

Esercizio 1

Specificare la definizione regolare relativa ai simboli lessicali **intconst** (costante intera) e **id** (identificatore), sulla base dei seguenti vincoli:

- Una costante intera con più di una cifra non può iniziare con uno zero;
- Una costante intera diversa da zero è opzionalmente qualificata da un segno (parte integrante della costante);
- Un identificatore inizia con un carattere alfabetico ed è seguito da una sequenza (anche vuota) di caratteri alfanumerici, eventualmente separati da caratteri underscore '_';
- In un identificatore, non sono ammesse sequenze di due o più underscore consecutivi e l'underscore non può terminare l'identificatore.

Esercizio 1

Specificare la definizione regolare relativa ai simboli lessicali **intconst** (costante intera) e **id** (identificatore), sulla base dei seguenti vincoli:

- Una costante intera con più di una cifra non può iniziare con uno zero;
- Una costante intera diversa da zero è opzionalmente qualificata da un segno (parte integrante della costante);
- Un identificatore inizia con un carattere alfabetico ed è seguito da una sequenza (anche vuota) di caratteri alfanumerici, eventualmente separati da caratteri underscore '_';
- In un identificatore, non sono ammesse sequenze di due o più underscore consecutivi e l'underscore non può terminare l'identificatore.

lettera $\rightarrow [A-Za-z]$

cifra $\rightarrow [0-9]$

alfanum $\rightarrow \text{lettera} \mid \text{cifra}$

cifra-iniziale $\rightarrow [1-9]$

segno $\rightarrow + \mid -$

intconst $\rightarrow \text{segno?} (\text{cifra-iniziale cifra}^*) \mid 0$

id $\rightarrow \text{lettera alfanum}^* (_ \text{alfanum}^+)^*$

Esercizio 2

Specificare la definizione regolare relativa ai simboli lessicali **realconst** (costante reale) e **id** (identificatore), sulla base dei seguenti vincoli:

- Una costante reale si compone di una parte intera (obbligatoria) ed una parte decimale (opzionale);
- La parte intera è separata dalla parte decimale (se espressa) da un punto;
- Una costante reale include il segno '-' se negativa (ma non il segno '+', se positiva);
- La parte intera con più di una cifra non può iniziare con uno zero;
- Un identificatore inizia con un carattere alfabetico ed è seguito da una sequenza (anche vuota) di caratteri alfanumerici;
- Un identificatore non può essere più lungo di quattro caratteri.

Esercizio 2

Specificare la definizione regolare relativa ai simboli lessicali **realconst** (costante reale) e **id** (identificatore), sulla base dei seguenti vincoli:

- Una costante reale si compone di una parte intera (obbligatoria) ed una parte decimale (opzionale);
- La parte intera è separata dalla parte decimale (se espressa) da un punto;
- Una costante reale include il segno '-' se negativa (ma non il segno '+', se positiva);
- La parte intera con più di una cifra non può iniziare con uno zero;
- Un identificatore inizia con un carattere alfabetico ed è seguito da una sequenza (anche vuota) di caratteri alfanumerici;
- Un identificatore non può essere più lungo di quattro caratteri.

lettera $\rightarrow [A-Za-z]$

cifra $\rightarrow [0-9]$

alfanum $\rightarrow \text{lettera} \mid \text{cifra}$

cifra-iniziale $\rightarrow [1-9]$

segno $\rightarrow '-'$

intero $\rightarrow \text{cifra-iniziale cifra}^* \mid 0$

decimale $\rightarrow \text{cifra}^+$

realconst $\rightarrow \text{segno? intero ('.' decimale)?}$

id $\rightarrow \text{lettera alfanum? alfanum? alfanum?}$

Esercizio 3

Specificare la definizione regolare relativa al simbolo lessicale **indirizzo**, che rappresenta l'indirizzo civico di una persona, come nei seguenti esempi:

```
Angelo B. Rossi  
Via Europa 145  
25100 Brescia
```

```
Angelo B. Matteo Rossi  
Via Africa 12  
27100 Pavia
```

```
T. Bianchi  
Via Asia 2  
64123 Bastia Franca
```

