Introduzione alla calcolabilità

... e al corso

Lezione del 07/03/2023

Premessa

- In genere, quando vengono resi disponibili agli studenti i lucidi delle lezioni, gli studenti si limitano a studiare sui lucidi.
- Al fine di impedire questo comportamento (che risulta sempre e necessariamente in una preparazione superficiale e insufficiente), nei miei lucidi si farà un costante riferimento alle dispense
- Ricordo anche che le dispense contengono **tutto** il materiale necessario per conseguire una preparazione adeguata
  - Rispetto alle dispense, altererò, talvolta, l'ordine degli argomenti ma indicherò sempre le pagine e/o i paragrafi delle dispense cui faccio riferimento
- Ciascuna di queste "lezioni a distanza" è pensata come sostituto di una lezione frontale: pertanto, assocerò una data a ciascuna di esse.
- In questi lucidi assegnerò frequentemente (come faccio nel corso delle lezioni frontali) esercizi:
  - gli studenti sono invitati a risolverli e ad inviarmeli (se lo desiderano) per una eventuale correzione a mezzo posta elettronica
  - entro una settimana dalla data della lezione nella quale sono inseriti
- Sono sempre a disposizione per chiarimenti (a mezzo posta elettronica o incontri telematici o incontri nel mio studio)

Contenuti del corso

- Il modulo 2 del corso di Fondamenti di Informatica è suddiviso in due parti:
- nella prima parte ci occuperemo di Calcolabilità
  - ossia, di capire quali problemi possono essere risolti automaticamente
    - e, strada facendo, ci accorgeremo che esistono problemi che proprio non possono essere risolti
  - e, per farlo, dovremo capire cosa significa risolvere ***automaticamente*** un problema
    - e, a dirla tutta, cosa significa, in assoluto, ***risolvere*** un problema
    - e, persino, *cos'è un* ***problema***
- nella seconda parte ci occuperemo di Complessità
  - ossia, di capire quali dei problemi che possono essere risolti, possono proprio essere risolti ***per davvero***
    - ohibò! Pare una contraddizione!
  - ma, di questo ci occuperemo più avanti...

Problemi e istanze

- Cos'è un problema? Facile!
  - "Quanto fa 5 +2?" oppure "Quanto misura l'area di un rettangolo la cui base è lunga 28 e la cui altezza è lunga 12?": ecco due esempi di problema!
- Sbagliato! Nell'esempio, sono illustrate due ***istanze*** di due problemi diversi
- I problemi cui corrispondono quelle istanze sono:
  - PROBLEMA SOMMA: dati due numeri naturali, n e k, calcolare il valore della somma di n con k (ossia, n + k)
  - PROBLEMA AREA: dato un rettangolo, la cui base è lunga b e la cui altezza è lunga h, calcolare l'area A di quel rettangolo
  - Chiara la distinzione?
- Un problema è la descrizione di un insieme di parametri, che chiameremo **dati**, collegati da un certo insieme di relazioni, associata alla richiesta di derivare da essi un altro insieme di parametri, che costituiscono la soluzione
- Un'istanza di un problema è un particolare insieme di valori associati ai dati

Su queste questioni torneremo, abbondantemente, (parecchio) più avanti

Trovare la soluzione di un'istanza

- Per trovare la soluzione di talune istanze di taluni problemi posso sfruttare le caratteristiche di quelle istanze
  - se chiedi a un bambino quanto fa 2 + 5, quello può contare sulle dita
  - se hai bisogno di trovare sen $\frac{\pi}{2}$, puoi disegnare la circonferenza goniometrica e vederlo
- D'altra parte, a volte non è così semplice
  - le dita non bastano per calcolare 49856739902+50672143559986
  - e calcolare sen $\frac{\sqrt[3]{\pi^{\pi}+8\ \ln \pi}}{5}$ … non è proprio una passeggiata
- Altre volte, è proprio impossibile
  - per quanto tu sia bravo in matematica, un numero reale che corrisponda a $\sqrt{-4}$, non c'è verso, non riuscirai mai a trovarlo
  - quando l'istanza di un problema non ha soluzione diciamo che essa è una **istanza negativa**
  - e cominciate a tenerla a mente questa cosa delle **istanze negative**, ché vi tornerà utile (eccome!)

