Francesca Parpinel - Claudio Pizzi

Dal Problema alla Soluzione

Guida Pratica per Principianti alla Programmazione in R

Giappichelli

Prefazione

In questo libro sono stati raccolti i materiali sviluppati e gli esercizi svolti dagli autori per il corso di Elementi di Informatica per l'Economia del corso di laurea triennale in Commercio Estero e Turismo dell'ateneo veneziano Ca' Foscari a partire dall'A.A. 2017/18.

Nel nostro mondo in continua e veloce evoluzione, in cui ormai sembra che tutto intorno possa essere gestito e governato da quella che oggigiorno è chiamata Intelligenza Artificiale (IA), è estremamente importante riuscire a districarsi e capire anche le dinamiche che sono proposte da tale tecnologia. Sempre più frequentemente si sente parlare di errori di interrogazioni a software di IA, quindi è immediato porsi alcuni quesiti. Ad esempio, come è possibile saper riconoscere un errore ed eventualmente correggerlo? A monte è necessario che l'utilizzatore di questo moderne tecnologie conosca il loro linguaggio per comprendere ed eventualmente intervenire con aggiustamenti, qualora ce ne fosse la necessità. Tra gli obbiettivi che hanno spinto gli autori a comporre questo libro, c'è proprio l'ambizione di fornire le basi per poter disporre di risposte.

L'intento del testo è quello di introdurre il lettore alla programmazione per sviluppare alcune competenze trasversali note come *soft skills* tra le quali ad esempio la capacità di risolvere problemi, o *problem solving*, Dunker (1969), e la capacità di avere una visione d'insieme del problema, il cosiddetto *system thinking*, Goleman (1995).

La scelta del linguaggio di programmazione da utilizzare nel corso è stata difficile poiché diversi linguaggi si prestavano a tale scopo. La scelta finale è ricaduta su *R* per varie ragioni, prima di tutto perché è uno tra i più utilizzati linguaggi di programmazione[1] e ormai noto anche a livello aziendale. Inoltre, si tratta di un software *freeware* rilasciato sotto la licenza Free Software Foundation's GNU General Public License (GPL) e multipiattaforma, cioè può essere eseguito su un'ampia varietà di piattaforme UNIX (come ad

[1] Ci sono fonti che forniscono un ranking per i linguaggi di programmazione, essi differiscono per la metrica utilizzata ma generalmente a parte qualche posizionamento diverso i linguaggi di programmazione più utilizzati sono gli stessi. Tra le fonti più note si ricorda il PYPL Popularity of Programming Languages (pypl.github.io/PYPL.html), il TIOBE Index (www.tiobe.com), IEEE Spectrum (spectrum.ieee.org) e il RedMonk (https://redmonk.com/). Per tutte le fonti R è tra i primi.

esempio Linux, FreeBSD), Windows e MacOS e più recentemente anche Android. Infine, *R* può essere utilizzato come valido strumento per lo studio di altre materie quantitative; si ricorda che *R* è nato come pacchetto statistico e linguaggio di programmazione completamente *Open Source*, R Core Team (2023). Tutto il software è liberamente scaricabile da https://www.r-project.org/.

Al fine di guidare il lettore nell'apprendimento attivo-esperienziale di quanto presentato nel testo, oltre ad esempi, di tanto in tanto, il lettore troverà sezioni come la seguente:

E ADESSO PROVA TU! 0.1.
Cerca l'attuale versione di *R*.

Si tratta di un invito ad esercitarsi in modo autonomo, attività fondamentale per apprendere e migliorare passo a passo le proprie conoscenze sul linguaggio di programmazione *R* e sviluppare le proprie competenze.

Al fine di semplificare l'utilizzo del software *R* si suggerisce di avvalersi dell'ambiente di sviluppo integrato (IDE) *Rstudio*, RStudio Team (2019)[2], che si basa sul programma *R* ma organizza lo spazio di lavoro in finestre dedicate agli *script*, alla gestione dei file, ai pacchetti e ai grafici, consentendo una gestione più snella e tracciabile del proprio lavoro.

