

Fondamenti

di informatica



Secondo alcuni studi, la media di accrescimento della felicità durante la consumazione di questa materia al fine di conseguire esami è decaduta del -125,079% nell'arco di tre mesi, provocando così una forte recessione che ha causato il crollo mondiale di vari mercati economici.

La matematica discreta

Discreto vuol dire *composto di elementi distinti separati tra di loro*, mentre matematica vuol dire *insieme di punti costituito da un numero finito o da un infinità numerabile di punti*.

La relazione fra discreto e continuo è una delle problematiche più antiche del pensiero umano; i pitagorici svilupparono un modello fondato sugli opposti, dove il finito è **misurabile e positivo**, e l'infinito è **incommensurabile e negativo**.

Intuitivamente, un sistema è *discreto* se costituito da elementi isolati con vuoti fra gli elementi, *continuo* se non ci sono vuoti fra gli elementi.

Da questo, deriva il concetto di **discretizzazione**. Tutti i sistemi informatici sono basati sul codice binario, sistema discreto per definizione (0 e 1). Possiamo approssimare un sistema continuo, dividendolo in piccole parti, ed è questo il processo.

Un esempio di discretizzazione può essere il passaggio dalle macchine fotografiche analogiche a quelle digitali. Quelle analogiche erano un sistema continuo, mentre con le macchine fotografiche digitali le immagini sono divise in pixel, quindi **divisa in piccole parti**.

Quando si parla della matematica discreta individuiamo tre fasi:

- **classificazione**: individuare le caratteristiche comuni di entità diverse
 - o **Teoria degli insiemi**
- **Enumerazione**: assegnare ad ogni oggetto un numero naturale
 - o **Contare**
- **Combinazione**: permutarne e combinarne gli elementi.
 - o **Grafi**

Il discreto si basa sul concetto di **cardinalità**. Nella teoria degli insiemi, per cardinalità di un insieme finito si intende il *numero dei suoi elementi*. Dato un insieme A, la cardinalità si indica con $|A|$. si può definire la cardinalità anche di un insieme infinito, ed è legata al concetto di numerabilità.

Un insieme è discreto se e solo se i suoi elementi si possono numerare, ovvero che se ad ogni elemento di un insieme posso associare un numero naturale. Se c'è la cardinalità dei numeri naturali, allora è numerabile.

Alcuni esempi di insiemi numerabili sono: numeri naturali **N**, gli insiemi finiti, i numeri razionali **Q**.

La logica

In matematica, il tipo di logica che viene usato è la **logica formale**, che vede i procedimenti del ragionamento, argomentazione e procedimenti inferenziali come calcoli formali e li trasforma in formule. Formalizzazione, infatti, significa trasformare in formula.

L'algebra booleana è il fondamento dell'informatica. Il suo inventore, Boole, ha vissuto in anni dove i computer non c'erano, e ha definito le basi di essi.

I numeri naturali

Nascono dall'attività di contare, essi formano l'insieme dei numeri naturali, che si denota con **N**. N è un numero generico, ed è la formalizzazione di una variabile. Legato al concetto dei numeri naturali abbiamo il concetto del **contare**, che vuol dire assegnare ad ogni oggetto un numero naturale, ovvero un ordine.

L'insieme N ha un limite inferiore, **0**, ma non ha un limite superiore, ovvero è infinito. Formalmente, i numeri naturali hanno le seguenti caratteristiche:

- i numeri naturali includono lo 0
- Ogni elemento n ha esattamente un successore ($s(n)$)
- 0 non è successore di nessun elemento
- due elementi diversi hanno successori diversi

Si parte dal definire lo *zero*, che deve avere un successore. Abbiamo qua un esempio di come funziona la successione numerica nell'insieme N.

