Storia dell'Informatica - Seconda Parte

Appunti

Luca Barra

Anno accademico 2023/2024

1 Premessa

1.1 Licenza

Questi appunti sono rilasciati sotto licenza Creative Commons Attribuzione 4.0 Internazionale (per maggiori informazioni consultare il link: https://creativecommons.org/version4/). Sono basati sulle slides del corso "Storia dell'Informatica" del prof. Felice Cardone.

1.2 Formato utilizzato

In questi appunti vengono utilizzati molti *box*. Questa è una semplice rassegna che ne spiega l'utilizzo:

Box di "Concetto sbagliato":

Concetto sbagliato 1.1: Testo del concetto sbagliato

Testo contente il concetto giusto.

Box di "Corollario":

Corollario 1.2.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

Definizione 1.2.1: Nome delle definizione

Testo della definizione.

Box di "Domanda":

Domanda 1.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

Esempio 1.2.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

Box di "Note":

Note:-

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

Box di "Meta-nota":

Meta-nota 1.2.1

Testo della meta-nota. Le meta-note utilizzate com note più personali per fornire riflessioni e motivazioni al perchè si sta facendo ciò che si sta facendo. Vengono usate per simulare lo stile espositivo del prof. Cardone che spazia in poco tempo su molti argomenti. Possono essere scollegate direttamente dal contesto in cui si trovano e ai fini degli appunti possono essere tranquillamente ignorate.

Box di "Teorema":

Teorema 1.2.1 Nome del teorema

Testo del teorema.

2 Introduzione

Il corso si divide in una serie di sezioni:

⇒ Macchine mai esistite (Knowledge Navigator, Dynabook, Memex): poichè la storia dell'informatica è una storia di idee, di pionieri e di visionari;

⇒ Il modo di organizzare i testi in rapporto all'inondazione informativa: workstation di Otlet (non realizzata), the mother of all demos (Doug Engelbart, 1968), gli ipertesti di Ted Nelson, libraries of the future;

⇒ La cybernetica: originata da Norbert Wiener, ci si concentra sugli aspetti matematici della neurofisiologia che portarono al modello formale di neurone (McCulloch e Pitts, 1943) utilizzato da Von Neumann per la definizione della sua architettura di calcolatore;

⇒ Gli hippies e la controcultura: si parla di come la controcultura abbia influenzato la nascita dell'informatica;

⇒ La semiotica: si parte in ordine cronologico dalla preistoria, passando per Lullo fino a Leibniz, per arrivare ai sistemi formali.

2.1 La preistoria dell'informatica

Domanda 2.1

Perchè studiare cosi indietro nel tempo?

Risposta: perchè si vuole caratterizzare il concetto di calcolo e si vogliono comprendere le abilità cognitive su cui tale concetto si basa.

Circa 110.000 anni fa si ha, in Cina, il primo esempio di astrazione con degli *ossi intagliati*. Essa è la produzione consapevole di una traccia relativamente stabile. Successivamente, intorno al 8000 a. C., sono stati usati dei *gettoni* in argilla. Presumibilmente questa è la nascita simultanea della scrittura e del calcolo. I gettoni vennero affiancati dalle *tavolette d'argilla*, modellate usando dei calchi (come un rudimentale libro contabile).

I gettoni:

⇒ Favoriscono la raccolta di dati;

⇒ Sono immediati da comprendere;

⇒ Permettono le operazioni aritmetiche come manipolazioni concrete.

Un ulteriore fenomeno è quello delle *bolle* che contengono dei gettoni. Venivano usate come registrazione di un contratto quando un proprietario di bestiame dava in affitto la sua mandria per la transumanza.

Oltre agli strumenti d'argilla vennero usati dei *bastoncini di legno* che venivano divisi in due metà: una ricevuta e un titolo di credito. Il titolo di credito si chiamava *stock* (da qui il termine "stock market") e la ricevuta si chiamava stub.

2.1.1 Semiotica

Nei meccanismi illustrati nella sezione precedente i segni hanno un ruolo importante.

Definizione 2.1.1: Semiotica

La *semiotica* è la scienza che studia i linguaggi e i segni che li costituiscono.

Corollario 2.1.1 Il linguaggio

In ogni linguaggio sono presenti tre componenti:

⇒ Chi produce i segni (studiati dalla pragmatica);

⇒ I segni prodotti (studiati dalla sintassi);

⇒ Il significato dei segni (studiati dalla semantica).

La semiotica è direttamente collegata all'informatica (computer semiotics):

⇒ Un computer è spesso utilizzato per generare, trasformare e visualizzare segni (il PC come medium);

⇒ Un computer viene programmato utilizzando un linguaggio.

Kenneth Iverson fu un importate semiotico e fu il fondatore del linguaggio esoterico APL, che nacque come linguaggio logico. APL influenzo molti linguaggi funzionali, come Haskell, e il paradigma di parallelismo.

2.2 Leibniz

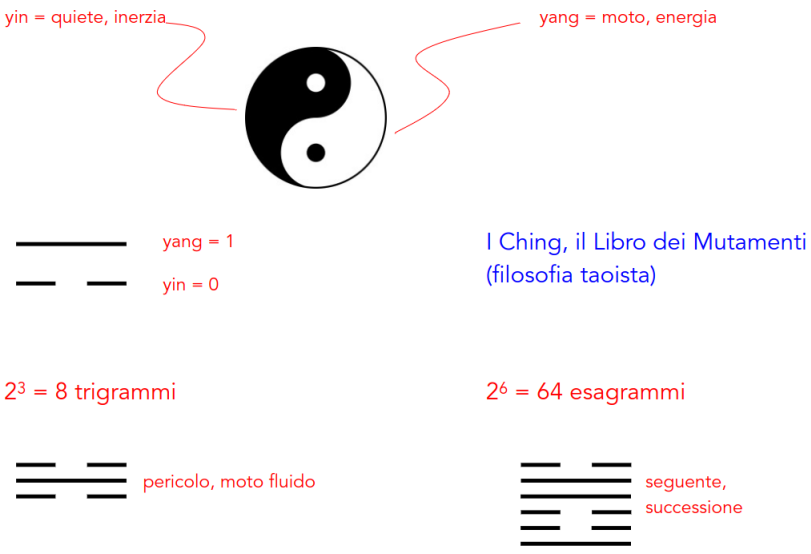
Nasce a Lipsia il 21 Giugno 1646, si laurea in Giurisprudenza nel 1666. I suoi primi scritti sono finalizzati al conseguimento di titoli accademici. Importante in questo periodo è la Dissertatio de Arte Combinatoria del 1666. Negli anni immediatamente successivi alla laurea diventa consigliere dell'Elettore di Magonza ed assume diversi incarichi politici.