Il libro è stato composto dagli autori usando il linguaggio della libreria `rmarkdown`, definita dai suoi stessi autori come *costruita per R Markdown, un ecosistema di pacchetti per creare documenti di calcolo in R*, Xie *et al.* (2018), e adattando alcune caratteristiche del linguaggio LaTeX, Lamport (1986).

Gli autori vogliono ringraziare i colleghi con cui hanno scambiato idee e suggerimenti per la scrittura del libro, in particolare ricordano fra tutti Antonella Basso, Silvia Bozza, Marco Corazza, Raffaele Pesenti e Paolo Pellizzari. Ma sono grati soprattutto a tutti gli studenti che si sono avvicendati in questi anni nella frequenza dei loro corsi, ispiratori di quello che è nato come un semplice eserciziario e qui ha assunto invece una connotazione più completa.

[2] Rstudio è scaricabile da https://posit.co/download/rstudio-desktop/.

Introduzione

Come ci spiega l'Enciclopedia Treccani, il termine informatica, coniato nel 1962 da P. Dreyfus, deriva dal francese **informatique**, contrazione di **information automatique**, ed indica la scienza che si occupa dell'utilizzo di procedure automatizzate per il trattamento dell'informazione.

Dal punto di vista etimologico, il vocabolo informatica deriva dai due vocaboli: *informazione* (dal latino informare cioè dare forma) e *automatica* (dal greco con il significato: che pensa da solo).

Quindi potremmo anche definire l'informatica come quella scienza applicata che si occupa del trattamento automatico dei dati (qui si preferisce utilizzare il termine dati piuttosto che informazione perché sono i dati che sono trattati e ai quali si *dà forma* trasformandoli in informazioni) mediante l'ausilio di strumenti ad hoc.

L'informatica ha quindi per oggetto lo studio dei fondamenti teorici dell'informazione, di come a livello logico l'informazione viene creata e delle tecniche per la sua implementazione e applicazione in sistemi elettronici automatizzati. Tali sistemi elettronici automatizzati sono detti sistemi informatici. Lo strumento principale utilizzato dall'informatica per rendere possibile il trattamento automatico dell'informazione è il computer.

L'utilizzo del computer permette quindi di progettare procedure automatizzate, cioè che possono essere svolte in modo automatico le quali consentono di ridurre al minimo la necessità di intervento umano. Questo comporta la riduzione degli errori riconducibili all'intervento umano e la possibilità di contenere il tempo necessario allo svolgimento di operazioni complesse o ripetitive. Ulteriore conseguenza è il miglioramento all'accesso ad elaborazioni che richiedono competenze poco comuni.

L'organizzazione di questo libro prevede un primo capitolo in cui è presentato il processo di formalizzazione del problema con alcuni richiami ad elementi base dell'informatica. Il capitolo successivo è invece più tecnico e rivolto alla familiarizzazione del linguaggio *R*.

Dal terzo capitolo si presentano alcuni problemi iniziando con esercizi semplici e legati

al mondo matematico a cui seguono due capitoli con problemi sempre più complessi. Per tutti questi quesiti si proporrà una soluzione predisponendo un programma scritto in linguaggio di programmazione *R*. L'ultimo capitolo è dedicato ad esempi più articolati e relativi all'ambito economico/aziendale. Come il lettore potrà vedere, nei primi esercizi le soluzioni ai problemi sono svolte in modo più approfondito rispetto ai quelli successivi; sono illustrati il *flowchart* e lo *pseudocodice*, spiegando in modo dettagliato la soluzione del problema in dettaglio. Negli esercizi a seguire è stata fatta la scelta di presentare uno solo dei due metodi di illustrazione dell'algoritmo lasciando al lettore il compito di risolvere il problema con l'altro metodo.

Un ulteriore capitolo, il settimo, è dedicato a proposte di esercizi con i quali il lettore può iniziare ad esercitarsi in completa autonomia.

Infine le Appendici riguardano l'installazione dei software usati nel libro per la programmazione, alcuni cenni di Algebra Booleana e le soluzioni della maggior parte degli esercizi proposti (E ADESSO PROVA TU!).

