Appunti di Fondamenti

Fondamenti dell'Informatica (prof. Peñaloza) - CdL Informatica Unimib - 23/24

Federico Zotti

08 Nov 2023

Indice

1 Matematica discreta

> **Discreto**: composto di elementi distinti, separati tra di loro.

Un sistema è:

- **Discreto** se è costituito da elementi isolati
- **Continuo** se non ci sono *vuoti* tra gli elementi

I sistemi informatici si basano su un sistema *binario*, perciò discreto.

Possiamo approssimare un sistema continuo dividendolo in piccole parti (*discretizzazione* o *digitalizzazione*).

1.1 Fasi della matematica discreta

- **Classificazione**: individuare le caratteristiche comuni di entità diverse (*teoria degli insiemi*)
- **Enumerazione**: assegnare ad ogni oggetto un numero naturale (*contare*)
- **Combinazione**: permutarne e combinarne gli elementi (*grafi*)

Queste fasi guidano un **algoritmo**.

1.2 Logica

In filosofia, la **logica** è lo studio del ragionamento, dell'argomentazione, e dei procedimenti **inferenziali** per distinguere quelli *validi* da quelli *non validi*.

La **logica matematica** vede questi procedimenti come calcoli formali, con una struttura algoritmica.

Infatti, è tutto basato sull'**algebra di Boole**.

1.2.1 Algebra astratta

L'algebra astratta studia le **strutture algebriche**, ovvero insiemi muniti di operazioni.

2 Insiemi e Operazioni

2.1 Numeri

2.1.1 Numeri naturali

I numeri **naturali** sono i primi che impariamo, e nascono dall'attività di contare.

Essi formano un **insieme**, chiamato *insieme dei numeri naturali* ($\mathbb{N}$).

$$\mathbb{N} = \{ 0, 1, 2, 3, 4, \dots, n, n + 1, \dots \}$$

Contare non è altro che assegnare ad ogni oggetto un numero naturale (in ordine).

$\mathbb{N}$ ha un *limite inferiore* (0), ma non ha un *limite superiore*, quindi $\mathbb{N}$ è infinito.

2.1.1.1 Definizione semiformale

- I numeri naturali hanno l'elemento 0
- Ogni elemento n ha (**esattamente**) un successore $s(n)$
- 0 non è un successore di nessun elemento
- Due elementi diversi hanno successori diversi

Questa definizione è la base del **processo di induzione**.

Una proprietà è vera in tutto $\mathbb{N}$ se e solo se:

- È vera in 0
- Se è vera in n allora è vera in $s(n)$

È possibile anche iniziare da un numero arbitrario.

2.1.2 Numeri interi

I numeri **interi** (relativi) è l'insieme dei numeri naturali preceduti da un segno "$+$" o "$-$".
Questo insieme si denota con il simbolo $\mathbb{Z}$.

$$\mathbb{Z} = \{ \ldots, -(n+1), -n, \ldots, -2, -1, 0, 1, 2, \ldots, n, n+1, \ldots \}$$

Ogni intero ha un successore, ma anche un **predecessore** (non c'è un *minimo*).

I numeri interi positivi (più 0) formano $\mathbb{N}$.

$$\mathbb{N} \subset \mathbb{Z}$$
$$\mathbb{N} = \mathbb{Z}^{+} \cup \{ 0 \}$$

2.1.2.1 Valore assoluto

Il **valore assoluto** di un numero intero è il numero privo di segno.

$$|-n| = n$$
$$|n| = n$$

L'**opposto** di un numero si ottiene cambiandogli il segno.

2.1.3 Numeri razionali

Razionale in questo caso si riferisce a **ratio** ossia **proporzione**. Indicano dunque una proporzione risultante da una divisione.

Si esprimono come rapporto di due numeri interi (*frazioni*).

$$\frac{m}{n}$$

Si indicano con il simbolo $\mathbb{Q}$.

2.1.3.1 Rappresentazioni e Relazioni

Ogni numero razionale può essere rappresentato da un numero decimale finito o periodico.

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q}$$

2.1.3.2 Densità

I numeri razionali sono **densi**: fra due razionali c'è sempre un altro numero.

Sono comunque **discreti**.

2.1.4 Numeri reali

I **numeri irrazionali** ($\mathbb{I}$) sono quelli che non si possono esprimere tramite frazioni: hanno un'espansione decimale infinita e non periodica.

L'insieme dei **numeri reali** ($\mathbb{R}$) contiene tutti i numeri che ammettono una rappresentazione decimale.

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$$
$$\mathbb{R} = \mathbb{Q} \cup \mathbb{I}$$

2.1.4.1 La Retta reale

L'insieme dei numeri reali spesso viene rappresentato su una **retta** (ordine implicito).

A ogni punto della retta è associato un numero reale e viceversa (*corrispondenza biunivoca*).

2.1.5 Numeri complessi

I **numeri complessi** ($\mathbb{C}$) estendono i reali per eseguire operazioni che non sono ben definite altrimenti.

Nascono dalla necessità di estrarre radici a numeri negativi.

Definiscono l'**unità immaginaria** $i = \sqrt{-1}$. Un numero complesso è $a + bi$, con $a, b \in \mathbb{R}$

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

2.1.6 Numeri booleani

L'insieme dei **numeri booleani** è

$$\mathbb{B} = \{ 0, 1 \}$$

2.2 Insiemi

Gli **insiemi**, le loro proprietà e le loro **operazioni** sono alla base della matematica moderna e dell'informatica.

Un sistema è **discreto** se costituito da elementi isolati e **continuo** se non vi sono spazi vuoti. In matematica, discreto si basa sul concetto di **cardinalità** (il "numero" di elementi che contiene).

Un insieme è discreto se (e solo se) i suoi elementi si possono **numerare**.

> Un insieme è un raggruppamento di oggetti distinti e ben definiti.

Gli oggetti che formano l'insieme sono i suoi **elementi**. In un insieme, tutti gli elementi sono **distinti** e l'ordine non è rilevante.

> Gli elementi di un insieme possono essere anch'essi insiemi.

Un tempo si pensava che la **teoria degli insiemi** poteva dare una base solida alla matematica. Esistono paradossi però che dicono il contrario.

Per esempio il paradosso del barbiere

> In un villaggio vi è un solo barbiere, che rade tutti e soli gli uomini del villaggio che non si radono da soli. *Chi rade il barbiere?*

o il paradosso eterologico

> Una parola è **autologica** se descrive se stessa (*"polisillabica"*, *"corta"*, *"leggibile"*).
> Una parola è **eterologica** se non è autologica (*"polisillabica"*, *"lunga"*, *"illeggibile"*).
> "*Eterologica*" è eterologica?

Il più famoso di essi è il paradosso degli insiemi (*Bertrand Russel*)

> Considerate l'insieme N di tutti gli insiemi che non appartengono a se stessi. N appartiene a se stesso?

Per costruire questo tipo di paradossi è necessario usare un'**autoreferenza** e una **negazione**.

Questa idea torna in diversi contesti per dimostrare l'impossibilità o inesistenza di certe strutture.

2.2.1 Notazione

Gli insiemi generici saranno denotati da lettere latine maiuscole

$$A, B, C, \dots$$

e i loro elementi con lettere latine minuscole

$$a, b, c, \dots$$

L'insieme senza elementi si chiama **vuoto** e si denota con $\emptyset$.

L'**uguaglianza** fra oggetti (elementi, insiemi, entità, ecc.) si denota con "$=$". La **disuguaglianza** si denota con "$\neq$".

L'uguaglianza ha tre importanti proprietà:

- **Riflessività**: $A = A$
- **Simmetria**: $A = B \iff B = A$
- **Transitività**: se $A = B$ e $B = C$ allora $A = C$

Un insieme può avere diverse rappresentazioni:

- **Diagramma Eulero-Venn**
- **Rappresentazione estensionale**: elenco di tutti gli elementi ($\{ x, y, z \}$)
  - $\{$ rosso, giallo, arancio $\}$: insieme con tre elementi

- $\{ \text{rosso, giallo, rosso} \}$: insieme con due elementi
- $\{ \emptyset \}$: insieme con un elemento
- $\{ 0, 1, 2, 3, \dots \}$: insieme dei numeri naturali
- $\{ \emptyset, 1, 2, \{ 3 \} \}$

- **Rappresentazione intensionale**: consiste nel formulare una proprietà $\mathscr{P}$caratteristica che distingue precisamente gli elementi dell'insieme ($S = \{ x | \mathscr{P}(x) \}$)
  - $\{ x | x \in \mathbb{Z}, x > 0 \}$: insieme dei numeri interi positivi
  - $\{ x | x \text{ è un colore dell'arcobaleno} \}$
  - $\{ x | x \in \mathbb{Z}, x > 3, x \leq 100 \} = \{ 4, 5, \dots, 99, 100 \}$
  - $\{ x | x \text{ è un numero primo} \}$

Per ogni elemento x esiste l'insieme **singoletto** $\{ x \}$.

Proprietà complesse si possono costruire combinando proprietà più semplici mediante operazioni **vero-funzionali**.

Un **sottoinsieme** di A è un insieme formato unicamente per (alcuni) elementi di A. Un sottoinsieme B di A è **proprio** se è diverso da A e da $\emptyset$.

L'insieme vuoto ammette esattamente un sottoinsieme: $\emptyset$ (*sottoinsieme non proprio*). Un singoletto $\{ a \}$ ammette due sottoinsiemi: $\emptyset$ e $\{ a \}$ (*sottoinsiemi non propri*).

Se A e B hanno gli stessi elementi, sono mutuamente sottoinsiemi

$$A = B \text{ se } A \subseteq B, B \subseteq A$$

L'inclusione soddisfa le proprietà:

- **Riflessività**: $A \subseteq A$
- **Antisimmetria**: $A \subseteq B \land B \subseteq A \iff A = B$
- **Transitività**: $A \subseteq B \land B \subseteq C \iff A \subseteq C$

L'insieme potenza (o insieme delle parti) di un insieme S, scritto $\mathscr{P}(S)$ è l'insieme formato da tutti i sottoinsiemi di S.

$$\mathscr{P}(S) = \{ x | x \subseteq S \}$$

Esempi:

- $\mathscr{P}(\varnothing) = \{\varnothing\}$
- $\mathscr{P}(\{\varnothing\}) = \{\varnothing, \{\varnothing\}\}$
- $\mathscr{P}(\{x, y\}) = ?$

Se S ha n elementi ($n \geq 0$) allora $\mathscr{P}(S)$ ha 2^n elementi.