0	S(0)	S(S(0))
Deve avere successore	Non può essere 0, deve avere un successore	Non può essere 0, non può essere $s(0)$, e deve avere un successore

Numeri interi

È un altro insieme. Anche relativi, rappresentano l'insieme dei numeri naturali preceduti da un segno + o -, e si indicano con **Z**. L'insieme **Z** contiene all'interno l'insieme N, da zero in avanti, ovvero *tutti i positivi*. Senza segno o col + sono positivi in questo insieme.

Altra caratteristica di questo insieme è che ogni intero ha un successore ma anche un predecessore; non c'è né un minimo né un massimo, poiché non hanno né un minimo né un massimo.

I numeri razionali

L'insieme dei numeri razionali rappresenta le frazioni, indicano un rapporto o una proporzione risultante di una divisione, e si esprimono come rapporto di due numeri interi. Si chiama **Q**, e Q contiene Z come sottoinsieme proprio.

Numeri irrazionali

Sono quelli che possono non essere espressi da frazioni.

Numeri reali

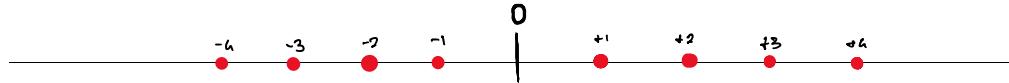
L'insieme dei numeri reali è indicato con **R**, e contengono tutti i numeri che ammettono una rappresentazione decimale.

Numeri complessi

I numeri complessi sono un insieme che si indica con **i**, ed estendono i reali per eseguire operazioni che non sono ben definite altrimenti.

La retta reale

La retta reale è una retta immaginaria dove si può rappresentare l'insieme \mathbb{R} . A ogni punto della retta è associato un numero reale e viceversa, ovvero con una corrispondenza biunivoca; è una rappresentazione 1:1.



Induzione

L'induzione è il procedimento di dimostrazione di un'affermazione dipendente da un numero naturale: se l'affermazione è vera nel caso $n=1$ (base dell'induzione), e se, essendo vera per n , è vera anche per $n+1$, allora è vera per ogni n .

È una proprietà generica, ed è vera in tutto l'insieme \mathbb{N} **se e solo se**, quindi condizioni necessarie e sufficienti:

- è vera in Z
- se è vera in n allora è vera in $s(n)$

Questa definizione di successore definisce implicitamente una nozione di ordine, ovvero che *un elemento è sempre più piccolo del suo successore*.

Valore assoluto

Il valore assoluto è il numero privo di segno, lo zero è di per sé privo di segno. L'opposto di un numero si ottiene cambiandogli il segno.

Gli insiemi

Un insieme è una collezione di oggetti, detti *elementi dell'insieme*. Per nominare un insieme, dobbiamo seguire alcune denominazioni:

- $X \in S$: l'oggetto x è elemento dell'insieme S
- $X \notin S$: l'oggetto x non è elemento di S
- $X = Y$ = l'oggetto x è uguale all'oggetto y
- $X \neq Y$: oggetto x non è uguale all'oggetto y
- $S = T$, s e t insiemi:
 - $X \in S$ sse (*se e solamente se*) $X \in T$
- \emptyset : insieme vuoto
- **Singololetto**: per ogni oggetto x esiste un insieme $\{x\}$, il cui unico elemento è x .

Un po' di storia

Secondo la definizione di Contor, *gli oggetti che formano l'insieme sono i suoi elementi*.

Per molto tempo, si è creduto che la teoria degli insiemi potesse fornire una base solida e univoca alla matematica. **Bertrand Russell** pensò, invece, che questo non potesse essere la soluzione, e quindi sviluppò il *paradosso del barbiere*.

“In un villaggio vi è un solo barbiere che rade tutti e soli gli uomini del villaggio che non si radano da soli.
Chi è il barbiere?”

A seguito di un tale paradosso, abbiamo due scenari possibili:

1. Il barbiere si rade da solo → lui rade **solo** quelli che non si radano da soli; quindi, non può radere sé stesso
2. Il barbiere non si rade da solo → lui rade **tutti** quelli che non si radono da soli, deve radere sé stesso, ma per il primo punto ciò non è possibile.