Informazioni su R

R nasce negli anni '90 ad opera di due studiosi dell'Università di Auckland, Ross Ihaka e Robert Gentlman ispirandosi al linguaggio *S*. Il linguaggio di programmazione *S* fu sviluppato negli anni '70 presso i Bell Laboratories da John M.Chambers e colleghi. Ihaka e Gentlman hanno ideato il sistema *R* come progetto *open source* nel 1995. Dal 1997, il progetto *R* fa riferimento al *R Core Group* mentre la prima versione di *R* è stata resa disponibile nel Febbraio 2000. *R* si diffonde velocemente come strumento operativo nella comunità dei ricercatori statistici data la sua semplicità di applicazione per le analisi statistiche e la sua disponibilità per diversi sistemi operativi (Microsoft Windows, macOS, Unix e Linux). In Tabella 1 è riportato un elenco delle versioni di *R* che sono state prodotte nel tempo.

Attualmente è ancora software libero a disposizione della comunità scientifica ma si sta diffondendo come strumento di elaborazione dati anche in alcune aziende e enti. A febbraio 2024 il PopularitY of Programming Language Index (PYPL index) posiziona *R* al 6 posto tra i linguaggi di programmazione, mentre spectrum, una rivista dall'Institute of Electrical and Electronics Engineers (IEEE) nel 2023 lo posiziona al 11 posto. L'R Core Group si preoccupa del costante aggiornamento delle versioni di *R* e dell'adeguamento ai diversi sistemi operativi dei desktop e dei server.

R è un programma che permette di effettuare analisi statistiche di base ed avanzate, ma non solo. Una sua caratteristica è quella di essere anche un ambiente di sviluppo for-

nendo un linguaggio di programmazione orientato agli oggetti e consentendo di ampliare e personalizzare l'applicabilità del *software* con la possibilità di costruire proprie funzioni ed espressioni.

La console di *R* a prima vista risulta poco attrattiva, in quanto anche le attuali versioni aggiornate di *R* sono fornite con un'interfaccia a cosiddetta *riga di comando*, cioè una finestra bianca in cui un cursore lampeggia aspettando l'inserimento di comandi. Per cercare di ovviare a questo problema, sono state implementate negli anni diverse interfacce grafiche; tipica è *Gretl* per le elaborazioni econometriche, e più recentemente *RStudio* che consente di lavorare agevolmente anche con il linguaggio di markup[3] per la produzione di documenti in diversi formati. Ad esempio il documento che state leggendo è stato creato con la suite **Rmarkdown**, della quale trovate maggiori informazioni al seguente link https://rmarkdown.rstudio.com. Per completezza, ricordiamo anche la presenza di una risorsa in *cloud* di *RStudio* (attualmente https://posit.cloud/) che offre la possibilità lavorare con il linguaggio *R* anche senza installare sul proprio computer il programma stesso.

La scelta di usare tale software in un corso d'*Informatica per l'Economia* è motivata dal fatto che oltre ad essere un programma open source, è uno strumento molto flessibile per gestire e analizzare dati, anche economici, che permette inoltre di creare dei codici di programmi e quindi *programmare* con le strutture tipiche della programmazione. Si sottolinea che la capacità di analizzare grandi insiemi di dati sta diventando una preziosa competenza nel mercato del lavoro, dato che le aziende dispongono di enormi quantità di dati che, se opportunamente elaborati, possono generare informazioni utili per le decisioni aziendali sia operative che strategiche.

[3] con il termine *linguaggio di markup* ci si riferisce ad un insieme di regole che permettono di creare una formattazione automatica di un testo attraverso l'utilizzo di marcatori.