2.2.2 Operazioni

2.2.2.1 Unione

L'**unione** di due insiemi A e B si denota

$$A \cup B$$

ed è definita come

$$A \cup B = \{x | x \in A \vee x \in B\}$$

Le proprietà dell'unione sono:

- **Idempotenza**: $A \cup A = A$
- **Commutatività**: $A \cup B = B \cup A$
- **Associatività**: $A \cup (B \cup C) = (A \cup B) \cup C$
- **Esistenza del neutro**: $A \cup \varnothing = A$
- **Assorbimento**: $A \cup B = B$ se $A \subseteq B$
- **Monotonicità**: $A \subseteq A \cup B$ e $B \subseteq B \cup A$

2.2.2.2 Intersezione

L'**intersezione** di due insiemi A e B si denota

$$A \cap B$$

ed è definita come

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

Le proprietà dell'intersezione sono:

- **Idempotenza**: $A \cap A = A$
- **Commutatività**: $A \cap B = B \cap A$
- **Associatività**: $A \cap (B \cap C) = (A \cap B) \cap C$
- **Annichilazione**: $A \cap \emptyset = \emptyset$
- **Assorbimento**: $A \cap B = B$ se $A \subseteq B$
- **Monotonicità**: $A \cap B \subseteq A$ e $A \cap B \subseteq B$

L'unione e l'intersezione distribuiscono una sull'altra

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

2.2.2.3 Sottrazione

La **sottrazione** tra due insiemi A e B è definita come

$$A \setminus B = \{x | x \in A \wedge x \notin B\}$$

Le proprietà della sottrazione sono:

- $A \setminus A = \emptyset$
- $A \setminus \emptyset = A$
- $\emptyset \setminus A = \emptyset$
- $A \setminus B = A \cap \overline{B}$
- $(A \setminus B) \setminus C = A \setminus (B \cup C) = (A \setminus C) \setminus B$
- $A \setminus B \neq B \setminus A$

2.2.2.4 Differenza simmetrica

La differenza simmetrica tra A e B è

$$A \triangle B = (A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B)$$

Proprietà:

- $A \triangle A = \emptyset$
- $A \triangle \emptyset = A$
- $A \triangle B = B \triangle A$

2.2.2.5 Complementazione

Dato un insieme di riferimento U (chiamato **Universo**), il **complemento** assoluto di A è definito come:

$$\overline{A} = \{ x \mid x \in U, x \notin A \} = U \setminus A$$

Le proprietà della complementazione sono:

- $\overline{U} = \emptyset$
- $\overline{\emptyset} = U$
- $\overline{\overline{A}} = A$
- $A \cap \overline{A} = \emptyset$ (*terzo escluso*)
- $A \cup \overline{A} = U$
- $\overline{(A \cup B)} = \overline{A} \cap \overline{B}$ (*legge di De Morgan*)
- $\overline{(A \cap B)} = \overline{A} \cup \overline{B}$ (*legge di De Morgan*)
- $A \subseteq B \iff \overline{B} \subseteq \overline{A}$

2.2.3 Famiglie di insiemi

Un insieme i cui elementi sono tutti insiemi viene chiamato **famiglia di insiemi** ($\mathscr{F}$).

Le operazioni su una famiglia di insiemi sono:

$$\cup\mathscr{F} = \{x \mid x \in A \text{ per almeno un insieme } A \in \mathscr{F}\}$$
$$\cap\mathscr{F} = \{x \mid x \in A \ \forall A \in \mathscr{F}\}$$

Dunque

$$\cup\mathscr{P}(A) = A \ \forall A$$

2.2.4 Partizioni

Una partizione di un insieme $A \neq \emptyset$ è una famiglia $\mathscr{F}$ di sottoinsiemi di A tale che:

- $\forall c \in \mathscr{F}, c \neq \emptyset$ (*non trivialità*)
- $\cup\mathscr{F} = A$ (*copertura*)
- se $c \in \mathscr{F}$, $D \in \mathscr{F}$ e $C \neq D$, allora $C \cap D = \emptyset$ (*disgiunzione*)

2.3 Relazioni

2.3.1 Ordinamenti negli insiemi

Ricordate che gli insiemi **non** sono ordinati

$$\{x, y\} = \{y, x\}$$

A volte è utile poter ordinare i loro elementi in modo chiaro.

2.3.1.1 Coppia ordinata

Una **coppia ordinata** è una collezione di due elementi, dove si può distinguere il **primo** e il **secondo** elemento

$$\langle x, y\rangle$$

Il primo elemento è x e il secondo è y. Notare che esiste la coppia ordinata $\langle x, x\rangle$.

2.3.1.1.1 Formulazione Insiemistica

La coppia ordinata $\langle x, y\rangle$ non è altro che l'insieme

$$\{\{x\}, \{x, y\}\}$$

Sia $\mathscr{F} = \{\{x\}, \{x, y\}\}$. x è il **primo elemento** $\iff x \in \cap\mathscr{F}$ (*appartiene a tutti gli insiemi*). y è il **secondo elemento** $\iff y \in \cup\mathscr{F} \setminus \cap\mathscr{F}$ (*non appartiene a tutti gli insiemi*) oppure $\{y\} = \cup\mathscr{F}$ ($\mathscr{F} = \{\{y\}\}$).

Notare che $\langle x, x\rangle = \{\{x\}, \{x, x\}\}$.

2.3.1.1.2 Definizione giusta

Vogliamo vedere che questa definizione **caratterizza** le coppie ordinate. Cioè, che

$$\langle a, b\rangle = \langle x, y\rangle \iff \{\{a\}, \{a, b\}\} = \{\{x\}, \{x, y\}\}$$

Le coppie ordinate sono **ben definite**.

2.3.1.1.3 Generalizzazione

Possiamo generalizzare le coppie ordinate a **tuple ordinate** di lunghezza $n \geq 2$ (n-tuple ordinate) definendo

$$\langle x_1, x_2, \dots, x_n, x_{n+1}\rangle = \langle\langle x_1, x_2, \dots, x_n\rangle, x_{n+1}\rangle$$

2.3.1.2 Prodotto cartesiano

Dati due insiemi A e B, definiamo il prodotto cartesiano come

$$A \times B = \{\langle x, y\rangle \mid x \in A, y \in B\}$$

$A \times B$ è l'insieme di tutte le coppie ordinate dove:

- il primo elemento appartiene ad A
- il secondo elemento appartiene a B

Notare che:

- $A \times B \neq B \times A$
- $A \times \emptyset = \emptyset = \emptyset \times A$

$A \times A$ è a volte denotato con A^2.

2.3.1.3 Sequenze

S^n è l'insieme di tutte le n-tuple di elementi di S definito tramite prodotti cartesiani di S. Una **sequenza finita** di elementi di S è un elemento di S^n per qualche $n \in \mathbb{N}$.

In altre parole, una sequenza è una tupla ordinata

$$\langle s_1, \dots s_n \rangle$$

dove $n \in \mathbb{N}$ e ogni $s_i \in S$.

2.3.1.4 Segmento

Data una sequenza finita $\sigma = \langle s_1, \dots, s_n \rangle$, una sequenza $\sigma' = \langle s_k, s_{k+1}, \dots, s_\ell \rangle$ dove $1 \leq k \leq \ell \leq n$ è chiamata un **segmento** di σ.

Il segmento è **iniziale** sse $k = 1$.

2.3.2 Relazioni

Una **relazione** tra gli elementi di due insiemi A e B non è altro che un sottoinsieme di $A \times B$.

Una relazione rappresenta un **collegamento** tra gli elementi di A e quelli di B.

2.3.3 Relazioni tra oggetti

Se la coppia ordinata $\langle x, y\rangle$ appartiene a una relazione $R \subseteq A \times B$, si dice che $x \in A$ ha come **corrispondente** $y \in B$ nella relazione R oppure che x è *in relazione con* y.

2.3.4 Rappresentazione tabulare

Ogni relazione si può rappresentare graficamente tramite una tabella.

2.3.5 Rappresentazione matriciale

R si può anche rappresentare tramite una **matrice booleana**.

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Ogni riga rappresenta un elemento dell'insieme A e ogni colonna rappresenta un elemento di B.

2.3.6 Elementi di una relazione

Sia $R \subseteq A \times B$ una relazione

- Il **dominio** di R ($\text{dom}(R)$) è l'insieme di tutti gli oggetti $x \in A$ tali che $\langle x, y\rangle \in R$ per qualche $y \in B$.

  $\text{dom}(R) = \{ x \in A \mid \exists y \in B, \langle x, y\rangle \in R \}$.
- Il **codominio** è l'insieme di tutti gli oggetti $y \in B$ tali che $\langle x, y\rangle \in R$ per qualche $x \in A$.

  $\text{codom}(R) = \{ y \in B \mid \exists x \in A, \langle x, y\rangle \in R \}$.
- Il **campo** o **estensione** di R è $\text{dom}(R) \cup \text{codom}(R)$.

2.3.7 Relazioni n-arie

Il concetto di relazione può estendersi a tuple ordinate con **più di due** elementi.

Se gli elementi delle tuple appartengono allo stesso insieme A, allora una relazione n-aria è un sottoinsieme di A^n.

Esempi:

- $\{\langle x, x\rangle \mid x \in A\}$ è una relazione binaria su A
- $\{\langle x, y\rangle \mid x, y \in \mathbb{N}, x \leq y\}$ è la relazione d'ordine naturale su $\mathbb{N}$
- $\{\langle x, y, z\rangle \mid x, y, z \in \mathbb{R}, x^2 + y^2 = z^2\}$ è un'area geometrica

2.3.8 Operazioni su relazioni

Siano $R, S \subseteq A \times B$ due relazioni

- $R \cup S$ ha tutte le coppie che appartengono a R o a S
- $R \cap S$ ha tutte le coppie che appartengono ad entrambi R e S
- $\overline{R} = \{\langle x, y\rangle \mid \langle x, y\rangle \notin R\} \subseteq A \times B$ è il **complemento** di R
- $R^{-1} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R\} \subseteq A \times B$ è la **relazione inversa** di R

2.3.9 Proprietà delle relazioni

Siano $R, S \subseteq A \times B$ due relazioni

- Se $R \subseteq S$ allora $\overline{S} \subseteq \overline{R}$
- $\overline{(R \cap S)} = \overline{R} \cup \overline{S}$
- $\overline{(R \cup S)} = \overline{R} \cap \overline{S}$
- se $R \subseteq S$ allora $R^{-1} \subseteq S^{-1}$
- $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$
- $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$

2.3.9.1 Esempi

Siano $A = \{a, b\}, R = \{\langle a, b\rangle, \langle b, a\rangle\}, S = \{\langle a, b\rangle, \langle a, a\rangle\}$ $(R \subseteq A^2; S \subseteq A^2)$.

1. $R \cap S = \{\langle a, b\rangle\}$
2. $\overline{R \cup S} = \{\langle b, b\rangle\}$
3. $R^{-1} = R$
4. $S^{-1} \neq S$

2.3.10 Identità

Dato un insieme A, la relazione

$$I_A = \{\langle x, x\rangle \mid x \in A\}$$

dove ogni elemento è in relazione con se stesso è chiamata l'**identità** su A.

2.3.11 Proprietà delle relazioni binarie

Una relazione $R \subseteq A^2$ è

- **Riflessiva** se $\langle x, x\rangle \in R \; \forall x \in A \; (I_A \subseteq R)$
- **Simmetrica** se $\langle x, y\rangle \in R \implies \langle y, x\rangle \in R \; (R = R^{-1})$
- **Antisimmetrica** se $\langle x, y\rangle, \langle y, x\rangle \in R \implies x = y \, (R \cap R^{-1} \subseteq I_A)$
- **Antisimmetrica** (*def alternativa*) se $x \neq y \wedge \langle x, y\rangle \in R \implies \langle y, x\rangle \notin R \; (R \cap R^{-1} \subseteq I_A)$
- **Transitiva** se $\langle x, y\rangle, \langle y, z\rangle \in R \implies \langle x, z\rangle \in R$

2.4 Funzioni

Una classe di relazioni binarie di particolare importanza sono le **funzioni** (o **applicazioni**).

Una funzione è una relazione $R \subseteq A \times B$ tale che ad ogni $a \in A$ corrisponde **al più** un elemento $b \in B$.

Formalmente: se $\langle a, b\rangle, \langle a, c\rangle \in R$ allora $b = c$.

Notazione: $f : A \to B$

Se per ogni $a \in A$ esiste **esattamente un** $b \in B$ tale che $\langle a, b\rangle \in R$, allora f è una **funzione totale**.

Riformulazione: una relazione $f \subseteq A \times B$ è una funzione se per ogni $x \in \text{dom}(f)$ esiste un unico $y \in B$ tale che $\langle x, y\rangle \in f$. $f(x)$ denota tale elemento y.

Se $x \in \text{dom}(f)$, allora si dice che f è **definita** in x. Se $A = \text{dom}(f)$ allora f è una funzione **totale**.

2.4.1 Funzione iniettiva

Una funzione f è **iniettiva** se porta elementi distinti del dominio in elementi distinti del codominio (immagine).

> $f : A \to B$ è iniettiva sse per ogni $x, y \in A, x \neq y \implies f(x) \neq f(y)$.