L'indirizzo deve rispettare i seguenti vincoli lessicali:

- La prima riga specifica il nome (in generale, uno o più nomi) e il cognome (uno solo) della persona;
- La seconda riga specifica la via (unico nome preceduto dalla keyword Via) e il numero civico;
- La terza riga specifica il CAP e la città (eventualmente composta da più nomi);
- Ogni elemento sulla stessa riga è separato dal successivo mediante un **blank**;
- Ogni riga è separata dalla successiva mediante un **newline**;
- Ogni identificatore (nome, cognome, indirizzo, città) inizia con una maiuscola ed è seguito da una o più minuscole;
- Un nome (o più) della persona può essere abbreviato dal primo carattere seguito dal punto;
- Il numero civico è composto da non più di 3 cifre e non può iniziare con una sequenza di zeri;
- Il CAP è composto da cinque cifre.

Esercizio 3

Specificare la definizione regolare relativa al simbolo lessicale **indirizzo**, che rappresenta l'indirizzo civico di una persona, come nei seguenti esempi:

```
Angelo B. Rossi  
Via Europa 145  
25100 Brescia
```

```
Angelo B. Matteo Rossi  
Via Africa 12  
27100 Pavia
```

```
T. Bianchi  
Via Asia 2  
64123 Bastia Franca
```

maiuscola $\rightarrow [A-Z]$

minuscola $\rightarrow [a-z]$

cifra $\rightarrow [0-9]$

nonzero $\rightarrow [1-9]$

id $\rightarrow \text{maiuscola minuscola}^+$

nome $\rightarrow \text{id} \mid \text{maiuscola}\backslash.$

num $\rightarrow \text{nonzero cifra? cifra?}$

cap $\rightarrow \text{cifra cifra cifra cifra cifra}$

indirizzo $\rightarrow \text{nome (blank nome)}^* \text{blank id newline}$

$\text{Via blank id blank num newline}$

cap (blank id)^+

Esercizio 4

Specificare la definizione regolare relativa ad una tabella (non vuota), in cui ogni riga rappresenta nome, cognome e codice fiscale di una persona, come nel seguente esempio,

```
Luigi Rossi ROSLGI84L14M634U  
Anna Bianchi BCIANN92A07H584X  
Enrico Verdi VRDERC67M13S124Y
```

sulla base dei seguenti vincoli lessicali:

- Nome e cognome iniziano con una lettera maiuscola, seguita da una o più lettere minuscole;
- Il codice fiscale, composto da lettere maiuscole e cifre, è suddiviso in quattro parti: tre lettere per il cognome, tre lettere per il nome, la data di nascita (due cifre per l'anno, una lettera per il mese, giorno del mese compreso tra 01 e 31) ed una stringa alfanumerica di cui il primo carattere e l'ultimo sono lettere mentre i tre caratteri intermedi sono cifre;
- Nome e cognome sono seguiti da uno spazio;
- Ogni riga, ad eccezione dell'ultima, è seguita da un newline.

Esercizio 4

Specificare la definizione regolare relativa ad una tabella (non vuota), in cui ogni riga rappresenta nome, cognome e codice fiscale di una persona, come nel seguente esempio:

```
Luigi Rossi ROSLGI84L14M634U
Anna Bianchi BCIANN92A07H584X
Enrico Verdi VRDERC67M13S124Y
```

maiuscola $\rightarrow [A-Z]$

minuscola $\rightarrow [a-z]$

cifra $\rightarrow [0-9]$

spazio $\rightarrow ' '$

newline $\rightarrow '\n'$

id \rightarrow **maiuscola minuscola**⁺

anno \rightarrow **cifra cifra**

mese \rightarrow **maiuscola**

giorno $\rightarrow 0 [1-9] \mid [1-2] [0-9] \mid 3 [0-1]$

cf \rightarrow **maiuscola maiuscola maiuscola maiuscola maiuscola maiuscola**
anno mese giorno

maiuscola cifra cifra cifra maiuscola

persona \rightarrow **id spazio id spazio cf**

tabella \rightarrow **persona (newline persona)***

Esercizio 5

Specificare la definizione regolare relativa ad una lista (anche vuota) di elementi, in cui ogni elemento è un identificatore o un numero, come nel seguente esempio,

```
[a2b, 25, 0, AAzxy, 0.12, -13.00, stella, 100, S289aa, 123.65, zeta]
```

sulla base dei seguenti vincoli lessicali:

- Un identificatore inizia con una lettera ed è seguito da almeno due caratteri alfanumerici.
- Un numero è composto da una parte intera (una o più cifre) ed opzionalmente da una parte decimale (un punto seguito da due cifre); la parte intera con più di una cifra non può iniziare con uno zero; un numero può avere un segno negativo (ma non positivo).
- Gli elementi della lista sono separati tra loro da una virgola e da uno spazio.

Esercizio 5

Specificare la definizione regolare relativa ad una lista (anche vuota) di elementi, in cui ogni elemento è un identificatore o un numero, come nel seguente esempio,

```
[a2b, 25, 0, AAzxy, 0.12, -13.00, stella, 100, S289aa, 123.65, zeta]
```

sulla base dei seguenti vincoli lessicali:

- Un identificatore inizia con una lettera ed è seguito da almeno due caratteri alfanumerici.
- Un numero è composto da una parte intera (una o più cifre) ed opzionalmente da una parte decimale (un punto seguito da due cifre); la parte intera con più di una cifra non può iniziare con uno zero; un numero può avere un segno negativo (ma non positivo).
- Gli elementi della lista sono separati tra loro da una virgola e da uno spazio.

lettera \rightarrow [a-zA-Z]

nonzero \rightarrow [1-9]

cifra \rightarrow 0 | nonzero

spazio \rightarrow ' '

punto \rightarrow '.'

alfanum \rightarrow lettera | cifra

id \rightarrow lettera alfanum alfanum alfanum*

intero \rightarrow 0 | nonzero cifra*

num \rightarrow -? intero (punto cifra cifra)?

elem \rightarrow id | num

lista \rightarrow [(elem (, spazio elem)*)?]

Esercizio 6

Specificare la definizione regolare relativa ad una lista (anche vuota) di tuple, in cui ogni elemento di una tupla può essere una stringa, un numero intero, un numero reale o una costante booleana, come nel seguente esempio,

```
[ ( "alfa",+10,12.34,true) ( "\"beta\"",false) ( "Gamma 25",-0.12,false) ]
```

sulla base dei seguenti vincoli lessicali:

- Ogni tupla (racchiusa tra parentesi) contiene almeno due elementi, separati tra loro da una virgola.
- Le tuple non hanno separatori tra loro.
- Una stringa (anche vuota) è delimitata da due doppi apici e può contenere caratteri alfanumerici, spazi e doppi apici (per evitare ambiguità, questi ultimi devono essere preceduti dal backslash).
- Un numero intero diverso da zero non può iniziare con la cifra 0.
- Un numero reale ha una parte intera ed una parte decimale, separate da un punto.
- La parte intera diversa da zero di un numero reale non può iniziare con la cifra 0.
- I numeri possono essere preceduti dal segno.

Esercizio 6

Specificare la definizione regolare relativa ad una lista (anche vuota) di tuple, in cui ogni elemento di una tupla può essere una stringa, un numero intero, un numero reale o una costante booleana, come nel seguente esempio:

```
[ ("alfa",+10,12.34,true) ("\"beta\"",false) ("Gamma 25",-0.12,false) ]
```

lettera \rightarrow [a-zA-Z]

nonzero \rightarrow [1-9]

cifra \rightarrow 0 | **nonzero**

punto \rightarrow '.'

apici \rightarrow ''

alfanum \rightarrow **lettera** | **cifra**

stringa \rightarrow **apici** (**alfanum** | **spazio** | '\bapici')* **apici**

intero \rightarrow (+|-)?(0 | **nonzero cifra***)

reale \rightarrow **intero** **punto** **cifra**⁺

booleano \rightarrow true | false

elemento \rightarrow **stringa** | **intero** | **reale** | **booleano**

tupla \rightarrow '('(**elemento** (**elemento**)⁺)'

lista \rightarrow '['**tupla***']'

