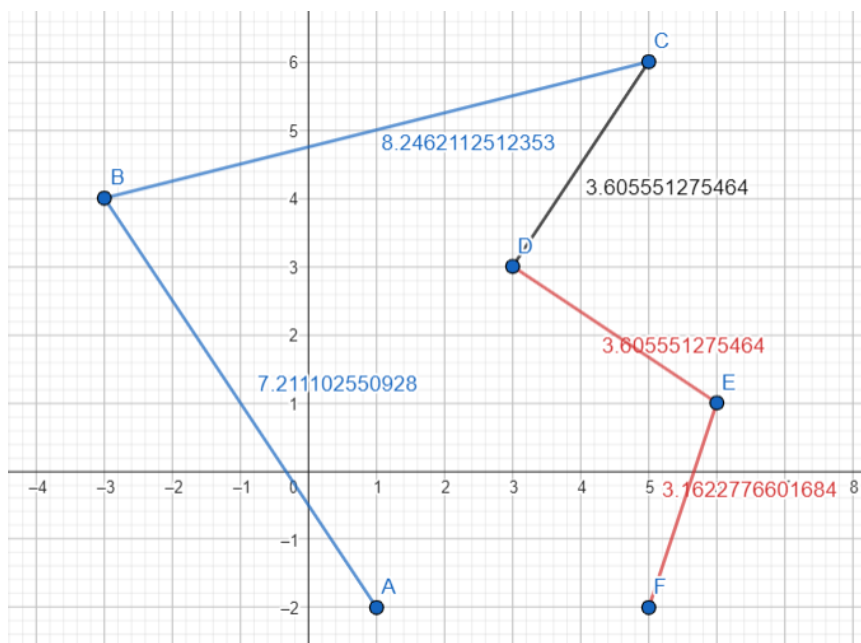


Figura 4: $P3 = P1 + P2$

2. Tradurre un testo valido nel linguaggio delle polilinee in un Abstract Syntax Tree.

Nel caso dell'esempio:

`simx (1 2 sim0(3 4)) 5 6`

Si deve stampare una struttura del tipo:

`simx`

`-----1`

`-----2`

`-----sim0`

`-----3`

`-----4`

`5`

`6`

Note sulla consegna: Scrivere un file README per il progetto. Inviare il progetto via e-mail due giorni prima della data dell'esame mettendo nell'oggetto il proprio numero di matricola o quello di tutti i membri del gruppo se si è lavorato in gruppo (preferibile massimo 2 persone). Mettere tutti i membri del gruppo in copia alla e-mail. Mandare una sola e-mail per gruppo.