Risolvere un problema

- Risolvere un problema significa individuare un metodo che *sappia trovare la soluzione di **qualunque** istanza positiva del problema*
  - e, in più, che sappia riconoscere se un'istanza è negativa
- ossia, significa trovare un procedimento che, data una qualunque istanza del problema, indichi la sequenza di azioni che devono essere eseguite per trovare la soluzione di quell'istanza
  - o per poter concludere che, quell'istanza, una soluzione non ce l'ha
- E qui sorgono un (bel) po' di questioni:
  - innanzi tutto, cos'è un *procedimento*?
  - E, poi, che cos'è una *azione*?
  - E, infine, *chi* è supposto debba eseguire le azioni indicate?
- Come stiamo per vedere, queste questioni sono fra loro interconnesse

Risolvere un problema

- Cos'è un procedimento?
  - **Un procedimento è la descrizione di un insieme di azioni unita alla specifica dell'ordine con il quale le azioni devono essere eseguite**
- E che cos'è una azione?
  - Qualcosa che deve esser fatto, ovvio! Tuttavia...
  - Anche "data un'istanza del problema, trova la soluzione di quell'istanza" è una azione
  - Allora, dobbiamo dire che le azioni indicate in un procedimento, devono essere azioni *semplici*, azioni, cioè, che possono essere eseguite con facilità
- ESEMPIO: data una funzione f: $\mathbb{R} \rightarrow \mathbb{R}^+$ e dati due numeri reali a e b, calcolare la misura dell'area della regione di piano compresa fra la funzione, l'asse x e le rette y=a e y=b
- PROCEDIMENTO: 1) calcola la funzione primitiva F(x) di f(x)
  2) calcola F(b) – F(a)

Risolvere un problema

- Un procedimento è la descrizione di un insieme di *azioni* unita alla specifica dell'ordine con il quale le azioni devono essere eseguite
  - e, a ciascuna di quelle azioni, viene dato il nome di ***istruzione***
- e le istruzioni indicate in un procedimento, devono essere *elementari*, devono, cioè, essere azioni che possono essere eseguite con facilità
- ESEMPIO: data una funzione f: $\mathbb{R} \rightarrow \mathbb{R}^+$ e dati due numeri reali a e b, calcolare la misura dell'area della regione di piano compresa fra la funzione, l'asse x e le rette y=a e y=b
- PROCEDIMENTO: 1) calcola la funzione primitiva F(x) di f(x)
  2) calcola F(b) – F(a)
- Certo, quello indicato è un procedimento che risolve il problema nell'esempio
- Tuttavia, "calcola la funzione primitiva F(x) di f(x)" è <u>davvero</u> un'*istruzione elementare*?
  - per me (che sono una matematica) sì, per un bambino in prima elementare no...
- Cioè, che sia elementare o no, *dipende da chi è supposto debba eseguire le azioni indicate*

L'istruzione elementare

- Dunque, se vogliamo **svincolare la definizione di procedimento risolutivo di un problema da quello di esecutore delle azioni in esso indicate**, è necessario, prima di tutto, chiarire formalmente cosa si intende con *istruzione elementare*
- Vediamo, a tal proposito, la soluzione individuata da Alan Turing a questa questione
- Turing, osservò che, indipendentemente dall'esecutore, qualunque istruzione, per potere essere definita *elementare*, deve avere le seguenti caratteristiche:
  - deve essere scelta in un insieme di "poche" istruzioni disponibili
  - deve scegliere l'azione da eseguire all'interno di un insieme di "poche" azioni possibili
  - deve poter essere eseguita ricordando una quantità limitata di dati, ossia, in termini più tecnici, utilizzando una quantità limitata di memoria.
- Osserviamo che le caratteristiche individuate da Turing indicano come istruzione elementare una operazione che possa essere eseguita... a mente!
- Chiariamo con un esempio

L’istruzione elementare

- Consideriamo il PROBLEMA SOMMA: dati due interi n e k, ci viene richiesto di calcolare il numero n + k
- Vogliamo progettare un procedimento che risolva questo problema
- Ebbene: calcolare la somma di due interi è certamente facile
  - abbiamo imparato a calcolarla in prima elementare!
- allora, potremmo pensare che l’istruzione “calcola n + k” sia un’istruzione elementare
- ATTENZIONE: stiamo cercando un procedimento che risolva un problema (il PROBLEMA SOMMA), quindi “calcola n + k” deve essere un’istruzione elementare ***qualunque*** valore venga assegnato a n e k
- Però, se n = 37895 e k = 441238 ...
- a nessuno di noi, soltanto guardando i due addendi, salta in mente il risultato
  - anche se le addizioni le sappiamo fare benissimo!