2.4.2 Funzione suriettiva

Una funzione f è **suriettiva** quando ogni elemento di B è immagine di almeno un elemento di A ossia, quando $B = \text{codom}(f)$.

> $f : A \to B$ è suriettiva sse per ogni $y \in B$ esiste un $x \in A$ tale che $f(x) = y$.

2.4.3 Funzione biiettiva

Una funzione $f : A \to B$ è **biettiva** sse è iniettiva e suriettiva.

Attenzione: f può non essere totale.

- Ad ogni $x \in \text{dom}(f)$ corrisponde esattamente un $y \in B$
- Ad ogni $y \in B$ corrisponde esattamente un $x \in \text{dom}(f)$

2.4.4 Corrispondenza biunivoca

Una **corrispondenza biunivoca** tra A e B è una relazione binaria $R \subseteq A \times B$ tale che ad ogni elemento di A corrisponde uno ed un solo elemento di B e viceversa, ad ogni elemento di B corrisponde uno ed un solo elemento di A.

Tale R deve essere una funzione *totale*, *iniettiva* e *suriettiva*.

2.4.5 Formalizzazione

$f \subseteq A \times B$

$$\mathrm{dom}(f) = \{ x \in A \mid \exists y \in B.\langle x, y\rangle \in f\}$$
$$\mathrm{codom}(f) = \{ y \in A \mid \exists x \in B.\langle x, y\rangle \in f\}$$

Funzione (parziale)

$$\forall a \in A.\forall x, y \in B.(\langle a, x\rangle \in f \land \langle a, y\rangle \in f) \implies x = y$$

Funzione totale

$$\forall a \in A.\exists!\, x \in B.\langle a, x\rangle \in f$$

Funzione iniettiva

$$\forall a \in A.\forall x, y \in B.(\langle a, x\rangle \in f \land \langle a, y\rangle \in f) \implies x = y \land$$
$$\forall a, b \in A.\forall x \in B.(\langle a, x\rangle \in f \land \langle b, x\rangle \in f) \implies a = b$$

Funzione suriettiva

$$\forall a \in A.\forall x, y \in B.(\langle a, x\rangle \in f \land \langle a, y\rangle \in f) \implies x = y \land$$
$$\forall x \in B.\exists a \in A.\langle a, x\rangle \in f$$

Funzione biiettiva

$$\forall a \in A.\forall x, y \in B.(\langle a, x\rangle \in f \land \langle a, y\rangle \in f) \implies x = y \land$$
$$\forall a, b \in A.\forall x \in B.(\langle a, x\rangle \in f \land \langle b, x\rangle \in f) \implies a = b \land$$
$$\forall x \in B.\exists a \in A.\langle a, x\rangle \in f$$

2.4.6 Punto fisso

Sia A un insieme e $f : A \to A$ una funzione.

Un **punto fisso** di f è un elemento di A che coincide con la sua immagine

$$x = f(x)$$

2.4.7 Operazioni

Sia A un insieme.

Un'**operazione** (n-aria) su A è una funzione $A^n \to A$.

L'operazione è totale sse la funzione è totale.

2.4.8 Immagine inversa

Sia $f : A \to B$ una funzione e $y \in B$ l'**immagine inversa** di f in y è

$$f^{-1} : B \to \mathscr{P}(A)$$
$$f^{-1}(y) = \{ x \in A \mid f(x) = y \}$$

Nota: f è iniettiva sse per ogni $y \in B$, $f^{-1}(y)$ ha al più un elemento.

2.4.9 Funzione inversa

Una funzione $f : A \to B$ è **invertibile** se esiste una funzione $g : B \to A$ tale che per ogni $x \in A$ e ogni $y \in B$o

$$g(f(x)) = x$$
$$f(g(y)) = y$$

In questo caso, g è l'**inverso** di f e si rappresenta come f^{-1}.

Una funzione f è invertibile sse è iniettiva. f_{-1} è totale sse f è suriettiva.

2.4.10 Composizione di Funzioni

La **composizione** di due funzioni si riferisce all'applicazione di una funzione al risultato di un'altra.

Siano $f : A \to B$ e $g : B \to C$ due funzioni. La funzione composta $g \circ f : A \to C$ è definita per ogni $x \in A$ da

$$(g \circ f)(x) = g(f(x))$$

$(g \circ f)(x)$ è definita sse $f(x)$ e $g(f(x))$ sono definite.

Se $f : A \to B$ e $g : C \to D$ sono due funzioni, allora la composizione $g \circ f$ è solo definibile se $\text{codom}(f) \subseteq C$.

Le proprietà della composizione:

- **Associativa:** $f \circ (g \circ h) = (f \circ g) \circ h$
- Se f e g sono entrambe iniettive, allora $f \circ g$ è **iniettiva**
- Se f e g sono entrambe suriettive, allora $f \circ g$ è **suriettiva**
- Se f e g sono entrambe invertibili, allora $f \circ g$ è **invertibile** ($(g \circ f)^{-1} = f^{-1} \circ g^{-1}$)

2.4.11 Funzione caratteristica

I sottoinsiemi di un insieme A si possono anche rappresentare tramite una funzione detta **caratteristica**.

La funzione caratteristica di un insieme $S \subseteq A$ è la funzione $f_S : A \to \{0, 1\}$ dove

$$f_S(x) = \begin{cases} 0 & x \notin S \\ 1 & x \in S \end{cases}$$

Per ogni $x \in A$

- $f_{S \cap T}(x) = f_S(x) \cdot f_T(x)$
- $f_{S \cup T}(x) = f_S(x) + f_T(x) - f_S(x) \cdot f_T(x)$
- $f_{S \triangle T}(x) = f_S(x) + f_T(x) - 2 \cdot f_S(x) \cdot f_T(x)$

2.4.12 Multinsiemi

Un **multinsieme** è una variante di un insieme dove gli elementi si possono ripetere

$$\{ a, a, b, c, c, c \} \neq \{ a, b, c \}$$

Formalmente un multinsieme è una funzione da un insieme a $\mathbb{N}$

$$f : A \to \mathbb{N}$$

che esprime quante volte si ripete ogni elemento nel multinsieme ($A = \{ a, b, c, d \}$)

$$\{ \langle a, 2 \rangle, \langle b, 1 \rangle, \langle c, 3 \rangle, \langle d, 0 \rangle \}$$

2.5 Cardinalità

I **numeri cardinali** si utilizzano per misurare gli insiemi (indicare la loro *grandezza*). Se un insieme è **finito**, la sua cardinalità è un numero naturale (il numero di elementi). Con i numeri cardinali, possiamo anche misurare e classificare insiemi **infiniti**.

2.5.1 Cardinalità tramite funzioni

Georg Cantor utilizzò le proprietà delle funzioni per paragonare la cardinalità degli insiemi.

Sia f una funzione $f : A \to B$

- Se f è *suriettiva* allora B non è "più grande" di A
- Se f è *totale* e *iniettiva* allora A non è "più grande" di B

Due insiemi sono **equipotenti** (hanno la stessa cardinalità) sse esiste una funzione **biunivoca** fra di loro.

$$A \sim B$$

2.5.2 Cardinalità finite

Se A ha n elementi, allora $A \sim \{1, \ldots, n\}$. In questo caso si dice che A è **finito** e ha **cardinalità** (o potenza) n.

Utilizziamo la notazione

$$|A| = n$$

I numeri naturali si utilizzano come cardinali finiti.

Se $|A| = n$ allora $|\mathscr{P}(A)| = 2^n$.

2.5.3 Numerabili

Basati su questa definizione, chiamiamo **numerabili** tutti gli insiemi che hanno la cardinalità di $\mathbb{N}$. I suoi elementi possono essere posti in corrispondenza biunivoca con i naturali.

$$A \sim \mathbb{N} \sim \mathbb{N}^+$$

La cardinalità di $\mathbb{N}$ è chiamata $\aleph_0$.

$$|\mathbb{N}| = \aleph_0$$

$\aleph_0$ è il più piccolo dei numeri cardinali **transfiniti** (i cardinali per misurare insiemi infiniti). Ovviamente $\aleph_0$ non è un numero naturale.

I seguenti insiemi sono numerabili:

- L'insieme dei numeri pari
- L'insieme dei numeri primi
- L'insieme dei numeri interi $\mathbb{Z}$

$$f : \mathbb{N} \to \mathbb{Z}$$

$$f(x) = \begin{cases} -\frac{x}{2} & \text{se } x \text{ pari} \\ \left\lceil \frac{x}{2} \right\rceil & \text{se } x \text{ dispari} \end{cases}$$

- Il prodotto cartesiano $\mathbb{N} \times \mathbb{N}$
- I numeri razionali $\mathbb{Q}$ ($\subset \mathbb{N} \times \mathbb{N}$)

2.5.4 Il continuo

$$[0, 1] = \{ x \in \mathbb{R} \mid 0 \leq x \leq 1 \} \sim \mathscr{P}(\mathbb{N})$$

Denotiamo per convenzione $|\mathscr{P}(\mathbb{N})| = 2^{\aleph_0}$. Allora $|\mathbb{R}| \geq 2^{\aleph_0}$.

Cantor dimostro che $\aleph_0 < 2^{\aleph_0}$ (in realtà che $|A| < |\mathscr{P}(A)|$). Dunque $\mathbb{R}$ non è numerabile.

2.5.4.1 Teorema di Cantor

$$\aleph_0 < 2^{\aleph_0}$$

Dobbiamo dimostrare che *non esiste* una funzione biunivoca $f : \mathbb{N} \to \mathscr{P}(\mathbb{N})$.

Supponiamo che esiste una tale funzione f. Definiamo

$$Z = \{ z \in \mathbb{N} \mid n \notin f(n) \} \subseteq \mathbb{N}$$

Siccome f è biunivoca (quindi suriettiva), esiste $k \in \mathbb{N}$ tale che $f(k) = Z$.

Domanda: $k \in Z$?

Se $k \in Z$, allora per definizione $k \notin f(k) = Z$. Se $k \notin Z$, allora $k \notin f(x)$ e quindi per definizione $k \in Z$.

Conclusione: la funzione f non può esistere.

2.5.5 Gerarchia transfinita

Cantor definì la gerarchia dei numeri transfiniti

$$\aleph_0 < \aleph_1 < \aleph_2 < \dots$$

L'**ipotesi del continuo** dice che $\aleph_1 = 2^{\aleph_0}$. Non ci sono insiemi di cardinalità intermedia fra $\mathbb{N}$ e $\mathbb{R}$.

3 Strutture relazionali, Grafi e Ordinamenti

3.1 Rappresentazioni

Le relazioni possono essere rappresentate da diverse forme:

- **Rappresentazione per elencazione:** descrivere l'insieme di coppie ordinate ($R = \{\langle 1, 2\rangle, \langle 2, 4\rangle, \langle 3, 6\rangle\}$)
- **Rappresentazione sagittale:** collegare con delle frecce gli elementi che verificano la relazione
- **Rappresentazione tramite diagramma cartesiano:** se S e T sono sottoinsiemi di $\mathbb{R}$, rappresentare le coppie come coordinate sul piano cartesiano
- **Rappresentazione tramite tabella:** una matrice booleana con per colonne gli elementi dell'insieme di arrivo e per righe l'insieme di partenza.

3.1.1 Relazioni in un insieme

Una relazione $R \subseteq S \times S$ è detta **relazione in S**. In una relazione in S, la rappresentazione sagittale collassa in un **grafo**. Usiamo lo stesso insieme per l'origine e la destinazione di ogni freccia. Formalmente un grafo è costituito da **nodi** collegati fra loro da frecce (o **spigoli**). Se $\langle x, y\rangle \in \mathbb{R}$, disegnamo uno spigolo da x a y.