Nel 1672 viene inviato a Parigi in missione diplomatica, per distogliere Luigi XIV dalla progettata invasione dell'Olanda ed invogliarlo invece alla conquista dell'Egitto. Fallita la missione, ottiene il permesso di fermarsi a Parigi (e Londra), dove rimane per 4 anni (marzo 1672-ottobre 1676) avendo la possibilità di conoscere la matematica e la fisica più avanzate. Nel 1676 scopre il calcolo infinitesimale (che verrà pubblicato solo ne 1684), già introdotto da Newton indipendentemente 10 anni prima. Nel 1705 inizierà tra i due una polemica che finirà solo con la morte di Leibniz. Nello stesso anno torna ad Hannover come bibliotecario presso il Duca di Hannover.

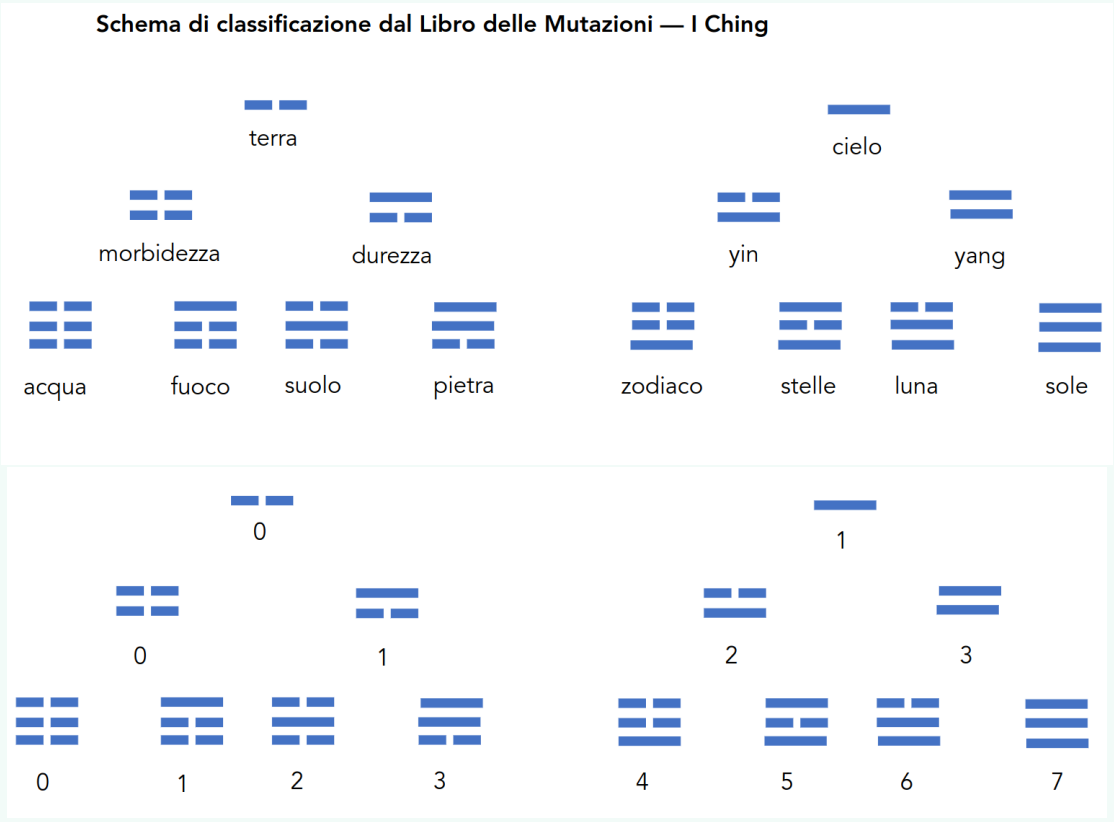
Tra il 1685 e il 1694: migliora la scatola di Pascal (*pascalina*) per l'addizione e la sottrazione per realizzare anche la moltiplicazione e la divisione (e l'estrazione di radice). La macchina, che opera mediante pulegge e ruote dentate, è conservata nella biblioteca di Hannover.

2.2.1 L'origine del bit

Leibniz fece un parallelismo tra i simboli dello yin e dello yang e i numeri binari. Le linee piene potevano essere associate al 1, mentre le linee vuote al 0. Così facendo si ottiene un sistema di numerazione binario.



Esempio 2.2.1 (L'interpretazione di Leibniz)



2.2.2 La caratteristica universale

La *caratteristica universale*, concepita come lingua o scrittura universale, si fonda sui seguenti principi:

⇒ Le idee sono analizzabili fino a idee semplici (atomiche);

⇒ Le idee possono essere rappresentate da simboli;

⇒ Le relazioni tra idee possono essere rappresentate da simboli;

⇒ Le idee si combinano mediante regole.

Definizione 2.2.1: Caratteristica universale

La *caratteristica universale* è un sistema di segni che rappresentano nozioni e cose, ma non parole. Le connessioni tra i segni rappresentano le relazioni tra le nozioni e le cose. Il nome di una nozione serve a:

⇒ Relazionarla con altre nozioni;

⇒ Relazionarla con lo schema dell'universo;

⇒ Indicare le esperienze necessarie per la conoscenza.

Note:-

L'apprendimento della lingua universale coincide con l'*enciclopedia*.

Il calculus ratiocinator

Definizione 2.2.2: Calculus ratiocinator

Il *calculus ratiocinator* è un insieme di tecniche per manipolare i segni della caratteristica universale. Può essere considerato come un *sistema formale*. Il frammento XX offre un'idea di questo calcolo: in esso vengono trattati molti assiomi e teoremi come il principio degli indiscernibili, la simmetria, la transitività, etc.

2.2.3 La mereologia

Definizione 2.2.3: Mereologia

La *mereologia* è la "dottrina delle parti".

Definizione 2.2.4: Principio di identità degli indiscernibili

Soo identici i termini che possono essere sostituiti a vicenda senza alterare la verità degli enunciati in cui compaiono. Si scrive A = B quando A e B sono identici.

$\Rightarrow$ $B + N = L$ significa che B e N compongono L, cioè che B è parte di L;

$\Rightarrow$ $A \leqslant B$ significa che A è parte di B.

Teorema 2.2.1 Simmetria

Se $A = B$ allora $B = A$.

Teorema 2.2.2 Transitività

Se $A = B$ e $B = C$ allora $A = C$.

Teorema 2.2.3

Se $A + B = B$ allora $A \leqslant B$.

Teorema 2.2.4

Se $A = B$ allora $A + L = B + L$.

2.3 Peano

Peano nacque a Cuneo il 27 agosto 1858. Si laureò in matematica nel 1880 e divenne professore di analisi matematica a Torino nel 1890. Nel 1889 pubblicò i suoi famosi *Principi di aritmetica*, in cui formulò un sistema assiomatico per i numeri naturali.

Definizione 2.3.1: Principi di aritmetica

I *Principi di aritmetica* sono un insieme di assiomi che definiscono i numeri naturali.