Esercizio 7

Specificare la definizione regolare relativa ad un insieme (anche vuoto) di identificatori, come nel seguente esempio,

```
{Alfa, Beta22, Gamma_1_delta, Luna_Sole_stelle, Xilofono_1_2_345_omega}
```

sulla base dei seguenti vincoli lessicali:

- ogni identificatore è separato dal successivo da una virgola seguita da uno spazio;
- un identificatore inizia con una lettera maiuscola ed è seguito da zero o più caratteri alfanumerici;
- un identificatore può includere caratteri underscore;
- un underscore non può seguire un altro underscore e nemmeno essere l'ultimo carattere dell'identificatore.

Esercizio 7

Specificare la definizione regolare relativa ad un insieme (anche vuoto) di identificatori, come nel seguente esempio,

```
{Alfa, Beta22, Gamma_1_delta, Luna_Sole_stelle, Xilofono_1_2_345_omega}
```

sulla base dei seguenti vincoli lessicali:

- ogni identificatore è separato dal successivo da una virgola seguita da uno spazio;
- un identificatore inizia con una lettera maiuscola ed è seguito da zero o più caratteri alfanumerici;
- un identificatore può includere caratteri underscore;
- un underscore non può seguire un altro underscore e nemmeno essere l'ultimo carattere dell'identificatore.

maiuscola $\rightarrow [A-Z]$

minuscola $\rightarrow [a-z]$

lettera \rightarrow **maiuscola** | **minuscola**

cifra $\rightarrow [0-9]$

underscore $\rightarrow \text{'_}'$

spazio $\rightarrow \text{' '}$

alfanum \rightarrow **lettera** | **cifra**

id \rightarrow **maiuscola** ((**underscore** ? **alfanum**)⁺)*

insieme $\rightarrow \text{'\{ (id (, spazio id)*) ? '\}'}$

Esercizio 8

Specificare la definizione regolare relativa ad una lista (anche vuota) di record composti da tre elementi: una parola, un numero (senza segno) ed un insieme (anche vuoto, eventualmente disomogeneo) di parole o numeri,

```
[ (sole, 10, {}), (12_beta, 0.12, {0, 12.35, 21}), (Gamma_1_2_tre, 128.09, {luna, 2_stelle}) ]
```

sulla base dei seguenti vincoli lessicali:

- Non è ammessa spaziatura;
- Una parola è composta da caratteri alfanumerici ed underscore (questi ultimi non possono essere consecutivi, né iniziare o terminare la parola);
- Un numero può essere un intero o un reale (con parte decimale obbligatoria);
- Non sono ammessi numeri con prefisso di zeri non significativi.

Esercizio 8

Specificare la definizione regolare relativa ad una lista (anche vuota) di record composti da tre elementi: una parola, un numero (senza segno) ed un insieme (anche vuoto, eventualmente disomogeneo) di parole o numeri:

```
[ (sole,10,{ }),(12_beta, 0.12,{0,12.35,21}),(Gamma_1_2_tre,128.09,{luna,2_stelle}) ]
```

lettera \rightarrow [a-zA-Z]

nonzero \rightarrow [1-9]

cifra \rightarrow 0 | nonzero

underscore \rightarrow '_'

punto \rightarrow '.'

alfanum \rightarrow lettera | cifra

parola \rightarrow alfanum⁺ (underscore alfanum⁺)^{*}

intera \rightarrow 0 | (nonzero cifra^{*})

decimale \rightarrow cifra⁺

numero \rightarrow intera (punto decimale)?

atomo \rightarrow parola | numero

insieme \rightarrow { (atomo (, atomo)^{*})? }

record \rightarrow '(' parola, numero, insieme ')'

lista \rightarrow '[' (record (, record)^{*})? ']'

Esercizio 9

Specificare la definizione regolare relativa ad una lista (anche vuota) di numeri complessi, come nel seguente esempio:

```
[ (1, 24.66), (0.12, +3) ( -1.845, -1.23E20 ) , (33E4, 26.80E-3) ]
```

sulla base dei seguenti vincoli lessicali:

- Un numero complesso è rappresentato da una coppia (parte reale, parte immaginaria);
- Ognuna delle due parti è rappresentata da un numero, eventualmente con segno, avente una parte intera, opzionalmente una parte decimale (almeno una cifra) e, infine, opzionalmente una parte esponenziale (rappresentata da E seguita da un intero, eventualmente con segno);
- La parte intera non contiene zeri non significativi;
- Il linguaggio è libero dal formato: ogni elemento lessicale (numero, parentesi, virgola) può essere separato dagli altri mediante spazi bianchi e/o tab e/o newline.