L'istruzione elementare

- Se n = 37895 e k = 441238, a nessuno di noi, guardando i due addendi, salta in mente quanto fa n + k
- Questo perché la nostra memoria è limitata
- Chiariamo:
  - In qualche modo, quando abbiamo imparato a fare le addizioni, abbiamo **memorizzato** la tabella che ci permette di calcolare a mente la somma di qualunque coppia di numeri di una cifra ciascuno

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

L'istruzione elementare

- Se n = 37895 e k = 441238, a nessuno di noi, guardando i due addendi, salta in mente quanto fa n + k
- Ma se disponessimo di una tabella *sufficientemente grande* che indica le somme di tutti i numeri naturali compresi fra 0 e 1000000 (ad esempio), ci basterebbe guardare nella cella opportuna e avremmo la somma cercata: al volo, ad occhio...

+	0	1	2	…	**37895**		1000000
0	0	1	2	...	37895	...	1000000
1	1	2	3	...	37896	...	1000001
2	2	3	4	...	37897	...	1000002
…	…	…	…	…		…	
441238	441238	441239	441240	...	**479133**	...	1441238
…	…	…	…	…		…	
1000000	1000000	1000001	1000002	...	1037895	...	2000000

L'istruzione elementare

- Se n = 37895 e k = 441238, a nessuno di noi, guardando i due addendi, salta in mente quanto fa n + k
- Ma se disponessimo di una tabella che indica le somme di tutti i numeri naturali compresi fra 0 e 1000000 (ad esempio), ci basterebbe guardare nella cella opportuna e avremmo la somma cercata: al volo, ad occhio...
- Ossia, disporre di questa nuova tabella ci permetterebbe di considerare istruzione elementare la somma di qualunque coppia di numeri naturali compresi fra 0 e 1000000
- Allora, è fatta! Basta predisporre una tabella *sufficientemente grande* e qualunque somma diventa un'istruzione elementare!
- Ma **NO, NON FUNZIONA IN QUESTO MODO!!!!**
- Il problema è che, per risolvere il PROBLEMA SOMMA, occorre indicare un procedimento che sappia addizionare *qualunque* coppia di numeri naturali
  - per quanto grandi essi siano
- e, quindi, se volessimo considerare istruzione elementare la somma di qualunque coppia di numeri, **dovremmo costruire una tabella infinita!**

L’istruzione elementare

- Ecco perché la somma di qualunque coppia di numeri naturali non può essere considerata un’operazione elementare: perché avremmo bisogno di memorizzare una tabella di dimensioni illimitate
- mentre, invece, la nostra memoria è limitata!

- Per questa ragione, per eseguire la somma di qualunque coppia di numeri naturali, utilizziamo un *procedimento* che
  - utilizza un numero limitato di operazioni elementari (le somme di coppie di numeri di una sola cifra)
  - e in cui ogni operazione elementare utilizza una quantità limitata di dati (due cifre e l’eventuale riporto)
- In accordo alle caratteristiche enunciate da Turing

- E adesso andiamo a ripassare questo procedimento ...

La somma di due numeri naturali

- Per calcolare il valore della somma 37895 + 441238, innanzi tutto scriviamo l'operazione in colonna:

```
      3   7   8   9   5   +
  4   4   1   2   3   8   =
-----------------------------
```

- poi, osserviamo le due cifre più a destra, e calcoliamo la loro somma e l'eventuale riporto

```
      3   7   8   9   5   +
  4   4   1   2   3   8   =
-----------------------------
                      3   con riporto di 1
```

- poi, osserviamo le due cifre più a destra non ancora considerate, e calcoliamo la loro somma più l'eventuale riporto, e il nuovo eventuale riporto

```
      3   7   8   9   5   +
  4   4   1   2   3   8   =
-----------------------------
                  3   3   con riporto di 1
```

- ... e così via ...