2.3.1 Sistemi formali

Note:-

Sono trattati più in dettaglio nel corso "Metodi formali dell'Informatica" e, in misura minore, nel corso di "Linguaggi e paradigmi di programmazione".

Definizione 2.3.2: Sistema formale

In un *sistema formale* si definiscono:

- ⇒ *Oggetti*, che sono tutti i termini costruibili a partire da atomi mediante operazioni;
- ⇒ *Proposizioni*, della forma $P(a_1, ..., a_n)$ dove P è un *predicato* e a_k sono termini;
- ⇒ *Regole di inferenza*, che permettono di dedurre nuove proposizioni. La conclusione di una regola di inferenza senza premesse è un'*assioma*.

$$\frac{A_1 \quad ... \quad A_v}{A}$$

Domanda 2.2

Come si usano le regole di inferenza?

Risposta: Per costruire derivazioni:

$$\frac{\frac{A_1 \qquad A_2}{A} \qquad B}{C}$$

Esempio 2.3.1 (Numeri naturali)

⇒ *Oggetti*: un atomo 0, un'operazione S, con un solo argomento;

⇒ *Proposizioni*: un predicato Num con un solo argomento;

⇒ *Regole di inferenza*: $\frac{}{\text{Num}\{0\}}$ e $\frac{\text{Num}\{n\}}{\text{Num}\{S(n)\}}$.

2.3.2 Confronto tra logica e informatica

Logica:

⇒ Si studiano modelli di processi deduttivi;

⇒ Si studiano modelli di ragionamento;

⇒ Operatori modali.

Informatica:

⇒ Si studiano modelli di processi trasformazionali;

⇒ Si studiano modelli di calcolo (sequenziali, paralleli, distribuiti);

2.4 Kleene e le funzioni ricorsive

Nel 1981 Stephen Cole Kleene, matematico e logico statunitense, pubblicò l'articolo "*Origins of recursive function theory*" in cui descrisse l'attività di ricerca svolta da alcuni matematici e logici negli anni '30. In particolare, Kleene descrisse una serie di lezioni di Gödel tenute nel 1934 a Princeton sullo sviluppo di definizioni ricorsive primitive di funzioni numeriche. Lo *schema di ricorsione primitiva* permette di definire una funzione di numeri naturali f($\mathbf{x}$, n) a partire da funzioni predefinite b($\mathbf{x}$) e h($\mathbf{x}$, n, m) mediante lo schema:

$$f(x,0) = b(x)$$

$$f(x,n+1) = h(x,n,f(x,n))$$

Inoltre si ha la composizione:

$$f(x) = h(g_1(x),\cdots,g_k(x))$$

Gödel usava la nozione di *sistema matematico formale* basata sulla nozione informale di *regola costruttiva* la cui applicazione si basava su una *procedura finita* del tipo necessario per calcolare funzioni definite mediante ricorsioni generali.

Nasce il problema della caratterizzazione delle ricorsioni ammissibili. Kleene aggiunse un *operatore di ricerca* non limitato che dato un predicato P(x, y) restituisce il valore minimo che soddisfa il predicato.

Nel 1938, Kleene ammise la possibilità di definire funzioni parziali.

2.5 Curry e la logica combinatoria

Nel 1927, Haskell Brooks Curry, matematico e logico statunitense, riscopri la nozione di *combinatore* (introdotta da Moses Schönfinkel nel 1920).

Definizione 2.5.1: Combinatore

Un *combinatore* è una funzione senza variabili libere.

Note:-

I sistemi di combinatori attuali usano K e S come combinatori, perchè sono sufficienti a generare tutti gli altri combinatori.

Esempio 2.5.1 (Combinatori)

- ⇒ K(x, y) = x;
- ⇒ S(x, y, z) = x(z)(y(z)).;
- ⇒ B(x, y, z) = x(y(z));
- ⇒ C(x, y, z) = x(z)(y);
- ⇒ W(x, y) = x(y)(y).
- ⇒ I(x) = x.

Inoltre riprende anche la possibilità di trattare funzioni a più argomenti come funzioni a un solo argomento che verrà chiamata *curryficazione*.

$$f(x, y) = f'(x)(y)$$

Nello stesso tempo, Alonzo Church, sviluppa il *lambda calcolo*[1].

Definizione 2.5.2: Lambda calcolo

Il *lambda calcolo* è un sistema formale per la definizione di funzioni intese come regole di corrispondenza. $\lambda x.M$ descrive la regola che assegna a un argomento x il valore M.

2.5.1 Paradosso di Russell e combinatori

$$WS(BWB) = Y = \lambda x.(\lambda z.x(zz))(\lambda z.x(zz))$$

Note:-

Si tratta del combinatore di punto fisso. Permette di definire funzioni ricorsive.

Definizione 2.5.3: Paradosso di Russell

$$(\lambda x.(\lambda z.x(zz))(\lambda z.x(zz)))\neg = (\lambda z.\neg(zz))(\lambda z.\neg(zz))$$

In cui $\{z|\neg(z \in z)\} \in \{z|\neg(z \in z)\}$. Ciò vuol dire che qualcosa appartiene a se stesso, ma non può appartenere a se stesso.

[1] Visto approfonditamente nei corsi "Metodi formali dell'informatica" e "Linguaggi e paradigmi di programmazione, per cui in questi appunti non andrò in dettaglio dato che esula gli obiettivi del corso

2.6 L'analisi dei processi di calcolo di Turing

Nel paragrafo 9 del suo articolo del 1936, Turing introduce il comportamento di un *computor*. Lo esemplifica con un foglio di carta che viene astratto come un "nastro infinito" suddiviso in caselle (quadratini):

⇒ Il numero di simboli che si possono stampare è finito, perchè se ci fossero infiniti simboli ci si potrebbe confondere[2];

⇒ Il comportamento è determinato dallo stato mentale e dal simbolo osservato, c'è un limite alla percezione delle caselle per lo stesso motivo di prima;

⇒ Le operazioni sono elementari, cioè non possono essere ulteriormente scomposte. Esse consistono in cambiamenti di stato mentale e del simbolo osservato[3].

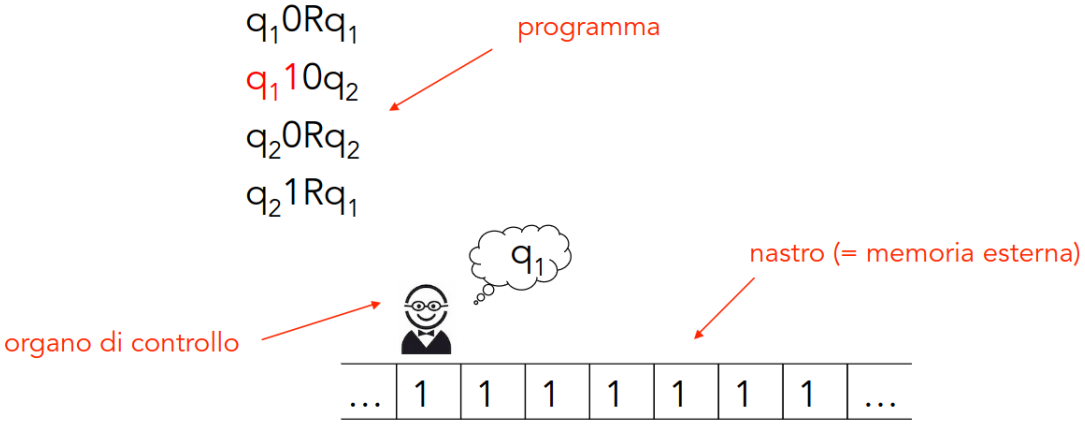


Figure 2.1: Macchine di Turing come Computer.