Vers.	Data	N.	Vers.	Data	Nickname	Vers.	Data	Nickname
0.60	04/12/97		2.0.0	04/10/04		3.1.1	10/07/14	Sock it to Me
0.61	21/12/97		2.0.1	15/11/04		3.1.2	31/10/14	Pumpkin Helmet
0.61.1	10/01/98		2.1.0	18/04/05		3.1.3	09/03/15	Smooth Sidewalk
0.61.2	14/03/98		2.1.1	20/06/05		3.2.0	16/04/15	Full of Ingredients
0.61.3	02/05/98		2.2.0	06/10/05		3.2.1	18/06/15	World-Famous Astronaut
0.62	14/06/98		2.2.1	20/12/05		3.2.2	14/08/15	Fire Safety
0.62.1	14/06/98		2.3.0	24/04/06		3.2.3	10/12/15	Wooden Christmas-Tree
0.62.2	10/07/98		2.3.1	01/06/06		3.2.4	10/03/16	Very Secure Dishes
0.62.3	28/08/98		2.4.0	03/10/06		3.2.5	14/04/16	Very, Very Secure Dishes
0.62.4	23/10/98		2.4.1	18/12/06		3.3.0	03/05/16	Supposedly Educational
0.63	13/11/98		2.5.0	24/04/07		3.3.1	21/06/16	Bug in Your Hair
0.63.1	04/12/98		2.5.1	28/06/07		3.3.2	31/10/16	Sincere Pumpkin Patch
0.63.2	11/01/99		2.6.0	03/10/07		3.3.3	06/03/17	Another Canoe
0.63.3	05/03/99		2.6.1	26/11/07		3.4.0	21/04/17	You Stupid Darkness
0.64	07/04/99		2.6.2	08/02/08		3.4.1	30/06/17	Single Candle
0.64.1	07/05/99		2.7.0	22/04/08		3.4.2	28/09/17	Short Summer
0.64.2	02/07/99		2.7.1	23/06/08		3.4.3	30/11/17	Kite-Eating Tree
0.65	27/08/99		2.7.2	25/08/08		3.4.4	15/03/18	Someone to Lean On
0.65.1	06/10/99		2.8.0	20/10/08		3.5.0	23/04/18	Joy in Playing
0.90	22/11/99		2.8.1	22/12/08		3.5.1	02/07/18	Feather Spray
0.90.1	15/12/99		2.9.0	17/04/09		3.5.2	20/12/18	Eggshell Igloo
0.99	07/02/00		2.9.1	26/06/09		3.5.3	11/03/19	Great Truth
1.0	29/02/00		2.9.2	24/08/09		3.6.0	26/04/19	Planting of a Tree
1.0.1	14/04/00		2.10.0	26/10/09		3.6.1	05/07/19	Action of the Toes
1.1	15/06/00		2.10.1	14/12/09		3.6.2	12/12/19	Dark and Stormy Night
1.1.1	15/08/00		2.11.0	22/04/10		3.6.3	29/02/20	Holding the Windsock
1.2	15/12/00		2.11.1	31/05/10		4.0.0	24/04/20	Arbor Day
1.2.1	15/01/01		2.12.0	15/10/10		4.0.1	06/06/20	See Things Now
1.2.2	26/02/01		2.12.1	16/12/10		4.0.2	22/06/20	Taking Off Again
1.2.3	26/04/01		2.12.2	25/02/11		4.0.3	10/10/20	Bunny-Wunnies Freak Out
1.3	22/06/01		2.13.0	13/04/11		4.0.4	15/02/21	Lost Library Book
1.3.1	31/08/01		2.13.1	08/07/11		4.0.5	31/03/21	Shake and Throw
1.4	19/12/01		2.13.2	30/09/11		4.1.0	18/05/21	Camp Pontanezen
1.4.1	30/01/02		2.14.0	31/10/11	Great Pumpkin	4.1.1	10/08/21	Kick Things
1.5.0	29/04/02		2.14.1	22/12/11	December Snowflakes	4.1.2	01/11/21	Bird Hippie
1.5.1	17/06/02		2.14.2	29/02/12	Gift-Getting Season	4.1.3	10/03/22	One Push-Up
1.6.0	01/10/02		2.15.0	30/03/12	Easter Beagle	4.2.0	22/04/22	Vigorous Calisthenics
1.6.1	01/11/02		2.15.1	22/06/12	Roasted Marshmallows	4.2.1	23/06/22	Funny-Looking Kid
1.6.2	10/01/03		2.15.2	26/10/12	Trick or Treat	4.2.2	31/10/22	Innocent and Trusting
1.7.0	16/04/03		2.15.3	01/03/13	Security Blanket	4.2.3	15/03/23	Shortstop Beagle
1.7.1	16/06/03		3.0.0	03/04/13	Masked Marvel	4.3.0	21/04/23	Already Tomorrow
1.8.0	08/10/03		3.0.1	16/05/13	Good Sport	4.3.1	16/06/23	Beagle Scouts
1.8.1	21/11/03		3.0.2	25/09/13	Frisbee Sailing	4.3.2	31/10/23	Eye Holes
1.9.0	12/04/04		3.0.3	06/03/14	Warm Puppy	4.3.3	29/02/24	Angel Food Cake
1.9.1	21/06/04		3.1.0	10/04/14	Spring Dance	-	-	-