La somma di due numeri naturali

- Pensandoci bene, potremmo descrivere il procedimento per calcolare la "somma in colonna" di due numeri naturali nel modo seguente
- 1) posizionati sulla coppia di cifre più a destra, e poni r = 0
- 2) fino a quando leggi <u>una coppia</u> di cifre, esegui la somma della coppia di cifre sulle quali sei posizionato, aggiungi r a tale valore e scrivi una cifra del risultato calcolando anche il nuovo valore di r, e poi spostati a sinistra – ossia:
  - se r = 0 e le due cifre sono 0 e 0, allora scrivi 0, poni r = 0, e spostati di una posizione a sinistra
  - se r = 1 e le due cifre sono 0 e 0 e allora scrivi 1, poni r = 0, e spostati di una posizione a sinistra
  - ...
  - se r = 0 e le due cifre sono 9 e 9, allora scrivi 8, poni r = 1,e spostati di una posizione a sinistra
  - se r = 1 e le due cifre sono 9 e 9, allora scrivi 9, poni r = 1, e spostati di una posizione a sinistra
- [... continua ...]

La somma di due numeri naturali

- Pensandoci bene, potremmo descrivere il procedimento per calcolare la “somma in colonna” di due numeri naturali nel modo seguente
- 1) posizionati sulla coppia di cifre più a destra, e poni r = 0
- 2) fino a quando leggi una coppia di cifre, esegui la somma della coppia di cifre sulle quali sei posizionato, aggiungi r a tale valore e scrivi una cifra del risultato calcolando anche il nuovo valore di r, e poi spostati a sinistra
- 3) fino a quando leggi una sola cifra (ossia, le cifre di uno dei due numeri sono terminate) aggiungi r ad essa e scrivi una cifra del risultato calcolando anche il nuovo valore di r, e poi spostati a sinistra – ossia,
  - se r = 0 e l’unica cifra è 0, allora scrivi 0, poni r = 0, e spostati di una posizione a sinistra
  - se r = 0 e l’unica cifra è 1, allora scrivi 1, poni r = 0, e spostati di una posizione a sinistra
  - ...
  - se r = 1 e l’unica cifra è 8, allora scrivi 9, poni r = 0, e spostati di una posizione a sinistra
  - se r = 1 e e l’unica cifra è 9, allora scrivi 0, poni r = 1, e spostati di una posizione a sinistra
- [... continua ...]

La somma di due numeri naturali

- Pensandoci bene, potremmo descrivere il procedimento per calcolare la "somma in colonna" di due numeri naturali nel modo seguente
- 1) posizionati sulla coppia di cifre più a destra, e poni $r = 0$
- 2) fino a quando leggi una coppia di cifre, esegui la somma della coppia di cifre sulle quali sei posizionato, aggiungi r a tale valore e scrivi una cifra del risultato calcolando anche il nuovo valore di r
- 3) fino a quando leggi una sola cifra (ossia, le cifre di uno dei due numeri sono terminate) aggiungi r ad essa e scrivi una cifra del risultato calcolando anche il nuovo valore di r, e poi spostati a sinistra
- 4) se le cifre di entrambi i numeri sono terminate, allora calcola l'eventuale ultima cifra del risultato e termina – ossia:
  - se $r = 0$ e le cifre di entrambi i numeri sono terminate, allora termina
  - se $r = 1$ e le cifre di entrambi i numeri sono terminate, allora scrivi 1 e termina.

La somma di due numeri naturali

- Ossia, il procedimento per calcolare la "somma in colonna" di due numeri naturali è una sequenza di
  "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
  - ad ogni coppia (***certe condizioni***, ***queste azioni***) corrisponde un'istruzione
  - dove ***certe condizioni*** è ciò che viene letto (la coppia di cifre dei due numeri, eventualmente assenti) e il valore del riporto
  - e ***queste azioni*** è ciò che viene scritto, la modifica del valore del riporto, e lo spostamento
    - o, in alcuni casi, ***queste azioni*** è l'indicazione che la somma è stata completata (termina)
- Pensandoci bene, questo procedimento potrebbe eseguirlo chiunque sappia leggere e scrivere e distinguere fra destra e sinistra
  - che sono nozioni davvero ***elementari***!
  - Su questo non c'è davvero dubbio!
- Ma, pur essendo *istruzioni elementari da un punto di vista intuitivo*, sono quelle appena individuate **istruzioni elementari nel senso indicato da Turing**?

