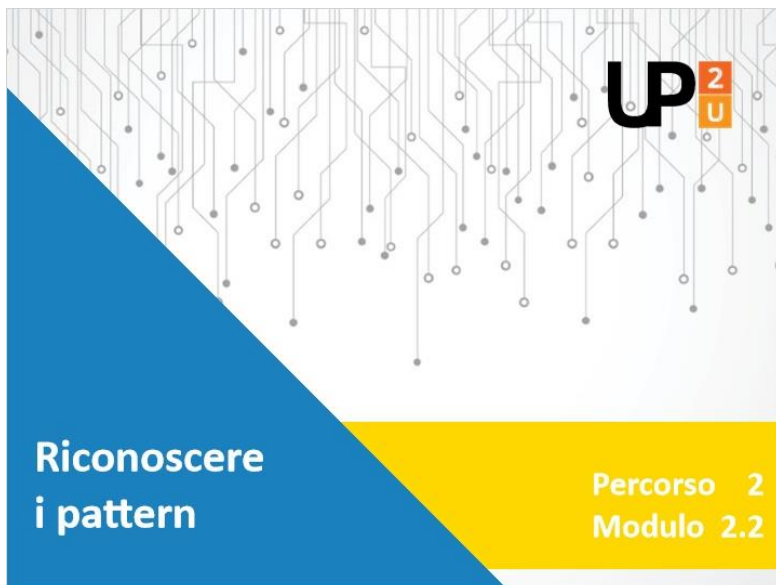


1.1 Riconoscere i pattern



1.2 La forma del testo

A cosa ti fa pensare questa riga di testo?

È evidente, penserai: è un indirizzo di posta elettronica!

Che sia vero o no, in questo momento quello che ci interessa è capire perché hai pensato subito a un indirizzo di posta. Probabilmente, il contenuto del testo per te non ha nessun significato.

Ma hai riconosciuto la “forma” della frase: c’è una chiocciola, preceduta da una parola o più parole separate da un punto, e seguita da almeno altre due parole, di cui le ultime sono sempre separate da un punto.

Decisamente, è un tipico indirizzo e-mail!

Senza neanche rendercene conto, prima ancora di leggere e capire il contenuto di un testo, ne osserviamo la struttura. Ed è così che a colpo d’occhio distinguiamo un numero di telefono da un codice fiscale, un sito web da un indirizzo stradale.

Quello che facciamo è riconoscere i “pattern”.

1.3 Cos'è un pattern

Non c'è un termine italiano per tradurre con esattezza "pattern".

Il pattern è una "forma", una disposizione stabile di elementi che può essere riconosciuta con certezza da un essere umano, ma anche da una macchina. L'operazione che abbiamo effettuato leggendo l'indirizzo di posta elettronica si chiama "pattern matching" e permette di verificare se un testo corrisponde a un determinato pattern. Il "pattern matching" è la base del riconoscimento di pattern (o "pattern recognition"), che a sua volta è il fondamento per l'analisi e la "comprensione" di un testo o di altri dati.

Questi sono indirizzi di posta elettronica validi?

È facile rispondere "no": nel primo caso ci sono due chiocciole, nel secondo uno spazio tra "ju" e "srtwax", nel terzo non c'è il punto prima di "biz", nel quarto manca del tutto la parte a sinistra della chiocciola.

Come possiamo insegnare il "pattern matching" a un computer?

Un modo elegante c'è: usando le "espressioni regolari"!

1.4 Le espressioni regolari

Le espressioni regolari sono un formalismo estremamente preciso e molto potente per descrivere i pattern:

- preciso per fare un modo che sia un computer a svolgere tutto il lavoro.
- potente per poterle utilizzare, come vedremo, in molte applicazioni diverse.

Un'espressione regolare è una sequenza di caratteri (tecnicamente si parla di "stringa di caratteri") che contiene informazioni su come una certa parte di testo deve essere strutturata.

La teoria delle espressioni regolari ha origine dallo studio del linguista Noam Chomsky sulle "gerarchie di grammatiche", sviluppato fin dagli anni '50. Ma sono state usate concretamente in informatica solo a partire dagli anni '80, con la creazione del Perl, un linguaggio di programmazione specializzato nella manipolazione dei testi.

Da quel momento, l'uso di espressioni regolari per il trattamento automatico dei testi si è diffuso ovunque e oggi tutti i linguaggi di programmazione ne fanno uso.

Ma come funziona un'espressione regolare?

1.5 Espressioni regolari al lavoro

Ecco un problema un po' più complesso di quello che abbiamo affrontato in precedenza:

riconoscere un codice fiscale.

Sappiamo che un codice fiscale contiene informazioni sul nome, sulla data e sul luogo di nascita. Per essere valido deve seguire alcune regole formali:

- È una stringa di 16 caratteri.
- I primi tre caratteri sono lettere dell'alfabeto, di solito sono consonanti, ma non sempre.
- I secondi tre caratteri sono lettere dell'alfabeto.
- Il settimo e l'ottavo carattere sono due cifre che vanno da "01" a "99".
- Il nono carattere è una lettera dell'alfabeto, ma non una lettera qualunque.
- Il decimo e l'undicesimo carattere sono due cifre, che vanno da "01" a "31" o da "41" a "71".
- Il dodicesimo carattere è una lettera dell'alfabeto. Le lettere accettate sono quelle da "A" a "M" più "Z".
- I successivi tre caratteri sono tre cifre, da "100" a "999" se il dodicesimo carattere è una "Z", o da "001" a "999" in caso contrario.
- Infine, il sedicesimo e ultimo carattere è una lettera dell'alfabeto.

