Esercizio - Grammatiche (1)
Espressioni algebriche

G = ⟨VT,VN,P,S⟩, dove:

VT = { +, -, *, /, (,), 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }

VN = {<E>, <T>,
<F>, <num>,
<cifra>, <cifra-non-nulla> }

S = <E>

Esercizio - Grammatiche (1)
Espressioni Algebriche

P = {

<E> ::= <E> + <T> | <E> - <T> | <T>
<T> ::= <T> * <F> | <T> / <F> | <F>
<F> ::= <cifra> | (<E>)
<cifra> ::= 0 | <cifra-non-nulla>
<cifra-non-nulla> ::= 1|2|3|4|5|6|7|8|9

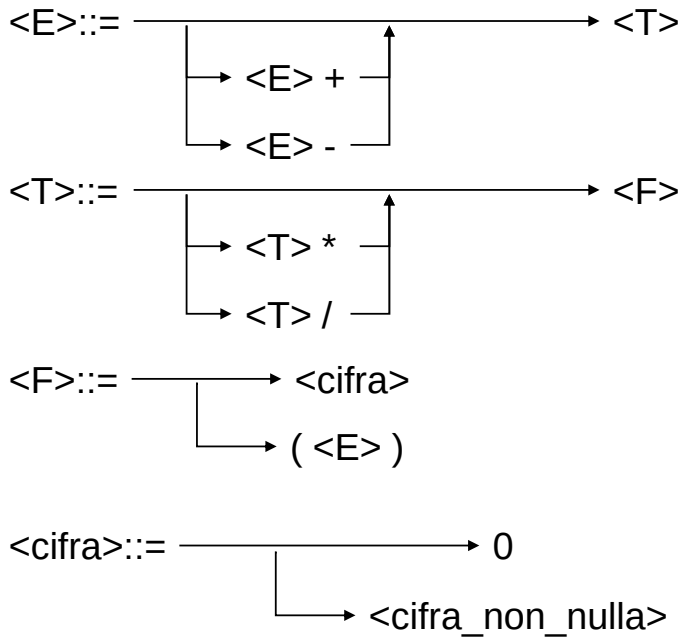
}

Disegnare il diagramma sintattico di tale grammatica.
Determinare poi se le seguenti frasi fanno parte del linguaggio generato da questa grammatica o no, e disegnarne l'albero di derivazione sintattica, e la derivazione left-most:

1. 1 + 4 / 2
2. (1 + 4) / 2

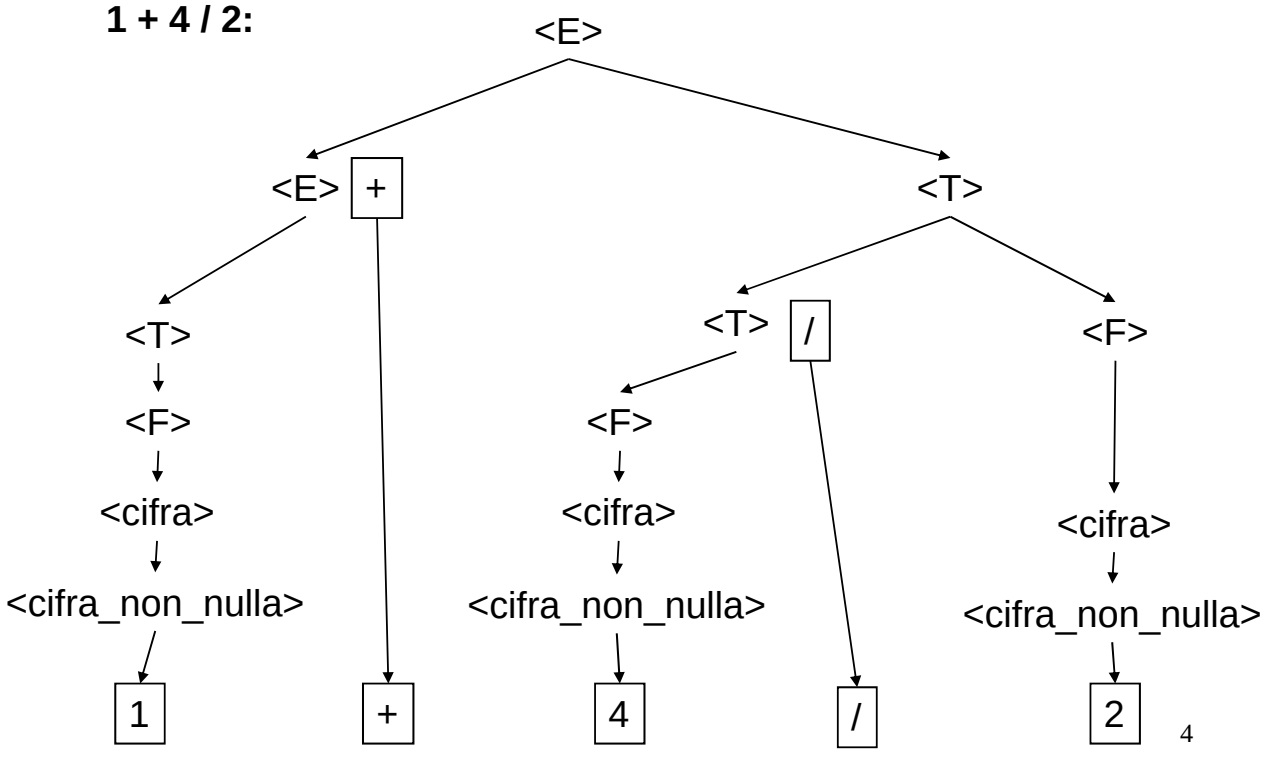
Esercizio - Grammatiche (1)
Soluzione

Diagramma sintattico:



Esercizio - Grammatiche (1)
Soluzione

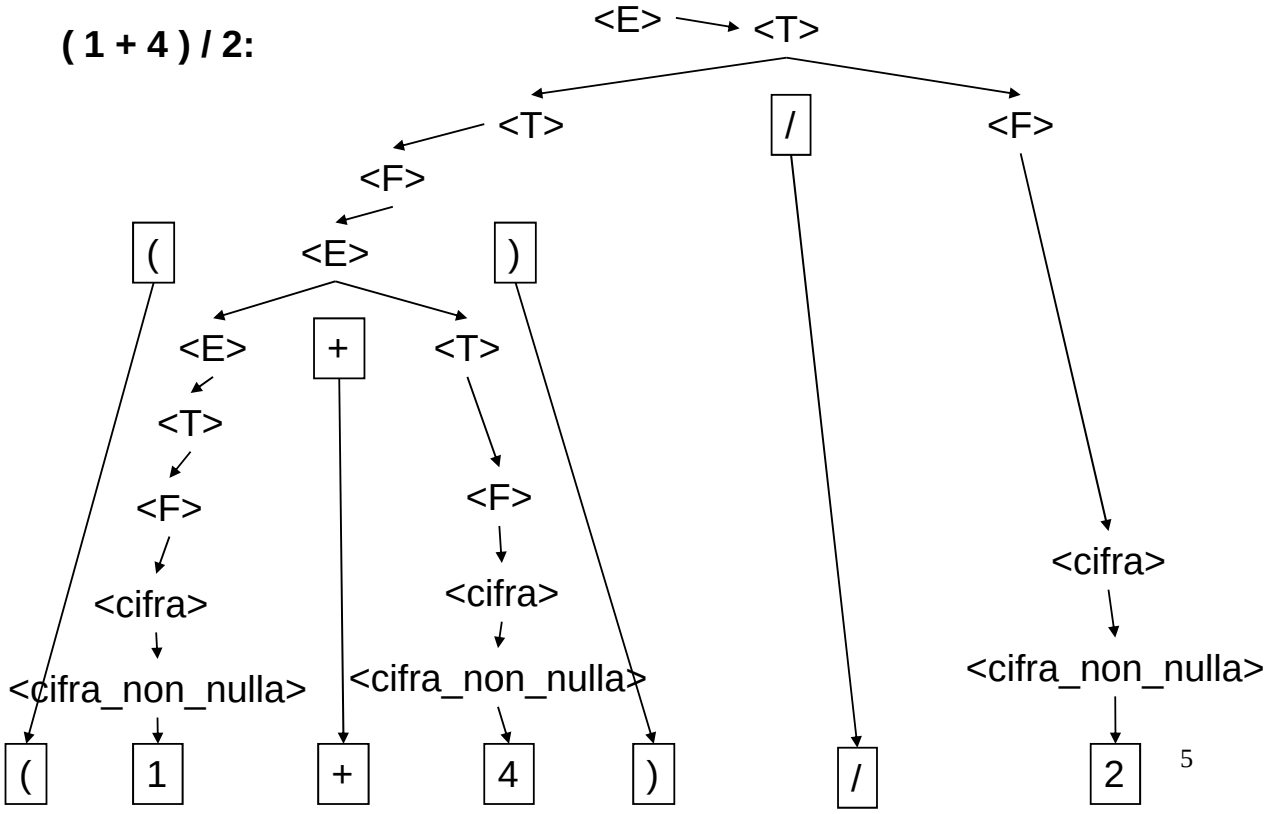
1 + 4 / 2:



Esercizio - Grammatiche (1)

Soluzione

(1 + 4) / 2:



Esercizio - Grammatiche (1)

Soluzione

Derivazione left-most:

1. 1 + 4 / 2

<E> → <E> + <T>
→ <T> + <T>
→ <F> + <T>
→ <cifra> + <T>
→ <cifra_non_nulla> + <T>
→ 1 + <T>
→ 1 + <T> / <F>
→ 1 + <F> / <F>
→ 1 + <cifra> / <F>
→ 1 + <cifra_non_nulla> / <F>
→ 1 + 4 / <F>
→ 1 + 4 / <cifra>
→ 1 + 4 / <cifra_non_nulla>
→ 1 + 4 / 2

Esercizio - Grammatiche (1)
Soluzione

Derivazione left-most:

2. (1 + 4) / 2

<E> → <T> → <T> / <F>
→ <F> / <F> → (<E>) / <F>
→ (<E> + <T>) / <F> → (<T> + <T>) / <F>
→ (<F> + <T>) / <F> → (<cifra> + <T>) / <F>
→ (<cifra_non_nulla> + <T>) / <F> → (1 + <T>) / <F>
→ (1 + <F>) / <F> → (1 + <cifra>) / <F>
→ (1 + <cifra_non_nulla>) / <F> → (1 + 4) / <F>
→ (1 + 4) / <cifra> → (1 + 4) / <cifra_non_nulla>
→ (1 + 4) / 2

Esercizio - Grammatiche (2)
Compito del 13 Settembre 2007

Si consideri la grammatica G con scopo S, simboli non terminali {X, Y, F, G} e simboli terminali {a,b,c,0,1,2}:

S ::= GXY | FX
X ::= FX | FYX
Y ::= G | GYF
F ::= 0 | 1 | 2
G ::= a | b | c

La stringa "12ac0b" appartiene al linguaggio generato da tale grammatica?

In caso affermativo, se ne mostri la derivazione left-most.

Esercizio - Grammatiche (2)
Compito del 13 Settembre 2007

La stringa data inizia col simbolo terminale '1' e termina col simbolo terminale 'b'.

Per poter ottenere una tale stringa, dovrebbe esserci una produzione con F come primo simbolo non terminale e G come ultimo simbolo non terminale…

… ma nessuna delle produzioni date ha tali caratteristiche. Infatti l'unica produzione che può terminare con G è S::= GXY, ma G come primo simbolo non terminale è incompatibile con la stringa data.