Le proprietà di una relazione sono (*again*):

- **Riflessiva** se: $\langle x, x\rangle \in R \;\forall x \in S$ (*ogni nodo ha un cappio*)

- **Irriflessiva** se: $\langle x, x \rangle \notin R \; \forall x \in S$ (*nessun nodo ha un cappio*)
- **Simmetrica** se: $\langle x, y \rangle \in R \implies \langle y, x \rangle \in R$ (*ogni spigolo ha il suo inverso*)
- **Asimmetrica** se: $\langle x, y \rangle \in R \implies \langle y, x \rangle \notin R$ (*nessuno spigolo ha il suo inverso e nessun nodo ha un cappio*)
- **Antisimmetrica** se: $\langle x, y \rangle \in R \land \langle y, x \rangle \in R \implies x = y$ (*nessuno spigolo ha il suo inverso (escluso il cappio)*)
- **Transitiva** se: $\langle x, y \rangle \in R \land \langle y, z \rangle \in R \implies \langle x, z \rangle \in R$

Una relazione $R \subseteq S \times S$ in S è

- **Connessa** se ogni due elementi sono collegati. $\forall x, y \in S$x se $x \neq y$ allora $\langle x, y \rangle \in R$ oppure $\langle y, x \rangle \in R$
- **Relazione di equivalenza** se è riflessiva, transitiva e simmetrica

La relazione vuota $\emptyset \subseteq S \times S$ è irriflessiva, simmetrica, asimmetrica, antisimmetrica e transitiva. L'identità I_S è riflessiva, simmetrica e transitiva (è una relazione di equivalenza).

3.1.2 Riflessività ed operazioni

Siano R ed R' due relazioni su S

1. Se R è riflessiva, R^{-1} è riflessiva (*stesso per irriflessibilità*)
2. R è riflessiva sse $\overline{R}$ è irriflessiva
3. Se R ed R' sono riflessive, allora anche $R \cup R'$ e $R \cap R'$ sono riflessive (*stesso per irriflessibilità*)

3.1.3 Simmetria ed operazioni

Siano R ed R' due relazioni su S

1. R è simmetrica sse $R = R^{-1}$
2. Se R è simmetrica, allora R^{-1} e $\overline{R}$ sono simmetriche
3. R è antisimmetrica sse $R \cap R^{-1} \subseteq I_S$
4. R è asimmetrica sse $R \cap R^{-1} = \emptyset$
5. Se R ed R' sono simmetriche, allora anche $R \cup R'$ e $R \cap R'$ sono simmetriche

3.1.4 Transitività ed operazioni

Se R ed R' sono transitive allora $R \cap R'$ è transitiva. $R \cup R'$ non è necessariamente transitiva.

3.1.5 Matrici booleane

Una matrice booleana è una matrice a valori $\{0, 1\}$. La matrice booleana associata a $R \subseteq S \times T$ si denota M_R. Se $|S| = n$ e $|T| = m$, M_R ha n righe e m colonne.

La riga i corrisponde all'elemento $s_i \in S$, la colonna j corrisponde all'elemento $t_j \in T$ ed è tale che

$$m_{ij} = \begin{cases} 1 & \langle s_i, t_j \rangle \in R \\ 0 & \text{altrimenti} \end{cases}$$

3.1.5.1 Proprietà di una matrice booleana

Se R è una relazione su S, M_R ha le stesse proprietà della visualizzazione tabulare.

- R è **riflessiva** sse M_R ha tutti 1 sulla diagonale principale
- R è **irriflessiva** sse M_R ha tutti 0 sulla diagonale principale
- R è **simmetrica** sse M_R è simmetrica
- R è **asimmetrica** sse per ogni i, j, se $m_{ij} = 1$, allora $m_{ji} = 0$
- R è **antisimmetrica** sse per ogni $i \neq j$, se $m_{ij} = 1$, allora $m_{ji} = 0$
- $M_{R^{-1}}$ è la trasposta di M_R
- $M_{\overline{R}}$ si ottiene scambiando 0 e 1 in M_R

$$R = \{ \langle 0, 0 \rangle, \langle 1, 2 \rangle \}$$

$$M_R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \longrightarrow M_{R^{-1}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

3.1.6 Operazioni su matrici booleane

Se M e N sono due matrici booleane di dimensioni $n \times m$, $M \sqcup N$ (il **join** di M e N) è la matrice booleana L didimensionee $n \times m$ i cui elementi sono

$$\ell_{ij} = \begin{cases} 1 & m_{ij} = 1 \vee n_{ij} = 1 \\ 0 & \text{altrimenti} \end{cases}$$

$M \sqcap N$ (il **meet** di M e N) è la matrice booleana L di dimensione $n \times m$ i cui elementi sono

$$\ell_{ij} = \begin{cases} 1 & m_{ij} = 1 \wedge n_{ij} = 1 \\ 0 & \text{altrimenti} \end{cases}$$

$\sqcup$ e $\sqcap$ sono commutative, associative e distributive fra di loro.

3.1.7 Prodotto booleano

Siano M e N matrici booleane di dimensioni $n \times m$ e $m \times p$ rispettivamente. Il loro **prodotto booleano** è la matrice $L = M \odot N$ di dimensioni $n \times p$ dove

$$\ell_{ij} = \begin{cases} 1 & \exists k, 1 \le k \le m \text{ t.c. } m_{ik} = 1 \wedge n_{kj} = 1 \\ 0 & \text{altrimenti} \end{cases}$$

Esempio:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \odot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Questa operazione è associativa ma non commutativa.

YT Link con spiegazione[1].

[1] https://youtu.be/BjTeDlpj-ts?si=snvhzdZvQByBGinl

3.2 Composizione di relazioni

Dati $R_1 \subseteq S \times T, R_2 \subseteq T \times Q$:

$$R_2 \circ R_1 = \{ \langle x, y \rangle \in S \times Q \mid \exists \in T. \langle x, z \rangle \in R_1, \langle z, y \rangle \in R_2 \}$$

$R_2 \circ R_1$ è la **composizione** di R_1 e R_2.

La composizione si può calcolare tramite il prodotto di matrici booleane.

$$M_{R_2 \circ R_1} = M_{R_1} \odot M_{R_2}$$

3.3 Relazioni di Equivalenza

Una **relazione di equivalenza** ci aiuta a creare blocchi di elementi che hanno *qualcosa* in comune. Sono relazioni che si comportano "come l'uguaglianza" tra oggetti. Dal punto di vista di una proprietà data, **non** esistono differenze tra due elementi in una relazione di equivalenza.

Def: una relazione riflessiva, simmetrica e transitiva è detta **relazione di equivalenza**.

Esempio:

- Appartenere alla stessa classe
- Essere nati nello stesso anno
- Essere parallele nell'insieme delle rette
- …

Se $f : A \to B$ è una funzione totale, allora la relazione

$$R := \{ \langle x, y \rangle \in A \times A \mid f(x) = f(y) \}$$

è una relazione di equivalenza.

La rappresentazione sagittale di una relazione di equivalenza consiste di diversi grafi totalmente collegati.

3.3.1 Partizioni e classi di equivalenza

Dividendo S in gruppi i cui elementi sono "uguali", possiamo studiare insiemi grandi osservando soltanto pochi elementi. Questi gruppi sono chiamati **classi di equivalenza**.

Sia S un insieme. Una partizione di S è una famiglia di insiemi $\mathscr{P} = \{T_1, \dots, T_n\}, T_i \subseteq S, 1 \leq i \leq n$ tali che:

- $T_i \neq \emptyset$ per ogni i, $1 \leq i \leq n$
- $T_i \cap T_j \neq \emptyset$ per ogni i, j, $1 \leq i \leq j \leq n$
- $\cup\mathscr{P} = S$

Se R è una **relazione di equivalenza** su S allora $T \neq \emptyset \subseteq S$ è una classe di equivalenza se per ogni $x \in S$:

$$x \in T \iff \{y \in S \mid \langle x, y\rangle \in R\} = T$$

Cioè, x è in relazione con tutti e soltanto quegli elementi di T.

Sia S un insieme e R una relazione di equivalenza su S. Ogni elemento $x \in S$ definisce una classe di equivalenza

$$[x]_R = \{y \in S \mid \langle x, y\rangle \in R\}$$

La famiglia di insiemi $\{[x]_R \mid x \in S\}$ (gli elementi sono le classi di equivalenza di S) è chiamato l'**insieme quoziente** di S rispetto a R (indicato con S/R). L'insieme quoziente è una partizione di S.

Esempio: Sia $n \in \mathbb{N}$. La relazione $\simeq_n \subseteq \mathbb{N} \times \mathbb{N}$ definita come

$$x \simeq_n y \iff x \equiv y \mod n \leftrightarrow (\text{ossia } (x \mod n) = (y \mod n))$$

è una relazione di equivalenza.

Per $n = 4$, $\simeq_4$ definisce 4 classi di equivalenza.

$$[x] = \{ x + 4k \mid k \in \mathbb{N} \}$$
$$[0] = \{ 0, 4, 8, 12, \dots \}$$
$$[1] = \{ 1, 5, 9, 13, \dots \}$$
$$[2] = \{ 2, 6, 10, 14, \dots \}$$
$$[3] = \{ 3, 7, 11, 15, \dots \}$$

L'insieme quoziente $\mathbb{N}/\simeq_4 = \{ [0], [1], [2], [3] \}$ è spesso indicato con $\mathbb{N}_4$.

3.4 Grafi

Un grafo è definito da

- Un insieme di **nodi** (chiamati anche *vertici*)
- Collegamenti tra vertici che possono essere:
  - Orientati (**archi**)
  - Non orientati (**spigoli**)
- (eventualmente) Dati associati ai nodi e collegamenti (**etichette**)

I grafi possono rappresentare **relazioni binarie**.

3.4.1 Gradi

Un arco che va da v a w è **uscente** da v ed entrante in w. Il numero di archi uscenti dal nodo v è il **grado di uscita** di v. Il numero di archi entranti in v è il **grado in ingresso** di v.

Un nodo è chiamato:

- **Sorgente** se non ha archi entranti (*grado di entrata* 0)
- **Pozzo** se non ha archi uscenti (*grado di uscita* 0)
- **Isolato** se non ha archi né uscenti né entranti

I nodi v e w sono **adiacenti** se c'è un arco tra v e w (in qualunque direzione). Questo arco è **incidente** su v e w. Il grado di v è il numero di nodi adiacenti a v.

3.4.2 Cammino

Un **cammino** è una sequenza **finita** di nodi

$$\langle v_1, v_2, \dots, v_n \rangle$$

tali che per ogni i, $1 \leq i < n$, esiste un arco uscente da v_i ed entrante in v_{i+1}. Questo cammino va da v a w se $v_1 = v$ e $v_n = w$.

3.4.3 Semicammino

Un **semicammino** è una sequenza finita di nodi

$$\langle v_1, v_2, \dots, v_n \rangle$$

tali che per ogni i, $1 \leq i < n$, esiste un arco che collega v_i e v_{i+1} in **direzione arbitraria**.

La **lunghezza** di un (semi)cammino è il numero di archi che lo compongono $(n - 1)$.

Un (semi)cammino è **semplice** se tutti i nodi nella sequenza sono diversi (anche se $v_1 = v_n$).

Un grafo è **connesso** se esiste sempre un semicammino tra due nodi qualsiasi.

3.4.4 Ciclo

Un **ciclo** intorno al nodo v è un cammino tra v e v. Un **semiciclo** intorno al nodo v è un semicammino tra v e v. Un **cappio** intorno a v è un ciclo di lunghezza 1.

3.4.5 Distanza

La **distanza** da v a w è la lunghezza del cammino *più corto* tra v e w.

- La distanza da v a v è sempre 0
- Se non c'è nessun cammino da v a w allora la distanza è infinita (∞)

In un grafo ordinato, la distanza da v a w **non** è sempre uguale alla distanza da w a v.

3.4.6 Trovare le distanze: Algoritmo

Ricerca in **ampiezza** delle distanze da v ad ogni nodo.