Esercizio 9

Specificare la definizione regolare relativa ad una lista (anche vuota) di numeri complessi, come nel seguente esempio:

```
[ (1, 24.66), (0.12, +3) ( -1.845, -1.23E20 ) , (33E4, 26.80E-3) ]
```

spazio \rightarrow ' ' | '\t' | '\n' | ϵ

nonzero \rightarrow [1-9]

cifra \rightarrow 0 | nonzero

punto \rightarrow '.'

intera \rightarrow 0 | (nonzero cifra*)

decimale \rightarrow cifra⁺

intera_con_segno \rightarrow (+|-)? intera

esponenziale \rightarrow E intera_con_segno

numero \rightarrow intera_con_segno (punto decimale)? esponenziale?

complesso \rightarrow spazio '(' spazio numero spazio , spazio numero spazio ')' spazio

lista \rightarrow spazio '[' (complesso (, complesso)*)? ']' spazio

Esercizio 10

Specificare la definizione regolare relativa ai seguenti simboli lessicali di un linguaggio di programmazione:

- *Identificatore*: stringa alfanumerica, di lunghezza compresa tra due e quattro caratteri, che inizia con una lettera e termina con una cifra;
- *Costante intera*: sequenza di cifre priva di zeri non significativi;
- *Costante stringa* (anche vuota): racchiusa tra doppi apici e contenente qualsiasi carattere diverso da doppi apici e newline;
- *Costante booleana*: **true** o **false**;
- *Commento*: stringa di caratteri che inizia con # e termina con newline.

Esercizio 10

Specificare la definizione regolare relativa ai seguenti simboli lessicali di un linguaggio di programmazione:

- *Identificatore*: stringa alfanumerica, di lunghezza compresa tra due e quattro caratteri, che inizia con una lettera e termina con una cifra;
- *Costante intera*: sequenza di cifre priva di zeri non significativi;
- *Costante stringa* (anche vuota): racchiusa tra doppi apici e contenente qualsiasi carattere diverso da doppi apici e newline;
- *Costante booleana*: **true** o **false**;
- *Commento*: stringa di caratteri che inizia con **#** e termina con newline.

```
lettera → [A-Za-z]
minuscola → [a-z]
nonzero → [1-9]
cifra → 0 | nonzero
alfanum → lettera | cifra
newline → '\n'
id → lettera alfanum? alfanum? cifra
intconst → nonzero cifra* | 0
strconst → \"~[\n\"]\"
boolconst → true | false
commento → #(\n)*\n
```

Esercizio 11

Specificare la definizione regolare relativa ai seguenti simboli lessicali di un linguaggio di programmazione:

- *Commento*: stringa (non vuota) di caratteri che inizia con -- e termina con newline;
- *Costante intera*: sequenza di cifre priva di zeri non significativi, con segno opzionale;
- *Costante reale*: costituita da una parte intera (obbligatoria, priva di zeri non significativi), un punto ed una parte reale (opzionale, priva di zeri non significativi), con segno opzionale;
- *Identificatore*: sequenza di caratteri alfanumerici, eventualmente separati dal carattere underscore, il quale però non può stare né all'inizio né alla fine dell'identificatore, e nemmeno dopo un altro underscore.