Definizione 2.6.1: Macchina universale

Una *macchina universale* (U) è una macchina di Turing che può simulare il comportamento di qualsiasi altra macchina di Turing (M) dato il suo programma (I_M) per ogni dato x:

$$U(I_M, x) = M(x)$$

È possibile che U sia più complessa di M. In informatica, il programma di U è chiamato *interprete*.

Note:-

Questo risultato suggerirà a von Neumann la possibilità di automi che generano automi di pari o maggiore complessità.

2.6.1 Non tutte le funzioni sono calcolabili da una macchina di Turing

Per dimostare che non tutte le funzioni sono calcolabili da una macchina di Turing si può utilizzare la *diagonalizzazione di Cantor*:

⇒ Si immagina di poter numerare le funzioni da $\mathbb{N}$ in $\mathbb{N}$;

⇒ Si considera una funzione da $\mathbb{N}$ in $\{0,1\}^{\mathbb{N}}$;

[2] Un po' come il fatto che molti caratteri Unicode non possono essere usati nei nomi dei siti web in quanto troppo "simili".
[3] Un solo simbolo viene alterato.

⇒ Questo elenco può essere visto come una tabella;

⇒ Si vuole dimostrare che esiste una sequenza che non è in questa tabella;

⇒ Si prende la diagonale, che interseca tutte le righe in un punto;

⇒ Si cambiano tutti i valori della diagonale (0 diventa 1 e viceversa);

⇒ La diagonale trasformata può essere una riga dell'originale?

⇒ No, perchè differisce da ogni riga in almeno un punto.

0	1	1	0	0	1	1	0	1	0	...
1	1	0	1	1	1	0	1	0	0	...
2	1	0	1	0	0	0	0	1	1	...
3	0	0	0	1	0	0	0	0	0	...
4	1	0	1	0	0	0	1	0	1	...
5	1	0	0	0	0	1	0	1	1	...
6	1	1	1	1	1	1	0	1	1	...
7	1	0	0	0	0	0	0	0	1	...
8	0	0	0	0	0	0	0	0	1	...
...	...	...	...	...	...	...	...	...	...	...

Figure 2.2: Diagonalizzazione di Cantor.

0	0	1	0	0	1	1	0	1	0	...
1	1	1	1	1	1	0	1	0	0	...
2	1	0	0	0	0	0	0	1	1	...
3	0	0	0	1	0	0	0	0	0	...
4	1	0	1	0	1	0	1	0	1	...
5	1	0	0	0	0	0	0	1	1	...
6	1	1	1	1	1	1	1	1	1	...
7	1	0	0	0	0	0	0	1	1	...
8	0	0	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...

Figure 2.3: Diagonalizzazione di Cantor, inversione della diagonale.

Corollario 2.6.1 Problemi indecidibili

Non esiste una macchina di Turing M che, operando su un nastro che contiene:

$\Rightarrow$ La descrizione di una qualsiasi macchina di Turing T;

$\Rightarrow$ Un input x;

termina sempre i suoi calcoli scrivendo sul nastro il valore $M(T, x)$ dove:

$\Rightarrow$ $M(T, x) = 1$ se $T(x)$ termina;

$\Rightarrow$ $M(T, x) = 0$ se $T(x)$ non termina.

Note:-

Il problema della fermata è indecidibile.

3
Informazione e cibernentica

3.1 Shannon

Claude Shannon è un matematico che ha lavorato per la Bell Labs, e ha scritto un articolo nel 1948, *A Mathematical Theory of Communication*, in cui ha definito la teoria dell'informazione[1]: "il problema fondamentale della comunicazione è quello di riprodurre ad un punto, esattamente o con qualche approssimazione, un messaggio scelto in un altro punto".

Shannon definisce un contesto tecnico di termini:

⇒ *Emittente ricevente*: il mittente è colui che invia il messaggio, il ricevente è colui che lo riceve.

⇒ *Messaggio*: è l'informazione che si vuole trasmettere.

⇒ *Rumore*: è tutto ciò che può interferire con la trasmissione del messaggio.

⇒ *Informazione*: è la quantità di incertezza che si riduce nel ricevente dopo aver ricevuto il messaggio.

Note:-

L'informazione in sé è considerata in base al numero di possibili messaggi che possono essere inviati.

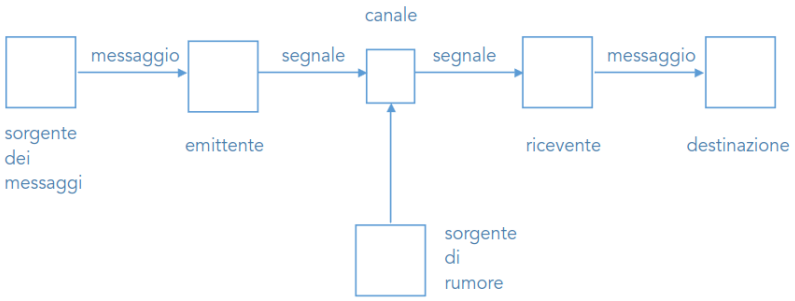


Figure 3.1: Schema di Shannon

[1]Nella laurea magistrale è presente il corso "Teoria dell'Informazione" che approfondice questi argomenti.

Meta-nota 3.1.1

L'informatica non è solamente uno scheletro che può essere riempito con qualsiasi cosa. Alla sua base vi sono idee e atteggiamenti fondanti che possono essere applicati in molti campi. Un esempio è la cibernetica che ha avuto una breve durata (circa 20 anni), ma ha avuto un impatto molto forte su molti campi.
Esiste un'epistemologia dell'informatica, che ne studia le basi e le fondamenta.
Se si vuole approfondire l'argomento, nella prima parte del corso "Metodologie e Tecnologie Didattiche dell'Informatica" (MTDI o PREFIT) si ha una panoramica "motivazionale" sulle basi dell'informatica e sul suo essere una scienza.

3.1.1 Il modello di comunicazione

Il modello di comunicazione di Shannon purtroppo non mette in evidenza il fattore temporale e il ritardo. Quando si comunica in un contesto asincrono bisogna utilizzare un sistema di feedback per capire se il messaggio è stato ricevuto correttamente[2]. Il modello di comunicazione di Shannon ha avuto varie interpretazioni, una delle quali è quella di Chapman.