Inizializzazione:

- Segnare v come **visitato** con distanza $d(v) = 0$
- Segnare altri nodi come **non visitato**

Ciclo:

- Trovare un nodo w **visitato** con distanza *minima* $d(w) = n$
- Segnare w come **esplorato**
- Per ogni nodo w' incidente da w: se w' è **non visitato**, segnare w' come **visitato** e $d(w') = n + 1$

Finalizzazione: ad ogni nodo w **non visitato** assegnare $d(w) = \infty$.

3.4.7 Definizione formale di grafo

Un **grafo orientato** è una coppia $G = (V, E)$ dove

- V è un insieme di **nodi**
- $E \subseteq V \times V$ è una relazione binaria in V (**archi**)

Un **grafo non orientato** è un grafo orientato dove E è una relazione **simmetrica**. In questo caso gli archi sono rappresentati come **coppie non ordinate** (v, w) $((v, w) = (w, v))$. Graficamente togliamo le frecce (l'ordine) agli archi.

3.4.8 Sottografo

Il grafo $G_1 = (V_1, E_1)$ è un **sottografo** di $G_2 = (V_2, E_2)$ sse $V_1 \subseteq V_2$ e $E_1 \subseteq E_2$. Un sottografo si ottiene togliendo nodi e/o archi dal grafo.

Sia $G = (V, E)$ un grafo. Il sottografo **indotto** da $V' \subseteq V$ è il grafo che ha soltanto archi adiacenti agli elementi di V'. Formalmente è il grafo $G = (V', E')$ dove

$$E' = \{ \langle v, w \rangle \in E \mid v, w \in V' \}$$

3.4.9 Grafo aciclico orientato (DAG)

Un grafo orientato senza cicli si chiama **grafo aciclico orientato**.

In un DAG non esiste nessun cammino da un nodo a se stesso

3.4.10 Grafi etichettati

Un **grafo etichettato** è una tripla $G = (V, E, \ell)$ dove

- (V, E) è un grafo
- $\ell : E \to L$ è una funzione totale che associa ad ogni arco $e \in E$ un'**etichetta** da un insieme L

Diamo un'etichetta ad ogni arco del grafo.

Un grafo etichettato può rappresentare una **relazione ternaria** (e viceversa).

I nomi e le etichette sono spesso irrilevanti.

3.4.11 Matrice di adiacenza

La **matrice di adiacenza** di un grafo $G = (V, E)$ è la matrice booleana della relazione E.

La matrice di adiacenza di grafi non orientati è **sempre simmetrica**.

3.4.12 Grafo completo

Un **grafo completo** collega ogni nodo con tutti gli altri nodi (ma non con se stesso).

La sua matrice di adiacenza ha 0 su tutta la diagonale ed 1 sulle altre posizioni.

3.4.13 Connettività

Ricordiamo che $G = (V, E)$ è **connesso** se per ogni $v, w \in V$ esiste un **semicammino** da v a w. G è **fortemente connesso** se per ogni due nodi $v, w \in V$ esiste un **cammino** da v a w.

In un grafo fortemente connesso:

- Esiste sempre un ciclo che visita **ogni** nodo (non necessariamente semplice)
- Non ci sono né sorgenti né pozzi

3.4.14 Isomorfismi tra grafi

Due grafi $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ sono **isomorfi** se esiste una funzione biunivoca $f : V_1 \to V_2$ tale che

$$\langle v, w \rangle \in E_1 \iff \langle f(v), f(w) \rangle \in E_2$$

L'isomorfismo f mantiene la struttura del grafo G_1, ma sostituisce i nomi dei vertici con quelli di G_2. Due grafi isomorfi sono in realtà lo **stesso grafo** con i nodi rinominati.

3.4.15 Chiusure

3.4.15.1 Chiusura riflessiva

La **chiusura riflessiva** di $R \subseteq S^2$ è la più piccola relazione riflessiva R^{refl} su S che contiene R.

$$R \subseteq R^{\text{refl}} = R \cup I_S$$

3.4.15.2 Chiusura transitiva

La **chiusura transitiva** di $R \subseteq S^2$ è la più piccola relazione transitiva R^{trans} su S che contiene R.

$$R \subseteq R^{\text{trans}} \subseteq S$$

3.4.15.3 Chiusura simmetrica

La **chiusura simmetrica** di $R \subseteq S^2$ è la più piccola relazione transitiva R^{simm} su S che contiene R.

$$R \subseteq R^{\text{trans}} = R \cup R^{-1}$$

3.5 Alberi

Un'**albero** è un DAG *connesso* tale che

- Esiste *esattamente un* nodo sorgente (*radice dell'albero*)
- Ogni nodo diverso dalla radice ha **un solo** arco entrante

I nodi pozzo di un albero sono chiamati **foglie** o **nodi esterni**. Tutti gli altri nodi sono chiamati **interni**. Per analogia con gli **alberi genealogici**, le relazioni tra i nodi usano nomi come *padre*, *figlio*, *discendente*, …

3.5.1 Proprietà

Il grado di **ingresso** di un nodo è:

- 1 se non è la radice
- 0 se è la radice

Il grado di **uscita** di un nodo non ha restrizioni.

Per ogni nodo v che non è la radice, esiste *esattamente un* cammino dalla radice a v.

Un albero non può essere mai vuoto (la radice esiste sempre).

Se un albero è finito, allora esiste *almeno* una foglia (che può essere anche la radice).

I nodi **intermedi** sono contemporaneamente padre e figlio.

3.5.2 Rappresentazione gerarchica

Gli alberi spesso rappresentano **strutture gerarchiche**. In questo caso, l'ordine è **implicito** (gli archi si disegnano **senza frecce**).

3.5.3 Cammini in un albero

In un albero c'è *esattamente un* **cammino** dalla radice a qualunque nodo v diverso dalla radice. Ogni nodo w in questo cammino è un **ascendente** di v (oppure *avo*) e v è un **discendente** di w (la radice è l'unico nodo senza discendenti). Se il cammino da w a v ha lunghezza 1, allora w è il *padre* di v e v è un figlio di w.

3.5.4 Profondità

La **profondità** di un nodo v è la lunghezza del cammino dalla radice a v.

L'**altezza** di un albero è la profondità massima dei suoi nodi.

3.5.5 Alberi binari

Un **albero binario** è un albero dove ogni nodo ha al massimo due figli. I figli di un nodo in un albero binario sono **ordinati** (*figlio sinistro* e *figlio destro*).

Un albero binario ha al massimo 2^p nodi di profondità p. Un albero di altezza n ha al più $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ nodi.

Un albero binario è una **struttura ricorsiva** composta da

- Un nodo (*radice*)
- Un albero binario sinistro (*eventualmente vuoto*)
- Un albero binario destro (*eventualmente vuoto*)

Possiamo rappresentare un albero binario sia

- Come una collezione di nodi, dove la radice è segnalata, e ogni nodo ha due puntatori (alle radici degli alberi sinistro e destro)
- Come una tabella con $2^{n+1} - 1$ righe, dove n è l'altezza dell'albero

Un albero binario è **pieno** se ogni nodo interno ha due figli.

Un albero binario è **completo** se

- Ha altezza n

- Ad ogni profondità $i, 0 \leq i < n$ ci sono 2^i nodi
- L'ultimo livello è riempito da sinistra a destra

In rappresentazione tabulare, i nodi vuoti sono soltando sulle ultime righe.

Un albero binario è **bilanciato** se per ogni nodo v la differenza fra

- Il numero di nodi nell'albero sinistro di v
- Il numero di nodi nell'albero destro di v

è *al massimo 1*.

3.5.5.1 Albero binario di ricerca

Un **albero di ricerca** è un albero binario $G = (V, E)$ tale che per ogni nodo z:

- $z \in \mathbb{Z}$
- Ogni nodo dell'albero sinistro di z è minore di z
- Ogni nodo dell'albero destro di z è maggiore di z

Essi sono utili per rappresentare liste ordinate dinamiche.

3.5.5.2 Attraversamento di un albero binario

Un **attraversamento** è un processo che visita tutti i nodi di un albero. Solitamente in un ordine particolare. Un attraversamento che elenca ogni nodo *esattamente una volta* è un'**enumerazione** (dei nodi).

Distinguiamo fra due tipi di attraversamento:

- In **profondità** esplora ogni ramo dell'albero fino in fondo (*figli prima dei fratelli*)
- In **ampiezza** esplora prima i nodi più vicini alla radice (*fratelli prima dei figli*)

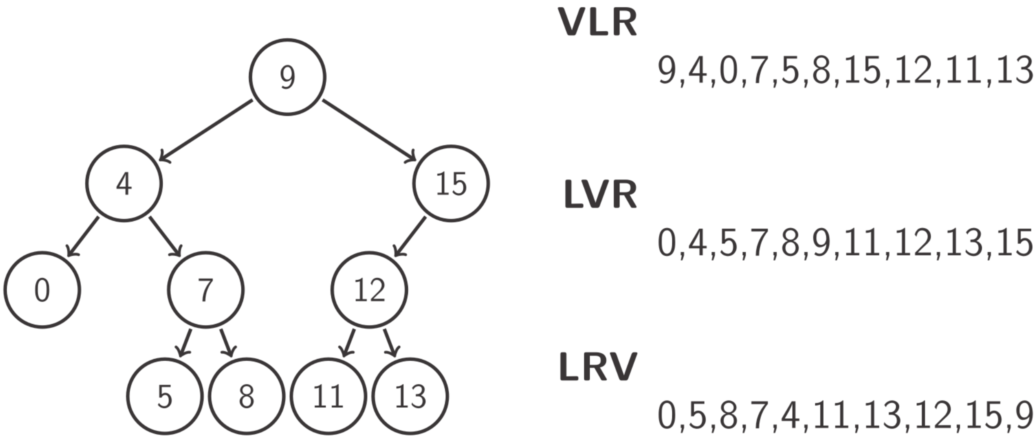
Ci sono tre tipi diversi di ordini in profondità, basati su *quando* enumeriamo un elemento. Si usa la notazione:

- **L** per sinistra
- **R** per destra
- **V** per enumerazione (*visit*)

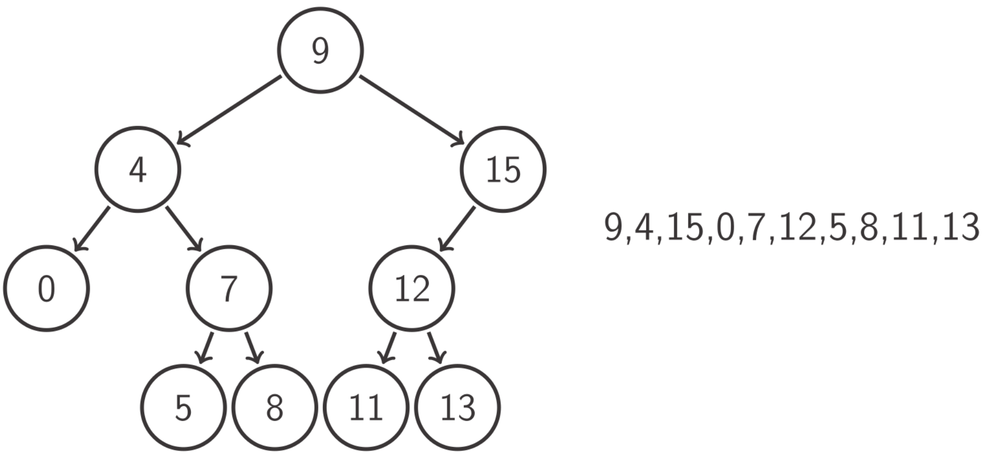
I tre ordini di enumerazione in profondità sono:

- In **preordine**: si visita un nodo prima di visitare i figli (*VLR*)
- In **ordine**: si visita l'albero sinistro, poi il nodo, poi l'albero destro (*LVR*)
- In **postordine**: si visitano prima i figli e poi il nodo (*LRV*)

Viene implementata come una **pila** che contiene gli elementi da esplorare (*LIFO*).



L'enumerazione in ampiezza visita *tutti* i nodi ad una profondità prima di esplorare altri livelli dell'albero. Viene implementata tramite una **coda** di elementi da esplorare (*FIFO*).