La stringa data non appartiene quindi al linguaggio generato dalla grammatica data.

Esercizio - RAPPRESENTAZIONE (3)

Un elaboratore rappresenta i numeri interi su 8 bit in complemento a 2. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

39 + (- 91)

Esercizio - RAPPRESENTAZIONE (3)

+39 -> 00100111

91 -> 01011011

10100100

10100101 -> -91

Conversione del valore assoluto in base 2

Inversione dei bit

Aggiungo 1

Si esegue la somma:

```
 00100111 +
 10100101
 ----------
11001100
```

11001100

00110011

00110100 -> 52 (base 10)

Complemento a 2 – quando c'è overflow?

Nella rappresentazione a complemento a 2, l'overflow può accadere solo in conseguenza di:

- Somma di due numeri positivi
- Somma di due numeri negativi

L'overflow si verifica quando uno dei due ultimi bit di riporto è a uno, ma non lo sono entrambi (xor degli ultimi due bit di riporto)

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri positivi senza overflow (caso su 4 bit):

```
(riporto)      0 0 0 0
3       ->     0011
4       ->     0100
              ---------
               0111
```

xor(0, 0) = 0, non c'è overflow!

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri positivi *con* overflow (caso su 4 bit):

```
(riporto)      0 1 1 1
3       ->     0011
5       ->     0101
              ---------
               1000
```

xor(0, 1) = 1, OVERFLOW!!!

Sommando due numeri positivi, è stato modificato il bit del segno…
… deve essere successo qualcosa!!!

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri negativi *senza* overflow (caso su 4 bit):

(riporto) 1 1 0 0

-3 -> 1101

-4 -> 1100

1001

xor(1, 1) = 0, no overflow

Il bit del segno è rimasto invariato…
… ed entrambi i riporti sono ad 1…

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri negativi *con* overflow (caso su 4 bit):

(riporto) 1 0 0 0

-3 -> 1101

-6 -> 1010

0111

xor(1, 0) = 1, OVERFLOW!!!

Il bit del segno è stato modificato…
…cosa è successo ?

Esercizio - RAPPRESENTAZIONE (4)

Un elaboratore rappresenta i numeri interi su 8 bit in complemento a 2. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

(-13) - (-91)

Esercizio - RAPPRESENTAZIONE (4)

```
13 ->  00001101
       11110010
       11110011 -> -13
```

```
91 ->  01011011
       10100100
       10100101 -> -91
```

```
(prestito) ->            0  1  10
-13     ->      1  1  1  1  0  0  1  1   -
-91     ->      1  0  1  0  0  1  0  1
                ---------------------------
                0  1  0  0  1  1  1  0
```

```
01001110 ->  78 in base 10
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    {  ++N; }

    do {
       if ((N%2) == 1)
              v = v + N;
       else
              v = v - N;
       N--;
    } while (v > 4);

    for (i=0; i<N; i++)
       v = v - 1;
    for (i=0; i < (i?N:0); )
       v = 10;
    return (0);
}
```


Esercizio analisi (5)

Il programma compila correttamente, e al termine dell'esecuzione le variabili `v` ed `N` valgono rispettivamente -2 e 3

- Le prime due istruzioni pongono v a 0, e poi incrementano il valore di N, che da 4 viene a valere 5.
- Il primo ciclo che poi si incontra nel codice è di tipo do..while, perciò almeno una volta viene eseguito. In particolare, l'istruzione di if controlla: se N è dispari allora v viene incrementato di N, altrimenti v viene decrementato di N. Il ciclo termina quando la condizione (v>4) non vale più. Alla prima iterazione dunque v viene a valere 5, ed N è decrementato a 4; alla seconda iterazione v viene a valere 1 ed N decrementato a 3: Siccome la condizione di continuazione del ciclo non è più verificata, il ciclo si interrompe.
- Il secondo ciclo che si incontra decrementa di 1 il valore di v, per N volte. Quindi al termine v viene a valere -2.
- Il terzo ciclo presente nel programma non viene eseguito, poichè la condizione è immediatamente non verificata.