In più, ci sono regole per evitare date inesistenti come il 31 aprile.

Pensi che questo insieme di regole sia troppo lungo? Allora possiamo concentrarle tutte in un'unica espressione regolare.

Troppo complicato? Non per un computer...

1.6 Come costruire espressioni regolari

Vediamo adesso come si costruisce un'espressione regolare.

In un'espressione regolare, i caratteri possono avere due funzioni:

- Rappresentare se stessi. In quel caso, la lettera "a" significa proprio "a".
- Rappresentare dei caratteri con funzioni speciali, chiamati anche "meta-caratteri".

Come si fa a capire se un certo carattere è un meta-carattere? Di solito, i meta-caratteri sono preceduti da una barra retroversa o backslash.

Così, per esempio:

- la lettera "d", da sola, significa "d";
- se preceduta da backslash, significa "una qualunque cifra decimale", cioè da "0" a "9".

Ma, attenzione: il backslash ha una funzione più complessa, perché viene usato anche per indicare che un meta-carattere deve essere trattato come un carattere letterale!

Ed ecco un elenco dei meta-caratteri più usati...

Come vedi le espressioni regolari possono essere piuttosto complicate. Ma... aiutiamoci con qualche esempio.

1.7 Come cercare "casa"

Ecco un'espressione regolare...

È composta dalla stringa "casa" che possiamo usare per verificare se nel testo che analizziamo questa stessa stringa c'è. L'esito sarà positivo con testi come "la casa nel bosco", "la finestra del casale", "casa mia, per piccina che tu sia" e "torno a casa".

Immaginiamo questi testi come scritti su righe diverse e proviamo a raffinare la ricerca. Se alla fine della mia espressione regolare aggiungo il metacarattere "dollar", che indica "fine riga" significa che sto cercando solo le parole "casa" che stanno a fine riga.

Puoi notare che solo l'ultima frase soddisfa questa condizione.

E se, invece, uso il metacarattere che indica "inizio riga", così... significa che voglio cercare solo le parole "casa" che stanno all'inizio di una riga di testo.

Soddisferà questa ricerca unicamente con "casa mia, per piccina che tu sia"...

1.8 Anche le "case"

Modifichiamo un po' la nostra espressione regolare di partenza...

Il punto con cui ho sostituito l'ultima "a" indica "un carattere qualsiasi".

L'espressione viene soddisfatta da tutte queste stringhe di testo...

... ma anche da queste...

Ma io sto cercando casa, non mi interessano "i casi della vita", né i formaggi sardi, né la cinematografia.

Allora devo affinare la ricerca inserendo le parentesi quadre, metacaratteri che definiscono una scelta.

Ecco, adesso ho un'espressione regolare che è soddisfatta solo quando la quarta lettera è "a" oppure "e"!

1.9 Qualcosa di utile

Questa espressione regolare indica una lettera maiuscola qualsiasi, cioè che è compresa nell'intervallo tra "A" e "Z". E se voglio non una ma due lettere maiuscole contigue? Potrei scrivere così:

... ma c'è un modo molto più elegante usando i quantificatori che indicano una ripetizione.

Come vedi, abbiamo usato un numero inserito tra i meta-caratteri "parentesi graffa", ponendolo immediatamente dopo l'elemento da moltiplicare".

E se invece di due lettere voglio cercare un numero a tre cifre?

Semplice, userò questa espressione regolare:

Le espressioni regolari si possono combinare tra loro. Per esempio, così:

Adesso, finalmente, abbiamo qualcosa di veramente utile e potente: un'espressione regolare che ci permette di riconoscere in un testo le attuali targhe automobilistiche italiane, formate da due lettere, tre cifre e altre due lettere!

1.10 Cosa ci facciamo con le espressioni regolari?

Abbiamo visto che con le espressioni regolari siamo in grado di insegnare a un computer a riconoscere determinati pattern in un testo.

E questo, a sua volta, apre la strada per elaborare il testo in molti modi:

- Cercare e contare quante volte appare un certo pattern, per esempio una qualunque data.
- Effettuare sostituzioni massive, per esempio aggiornare l'anno di tutte le date.
- Segmentare il testo in frasi o singole parole.
- Convalidare indirizzi di posta elettronica, date, numeri di telefono o codici fiscali in un modulo online.
- Creare filtri per guidare l'attività dei programmi che esplorano il web per analizzare i siti (si chiamano "spyder" e vengono usati dai motori di ricerca).
- Estrarre i dati significativi dalle pagine web (è un'operazione, chiamata "scraping", che serve sempre ai motori di ricerca).
- Effettuare ogni sorta di analisi e ricerche su testi in linguaggio naturale, su file di dati o su programmi.

E ora sicuramente avrai voglia di "giocare" un po' con questo strumento così potente!

1.11 Per fare esperienza

Tutti i moderni linguaggi di programmazione includono un sistema chiamato “regular expression engine” in grado di interpretare i formalismi delle espressioni regolari.

Ma se non sei un programmatore, ci sono in rete molti strumenti gratuiti con cui esercitarsi:

- Regex Crossword è un gioco on line, a metà strada tra le parole crociate e il popolare sudoku.
- Regular expressions 101 e Regex tester permettono di caricare un testo e scrivere espressioni regolari per individuare i pattern.

Ci sono poi diversi editor, cioè word processor specializzati per la stesura di programmi, che consentono di usare espressioni regolari per la ricerca e la sostituzione di stringhe di testo. Tra questi il diffusissimo Notepad++. [*pr. notpàd plas plas*]

Tutti strumenti per impadronirsi di uno strumento potente come le espressioni regolari. Uno strumento che ti aprirà le porte del mondo dell’analisi automatica del testo.