Esercizio 11

Specificare la definizione regolare relativa ai seguenti simboli lessicali di un linguaggio di programmazione:

- *Commento*: stringa (non vuota) di caratteri che inizia con -- e termina con newline;
- *Costante intera*: sequenza di cifre priva di zeri non significativi, con segno opzionale;
- *Costante reale*: costituita da una parte intera (obbligatoria, priva di zeri non significativi), un punto ed una parte reale (opzionale, priva di zeri non significativi), con segno opzionale;
- *Identificatore*: sequenza di caratteri alfanumerici, eventualmente separati dal carattere underscore, il quale però non può stare né all'inizio né alla fine dell'identificatore, e nemmeno dopo un altro underscore.

```
lettera → [A-Za-z]
nonzero → [1-9]
cifra → 0 | nonzero
newline → '\n'
dot → '.'
underscore → '_'
commento → -- (~newline)+ newline
intconst → (+|-)? (nonzero cifra* | 0)
realpart → cifra* nonzero
realconst → intconst dot realpart?
id → lettera+ (underscore lettera+)*
```

Esercizio 12

Specificare la definizione regolare relativa ai seguenti simboli lessicali di un linguaggio di programmazione:

- *Commento*: stringa (non vuota) di caratteri che inizia con -- e termina con un tab oppure con un newline;
- *Costante intera*: sequenza di cifre priva di zeri non significativi, con segno opzionale per numeri diversi da 0;
- *Identificatore*: sequenza illimitata di caratteri alfanumerici, che inizia con due lettere e termina con due o tre cifre.

Esercizio 12

Specificare la definizione regolare relativa ai seguenti simboli lessicali di un linguaggio di programmazione:

- *Commento*: stringa (non vuota) di caratteri che inizia con `--` e termina con un tab oppure con un newline;
- *Costante intera*: sequenza di cifre priva di zeri non significativi, con segno opzionale per numeri diversi da 0;
- *Identificatore*: sequenza illimitata di caratteri alfanumerici, che inizia con due lettere e termina con due o tre cifre.

lettera \rightarrow `[A-Za-z]`

nonzero \rightarrow `[1-9]`

cifra \rightarrow `0` | **nonzero**

tab \rightarrow `'\t'`

newline \rightarrow `'\n'`

commento \rightarrow `-- (~(\bnewline | \btab))^+(\bnewline | \btab)`

intconst \rightarrow `((+|-)? (\bnonzero \bcifra*)) | 0`

id \rightarrow `\blettera \blettera \balpha*\ bcifra \bcifra \bcifra?`

Esercizio 13

Specificare la definizione regolare relativa ad un assegnamento di una variabile con una espressione aritmetica che coinvolge altre variabili e numeri, come nel seguente esempio:

```
a12 = b + C3 - D2s * 24 / z + 1
```

sulla base dei seguenti vincoli lessicali:

- L'assegnamento è specificato su una sola linea e termina con un newline;
- Ogni elemento nell'assegnamento è separato dal successivo (se esiste) da uno spazio;
- Una variabile è una stringa alfanumerica, di lunghezza massima di tre caratteri, che inizia con una lettera;
- I numeri sono interi senza segno e senza zeri non significativi.

Esercizio 13

Specificare la definizione regolare relativa ad un assegnamento di una variabile con una espressione aritmetica che coinvolge altre variabili e numeri, come nel seguente esempio:

```
a12 = b + C3 - D2s * 24 / z + 1
```

sulla base dei seguenti vincoli lessicali:

- L'assegnamento è specificato su una sola linea e termina con un newline;
- Ogni elemento nell'assegnamento è separato dal successivo (se esiste) da uno spazio;
- Una variabile è una stringa alfanumerica, di lunghezza massima di tre caratteri, che inizia con una lettera;
- I numeri sono interi senza segno e senza zeri non significativi.

lettera → [A-Za-z]

nonzero → [1-9]

cifra → 0 | nonzero

alfanum → lettera | cifra

spazio → ' '

newline → '\n'

num → nonzero cifra* | 0

id → lettera alfanum? alfanum?

operando → id | num

operatore → '+' | '-' | '*' | '/'

assegnamento → id spazio = spazio operando (spazio operatore spazio operando)* newline

Esercizio 14

Specificare la definizione regolare relativa ad una tabella (non vuota), in cui ogni riga rappresenta codice, nome, quantità e prezzo di un prodotto, come nel seguente esempio,

```
8892 Sedia 4 28.50
1366 Tavolo 1 125.75
3244 Lampada 2 74.00
```

sulla base dei seguenti vincoli lessicali:

- Il codice è una sequenza di quattro cifre che non può iniziare con zero;
- Il nome inizia con una lettera maiuscola, seguita da una o più lettere minuscole;
- La quantità è un numero intero maggiore di zero e minore di 100;
- Il prezzo è un numero maggiore di zero, con parte decimale di due cifre, senza zeri non significativi nella parte intera.
- Ogni elemento nella riga è separato dal successivo da uno spazio;
- Ogni riga, ad eccezione dell'ultima, è separata dalla successiva da un newline.