3.5.5.3 Numero di foglie in un albero

Un albero finito ha sempre al meno una foglia. Per *massimizzare* il numero di foglie dobbiamo avere un albero **pieno**. Un albero pieno con n nodi interni ha $n + 1$ **foglie** (*dimostrazione per induzione*). Il numero di **puntatori nulli** in un albero binario con n nodi è $n + 1$. Basta sostituire i puntatori vuoti per foglie speciali, formando un albero pieno.

> Per la dimostrazione per induzione consultare le slide #view-slide

3.6 Ordinamenti

Molto spesso, gli elementi in un insieme hanno una struttura d'**ordine**. Per esempio, $\mathbb{N}$ e $\mathbb{R}$. Anche se ha volte non è possibile paragonare tutti gli oggetti (è più grande $\langle 0, 1\rangle$ o $\langle 1, 0\rangle$?).

Un'**ordinamento** è un tipo particolare di **relazione** fra elementi.

Una relazione R su un insieme S è un:

- **Preordine** sse R è riflessiva e transitiva
- **Ordine parziale** sse R è un preordine antisimmetrico (*riflessiva, antisimm. e transitiva*)
- **Ordine stretto** sse R è irriflessiva e transitiva (e quindi anche asimmetrica)

Un ordine parziale si rappresenta con $\leq$; uno stretto con $<$. La coppia $(S, \leq)$ si chiama **insieme parzialmente ordinato** (*poset*).

Un **ordine totale** è un ordine parziale *fortemente connesso*:

$$\forall x, y \in S \qquad x \leq y \lor y \leq x$$

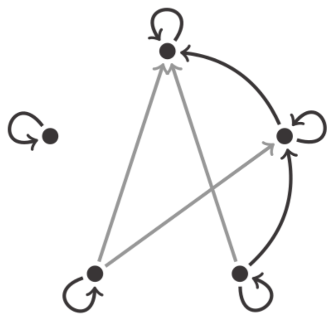
Un **ordine totale stretto** è un ordine stretto *connesso*:

$$\forall x, y \in S \text{t.c. } x \neq y \qquad x < y \lor y < x$$

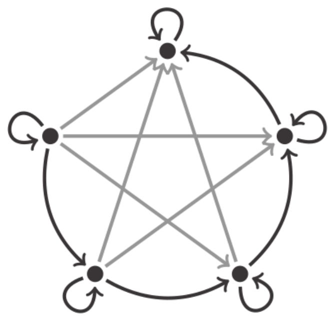
Ordini totali stretti e non stretti sono molto vicini:

- Se Ro è un ordine totale (non stretto), allora $R \setminus I_S$ è un ordine totale stretto
- Se R è un ordine totale stretto, allora $R \cup I_S$ è un ordine totale

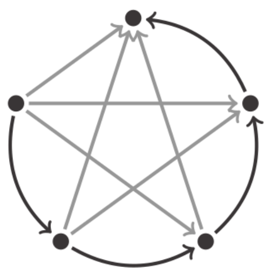
Uno è riflessivo, l'altro irriflessivo.



ordine parziale



ordine totale



ordine totale stretto

3.6.1 Tricotomia

In un **ordine totale stretto** R per ogni $x, y \in S$ si soddisfa *esattamente una* fra

1. $x = y$
2. $\langle x, y \rangle \in R$
3. $\langle y, x \rangle \in R$

3.6.2 Prodotto di ordinamenti

Siano $(S, \leq_S)$ e $(T, \leq_T)$ due *poset*. Definiamo la relazione $\leq_{S\times T}$ su $S \times T$ come

$$\langle s, t \rangle \leq_{S\times T} \langle s', t' \rangle \iff s \leq_S s' \ , \ t \leq_T t'$$

$(S \times T, \leq_{S\times T})$ è anche un *poset*.

3.6.3 Ordinamento lessicografico

L'ordine **lessicografico** paragona tuple di elementi posizione per posizione. La relazione $\leq_{\text{lex}}$ su $S \times T$ è definita da

$$\langle s, t \rangle \leq_{\text{lex}} \langle s', t' \rangle \iff \text{(i) } s <_S s' \text{ oppure (ii) } s = s', t \leq_T t'$$

$(S \times T, \leq_{\text{lex}})$ è anche un *poset* e preserva gli ordini totali.

Generalizza l'ordine alfabetico usuale e si può estendere a tuple di lunghezza arbitraria.

3.6.4 Copertura

In un poset $(S, \leq)$, una **copertura** di $x \in S$ è un elemento **minimo più grande** di x.

$y \in S$ è una copertura di $x \in S$ sse

- $x \leq y, x \neq y$
- $\nexists z, x \neq z \neq y$ tale che $x \leq z, z \leq y$

3.6.5 Elementi estremali

In un poset $(S, \leq)$, un elemento $s \in S$ è

- **Minimale** se non esiste un elemento $s' \neq s$ tale che $s' \leq s$
- **Massimale** se non esiste un elemento $s' \neq s$ tale che $s \leq s'$

Un poset può avere *nessuno*, *uno* o *tanti* elementi minimali e massimali.

3.6.6 Minoranti e maggioranti

Dato un poset $(S, \leq)$ e un insieme $X \subseteq S$, un elemento $s \in S$ è

- **Minorante** di X sse $s \leq x$ per ogni $x \in X$
- **Massimo minorante** di X ($\sqcap X$) sse $s' \leq s$ per ogni minorante s' di X e se s è un minorante

- **Maggiorante** di X sse $x \leq s$ per ogni $x \in X$
- **Minimo maggiorante** di X ($\sqcup X$) sse $s \leq s'$ per ogni maggiorante s' di X e se s è un maggiorante
- **Minimo** di X sse $s = \sqcap X \in X$
- **Massimo** di X sse $s = \sqcup X \in X$

3.6.7 Proprietà

Ogni $X \subseteq S$ ha **al più** un massimo minorante e un minimo maggiorante.

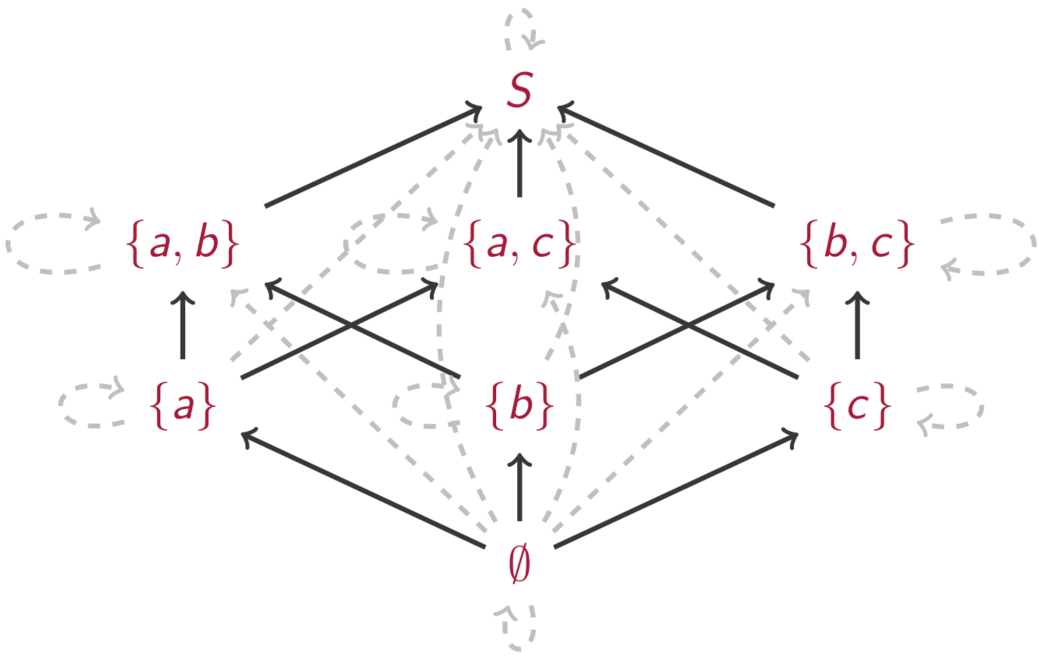
Se ogni $X \subseteq S$ ha un minimo, allora $(S, \leq)$ è un insieme **ben ordinato** (o ben fondato).

Se esiste, $\sqcap S$ è il minimo di S, denotato da $\underline{0}$. Se esiste, $\sqcup S$ è il massimo di S, denotato da $\underline{1}$.

3.6.8 Diagramma di Hasse

Un **diagramma di Hasse** è una rappresentazione *compatta* di un poset. Utilizza la **posizione** per rappresentare l'ordine e considera la riflessività e transitività **implicite**.

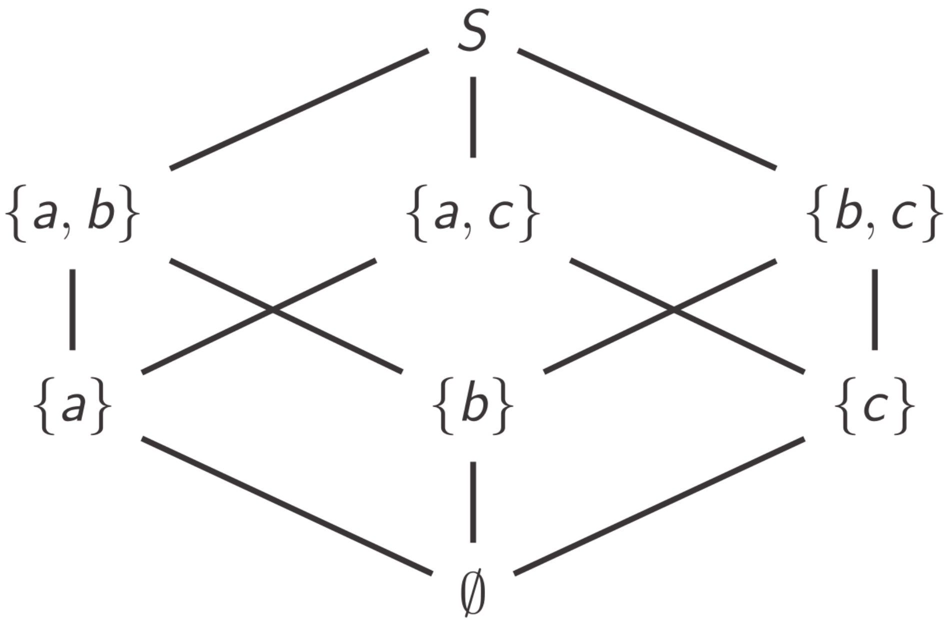
Sia $S = \{a, b, c\}$. Consideriamo il poset $(\mathscr{P}(S), \subseteq)$



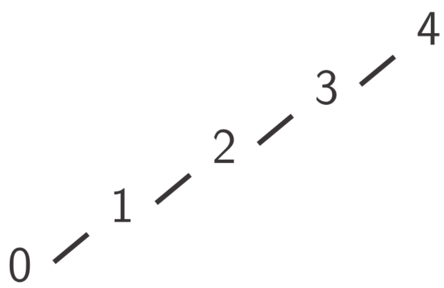
Un diagramma di Hasse è un grafo non orientato tale che per ogni x, y:

- Se $x \leq y$ allora x appare **sotto** y
- x e y sono collegati sse y è una **copertura** di x

L'ordine è la chiusura riflessiva e transitiva del grafo ordientato da giù verso su.