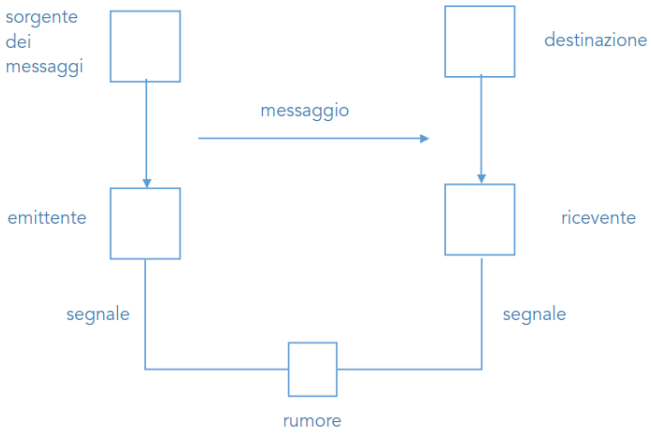


Figure 3.2: Modello di comunicazione di Shannon, reinterpretato da Chapman.

In questa interpretazione vengono identificati i livelli a cui i segnali possono esistere:

⇒ Nello strato basso è presente il rumore;

⇒ Al livello superiore è presente il messaggio.

3.2 Cibernetica e neurofisiologia

La cibernentica viene codificata da Norbert Wiener nel 1948, e si occupa di studiare i sistemi di controllo e di comunicazione nei sistemi biologici e nelle macchine.

Von Neumann aveva come obiettivo l'unificazione del lavoro di McCulloch e Pitts con il lavoro di Shannon e di Turing. Per Wiener, la nozione unificante era il feedback, mentre per von Neumann era la nozione di automa come elaboratore di informazione.

[2]Per esempio il sistema di ACK (Acknowledgement) di TCP visto nel corso "Reti I"/"Reti di calcolatori".

Lavori di Von Neumann sugli automi:

⇒ *The general and logical theory of automata*, Hixon Symposium, 1948;

⇒ *Theory and organization of complicated automata*, serie di 5 lezioni all'università dello Illinois, 1949;

⇒ *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, appunti delle lezioni alla Caltech, 1952;

⇒ *The Theory of Automata: Construction, Reproduction, Homogeneity*, 1952-1953;

⇒ *The computer and the brain*, Yale University Press, 1956 pubblicato postumo.

Per ricapitolare le relazioni tra i vari "attori":

⇒ *Shannon - Turing*: si incontrano nel 19343 ai Bell Labs, e discutono di comunicazione[3];

⇒ *Wiener - Von Neumann - McCulloch - Pitts*: partecipano alle conferenze di Macy la cui nozione centrale è il messaggio (con le accezioni di Shannon);

⇒ *Wiener - McCulloch - Pitts*: membri del RLE;

⇒ *Von Neumann - Turing*: si incontrano nel 1937 a Princeton. Nel 1939 Von Neumann offre a Turing un posto di lavoro come assistente a Princeton, ma Turing rifiuta;

⇒ *Shannon - Wiener*: sviluppano una teoria matematica e della comunicazione basata sulla concezione dell'informazione/entropia di un insieme di messaggi.

Meta-nota 3.2.1

Pare che Shannon adorasse il monociclo e che spesso tenesse dei party in cui si esibiva in numeri di equilibrismo.

Figure 3.3: Shannon sul monociclo

3.3 Von Neumann e la termodinamica del calcolo

Von Neumann cercò di valutare il costo energetico minimo per un atto elementare di generazione di informazioni. Da ciò deriva, nel 1949, la formula:

$$kT \log_e N$$

[3]Da ricordare Enigma e la crittografia.

⇒ k: costante di Boltzmann;

⇒ T: temperatura;

⇒ N: numero di alternative o possibili stati.

Questo calcolo attira le attenzioni di *Landauer*, un fisico teorico, che non riesce a dimostrarlo, ma osservo che le operazioni logicamente irreversibili (come l'operazione di cancellazione di un bit) generano entropia pari alla quantità di informazione cancellata. *Bennet*, un suo studente, nel 1973, dimostra che ogni calcolo può essere reso logicamente reversibile. *Fredkin* lavora a una base fisica per i calcoli reversibili.

Note:-

Da questo tipo di interessi nasce la computazione quantistica (che cosa succede se si considera la computazione come composta da operazioni maccaniche?).

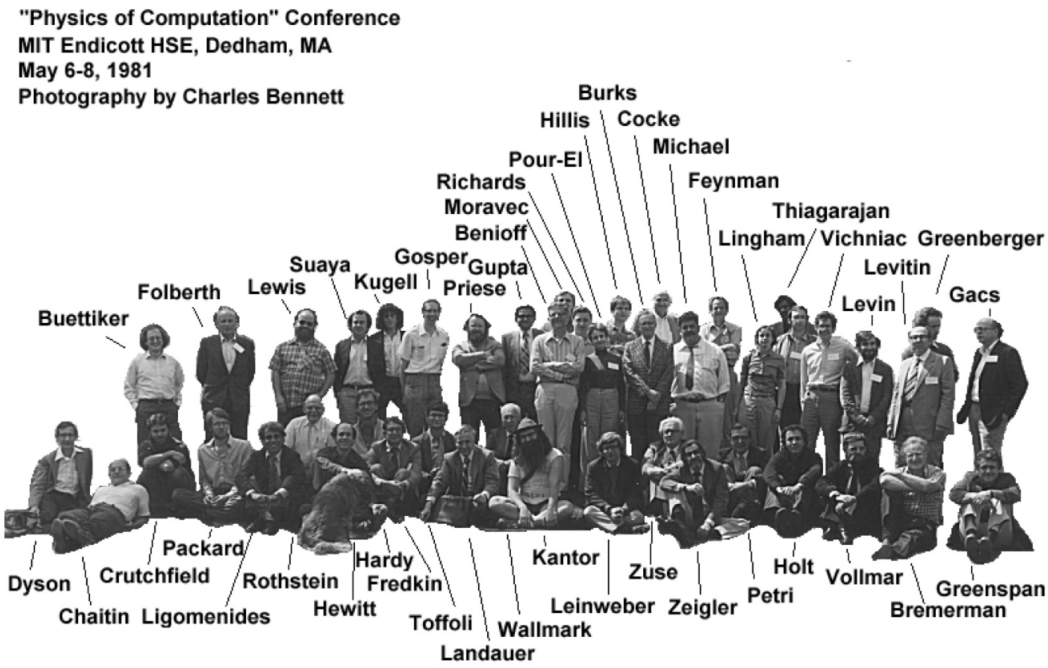


Figure 3.4: Physics of Computation

Meta-nota 3.3.1

⇒ *Bennet*: è un fisico teorico, non è direttamente presente nella foto perchè fu lui a scattarla;

⇒ *Landauer*: è presente nella foto in primo piano;

⇒ *Fredkin*: è presente nella foto, vicino a Landauer e Toffoli;

⇒ *Toffoli*: fu uno studioso di fisica della computazione che interagi con Burks;

⇒ *Burks*: in secondo piano;

⇒ *Feynman*, *Dysan* e altri fisici famosi;

⇒ *Zuse*: già visto nella prima parte del corso, inventore dello Z1;

⇒ *Kantor*: inventore e specialista dell'informazione come principio ontologico;

⇒ *Holt*: esperto di cibernentica;

⇒ *Gosper*: inventore della configurazione ad aliante, uno dei primi "hacker".