Tabella 1: Versioni di R

Capitolo 1

La formalizzazione dei problemi: l'informatica

Fin dalla nostra nascita la vita presenta problemi che devono essere risolti in autonomia o con l'aiuto di qualcuno. Per un bimbo che gattona il problema è come raggiungere il pelouche in fondo alla stanza, mentre per un bambino più grande è come costruire una macchinina con i mattoncini Lego; per una studentessa universitaria come superare l'esame di matematica, per una persona adulta come assemblare un mobile dell'Ikea. Quindi si può dire che il concetto di *problema* è innato nell'uomo perché nasce dell'esperienza di tutti i giorni. Ma come lo si potrebbe definire?

1.1 Dal problema alla soluzione

Un problema è una qualsiasi situazione che presentando difficoltà, ostacoli, discrepanza tra lo stato reale e quello desiderato richiede una soluzione o un'azione per superarlo.

Vale la pena sottolineare che affinché questa discrepanza tra reale e desiderato sia un problema è necessario che ci sia, o perlomeno sia immaginabile, una soluzione realisticamente implementabile. Al contrario ci troveremmo di fronte a un dato di fatto che non può essere alterato e pertanto non costituisce un problema.

Nelle prime fasi di vita di un essere umano la soluzione di un problema avviene mediante un processo di apprendimento per tentativi ed errori. Con il passare del tempo, lo sviluppo cognitivo tramite il ragionamento ipotetico deduttivo consente di trovare soluzioni che diventano sempre più sofisticate e articolate permettendo di affrontare e risolvere problemi sempre più complessi. Lo sviluppo di tale tipo di ragionamento permette di acquisire competenze di *problem solving* attraverso le quali si affronta, si analizza e risolve

un problema usando metodo e logica, Dunker (1969). Schematizzando tale procedimento potremo individuare le fasi descritte di seguito.

Il punto di partenza per risolvere un qualsiasi problema consiste nell'individuarlo cioè nel riuscire a mettere a fuoco la situazione che richiede un intervento o una decisione. La fase di *identificazione del problema* è importante perché permette di definire correttamente il problema delineandone i confini.

Nella seconda fase, quella di *analisi*, il problema viene studiato in tutti i suoi aspetti, individuando le variabili che hanno un qualche effetto su di esso facendo emergere quali dati ed informazioni sono indispensabili per la sua risoluzione.

Il terzo step prevede la *generazione di soluzioni* che si possono prospettare per risolvere il problema (individuazione di azioni e/o processi) e valutarne le conseguenze sulla situazione problematica identificata al punto iniziale.

Nel caso siano possibili molteplici soluzioni allora sarà necessario valutarle e confrontarle in termini di fattibilità, efficacia, efficienza, costi e benefici. Il risultato di questa fase di *valutazione delle soluzioni* è l'individuazione della soluzione migliore.

La fase di passaggio *dalla teoria alla pratica* è quella di implementazione in cui si traducono operativamente e si eseguono le azioni e i processi individuati nei passi precedenti. In questa fase potrebbe essere prevista un'attività di monitoraggio che verifichi e controlli che quanto attuato sia conforme a quanto prescritto.

Infine c'è la *valutazione dei risultati*, infatti l'applicazione di quanto previsto deve essere valutata per verificare se il problema è stato risolto oppure se la soluzione implementata richiede degli aggiustamenti o deve essere completamente rivista e sostituita.

Esempio Lara è una studentessa che si è iscritta al primo anno di Economia e vuole laurearsi. Per raggiungere il suo obiettivo dovrà sostenere degli esami per acquisire 180 crediti formativi universitari. Il primo esame previsto nel piano di studio è matematica.

1. `Identificazione problema`: Lara deve superare l'esame di matematica.