La somma di due numeri naturali

- Ricordiamo che, nell'accezione di Turing, un'istruzione, per potere essere definita *elementare*, deve avere le seguenti caratteristiche:
  - deve essere scelta in un insieme di "poche" istruzioni disponibili
  - deve scegliere l'azione da eseguire all'interno di un insieme di "poche" azioni possibili
  - deve poter essere eseguita ricordando una quantità limitata di dati, ossia, in termini più tecnici, utilizzando una quantità limitata di memoria.
- Ora, abbiamo già visto che nel procedimento che esegue la somma le azioni che vengono eseguite sono due: scrittura di una cifra e spostamento
  - e possiamo ben affermare che esse sono davvero "poche"!
- Ma è vero che Il procedimento che esegue la somma ha un insieme di "poche" istruzioni disponibili ciascuna delle quali utilizza una quantità limitata di memoria?
  - Che poi: ma cosa si intende con "poche" e con quantità limitata?

La somma di due numeri naturali

- Ma è vero che Il procedimento che esegue la somma ha un insieme di "poche" istruzioni disponibili ciascuna delle quali utilizza una quantità limitata di memoria?
- Riflettiamo:
  - il numero di istruzioni disponibili è pari al numero di coppie di cifre moltiplicato per il numero di possibili valori per il riporto, ossia, $10 \times 10 \times 2 = 200$
  - per sapere quale istruzione dobbiamo eseguire abbiamo bisogno di conoscere le due cifre da sommare e il valore del riporto, ossia, 3 numeri di una cifra
- Ricapitolando: per sommare qualunque coppia di interi (grandi quanto ci pare) abbiamo a disposizione 222 istruzioni (che eseguono 2 azioni) fra le quali scegliere quella da eseguire utilizzando una memoria di 3 cifre
- Indipendentemente da quanto sono grandi i due numeri che vogliamo sommare, sempre 222 istruzioni (che eseguono 2 azioni) disponibili che utilizzano una memoria di 3 cifre sono!
- **Ossia, il numero di istruzioni, azioni e la quantità di memoria necessaria sono costanti: non dipendono da quello che chiameremo input**
  - chiaro ora cosa si intende con "poche" e con quantità limitata?
  - Chiara, ora, la scelta di Turing delle sue tre caratteristiche!

La somma di due numeri naturali

- Il procedimento per calcolare la "somma in colonna" di due numeri naturali è una sequenza di
  "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
  - dove ***certe condizioni*** è ciò che viene letto (la coppia di cifre dei due numeri, eventualmente assenti) e il valore del riporto
  - e ***queste azioni*** è ciò che viene scritto, la modifica del valore del riporto, e lo spostamento
    - o, in alcuni casi, ***queste azioni*** è l'indicazione che la somma è stata completata (termina)
- Pensandoci bene, questo procedimento potrebbe eseguirlo chiunque sappia leggere e scrivere e distinguere fra destra e sinistra
- Pensandoci bene, *per eseguire questo procedimento non è necessario nemmeno sapere cosa significa "sommare due numeri naturali"*
- esegui le istruzioni del procedimento e, come per magia, alla fine ti ritrovi con il risultato in mano

La somma di due numeri naturali

- Il procedimento per calcolare la "somma in colonna" di due numeri naturali è una sequenza di "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
- Pensandoci bene, *per eseguire questo procedimento non è necessario nemmeno sapere cosa significa "sommare due numeri naturali":*
- esegui le istruzioni del procedimento e, come per magia, alla fine ti ritrovi con il risultato in mano!
- Perché, naturalmente, le istruzioni ti dicono, **per ogni condizione possibile**, **esattamente quali azioni devi eseguire in quelle condizioni**
- questo significa che l'insieme di istruzioni è *non ambiguo*: non può contenere due (o più) istruzioni che, a partire dalle stesse ***condizioni*** , ti indica diverse *azioni* da eseguire
  - non può succedere, ad esempio, che un'istruzione affermi "se è vero **a** allora scrivi 5" e un'altra istruzione affermi "se è vero **a** allora scrivi 6"
  - altrimenti, quando è vero **a** come devi comportarti tu che vuoi eseguire le istruzioni?