3.4 La cibernetica

Quando si *comunica* con un'altra persona gli si trasmette un messaggio, e quando l'altra persona risponde, si riceve un messaggio che contiene informazioni accessibili a sé stessi e all'altro. Quando si *controllano* le azioni di un'altra persona gli si comunica un messaggio (in forma imperativa). Tutto ciò si riduce a uno *scambio di messaggi*.

Definizione 3.4.1: Cibernetica

La *cibernetica* può essere vista come una teoria della trasmissione dei messaggi e delle sue condizioni che presentano un successo.

Note:-

Il computer diventa un oggetto cibernetico, in quanto è inserito in un contesto di comunicazione con esseri umani o con altri computer.

3.4.1 Il feedback e il flusso di messaggi nei sistemi

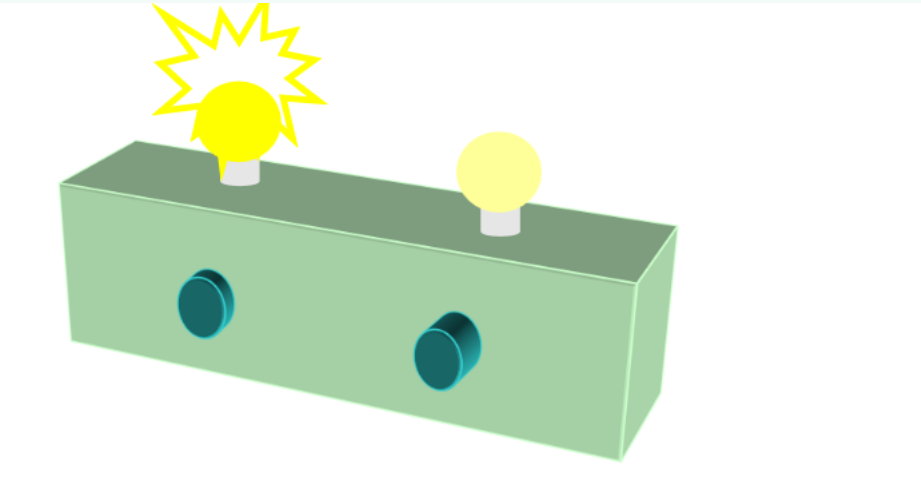
Wiener confrontava l'attività umana con quella di figure che danzano sopra un carillon secondo un modello. Tuttavia il modello in esame è un modello predisposto in cui il loro passato non ha influenza sul loro futuro. Non c'è feedback.

Figure 3.5: Modello IO

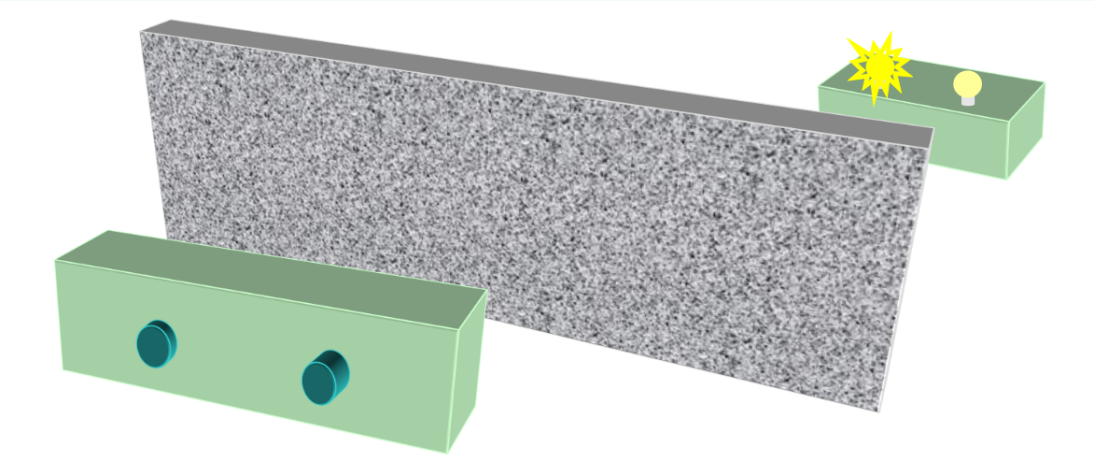
Holt propose l'idea che quando una persona scrive su una macchina premendo il tasto esso fornisce un feedback tornando indietro. La temporizzazione dei tasti è ciò che consente la scrittura. Questo si può estendere all'utente e all'interfaccia: la scelta di un'interfaccia comporta la progettazione dell'utente indicando cosa si può o non si può fare.

Esempio 3.4.1 (Scatola nera)

Si immagini una scatola con dei pulsanti che si possono premere. Quando si preme un pulsante si accende una luce. Prima di poter schiacciare un altro pulsante bisogna controllare che la luce si sia accesa. In questo caso la luce è il feedback.



Se i pulsanti sono premuti da un operatore che non può vedere le luci e le luci sono visibili a un altro operatore che non può premere i pulsanti, non si potrebbe stabilire una connessione tra luci e pulsanti e dunque la connessione non ha succeso.



3.4.2 Osservatori

In alcuni sistemi, per esempio un termostato, è l'osservatore a concettualizzarlo. Nei sistemi di secondo ordine, l'osservatore è esso stesso parte del sistema osservato.

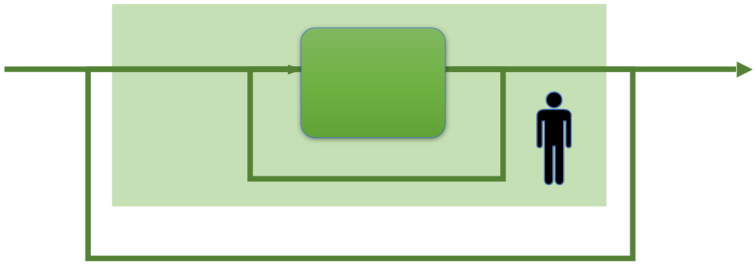


Figure 3.6: Osservatore

3.4.3 Comunicazione sincrona e asincrona

Definizione 3.4.2: Comunicazione sincrona

La *comunicazione sincrona* è una comunicazione in cui il mittente e il ricevente sono sincronizzati.
In una comunicazione sincrona è presente un unico sistema di riferimento temporale condiviso da tutti i partecipanti.