2. `Analisi del problema`: per superare l'esame Lara deve prendere un voto positivo alla prova scritta e per raggiungere tale obiettivo deve essere preparata, dovrà conoscere gli argomenti che vengono chiesti all'esame e a tale riguardo ha due fonti di dati: la prima è il syllabus pubblicato sul sito dell'ateneo nel quale viene riportato il programma dell'esame, la seconda è il testo di riferimento (e gli appunti) dove trova gli argomenti inerenti il programma da studiare.

3. `Generazione di soluzioni`: potrebbero esserci diversi modi per poter superare l'esame consideriamone tre: il primo, il più ovvio è quello di studiare in modo da apprendere le nozioni e le competenze richieste per superare l'esame, il secondo è quello di preparere un formulario e bigliettini da nascondere in tasca o nell'astuccio da poter utilizzare di nascosto senza farsi vedere dal professore, un terzo potrebbe essere quello di trovare qualche compagno di università compiacente disposto a "passare" la soluzione degli esercizi del compito d'esame[1]. In tutti e tre i casi si può raggiungere l'obiettivo anche se, verosimilmente, con diversi punteggi e probabilità di superamento dell'esame.

4. `Valutazione delle soluzioni`: delle tre soluzioni proposte la prima appare la migliore perché permette a Lara di aumentare le sue competenze che potrebbero tornare utili anche in altri esami. Inoltre, le soluzione due e tre sono eticamente deprecabili oltre che illecite.

5. `Implementazione`: si predispone un programma di studio in base alle informazioni tratte dal syllabus e al testo di riferimento oltre che ad eventuali appunti.

6. `Valutazione dei risultati`: i risultati dell'attività svolta per risolvere il problema saranno valutabili in sede di esame: nel caso di esito positivo il problema si può ritenere risolto, in caso contrario invece potrebbe esser opportuno modificare il metodo di studio o approfondirlo maggiormente.

Dall'esempio appena esposto emergono due aspetti che saranno approfonditi nei prossimi paragrafi: il primo è quello relativo al linguaggio utilizzato nel risolvere un problema che può essere quello naturale, come quello utilizzato nell'esempio, oppure un linguaggio di programmazione; il secondo aspetto è quello relativo alla descrizione dell'implementazione cioè alla definizione di un algoritmo risolutivo.

1.2 Il linguaggio

L'esempio proposto nel paragrafo precedente mette in evidenza come nella vita di tutti i giorni ci si trovi ad affrontare e risolvere problemi. C'è bisogno quindi di utilizzare strategie e soluzioni che devono essere espresse in modo comprensibile cioè con un linguaggio significativo per gli esseri umani. Tuttavia, ci possono essere situazioni nelle

[1] Le tre alternative proposte hanno a mero scopo esemplificativo in quanto solo la prima è lecita le altre due invece sono soggette a sanzione disciplinari.

quali il problema è più semplice oppure è semplicemente ripetitivo rispetto a quello presentato nell'esempio riguardante il superamento dell'esame universitario di matematica, come ad esempio nel caso in cui si volesse emettere una fattura elettronica. In quest'ultima circostanza ci si potrà avvalere di un computer per risolvere quello che si potrebbe definire il "problema fatturazione". Ne consegue la necessità di comunicare con il computer e quindi di utilizzare un linguggio che sia comprensibile dal computer dal momento che il linguaggio naturale non è adeguato. In questi ultimi anni sono stati fatti notevoli progressi nel rendere i computer in grado di comprendere il linguaggio naturale. Molti siti aziendali sono dotati di `chatbot` con cui i clienti possono interagire con un computer per ottenere informazioni. Una versione evoluta delle chatbot è ChatGPT, un sistema di intelligenza artificiale sviluppato da OpenAI basato sul modello di intelligenza artificiale GPT (Generative Pre-trained Transformer). "ChatGPT è progettato specificamente per sostenere conversazioni scritte in linguaggio naturale con gli utenti. Utilizza una variante della potente architettura GPT-3.5, ottimizzata per la generazione di testo coerente e contestualmente rilevante in risposta a domande o input degli utenti" (questa nota è stata generata automaticamente tramite il colloquio con ChatGPT).

Nonostante questi recenti sviluppi, al momento attuale c'è ancora la necessità di disporre di un altro *linguaggio* comprensibile dal sistema operativo del computer e dal computer stesso. Risulta evidente che il termine linguaggio può avere diversi significati a seconda del contesto in cui viene utilizzato ed è opportuno a questo punto soffermarci brevemente su questo concetto, senza voler essere esaustivi e rinviando a testi specifici per l'approfondimento, si veda ad esempio Graffi and Scalise (2002).