La somma di due numeri naturali

- Il procedimento per calcolare la "somma in colonna" di due numeri naturali è una sequenza di "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
- Le istruzioni ti dicono, **per ogni condizione possibile**, **esattamente quali azioni devi eseguire in quelle condizioni**
- questo significa che l'insieme di istruzioni è *non ambiguo*: non può contenere due (o più) istruzioni che, a partire dalle stesse ***condizioni*** , ti indica diverse ***azioni*** da eseguire
  - non può succedere, ad esempio, che un'istruzione affermi "se è vero **a** allora scrivi 5" e un'altra istruzione affermi "se è vero **a** allora scrivi 6"
- E, dunque, **l'ordine in cui eseguire le istruzioni è indicato implicitamente nel meccanismo stesso del "se ... allora ..."**
  - in ogni istante devi eseguire l'unica istruzione che è possibile eseguire, fino a quando non incontri un'istruzione che ti dice di terminare
  - non puoi fare altro!

La somma di due numeri naturali

- Il procedimento per calcolare la "somma in colonna" di due numeri naturali è una sequenza di "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
- Pensandoci bene, *per eseguire questo procedimento non è necessario nemmeno sapere cosa significa "sommare due numeri naturali":*
- esegui le istruzioni del procedimento e, come per magia, alla fine ti ritrovi con il risultato in mano!
- Attenzione, però: per ottenere il risultato **devi** eseguire le istruzioni
  - ossia, ogni volta che si verificano quelle ***condizioni*** tu quelle ***azioni*** **devi** eseguirle
  - senza se e senza ma, le esegui e basta!
- Cioè, le istruzioni sono una sorta di ordini
  - loro ti dicono di fare qualcosa e tu lo fai!
- Questa idea di istruzione, nata dall'analisi di Turing, è alla base di molti linguaggi di programmazione che, proprio per questo, vengono detti **imperativi**
  - il C, il Fortran, ma anche Java o Python...

La somma di due numeri naturali

- Il procedimento per calcolare la "somma in colonna" di due numeri naturali è una sequenza di
  "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
  - dove ***certe condizioni*** è ciò che viene letto (la coppia di cifre dei due numeri, eventualmente assenti) e il valore del riporto
  - e ***queste azioni*** è ciò che viene scritto, la modifica del valore del riporto, e lo spostamento
    - o, in alcuni casi, ***queste azioni*** è l'indicazione che la somma è stata completata (termina)
- Pensandoci bene, questo procedimento potrebbe eseguirlo chiunque sappia leggere e scrivere e distinguere fra destra e sinistra
- Pensandoci bene, *per eseguire questo procedimento non è necessario nemmeno sapere cosa significa "sommare due numeri naturali"*
  - esegui le istruzioni del procedimento e, come per magia, alla fine ti ritrovi con il risultato in mano
- cioè, *questo procedimento potrebbe anche essere eseguito da un automa*

Risolvere automaticamente un problema

- Eccoci al nocciolo della questione:
  - informalmente, *risolvere automaticamente un problema* significa progettare un **procedimento** che risolve **tutte** le istanze di quel problema e che *può essere eseguito da un automa*
    - ossia, da un esecutore che può non avere alcuna idea del problema né del significato delle istruzioni contenute nel procedimento
- Il resto della prima parte di questo modulo è dedicato a formalizzare questo concetto informale

Un nuovo linguaggio

- Ripensiamo alla somma di due numeri naturali:
- 1) il procedimento che abbiamo visto è costituito di sole istruzioni "se sono vere ***certe condizioni*** allora esegui ***queste azioni***"
  - ripetute fino a quando non si incontra il comando "termina"
- 2) in ciascuna istruzione le ***azioni*** da eseguire sono le 3 azioni seguenti
  - la scrittura di una cifra, la (eventuale) modifica del riporto, il movimento a sinistra per considerare le successive due cifre da sommare
- 3) infine, le ***condizioni*** di ognuna delle istruzioni dipendono da due tipi di parametri
  - il valore del riporto
  - le due cifre da sommare
- NOTA: mentre le due cifre da sommare le troviamo scritte sul foglio sul quale abbiamo indicato (in colonna) i due numeri che vogliamo sommare
- il valore del riporto lo teniamo a mente ad ogni coppia di cifre sommate
  - è, cioè, qualcosa che caratterizza il nostro "***stato interiore***"

Un nuovo linguaggio...

