

CineScript

CineScript è un linguaggio di programmazione ispirato alla sceneggiatura cinematografica. È stato sviluppato utilizzando **Flex** e **Bison** per i principi base dei compilatori, assieme a **C** per la gestione dello scope e delle strutture dati, e per implementare la semantica e la Symbol Table

Introduzione

CineScript consente di scrivere programmi usando una sintassi narrativa basata sul lessico del cinema. Il compilatore è in grado di interpretare ed eseguire dichiarazioni di variabili, operazioni aritmetiche, concatenazioni di stringhe, strutture condizionali (if) e comandi come la stampa, la gestione della Symbol Table e l'accesso a funzioni matematiche.

Funzionalità

- **Dichiarazione e riassegnazione di variabili**
 - AZIONE ID = expr → float
 - DRAMMA ID = expr → stringa
 - RIPRENDI ID = expr → riassegnazione
- **Stampa su output**
 - SCENA expr: stampa il valore.
 - ZOOM ID: stampa il valore completo.
 - FINALE : stampa l'output e termina il programma
 - CAST: stampa tutta la Symbol Table (ID, type, scope, value)
- **Strutture condizionali**
 - IF (condizione) {...} o SE (...) {...}
 - WHILE (condizione) {...} o MENTRE (...) {...}
 - Attualmente non sono stati implementati i salti condizionali, quindi non c'è l'ELSE che segue l'IF, e il WHILE viene eseguito una sola volta, quindi senza loop.
- **Funzioni matematiche**
 - SOMMA(a,b), DIFFERENZA(a,b), PRODOTTO(a,b), DIVISO(a,b)
 - PI, RADICEQ(x), ABS(x), POTENZA(x, y)
- **Operatori**
 - Aritmetici: + - * / %
 - Confronto: == != > < >= <=
- **Commenti**
 - Linee precedute da @ vengono ignorate.
- **Aiuto**
 - HELP o SCENEGGIATURA: mostra tutti i comandi disponibili.
- **Gestione dello scope**
 - Ogni variabile ha un campo scope che indica dove è stata dichiarata: scope = 0 se si trova fuori da blocchi if o while, scope = 1 (o maggiore) se si trova all'interno di uno di questi blocchi.
 - Il valore di currentScope cambia mentre il programma viene eseguito, in base a dove ci si trova nel codice. Per esempio, quando si entra in un blocco if, currentScope aumenta di 1; quando si esce, torna al valore precedente.
 - Quando cerchiamo una variabile con il lookup(), confrontiamo il suo scope con currentScope. In questo modo, il programma può decidere se usare la variabile dichiarata fuori o dentro il blocco, a seconda di quale è più adatta.

Symbol Table

- Implementata come **linked list**, tiene traccia di:
 - ID, Tipo (float|string), Scope, Valore o Testo
 - Tutte le variabili, anche quelle nei blocchi condizionali, vengono salvate nella Symbol Table.
-

Grammatica del linguaggio

- La grammatica parte da prog, che rappresenta l'intero programma e contiene una lista di istruzioni (lista_stmt).
- stmt rappresenta un'istruzione specifica (le sue produzioni includono dichiarazioni, riassegnazioni, comandi di stampa e blocchi condizionali).
- IF e WHILE aprono nuovi blocchi. Quando vengono letti, si entra in un nuovo scope (incrementando currentScope), e si esegue il blocco solo se la condizione è vera. Alla fine del blocco, currentScope viene decrementato.
- Le variabili vengono create tramite i comandi AZIONE e DRAMMA, e possono assumere valori numerici o stringa. Il non-terminal expr gestisce tutte le espressioni matematiche, concatenate, e l'uso delle funzioni.

prog → lista_stmt | lista_stmt FINALE

lista_stmt → stmt lista_stmt
|

stmt → AZIONE ID = expr
| DRAMMA ID = expr
| RIPRENDI ID = expr
| SCENA expr
| ZOOM ID
| IF (compare) { lista_stmt }
| WHILE (compare) { lista_stmt }
| CAST
| HELP

compare → expr == expr
| expr != expr
| expr > expr
| expr >= expr
| expr < expr
| expr <= expr

expr → ID
| NUM
| STRING
| expr + expr
| expr - expr
| expr * expr
| expr / expr
| expr % expr
| - expr
| (expr)
| funzione(expr, expr)
| PI

Compilare ed Esecuzione

Bisogna avere flex, bison e gcc installati. Poi, esegui:

make

Verrà generato l'eseguibile cinescript. Per eseguirlo, puoi usare:

./cinescript < TEST.cinema

File di test

Il file TEST.cinema contiene esempi completi di input. È utile per verificare il funzionamento del compilatore.

Esempi validi

@ Variabili Numeriche

azione a = 5

azione b = 3.20

@ Operazioni matematiche

azione sum = somma(a, b)

scena sum

@ Variabili di Testo

dramma s1 = "Ciao "

dramma s2 = "Mondo"

@ Concatenazione di stringhe

dramma saluto = s1 + s2

scena saluto

@ Blocco Condizionale

if (a > b){

azione m = 4.5

azione n = 2

azione prodotto = prodotto(m, n)

scena prodotto

}

@ Visualizzare la Symbol Table

CAST

@ Termina programma

FINALE