Esempio 3.4.2 (Comunicazione sincrona)

⇒ Chiamata telefonica;

⇒ Conversazione faccia a faccia;

⇒ Sistema di posta elettronica in cui i messaggi impiegano n secondi per essere consegnati;

⇒ Segnali di fumo.

Definizione 3.4.3: Comunicazione asincrona

La *comunicazione asincrona* è una comunicazione in cui il mittente e il ricevente non sono sincronizzati.
In una comunicazione asincrona non è presente un unico sistema di riferimento temporale condiviso da tutti i partecipanti.

Esempio 3.4.3 (Comunicazione asincrona)

⇒ Sistema di posta elettronica con garanzia di consegna entro n secondi;

⇒ Sistema postale ordinario;

⇒ Messaggi in bottiglia.

Note:-

Nella comunicazione asincrona è necessario un sistema di feedback per capire se il messaggio è stato ricevuto.

3.4.4 Circuiti asincroni

Definizione 3.4.4: Circuiti asincroni

I *circuiti asincroni* sono circuiti in cui i segnali di controllo non sono sincronizzati con un segnale di clock (orologio globale).

Corollario 3.4.1 C-Muller

Un elemento asincrono è il C-Muller per cui l'uscita è 1 se e solo se entrambi gli ingressi sono 1 e diventa 0 se entrambi gli ingressi sono 0.

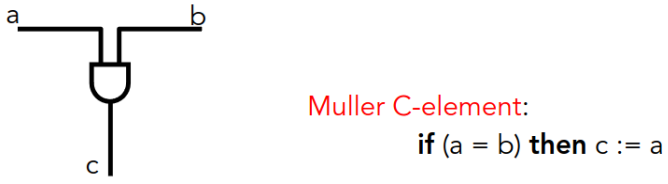


Figure 3.7: C-Muller

Note:-

Combinando più C-Muller si può costruire un circuito asincrono che implementa le micropipeline.

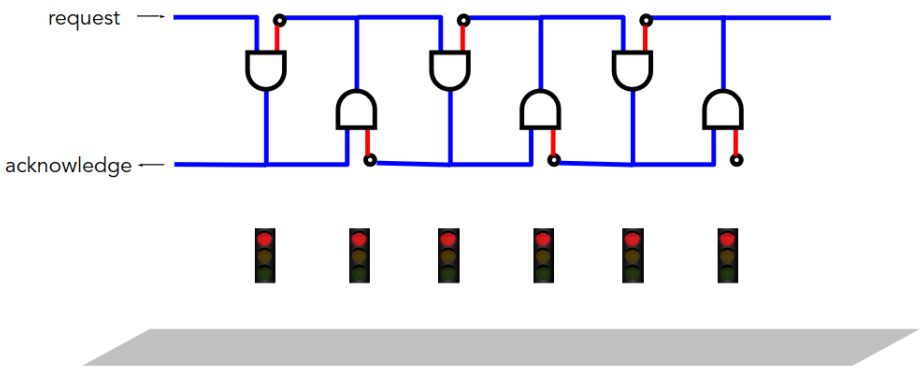


Figure 3.8: Micropipeline

Corollario 3.4.2 Reti di Petri

Grafi che generalizzano gli automi a stati finiti (DFA e NFA). Si hanno posti (condizioni) e transizioni (eventi).

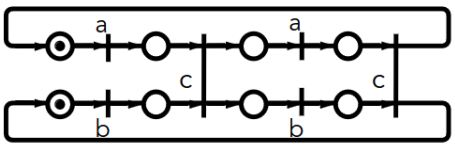


Figure 3.9: Rete di Petri

4

Bush e il Memex, Engelbard e "The Mother of All Demos"

4.1 Introduzione

Definizione 4.1.1: Knowledge Navigator

Un'idea di Apple, presentata nel 1987, di un assistente virtuale che aiuta l'utente a navigare tra le informazioni.

Il filmato di presentazione del Knowledge Navigator (1987):

⇒ *La comprensione del parlato*: il computer capisce il linguaggio naturale;

⇒ *La grafica e le finestre*: dietro questo video c'è Alan Kay, che ha lavorato a Xerox PARC e fu l'inventore delle finestre;

⇒ *Il touch screen*;

⇒ *La videochiamata*;

⇒ *La simulazione*: della desertificazione.

Domanda 4.1

Chi era Vannevar Bush?

Risposta: Vannevar Bush (1890 - 1974) era un ingegnere e scienziato americano. Lui teorizzò il Memex (non fu mai realizzato).

Note:-

Durante una conferenza in occasione del 50° anniversario di "As we may think" (1945) venne presentata un'animazione del Memex.

4.2 Il Memex

Definizione 4.2.1: Memex

Un sistema di archiviazione e ricerca delle informazioni, teorizzato da Vannevar Bush.

Note:-

Il Memex offre anche un'anticipazione di ciò che sarà l' *ipertesto*, che nascerà vent'anni dopo.

4.2.1 Problemi di organizzazione dell'informazione

Il problema che preocupa Bush è *la perdita di informazioni* che si accumula continuamente nel tempo[1]. Inoltre si aumenta progressivamente la specializzazione: le informazioni sono sempre più frammentate e sempre più specializzate (più difficili da comunicare). Bush ritiene che il problema non sia l'eccesso di pubblicazioni, ma il fatto che si siano estese oltre la capacità di gestione dei documenti. La *selezione*[2] è un problema per via delle enormi quantità di informazioni.

Esempio 4.2.1 (Addetto dell'ufficio informazioni)

L'addetto all'ufficio del personale di una fabbrica immette una pila di alcune migliaia di schede degli impiegati in una macchina selezionatrice, imposta un codice secondo una convenzione stabilita e produce in poco tempo una lista di tutti gli impiegati che vivono a Trenton e conoscono lo spagnolo.

Esempio 4.2.2 (Centralini telefonici automatici)

Si compone un numero e la macchina seleziona e connette solamente una tra un milione di possibili stazioni. Non le ispeziona tutte. Presta attenzione solo a una classe data dalla prima cifra, poi solo a una sottoclasse data dalla seconda cifra e così via; così procede rapidamente e quasi infallibilmente verso la stazione selezionata.

Prroblemi di indicizzazione:

⇒ Si cerca da sottoclasse a sottoclasse;

⇒ L'informazione si trova in un unico punto (a meno che non sia duplicata);

⇒ Bisogna avere regole per specificare il percorso, ma le regole sono complicate;

⇒ Quando si trova l'elemento bisogna riemergere dal sistema e rientrare attraverso un nuovo percorso.

Note:-

La situazione peggiore, ma più facilmente verificabile, è la ricerca in un albero.