- In virtù delle osservazioni 1), 2) e 3), possiamo scrivere il nostro procedimento in forma più compatta
  - poiché utilizziamo sole istruzioni "se *condizione* allora *azione*" possiamo anche evitare di scrivere "se ... allora ..." ogni santa volta
  - e scrivere, di seguito, le due condizioni seguite dalle tre azioni
- così, ad esempio, l'istruzione
  - se r = 0 e le due cifre sono 4 e 6, allora scrivi 0, poni r = 1, e spostati di una posizione a sinistra
- diventa
  - **⟨ q_0 , (4, 6), 0, q_1 , sinistra ⟩**
- dove q_0 e q_1 sono due simboli che indicano, rispettivamente, r = 0 e r = 1
- OSSERVAZIONE: in questo esempio sembrerebbe che anche "sinistra" possa essere omesso; vedremo che questa specifica, invece, occorre tenerla
- [... continua ...]

Un nuovo linguaggio...

- In virtù delle osservazioni 1), 2) e 3), possiamo scrivere il nostro procedimento in forma più compatta
- e l'istruzione
  - se r = 1 e l'unica cifra è 5, allora scrivi 6, poni r = 0, e spostati di una posizione a sinistra
- nella quale le cifre di uno degli operandi sono terminate, diventa la coppia di istruzioni
  - **⟨ q_1 , (5, $\square$), 6, q_0 , sinistra⟩**
  - **⟨ q_1 , ($\square$, 5), 6, q_0 , sinistra⟩**
- dove il simbolo $\square$ indica che non viene letto alcunché o che non deve essere scritto alcunché
- e abbiamo due diverse istruzioni perché l'operando le cui cifre sono terminate può essere il primo o il secondo
- [... continua ...]

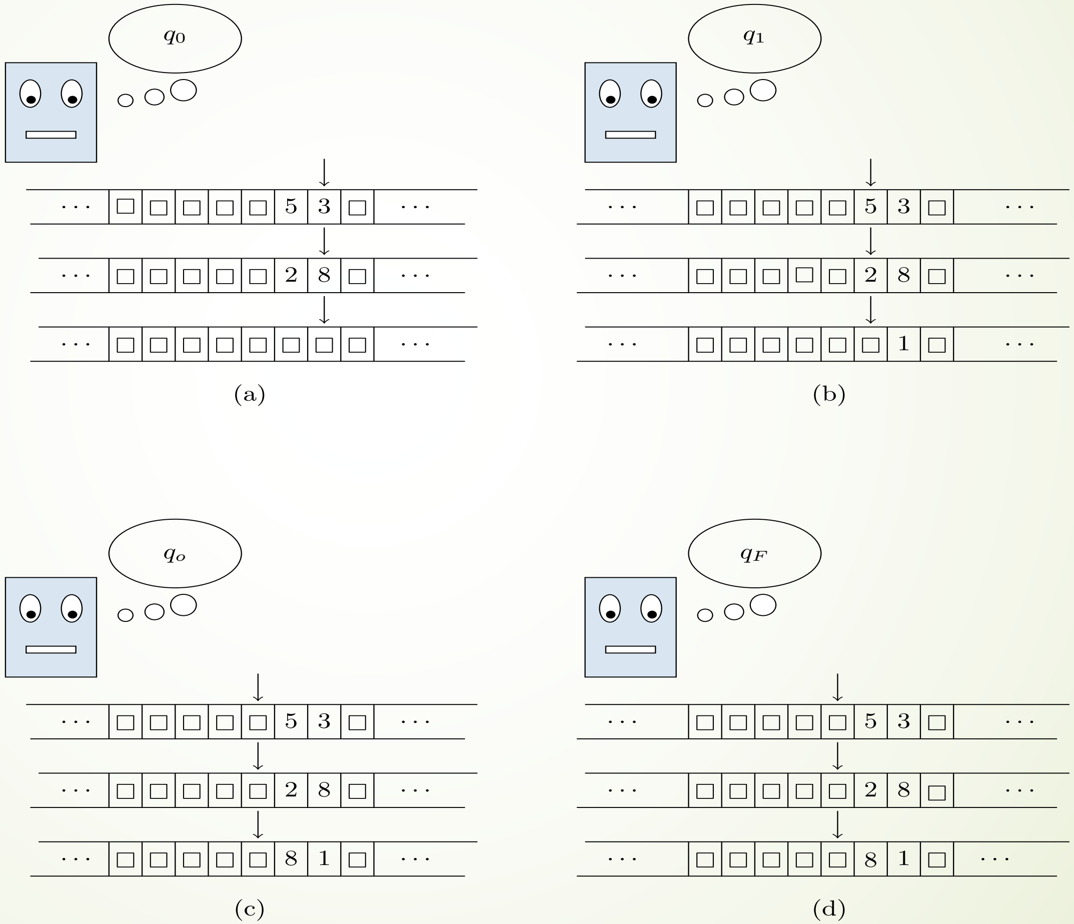
Un nuovo linguaggio...