Il diagramma di Hasse di un ordinamento **totale** formerà sempre una **catena**. Per esempio, $(\{0, 1, 2, 3, 4\}, \leq)$:

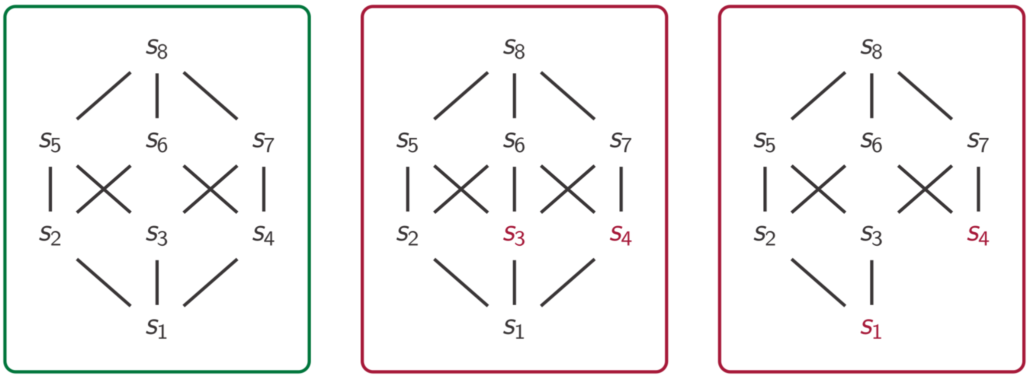


3.7 Reticoli

Un **reticolo** è un poset $(S, \leq)$ tale che per ogni $x, y \in S$:

- Esiste un **minimo maggiorante** $x \sqcup y$ (*join*)

- Esiste un **massimo minorante** $x \sqcap y$ (*meet*)



reticolo no reticolo no reticolo

3.7.1 Proprietà

- $a \sqcup a = a = a \sqcap a$ (*idempotenza*)
- $a \sqcup b = b \sqcup a$ (*commutatività*)
- $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$ (*associatività*)
- $a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c$ (*associatività*)
- $a \sqcup (a \sqcap b) = a = a \sqcap (a \sqcup b)$ (*assorbimento*)

Se $(L, \leq)$ è un reticolo, allora per ogni $a, b, c \in L$:

- $a \leq a \sqcup b$
- Se $a \leq c$ e $b \leq c$ allora $a \sqcup b \leq c$
- $a \sqcap b \leq a$
- Se $c \leq a$ e $c \leq b$ allora $c \leq a \sqcap b$
- $a \sqcup b = b$ sse $a \leq b$
- $a \sqcap b = a$ sse $a \leq b$

3.7.2 Monotonicità

Il join e il meet sono monotoni; cioè se $a \leq c$ e $b \leq d$, allora

- $a \sqcup b \leq c \sqcup d$
- $a \sqcap b \leq c \sqcap d$

3.7.3 Tipi di reticoli

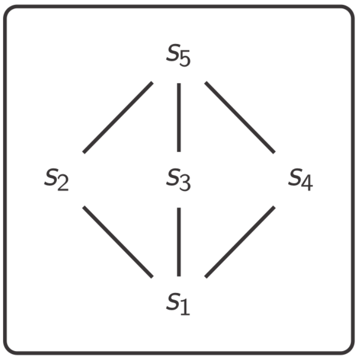
Un reticolo $(L, \leq)$ è

- **Completo** sse per ogni $M \subseteq L$, $\sqcup M$ e $\sqcap M$ esistono
- **Limitato** sse $\underline{1} = \sqcup L$ e $\underline{0} = \sqcap L$ esistono
- **Distributivo** sse meet e join distribuiscono fra di loro:

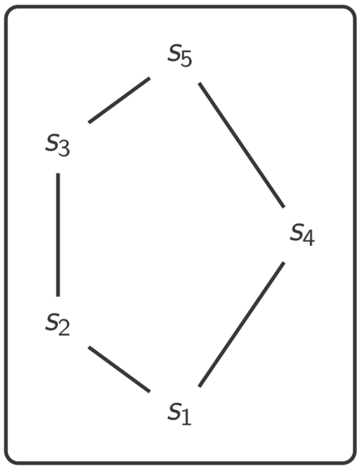
$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$$
$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$$

Ogni reticolo completo è limitato. Ogni reticolo finito è completo e limitato.

I due reticoli *non* distributivi prototipici sono



M_3



N_5

Per sapere se un reticolo è distributivo basta controllare che non sia presente una di queste due strutture.

3.7.4 Complemento

Siano $(L, \leq)$ un reticolo **distributivo limitato** e $a \in L$. Un elemento $b \in L$ è il **complemento** di a sse

$$a \sqcap b = \underline{0} \qquad \wedge \qquad a \sqcup b = \underline{1}$$

Se $a \in L$ ha un complemento, allora questo è **unico**.

$(L, \leq)$ è un **reticolo (distributivo) complementato** sse ogni $a \in L$ ha un complemento.

Il complemento del minimo ($\underline{0}$) è sempre il massimo ($\underline{1}$), e viceversa.

3.8 Algebra di Boole

Boole provò a formalizzare le regole di **ragionamento** combinando proposizioni in base al loro **valore di verità**.

Corrispondono alla prima **formalizzazione** delle **operazioni logiche**:

- **Congiunzione** (*"e"*)
- **Disgiunzione** (*"oppure inclusivo"*)
- **Negazione** (*"non"*)

Boole prima considerò due valori di verità:

- **Vero**: 1
- **Falso**: 0

Ma presto generalizzò a strutture più complesse chiamate **algebre di Boole**.

3.8.1 Reticolo booleano

Un **reticolo booleano** è un reticolo:

- Limitato
- Distributivo
- Complementato

Un reticolo booleano $(L, \leq)$ definisce l'**algebra di Boole**

$$(L, \sqcup, \sqcap, \bar{\cdot}, \underline{0}, \underline{1})$$

Con operazioni per disgiunzione, congiunzione e negazione.

3.8.1.1 Esempio tipico

Per ogni insieme S, il reticolo $(\mathscr{P}(S), \subseteq)$ è un reticolo booleano.

Per ogni $T, T' \subseteq S$:

- $T \sqcup T' = T \cup T'$
- $T \sqcap T' = T \cap T'$
- $\overline{T} = S \setminus T$

La struttura dipende soltanto dalla **cardinalità** di S.

> Aggiungere diagrammi di Hasse #todo-uni

3.8.2 Algebra di Boole tradizionale

L'algebra di Boole "tradizionale" è definita dal reticolo $(\mathscr{P}(\{a\}), \subseteq)$. Invece di chiamare gli elementi $\emptyset$ e $\{a\}$, saranno 0 e 1 ("falso" e "vero").

Operazioni:

- La **disgiunzione** è data dal **join** ($\vee$)
- La **congiunzione** è data dal **meet** ($\wedge$)
- La **negazione** è data dal **complemento** ($\neg$)

Non è casuale che le operazioni su insiemi e su valori logici si somiglino. Infatti le operazioni su insiemi definiscono un reticolo di Boole. Di conseguenza, le proprietà delle operazioni si mantengono.

3.8.3 Proprietà delle operazioni logiche

- $\wedge$ e $\vee$ sono **idempotenti**, **commutative** e **associative**
- $\neg$ è **involutivo** ($\neg\neg x = x$)
- $\wedge$ e $\vee$ **distribuiscono** fra di loro

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

- si soddisfano le **leggi di De Morgan**

$$\neg(x \land y) = \neg x \lor \neg y$$
$$\neg(x \lor y) = \neg x \land \neg y$$

4 Automi a stati finiti e Linguaggi regolari

4.1 Automi

Gli automi sono una rappresentazione formale di un modello di calcolo. Dispositivi che **leggono** una sequenza di simboli, ed **eseguono** istruzioni basate su di essa. È un tipo particolare di una macchina di Turing.

In particolare, gli automi a stati finiti hanno tre proprietà:

- Memoria finita
- Leggono senza scrivere
- Leggono in ordine, senza tornare indietro

Vengono utilizzati per:

- Progettare circuiti digitali
- Analizzare espressioni lessicali
- Cercare parole in un file
- Verificare sistemi temporali
- …

4.1.1 Elementi di un automa

Un automa è composto da:

- Un **alfabeto** (istruzioni)
- Un insieme finito di **stati** (memoria)
- Un insieme di **regole di transizione** (azioni)
- Uno o più stati **iniziali**

- Stati designati come **finali**

L'automa comincia in uno stato iniziale e legge **un'istruzione alla volta**. Le regole di transizione descrivono il **nuovo stato** della memoria in base all'istruzione. Dopo aver letto la sequenza, può finire in uno stato finale (*accetta*), o no (*rifiuta*).

4.1.2 Definizione formale

Un **automa a stati finiti** è una quintupla $\mathscr{A} = \langle Q, \Sigma, \Delta, I, F \rangle$:

- Q è un insieme finito non vuoto di **stati**
- Σ è un insieme finito non vuoto di **simboli** (*alfabeto*)
- $\Delta \subseteq Q \times \Sigma \times Q$ è la **relazione di transizione**
- $I \subseteq Q$ è l'insieme degli **stati iniziali**
- $F \subseteq Q$ è l'insieme degli **stati finali**

Esempio:

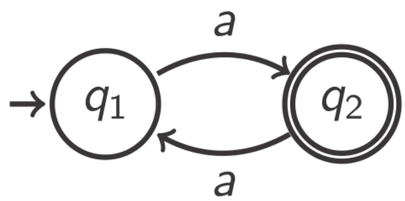
$$\langle \{ q_1, q_2 \}, \{ a \}, \{ \langle q_1, a, q_2 \rangle, \langle q_2, aq_1 \rangle \}, \{ q_1 \}, \{ q_2 \} \rangle$$

Accetta solo le sequenze dispari.

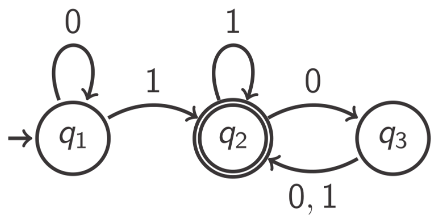
4.1.3 Rappresentazione grafica

Un automa si può rappresentare come un **grafo etichettato** $\langle Q, E, \ell \rangle$ con $\ell : E \to \mathscr{P}(\Sigma)$.

I **nodi** del grafo rappresentano gli stati e gli **archi** le transizioni. Gli stati iniziali si rappresentano con un **semiarco** (freccia senza "partenza") e gli stati finali con un **doppio bordo**.



Esempio:



È l'automa $\langle Q, \Sigma, \Delta, I, F \rangle$ dove

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $I = \{q_1\}$
- $F = \{q_2\}$

$$\Delta = \{ \langle q_1, 0, q_1 \rangle, \quad \langle q_1, 1, q_2 \rangle, \quad \langle q_2, 0, q_3 \rangle, \quad \langle q_2, 1, q_2 \rangle, \quad \langle q_3, 0, q_2 \rangle, \quad \langle q_3, 1, q_2 \rangle \quad \}$$

4.1.4 Linguaggi

Un **alfabeto** è un insieme finito non-vuoto Σ e i suoi elementi si chiamano **simboli**. Una **stringa** (o **parola**) è una sequenza finita di simboli. Essa può essere anche **vuota** (ε). Un **linguaggio** è un insieme di parole. Può essere anche vuoto o infinito. Il linguaggio vuoto $\emptyset$ è diverso dal linguaggio composto solo dalla parola vuota $\{\varepsilon\}$.

La **concatenazione** $x \cdot y$ di due **parole** x, y è la sequenza ottenuta mettendo y immediatamente dopo x

$$\text{ab} \cdot \text{aaab} = \text{abaaab}$$

La **concatenazione** $L \cdot M$ di due **linguaggi** L, M è il linguaggio ottenuto dal concatenare **ogni** parola di L con **ogni** parola di M

$$\{\varepsilon, \text{ ab}, \text{ aaa}\} \cdot \{\text{bb}, \text{ ba}\} = \{\text{bb}, \text{ ba}, \text{ abbb}, \text{ abba}, \text{ aaabb}, \text{ aaaba}\}$$

La concatenazione non è commutativa.