Il paradosso venne successivamente formalizzato con il **paradosso Russell-Zermelo**, dove venne ridefinito come:

“L’insieme di tutti gli insiemi che appartengono a sé stessi appartiene a sé stesso”

Come denominare gli insiemi

Gli insiemi sono denotati per lettere latine maiuscole di norma, come A, B, C, etc. I loro elementi, invece, sono denominati con lettere minuscoli. Gli insiemi possono essere anche elementi di altri insiemi.

Uguaglianza

L’uguaglianza fra oggetti si denota con $=$ e \neq , e ha tre proprietà:

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

Rappresentare gli insiemi

Ci sono varie modalità per rappresentare gli insiemi.

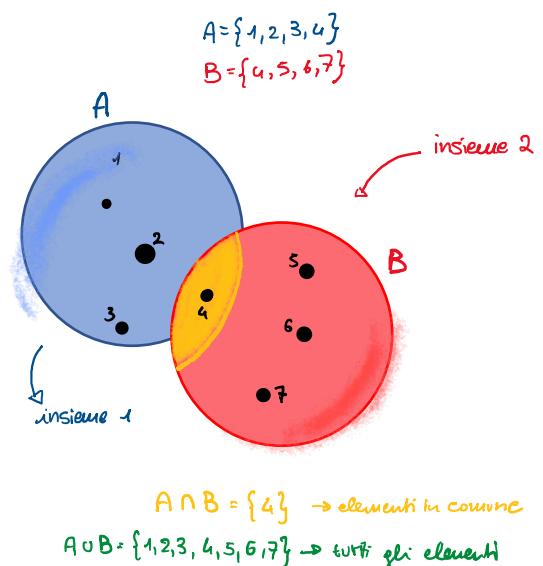
Diagramma di Eulero-Venn

È molto conosciuta, ma è limitata. Gli elementi sono punti nel suo interno, e si possono rappresentare anche elementi intersecati.

Rappresentazione estensionale

Consiste nell’elencare esplicitamente tutti gli elementi, racchiusi tra parentesi graffe, come $\{x, y, z\}$.

$$A = \{\text{elemento 1, elemento 2, elemento 3, ...}\}$$



Rappresentazione intensionale

È descritto in modo implicito come la collezione di elementi che condividono una certa proprietà P. Viene scritta come $\{x | P(x)\}$, per indicare l’insieme di tutti gli elementi che condividono una certa proprietà P. Si definisce dopo una proprietà che si vuole, e tutti gli elementi che sono validi per queste proprietà fanno parte dell’insieme.

Esempi:

1. $S = \{x \mid x \in \mathbb{Z}, x > 0\} \rightarrow S$ è l’insieme degli interi positivi, appartenenti all’insieme \mathbb{Z} e maggiori di zero.

$$A = \{\text{elementi} \mid (\text{tale che}) \text{ proprietà da soddisfare con elemento all'interno}\}$$

I sottoinsiemi

Il sottoinsieme è quando vuol dire che un insieme fa parte di un altro insieme. Come esempio, possiamo prendere il fatto che **N è un sottoinsieme di Z** → $N \subset Z$. Il segno con cui si include che un insieme è *sottoinsieme* di un altro insieme è appunto \subset .

Si dice **improprio** quando tutti gli elementi del sottoinsieme sono contenuti nell'insieme più grande, e si indica con \subseteq . Se due insiemi hanno gli stessi elementi, allora vengono detti **mutuamente insieme**.

Se $B \subseteq A$, ma $A \neq B$, allora B è strettamente contenuto in A, e si scrive $B \subset A$. Questo vuol dire che i due insiemi non sono uguali.