Esercizio 14

Specificare la definizione regolare relativa ad una tabella (non vuota), in cui ogni riga rappresenta codice, nome, quantità e prezzo di un prodotto, come nel seguente esempio,

```
8892 Sedia 4 28.50
1366 Tavolo 1 125.75
3244 Lampada 2 74.00
```

sulla base dei seguenti vincoli lessicali:

- Il codice è una sequenza di quattro cifre che non può iniziare con zero;
- Il nome inizia con una lettera maiuscola, seguita da una o più lettere minuscole;
- La quantità è un numero intero maggiore di zero e minore di 100;
- Il prezzo è un numero maggiore di zero, con parte decimale di due cifre, senza zeri non significativi nella parte intera.
- Ogni elemento nella riga è separato dal successivo da uno spazio;
- Ogni riga, ad eccezione dell'ultima, è separata dalla successiva da un newline.

maiuscola → [A-Z]

minuscola → [a-z]

nonzero → [1-9]

cifra → 0 | **nonzero**

spazio → ' '

newline → '\n'

nome → **maiuscola** **minuscola**⁺

codice → **nonzero** **cifra** **cifra** **cifra**

quantita → **nonzero** **cifra**[?]

prezzo → **nonzero** **cifra**^{*} '.' **cifra** **cifra** | 0 '.' (**nonzero** **cifra** | **cifra** **nonzero**)

prodotto → **codice** **spazio** **nome** **spazio** **quantita** **spazio** **prezzo**

tabella → **prodotto** (**newline** **prodotto**)^{*}

Esercizio 15

Specificare la definizione regolare relativa ad una tabella (non vuota), in cui ogni riga rappresenta codice, descrizione, dimensioni e peso di un prodotto, come nel seguente esempio:

```
324AX pila (2cm 8mm 1cm) 25g
158CE "lampada da tavolo" (20cm 18cm 28cm) 1.55Kg
208SG sedia (40cm 45cm 80cm) 6Kg
652PL "temperino in legno" (6mm 8cm 6mm) 22.5g
```

sulla base dei seguenti vincoli lessicali:

- Il codice inizia con tre cifre e termina con due lettere maiuscole, ma non può iniziare con uno zero.
- La descrizione è composta da una stringa non vuota di lettere minuscole, eventualmente separate da spazi. Nel caso in cui esistano spazi (e solo in tal caso), è richiesto l'uso dei doppi apici.
- Le dimensioni sono espresse da una terna di numeri interi (diversi da zero) racchiusa tra parentesi tonde, qualificati dall'unità di misura (mm o cm) e separati da uno spazio;
- Il peso si compone di una parte intera (diversa zero) ed opzionalmente da una parte decimale composta da una o due cifre. La parte decimale non può terminare con uno zero. Il peso è qualificato dall'unità di misura (g o Kg).
- Ogni elemento nella riga è separato dal successivo con uno spazio.
- Ogni riga, ad eccezione dell'ultima, è separata dalla successiva da un newline.

Esercizio 15

Specificare la definizione regolare relativa ad una tabella (non vuota), in cui ogni riga rappresenta codice, descrizione, dimensioni e peso di un prodotto, come nel seguente esempio:

```
324AX pila (2cm 8mm 1cm) 25g
158CE "lampada da tavolo" (20cm 18cm 28cm) 1.55Kg
208SG sedia (40cm 45cm 80cm) 6Kg
652PL "temperino in legno" (6mm 8cm 6mm) 22.5g
```

maiuscola → [A-Z]

minuscola → [a-z]

apici → \"

nonzero → [1-9]

cifra → 0 | **nonzero**

spazio → ' '