Esercizio - CICLI (6)

Dato un numero intero **a**, definito tramite una variabile, scrivere un programma che calcoli il valore della seguente espressione:

$$\sum_{i=1}^{a} \sum_{j=1}^{i} j$$

Esercizio - CICLI (6)

```
#include <stdio.h>

int main()
{       int a, i, j;
        int somma;

        printf("Inserisci un numero: ");
        scanf("%d", &a);

        somma = 0; //elemento neutro della somma
        for (i=1; i <= a; i++) {
                for (j=1; j <= i; j++) {
                        somma = somma+j;
                }
        }

        printf("La somma vale: %d", somma);

        return 0;
}
```

Esercizio – CICLI (7)

Si realizzi un programma che, partendo da una base **a** ed un limite **n**, calcoli la seguente funzione:

$$\sum_{i=0}^{n} a^i$$

Esercizio - CICLI (7)

```
#include <stdio.h>

int main()
{       int a, n, i, j;
        int somma;
        int prod;

        somma = 0; //elemento neutro della somma
        printf("Inserisci la base ed il numero di cicli: ");
        scanf("%d%d", &a, &n);

        for (i=0; i <= n; i++) {
                prod = 1; //elemento neutro del prodotto
                for (j=1; j <= i; j++) {
                        prod = prod * a;
                }
                somma = somma + prod;
        }

        return 0;
}
```

Esercizio – CICLI (7) - variante

```
#include <stdio.h>

int main()
{       int a, n, i;
        int somma;
        int prod;

        somma = 0; //elemento neutro della somma
        prod  = 1; //elemento neutro del prodotto
        printf("Inserisci la base ed il numero di cicli: ");
        scanf("%d%d", &a, &n);

        for (i=0; i <= n; i++) {
                if (i>0) {
                        prod = prod * a;
                }
                somma = somma + prod;
        }

        return 0;
}
```

Esercizio (8)

Si realizzi un programma che legga un intero N da tastiera, e stampi a video il risultato della seguente sommatoria:

$$\sum_{i=0}^{N} \left[(-1)^i \frac{4}{2 * i + 1} \right]$$

Una volta calcolato e stampato il valore a video, il programma deve chiedere un nuovo numero all'utente e ripetere il calcolo. Il programma deve terminare solo qualora l'utente inserisca un valore negativo.

Esercizio (8)

```
#include <stdio.h>

int main()
{
        int num, i, pari;
        double pi;

        do {
                pi = 0;
                printf("Inserire numero: ");
                scanf("%d", &num);
                if (num >=0) {
                        for (i=0; i<=num; i++) {
                                pari = i%2;
                                pi = pi + (pari?-1:1)*(4.0/(2*i +1));
                        }
                        printf("Pi Greco vale: %f\n\n", pi);
                }
        } while (num != -1);
        return 0;
}
```

Esercizio (9)

CALCOLO DEGLI INTERESSI BANCARI

Si progetti in C un programma che legge un intero, rappresentante un ammontare di euro; di seguito il programma deve leggere un tasso d'interesse (in percentuale), ed un numero di anni.
Il programma deve stampare, in uscita, per ogni anno, come l'ammontare cresce con gli interessi. Si ricordi che l'interesse si calcola con la seguente formula:

$$Cfin = Cin * \left(1 + \frac{r}{100}\right)^{N}$$

Dove Cfin è il capitale finale, Cin è quello iniziale, r è l'interesse, e N rappresenta il numero di anni in cui si applicano gli interessi.

Continua >

Esercizio (9)

Supponiamo che il capitale iniziale sia di 1000 €, con un tasso del 3%, per un periodo di 3 anni. L'output stampato deve avere all'incirca questo aspetto:

```
Capitale iniziale: 1000€
Dopo 1 anno: 1030 €
Dopo 2 anni: 1060.90 €
Dopo 3 anni: 1092.72 €
```