L insieme vuoto è **sempre** sottoinsieme di un altro insieme ($\emptyset \subset A$), un insieme è sempre **sottoinsieme proprio** dello stesso insieme ($A \subseteq A$), e l insieme vuoto è un **sottoinsieme proprio** dell'insieme vuoto ($\emptyset \subset \emptyset$)

Insieme potenza

L insieme potenza di un insieme S, scritto **P(S)**, è l insieme formato da tutti i sottoinsiemi di S. Scrivendolo con la rappresentazione intenzionale, sarebbe:

$$P(S) = \{X \text{ (insieme)} \mid X \subseteq S\}$$

L insieme vuoto fa sempre parte di qualsiasi insieme; successivamente, si passa ai singoletti, e successivamente ad ogni coppia possibile, fino ad arrivare all insieme di partenza.

Ecco vari esempi che possono essere utili:

- $P\{x,y\} \rightarrow \{\emptyset, \{x\}, \{y\}, \{x,y\}\}$
- $P\{a,b,c\} \rightarrow \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$
- $P\{\{\emptyset\}\} = \{\emptyset, \{\emptyset\}\} \rightarrow$ l insieme vuoto all interno del primo insieme viene considerato un **elemento** come tutti gli altri, quindi bisogna includere nell'insieme potenza sia l insieme vuoto che l insieme contenente l insieme vuoto.

Per quanto riguarda la cardinalità, se l insieme di partenza ha n elementi, con $n \neq 0$, allora $P(S)$ ha 2^n elementi. Per dimostrare questo, dobbiamo fare una dimostrazione per induzione.

Ipotesi: Se $|S|$ (cardinalità) = n allora $|P(S)| = 2^n$

Tesi 1: è vero per $n=0$

Se $n = 0$, allora $|S| = 0$

Se la cardinalità è 0, allora $S = \emptyset$, e $P(S) = \{\emptyset\}$.

La cardinalità $|P(S)| = 1$, che è uguale a 2^0 ,

Allora la tesi 1 è **vera**.

Tesi 2: se è vero per k, allora è vero per $k + 1 \rightarrow$ ovvero per tutti i numeri naturali.

Definiamo un insieme S la cui cardinalità è **$|S| = k + 1$**

Definiamo un insieme T togliendo a S l elemento $x \in S$

Definiamo l insieme $T = S \setminus \{x\}$ (insieme differenza)

La cardinalità di T, togliendo un elemento di S ($|S| = k + 1$), è **$|T| = k$**

In base all **ipotesi**, se $|T| = k$ allora $|P(T)| = 2^k$

L insieme potenza di S quindi diventa $P(S) = P(T) \cup Y$ (Y: terzo insieme) = $\{X \mid X \cup \{x\}, X \in P(T)\}$.

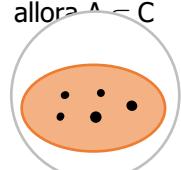
L insieme Y denominato qua è uguale a un insieme potenza, con aggiunto x ad ogni suo singolo elemento.

Con questo, la cardinalità è:

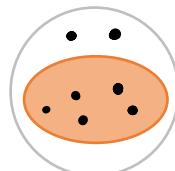
L'inclusione

L inclusione definisce un ordine, e soddisfa tre proprietà:

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



Improprio
Il sottoinsieme piccolo contiene tutti gli elementi del grande (può essere anche vuoto)



Proprio
Il sottoinsieme piccolo non contiene tutti gli elementi del grande, e non è vuoto

$$|P(S)| = |P(T)| + |Y| = 2^k + 2^k = 2^{k+1}$$

questo perché entrambi gli insiemi hanno n elementi (come detto nell'ipotesi), e per le proprietà delle potenze, si arriva al risultato.

OPERAZIONI TRA GLI INSIEMI

Unione

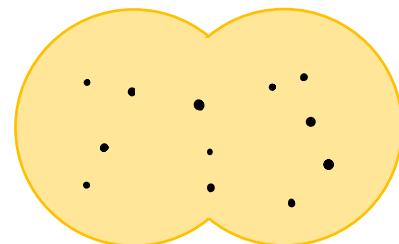
L unione di due insiemi si denota con \cup , ed è definito come:

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

$A \cup B$ è l insieme di tutti gli elementi che appartengono ad A , oppure a B .