Le **potenze** M^k di un linguaggio M sono definite da

- $M^0 = \{\varepsilon\}$

- $M^{k+1} = M \cdot M^k, k \geq 0$

I linguaggi M^* e M^+ sono

- $M^* = M^0 \cup M^1 \cup M^2 \cup \ldots$
- $M^+ = M^1 \cup M^2 \cup M^3 \cup \ldots$

In M^* è garantita la presenza di ε.

Nota: se Σ è un alfabeto, allora

- Σ^* è l'insieme di tutte le parole su Σ
- Σ^+ è l'insieme di tutte le parole non vuote

Esempio:

- $\{11\}^*$ è il linguaggio di tutte le parole di lunghezza pari su $\{1\}$
- $\{\mathbf{a}\} \cdot \{\mathbf{a}, \mathbf{b}\}^*$ è il linguaggio delle parole che iniziano con a su $\{\text{a}, \text{b}\}$

4.2 Linguaggi regolari

La famiglia dei **linguaggi regolari** è definita ricorsivamente

- Tutti i linguaggi **finiti** sono regolari
- Se L, M sono linguaggi regolari, allora sono regolari anche
  - $L \cup M$
  - $L \cdot M$
  - L^*
  - L^+

Esempio:

$$L = \{0,1\}^* \cdot \{01\} \cdot \{0,1\}^* = \{x01y \mid x, y \in \{0,1\}^*\} \text{ è regolare}$$

L'insieme di tutti i palindromi su un alfabeto Σ non è regolare.

4.3 Teorema di equivalenza

Il **linguaggio riconosciuto** da un automa $\mathscr{A}$ è l'insieme delle parole accettate da $\mathscr{A}$.

Teo: gli automi di stati finiti riconoscono *esattamente* i linguaggi regolari.

4.4 Costruzione di automi

Per vedere che ogni linguaggio regolare è **riconosciuto** da un automa, vediamo che

- Ogni **parola** è riconosciuta; $\emptyset$ e $\{\varepsilon\}$ sono riconosciuti
- Se L, M sono riconosciuti, allora $L \cup M$, $L \cdot M$ e L^+ sono riconosciuti

4.4.1 Unione

Se $\mathscr{A} = (Q_1, \Sigma, \Delta_1, I_1, F_1)$ e $\mathscr{B} = (Q_2, \Sigma, \Delta_2, I_2, F_2)$ riconsocono L e M ($Q_1 \cap Q_2 = \emptyset$), allora

$$(Q_1 \cup Q_2, \Sigma, \Delta_1 \cup \Delta_2, I_1 \cup I_2, F_1 \cup F_2)$$

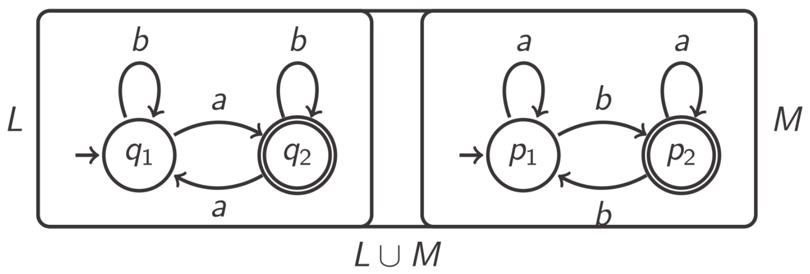
riconosce $L \cup M$.

Proviamo ad accettare con ogni automa *indipendentemente*.

Esempio:

- Alfabeto $\{$ a, b $\}$
- L sono le parole che hanno un numero dispari di a
- M sono le parole che hanno un numero dispari di b

Creare un automa per $L \cup M$



4.4.2 Concatenazione

Se $\mathscr{A} = (Q_1, \Sigma, \Delta_1, I_1, F_1)$ e $\mathscr{B} = (Q_2, \Sigma, \Delta_2, I_2, F_2)$ riconsocono L e M ($Q_1 \cap Q_2 = \emptyset$), allora

$$(Q_1 \cup Q_2, \Sigma, \Delta, I_1, F_2)$$

dove

$$\Delta := \Delta_1 \cup \Delta_2 \cup \{ \langle q, \sigma, p \rangle \mid \langle q, \sigma, q' \rangle \in \Delta_1, q' \in F_1, p \in I_2 \}$$

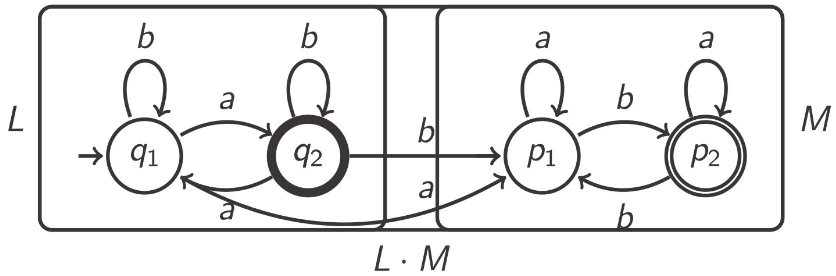
riconosce $L \cdot M$.

Accettiamo la parola in L e poi quella in M.

Esempio:

- Alfabeto $\{ \text{a}, \text{b} \}$
- L sono le parole che hanno un numero dispari di a
- M sono le parole che hanno un numero dispari di b

Creare un automa per $L \cdot M$



4.4.3 Iterazione

Se $\mathscr{A} = (Q, \Sigma, \Delta, I, F)$ riconosce L, allora

$$(Q, \Sigma, \Delta', I, F)$$

dove

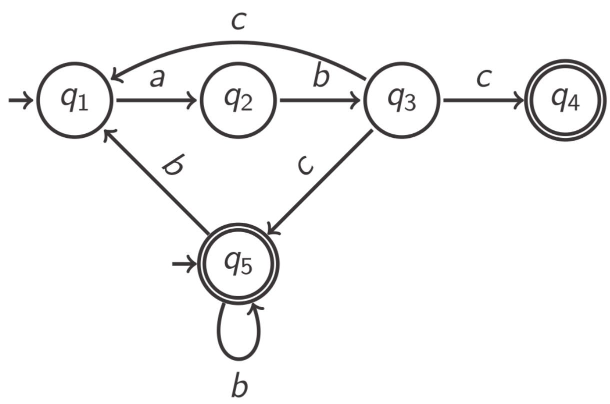
$$\Delta' := \Delta \cup \{ \langle q, \sigma, p \rangle \mid \langle q, \sigma, q' \rangle \in \Delta, q' \in F, p \in I \}$$

riconosce L^+.

Accettiamo una parola in L e ricominciamo.

Esempio:

Sia $L = \{ \text{abc}, \text{ b} \}$. Costruire un automa che riconosce L^+



4.5 Determinismo

In un automa, il processo per accettare una parola è **nondeterminista**. Quando si legge una parola, ci sono diverse strade da seguire per arrivare a uno stato finale.

In un automa **determinista** il processo di lettura è **prefissato**: c'è un solo cammino da seguire per parola.

Formalmente, in un automa determinista:

- Δ è una funzione $Q \times \Sigma \to Q$
- I è un singoletto $I = \{ q \}$

4.6 Linguaggi

Gli automi nondeterministi sono **più generali** dei deterministi. Eppure accettano esattamente la stessa classe di linguaggi: quelli **regolari**.

Gli automi nondeterministi si possono trasformare in automi deterministi che accettano lo stesso linguaggio. Ma l'automa determinista ha bisogno di molti più stati (*costruzione dell'insieme potenza*).

5 Ricorsione e Induzione

Ricorsione e induzione sono due **principi matematici** per definire, studiare e manipolare oggetti e strutture complesse a partire da elementi semplici.

La loro caratteristica principale è l'autoreferenza, che però deve essere *ben fondata*.

In matematica e informatica il termine **ricorsione** si riferisce alla **definizione** di strutture basate su sé stesse.

Induzione si refirisce invece al processo di derivare una proprietà generale a partire da casi particolari. In matematica è un metodo di dimostrazione per gestire le strutture definite ricorsivamente.

L'uso autoreferenziale in ricorsione e induzione è utile per

- **Definire** insiemi, strutture dati, … (*definizioni ricorsive*)
- **Verificare** proprietà di questi insiemi, … (*dimostrazioni per induzione*)
- **Descrivere** metodi di calcolo e programmi su di essi (*definizioni ricorsi e algoritmi*)

Una **definizione** caratterizza e descrive le proprietà che distringuono un oggetto di interesse dagli altri oggetti.

5.1 Assiomi

Un **assioma** è un principio che è considerato vero senza bisogno di dimostrarlo. "Verità evidenti" che forniscono il punto di partenza per lo sviluppo e studio di una disciplina formale.

La scelta degli assiomi può avere ripercussioni importanti. Per esempio la geometria euclidea si basa su cinque assiomi, l'ultimo dei quali è: *data una retta e un putno fuori da essa, esiste soltanto una parallela*. Diverse geometrie sono state "create" variando quest'ultimo assioma.

5.2 Ipotesi

Un'**ipotesi** è una proposizione considerata **temporaneamente vera** durante il processo di dimostrazione.

È fondamentale per l'induzione, ma anche utile in dimostrazioni dove ci sono diversi casi da analizzare.

5.3 Teorema

Un **teorema** è una conseguenza logica degli assiomi. Una proposizione che è **sempre vera** nella teoria definita da essi.

Per essere sicuri che siano teoremi, abbiamo bisogno di una **dimostrazione**.

A volte, un teorema è anche chiamato

- Lemma
- Corollario
- Proposizione

Questa scelta è guidata da una questione stilistica per distinguere la loro importanza o funzionalità.

5.4 Definizioni ricorsive

In generale, una definizione ricorsiva ha bisogno di

- Uno o più **casi base** (*base della ricorsione*)
- Una **funzione** per costruire nuovi casi da quelli esistenti (*passo ricorsivo*)

L'esempio più semplice è la definizione dei numeri naturali:

- 0 è un numero naturale
- Se n è un numero naturale, allora $s(n)$ (*il successivo di n*) è un numero naturale

5.5 Ordine naturale

I numeri naturali hanno un **ordine totale** che possiamo anche definire ricorsivamente:

- $\forall n \in \mathbb{N}, n < n + 1$
- Se $n < m$ e $m < l$, allora $n < l$

5.6 Buon ordinamento

In un poset $(S, \leq)$, $\leq$ è un **buon ordine** sse ogni sottoinsieme **non vuoto** $X \subseteq S$ ha un elemento **$\leq$-minimo**. In questo caso si dice che S è **ben ordinato** o **ben fondato**.

Teorema: $\mathbb{N}$ è ben ordinato.

Ogni definizione per ricorsione stabilisce un ordine naturale che è un buon ordine.

5.7 Principio di induzione (generale)

Sia S un insieme definito ricorsivamente e P una proprietà. Se:

1. Dimostriamo che P è vero in ogni caso base
2. Supponiamo che P è vero per elementi generici $T \subseteq S$
3. Dimostriamo che P è vero per elementi costruiti da T tramite il passo ricorsivo

allora P è vero per **tutti** gli elementi di S.

5.8 Principio di induzione in $\mathbb{N}$

Per dimostrare che una proprietà P è vera **per ogni** $n \in \mathbb{N}$ sfruttiamo la definizione ricorsiva di $\mathbb{N}$:

- P deve essere vera per 0
- **Se** P è vera per un generico $n \in \mathbb{N}$, allora P **deve** essere vera per $n + 1$

Ovvero se:

- $P(0)$ è vero

- $P(n)$ implica $P(n+1)$ per qualunque generico $n \in \mathbb{N}$

allora $P(n)$ è vera per ogni $n \in \mathbb{N}$.

Nota: cambiando il caso base si possono dimostrare proprietà per ogni $n \geq k$.

Una dimostrazione per induzione in $\mathbb{N}$ si svolge in **tre passi**:

1. Dimostrare il **caso base** $(n = 0)$
2. Supporre che la proprietà sia vera per un n (*ipotesi di induzione*)
3. Dimostrare che è vera anche per $n + 1$ (*passo induttivo*)