Definizione 4.2.2: Classificazione

La *classificazione* è una segmentazione spaziale, temporale o spazio-temporale del mondo.
Un sistema di classificazione è un insieme di scatole in cui si mettono cose per fare un qualche tipo di lavoro.

Proprietà di un sistema di classificazione:

⇒ Ci sono principi univoci e consistenti;

⇒ Le categorie sono reciprocamente esclusive;

⇒ Il sistema è completo.

[1] Questo problema non era nuovo. Era già stato affrontato da Paul Otlet.
[2] Processo di scelta delle informazioni.

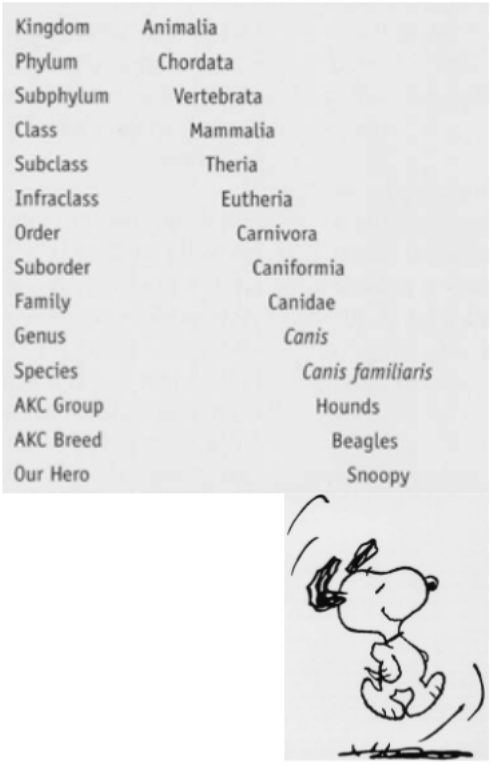


Figure 4.1: Snoopy secondo due sistemi di classificazione.

4.2.2 Indicizzazione

Definizione 4.2.3: Indicizzazione

L'*indicizzazione* è un'operazione è l'azione di descrivere o identificare un documento (o un oggetto) nei termini del suo contenuto concettuale[a].

[a] ISO 5963.

Note:-

Lo scopo generale dell'indicizzazione è quello di rappresentare oggetti in modo che possano essere efficacemente trovati e utilizzati.

Corollario 4.2.1 Linguaggio di indicizzazione

Un linguaggio di indicizzazione è un sistema di rappresentazioni simboliche (un codice) che consentono la classificazione e la ricerca di documenti attraverso i codici assegnati ai concetti che essi contengono (indicizzazione per concetti).

Note:-

Ted Nelson, in "As we may think", criticherà il fraintendimento per cui si vede nel lavoro di Bush un contributo alla information retrieval[a].

[a] Reperimento di oggetti informativi.

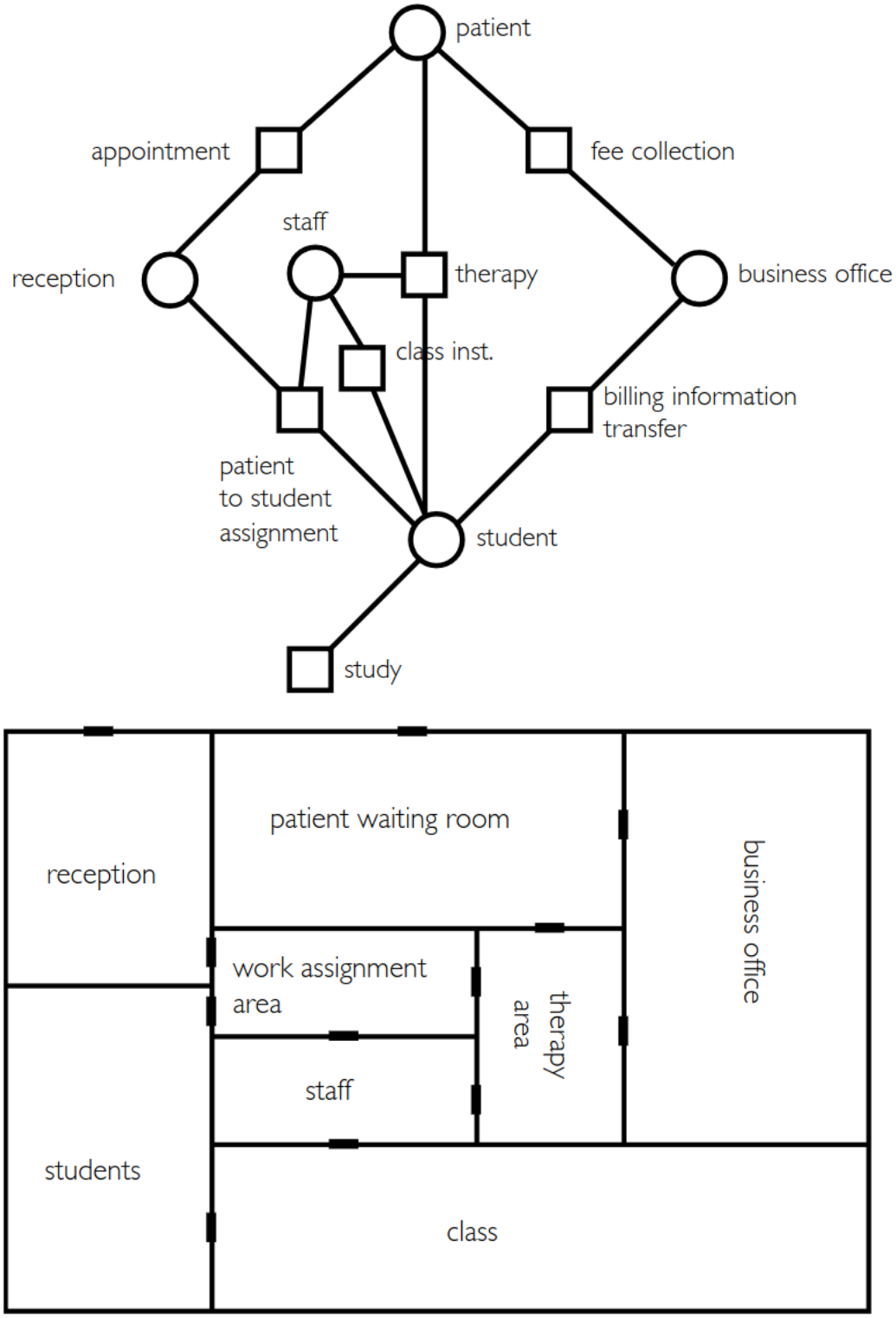


Figure 4.2: Il sistema di classificazione della scuola dentistica dell'Università di Boston.

5 Domande

5.1 Lullo, Leibniz e la storia del calcolo

5.2 Turing e la fisica del calcolo

5.3 Funzioni calcolabili e combinatori

5.4 "As we may think"

5.5 Engelbart e Nelson

5.6 Otlet, Lickider e Kay