Esempi:

- $\{x \in \mathbb{N} \mid x > 3, x < 10\} \cup \{x \in \mathbb{N} \mid x > 90, x < 91\} = \{4, 5, 6, 7, 8, 9\}$ (non ci sono numeri in comune tra i due, e non ci sono numeri naturali tra 90 e 91)



Le proprietà

Dati due insiemi A e B :

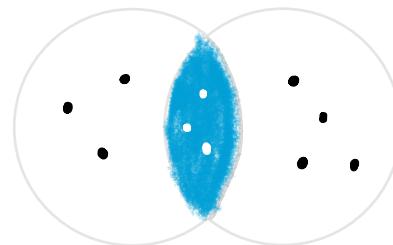
- **Idempotenza:** $A \cup A = A$
- **Commutatività:** $A \cup B = B \cup A$
- **Associatività:** $A \cup (B \cup C) = (A \cup B) \cup C$
- **Neutro:** $A \cup \emptyset = A$
- **Assorbimento:** $A \cup B = B$ sse $A \subseteq B$
- **Monotonicità:** $A \subseteq A$

Intersezioni

L intersezione di due insiemi A e B si denota con $A \cap B$, ed è definito come:

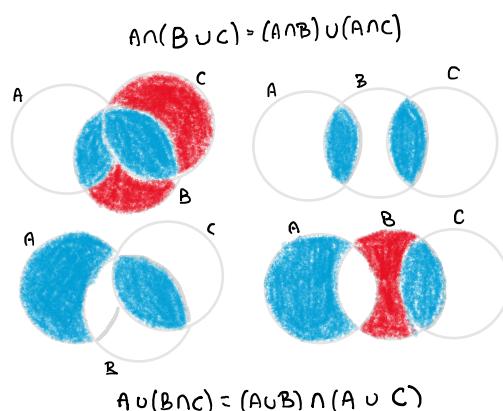
$$A \cap B = \{x \mid x \in A, x \in B\}$$

Esempi: $\{x \in \mathbb{N} \mid x > 3, x < 10\} \cap \{x \in \mathbb{N} \mid x > 90, x < 91\} = \emptyset$



Le proprietà

- **Idempotenza**
- **Commutatività**
- **Associatività**
- **Annichilazione:** $A \cap \emptyset = \emptyset$
- **Assorbimento:** $A \cap B = B$ sse $B \subseteq A$
- **Monotonicità:** $A \cap B \subseteq A$ e $A \cap B \subseteq B$



La proprietà distributiva

Questa proprietà vale sia per l unione che l intersezione.

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

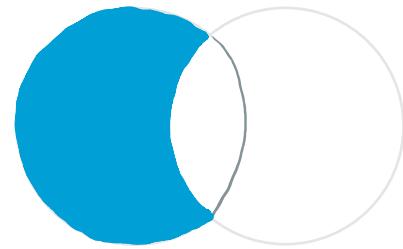
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Differenza

È definita come $A \setminus B$, o:

$$A \setminus B = \{x \mid x \in A, A \cap B, x \notin B\}$$

$A \setminus B$ è l'insieme di tutti gli elementi che appartengono ad A ma non a B .



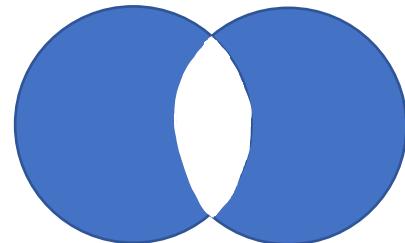
Esempi:

- $A = \{a, b, c, d, e\}, B = \{a, c, f, e, h\} \rightarrow A \setminus B = \{b, d\}$
(tolgo da A quelli comuni a tutte e due)
- Se P è l'insieme di tutti i numeri naturali pari, e D dei numeri naturali dispari, allora $N / D = P$