- In virtù delle osservazioni 1), 2) e 3), possiamo scrivere il nostro procedimento in forma più compatta
- e, infine, le istruzioni
  - se r = 1 e le cifre di entrambi i numeri sono terminate, allora scrivi 1 e termina
  - se r = 0 e le cifre di entrambi i numeri sono terminate, allora termina
- diventano, rispettivamente
  - **$\langle\, q_1$, ($\square$, $\square$), 1, q_F, fermo$\rangle$**
  - **$\langle\, q_0$, ($\square$, $\square$), $\square$, q_F, fermo$\rangle$**
- dove $\mathbf{q_F}$ è lo "*stato interiore*" che permette all'esecutore di comprendere che non deve più eseguire alcuna azione
  - ossia, non si deve "tornare al punto 2)"
- e qui l'utilizzo di "fermo" mostra anche perché è necessario specificare come ci si deve muovere

… e una macchina che lo comprende

- Possiamo, a questo punto, rappresentare graficamente l'esecuzione del procedimento che calcola la somma di due numeri qualsiasi
  - ad esempio, i numeri 53 e 28
- per farlo, immaginiamo di disporre di una sorta di automa
  - che rappresentiamo come una specie di "testa robotizzata"
  - e che può trovarsi in uno di tre possibili "*stati interiori*": q_0 , q_1 e q_F
- che utilizza, per leggere e scrivere , tre nastri
  - suddivisi ciascuno in un numero infinito di celle
  - tali che ciascuna cella, in ogni istante, può contenere o una cifra (un numero compreso fra 0 e 9) oppure può essere vuota (e indichiamo con ◻ il simbolo di cella vuota)
- e tre testine di lettura/scrittura
- Non appena viene scritto qualcosa sui nastri, dipendentemente dallo "stato interiore" dell'automa e da quello che viene letto, l'automa inizia a ***computare*** – ossia a eseguire le quintuple del procedimento

... e una macchina che lo comprende



(a) (b) (c) (d)

Quasi una macchina di Turing

- Quella che abbiamo visto è *quasi* una descrizione informale di una macchina di Turing
- *quasi*, perché abbiamo utilizzato tre nastri e in una macchina di Turing occorre descrivere cosa viene letto (nelle ***condizioni***) e cosa viene scritto (nelle ***azioni***) su <u>ogni</u> nastro
- così che l'istruzione

  se r = 0 e le due cifre sono 4 e 6, allora scrivi 0, poni r = 1, spostati di una posizione a sinistra e torna al punto 2)
- diventa **$\langle$ $\mathbf{q_0}$, (4, 6, □), (4, 6, 0), $\mathbf{q_1}$, sinistra $\rangle$**
  - che specifica cosa deve essere scritto sui 3 nastri (4, 6, □) e con cosa questi tre elementi devono essere sovrascritti (4, 6, 0)
- poiché specifica 2 condizioni e 3 azioni, essa prende il nome di ***quintupla***
- e, quelli che abbiamo chiamato sino ad ora "*stati interiori*", si chiamano propriamente ***stati interni***
- e l'esecuzione delle quintuple su un insieme fissato di dati (come nella figura) si chiama ***computazione***

Calcolabilità

- Quella che abbiamo visto è, dunque, una descrizione informale di una macchina di Turing
  - con la ‘m’ minuscola
- che è la *descrizione di un procedimento di risoluzione di un problema* espresso nel *linguaggio definito da Alan Turing*
- linguaggio che costituisce un *modello di calcolo*: il modello **Macchina di Turing**
  - con la ’M’ maiuscola
- E tutto ciò, che è necessario per parlare di Calcolabilità, inizieremo a vederlo *formalmente* nella prossima lezione