newline → '\n'

intero → **nonzero cifra***

decimale → '.' **cifra? nonzero**

undim → [mc]m

unpeso → K?g

codice → **nonzero cifra cifra maiuscola maiuscola**

parola → **minuscola⁺**

descrizione → **parola | (apici parola (spazio parola)⁺ apici)**

dimensione → **intero undim**

dimensioni → '(' **dimensione spazio dimensione spazio dimensione** ')'

peso → **intero decimale?**

prodotto → **codice spazio descrizione spazio dimensioni spazio peso**

tabella → **prodotto (newline prodotto)***

Esercizio 16

Specificare la definizione regolare relativa ai simboli **identifier** e **num**, sulla base dei seguenti vincoli lessicali:

- Un **identifier** è composto da almeno tre caratteri alfanumerici, ha un prefisso (non vuoto) di lettere ed un suffisso (anche vuoto) di cifre, come nei seguenti esempi: `alfa`, `ab3`, `abc`, `a34`, `M240`, `Lt2`, `gH27`, `z32158`.
- Un **num** è una costante (senza segno) intera o reale. Nel primo caso, non sono inclusi zeri non significativi. Nel secondo caso, sia la parte intera che la parte decimale (composta da almeno una cifra) non include zeri non significativi, come nei seguenti esempi: `0`, `2`, `10`, `124`, `0.2`, `12.01`, `345.3456700001`

Esercizio 16

Specificare la definizione regolare relativa ai simboli **identifier** e **num**, sulla base dei seguenti vincoli lessicali:

- Un **identifier** è composto da almeno tre caratteri alfanumerici, ha un prefisso (non vuoto) di lettere ed un suffisso (anche vuoto) di cifre, come nei seguenti esempi: `alfa`, `ab3`, `abc`, `a34`, `M240`, `Lt2`, `gH27`, `z32158`.
- Un **num** è una costante (senza segno) intera o reale. Nel primo caso, non sono inclusi zeri non significativi. Nel secondo caso, sia la parte intera che la parte decimale (composta da almeno una cifra) non include zeri non significativi, come nei seguenti esempi: `0`, `2`, `10`, `124`, `0.2`, `12.01`, `345.3456700001`

lettera $\rightarrow [A-Za-z]$

nonzero $\rightarrow [1-9]$

cifra $\rightarrow 0 \mid \text{nonzero}$

intero $\rightarrow \text{nonzero cifra}^* \mid 0$

decimale $\rightarrow '.' \text{ cifra}^* \text{ nonzero}$

num $\rightarrow \text{intero} \mid \text{decimale}$

identifier $\rightarrow \text{lettera}^+ (\text{lettera lettera} \mid \text{lettera cifra} \mid \text{cifra cifra}) \text{ cifra}^*$

Esercizio 17

Dato un alfabeto $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, si chiede di specificare l'espressione regolare delle stringhe che contengono al più due **b** non consecutivi.

Esercizio 17

Dato un alfabeto $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, si chiede di specificare l'espressione regolare delle stringhe che contengono al più due **b** non consecutivi.

$$(\mathbf{a} \mid \mathbf{c})^* \mid (\mathbf{a} \mid \mathbf{c})^* \mathbf{b} (\mathbf{a} \mid \mathbf{c})^* \mid (\mathbf{a} \mid \mathbf{c})^* \mathbf{b} (\mathbf{a} \mid \mathbf{c})^+ \mathbf{b} (\mathbf{a} \mid \mathbf{c})^*$$

Esercizio 18

Dato un alfabeto $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, si chiede di specificare l'espressione regolare delle stringhe che contengono esattamente tre **b** separati fra loro da altri caratteri.

Esercizio 18

Dato un alfabeto $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, si chiede di specificare l'espressione regolare delle stringhe che contengono esattamente tre **b** separati fra loro da altri caratteri.

$$(\mathbf{a} \mid \mathbf{c})^* \mathbf{b} (\mathbf{a} \mid \mathbf{c})^+ \mathbf{b} (\mathbf{a} \mid \mathbf{c})^+ \mathbf{b} (\mathbf{a} \mid \mathbf{c})^*$$