Differenza simmetrica

Viene indicata con Δ , e definita come:

$$A \Delta B = \{A \setminus B\} \cup \{B \setminus A\}$$



Le Proprietà

- $A \Delta B = B \Delta A$
- $(A \Delta B) \Delta C = A \Delta (B \Delta C)$
- $(A \Delta B) \Delta (B \Delta C) = A \Delta C$
- $A \Delta \emptyset = A$

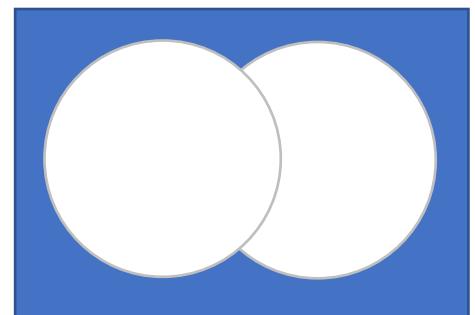
Complementazione

Se abbiamo un insieme di riferimento U chiamato **universo**, la differenza di tale insieme U rispetto ad un insieme A si chiama **complemento**.

Per facilità, il complementare di A verrà chiamato \underline{A} .

Alcune note riguardo la complementazione:

- Il complemento dell'universo è \emptyset
 - o Il complemento di \emptyset è l'insieme universo.
- $A \cup \underline{A} = \text{insieme universo}$
- $A \cap \underline{A} = \emptyset$
- A complementare del complementare = A
- $\underline{A \cup B} = \underline{A} \cap \underline{B}$
- $\underline{A \cap B} = \underline{A} \cup \underline{B}$. Questo viene chiamato anche **Legge di De Morgan**
- $A \setminus B = A \cap \underline{B}$
- $A \subset B$ sse $\underline{B} \subset \underline{A}$
- Bisogna sempre definire un **insieme universo** per poter fare il complementare di un insieme



Le famiglie di insiemi

Una famiglia di insiemi è un insieme F tale che ogni elemento di F è un insieme. Può essere rappresentato come:

$$F = \{\{insieme1\}, \{insieme2\}, \{insieme3\}, \dots\}$$

In queste famiglie si possono definire:

- **Unione:** $F = \{x \mid x \in A \text{ per qualche elemento } A \in F\}$
- **Intersezione:** $F = \{x \mid x \in A \text{ per ogni elemento } A \in F\}$

Esempi:

$$A = \{1,2\}$$

$$B = \{x,y,z,1\}$$

$$\{A, B\} = \{(1,2), (x,y,z,1)\}$$

Le partizioni

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

- $\forall C \in F \rightarrow C \in F, C \neq \emptyset$ (non ci sono partizioni con insieme vuoto)
- $\cup F = A$ (copertura, ovvero l'unione della famiglia di insiemi è l'insieme stesso)
- Se $C \in F, D \in F, C \neq D$, allora $C \cap D = \emptyset$ (il sistema non deve avere partizioni; C e D sono partizioni dell'insieme)

Esempi:

$$S = \{a,b,c,d,e,f,g,h\}$$

$$S_1 = \{a, b, c, d\}$$

$$S_2 = \{a,c,e,f,g,h\}$$

$$S_3 = \{a,c,e,g\}$$

$$S_4 = \{b,d\}$$

$$S_5 = \{f,h\}$$

$F_1 = \{S_1, S_2\}$ non è una partizione \rightarrow hanno degli elementi in comune

$F_2 = \{S_1, S_5\}$ non è una partizione \rightarrow non ci sono tutti gli elementi dell'insieme S

$F_3 = \{S_3, S_4, S_5\}$ è una partizione \rightarrow nessuna intersezione e tutti gli elementi dell'insieme iniziale

Le relazioni tra gli insiemi

Ordinamenti negli insiemi

Gli insiemi non sono ordinati; questo vuol dire che l'ordine non conta nell'insieme. Inoltre, si possono ripetere gli elementi nell'insieme; verranno contati una sola volta.