In generale quando si utilizza il vocabolo *linguaggio* ci si riferisce ad un sistema di simboli, parole, gesti o segni avente identico valore e significato per i soggetti appartenenti ad uno stesso gruppo, che viene utilizzato per comunicare informazioni, idee, pensieri, emozioni e permette un rapporto di interazione tra i soggetti, siano essi individui o macchine. Il linguaggio è quindi un mezzo fondamentale per la comunicazione e la trasmissione della conoscenza.

Da quanto detto ne consegue che nella progettazione e costruzione di un calcolatore oltre ad affrontare e risolvere problemi relativi alla componente `hardware`[2] deve essere definito anche il linguaggio con il quale comunicare al computer le istruzioni e i comandi da eseguire mediante la Central Process Unit (CPU)[3] o altra unità di elaborazione, come ad esempio la Graphical Process Unit (GPU). Tale linguaggio è detto linguaggio macchina.

[2]La parola *hardware* potrebbe essere tradotta letteralmente con ferraglia/ferramenta e costituisce tutto ciò che nel computer ha consistenza materiale, in altre parole è il supporto fisico del computer.

[3]Si ricorda che la CPU è una componente hardware dedicata all'esecuzione dei programmi.

Come si vedrà nei prossimi paragrafi, per la comunicazione tra utente finale e computer si utilizzano altri tipi di linguaggi, in particolare i linguaggi di programmazione.

Il linguaggio macchina è un linguaggio informatico, definito dagli ingegneri che progettano il computer, molto diverso dal linguaggio naturale a cui invece si avvicina maggiormente il linguaggio di programmazione. Le differenze tra linguaggio macchina e linguaggio di programmazione sono molte, dall'alfabeto utilizzato alle regole sintattiche, dalla loro origine agli obiettivi per cui sono stati creati.

I tre tipi di linguaggi citati in questo paragrafo, il linguaggio naturale, quello di programmazione e il linguaggio macchina, verranno presentati con maggior dettaglio nei prossimi paragrafi.

1.3 Il linguaggio naturale

Un linguaggio naturale è una lingua parlata e scritta che le persone utilizzano per comunicare tra di loro. Esistono molteplici linguaggi naturali che utilizzano un numero ristretto di simboli con cui si possono costruire vocaboli che possono essere messi assieme, seguendo regole grammaticali, per identificare oggetti, esprimere concetti ed idee. Con il linguaggio naturale è possibile anche contare ma questo non permette lo sviluppo di operazioni aritmetiche sulla base di regole generali precise.

L'insieme dei linguaggi naturali oggi esistenti è molto ampio ed è il risultato di una continua evoluzione, si pensi ai linguaggi utilizzati un tempo che sono stati abbandonati al giorno d'oggi, come ad esempio alla lingua fenicia o etrusca; altri sono stati introdotti, anche recentemente, ne ricordiamo qui di seguito due.

Il primo risale al 1820 quando il fabbro analfabeta Cherokee, Sequoyah, a partire dalla comunicazione verbale ha definito un numero ristretto di simboli (85 segni sillabici) con cui costruire delle parole con le quali, seguendo semplici regole grammaticali, esprimere concetti, Berder (2002). In Figura 1.1 è riportato l'alfabeto sillabico di Sequoyah.

Nel giro di poche generazioni tutti i Cherokee sono stati in grado di leggere e scrivere nella nuova lingua.

L'importanza di tale linguaggio è stata riconosciuta anche in ambito informatico infatti il cherokee è rappresentato in Unicode[4] con i caratteri compresi tra `U+13A0` e `U+13FD` e tra `U+AB70` e `U+ABFF`.

[4] *Unicode è un sistema di codifica che assegna un numero univoco ad ogni carattere usato per la scrittura di testi, in maniera indipendente dalla lingua, dalla piattaforma informatica e dal programma utilizzato* (https://it.wikipedia.org/wiki/Unicode).

Figura 1.1: L'alfabeto sillabico di Sequoyah e il relativo Unicode