Coppia ordinata

Una coppia ordinata è una collezione di due elementi, dove si può distinguere il primo e il secondo elemento, indicata con $\langle x,y \rangle$.

In una coppia ordinata, l'ordine con cui gli elementi sono posti è importante. Infatti, $\langle x,y \rangle$ è diverso da $\langle y,x \rangle$.

Può esistere anche una coppia ordinata con due volte lo stesso elemento, come $\langle x,x \rangle$.

Formulazione insiemistica

La coppia ordinata $\langle x,y \rangle$ è l'insieme: $\{\{x\}, \{x,y\}\}$, ovvero una famiglia degli insiemi.

In questo modo, possiamo distinguere il primo dal secondo elemento:

- il **primo elemento** (x) è quello che appartiene ad entrambi gli insiemi della famiglia
 - o x è il primo elemento $x \in \cap F$ (x appartiene a tutti gli insiemi, all'unione degli insiemi della famiglia)
- il **secondo elemento** (y) è quello che appartiene ad uno degli insiemi della famiglia, ma non nell'altro.
 - o y è il secondo elemento sse $y \in \cup F \setminus \cap F$ (y non appartiene a tutti gli insiemi)
 - $\{y\} = \cup F \rightarrow$ condizione posta per il caso in cui $F = \{\{y\}\}$, che può accadere nel caso una coppia ordinata ha come 1° e 2° elemento lo stesso elemento: $\langle y,y \rangle = \{\{y\}, \{y,y\}\} = \{\{y\}\}$

- Se è in tutti e due è il primo, se è in solo uno è il secondo.

Dimostrazione della formulazione insiemistica

Si vuole vedere se la definizione insiemistica caratterizzi correttamente le coppie ordinate, ovvero dimostrare che $(a,b) = (x,y)$ sse $\{\{a\}, \{a,b\}\} = \{\{x\}, \{x,y\}\}$. Quando queste due famiglie di insiemi sono uguali?

Possibilità 1:

$\{a\} = \{x\} \rightarrow a = x$
 $\{a,b\} = \{x,y\} \rightarrow$ dalla prima condizione, sostituendo x con a $\rightarrow \{a,y\} = \{a,b\} \rightarrow b = y$

Possibilità 2:

$\{a\} = \{x,y\} \rightarrow$ l'unica possibilità è che i due elementi x e y siano uguali ad a $\rightarrow a = x, a = y$
 $\{a,b\} = \{x\} = \{a\} \rightarrow$ le due famiglie di insiemi sono uguali

Generalizzazione: si possono generalizzare le coppie ordinate a **tuple ordinate** di lunghezza n, definendo:

$$\langle x_1, \dots, x_n, x_{n+1} \rangle = \langle x_n, \dots, x_1 \rangle, x(n+1) = n = 2$$

Prodotto cartesiano

Dati due insiemi A e B, possiamo definire il prodotto cartesiano.

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

$$A \times B \neq B \times A$$

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

Dato un insieme S, io posso avere $s^n = \{s^1, s^n, \text{etc}\} \mid s \in S$

Le funzioni

Una classe di relazioni binarie che hanno particolare importanza sono le **funzioni**. Una funzione è una relazione $R \subseteq A \times B$ tale che per ogni $a \in A$ (dominio) corrisponde **al più** un elemento $b \in B$ (codominio)

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

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

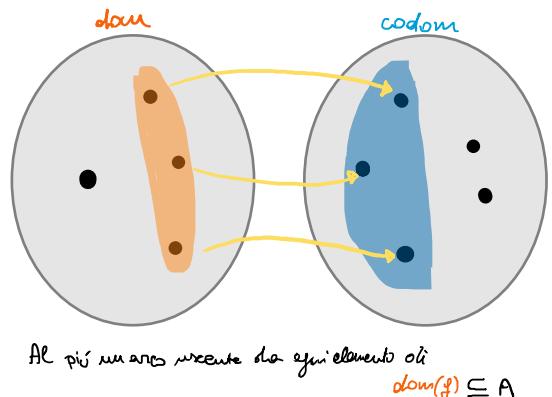
Le funzioni parziali

Ogni $a \in A$ è in relazione con **al più** un elemento di B , ma possono esistere elementi di A che non sono in relazione con nessun elemento di B .

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

Se $z \in \text{dom}(f)$, allora si dice che f è definita in z .

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



Le funzioni totali

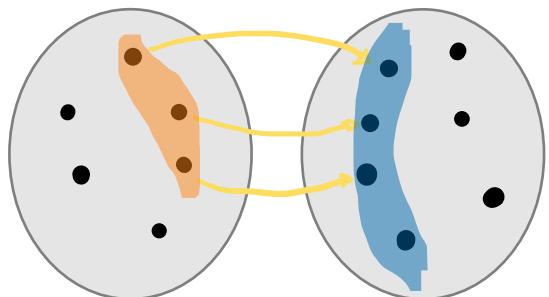
Se per ogni $a \in A$ esiste esattamente un $b \in B$ tale che $\langle a, b \rangle \in R$, allora R è una funzione totale. Se l'insieme $A = \text{dom}(f)$ allora f è una funzione totale.

Se $f \subseteq A \times B$ è una funzione, scriviamo: $f: A \rightarrow B$

In questo caso, $\text{dom}(f) \subseteq A$ e $\text{codom}(f) \subseteq B$

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

Elementi di A in relazione con un solo elemento di B

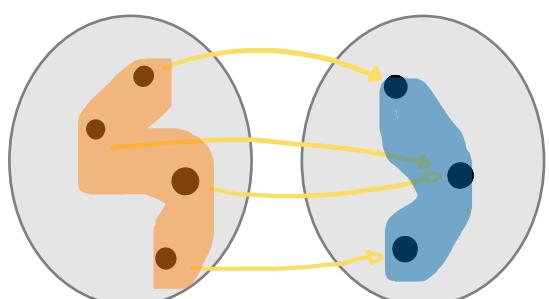


Funzione iniettiva

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

$f: A \rightarrow B$ è iniettiva sse per ogni $x, y \in A$, $x \neq y$ implica $f(x) \neq f(y)$

Due X diverse non **possono** indicare la stessa y .



Tutte le immagini nel codominio devono avere una relazione

Funzione suriettiva

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

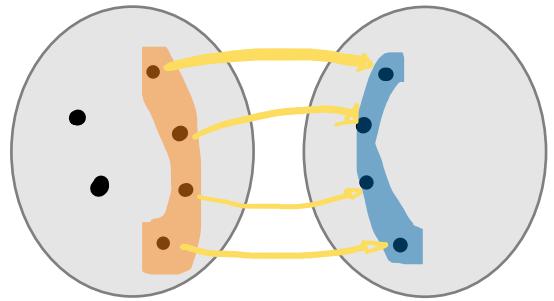
$f: A \rightarrow B$ è suriettiva sse per ogni $y \in B$ esiste un $X \in A$ tale che $f(x) = y$

Funzione biiettiva

Una funzione $f: A \rightarrow B$ è biiettiva sse è **iniettiva e suriettiva**.

F non può essere totale!

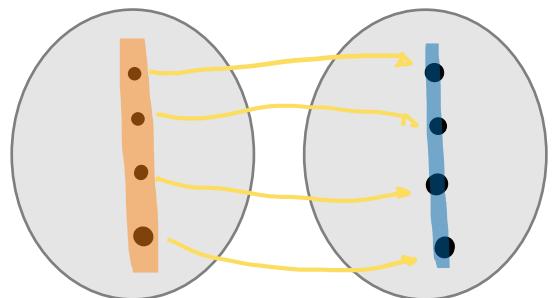
Ad ogni $x \in \text{dom}(f)$ corrisponde esattamente un $y \in B$. ad ogni $y \in B$ corrisponde esattamente un $x \in \text{dom}(f)$



Funzione biunivoca

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

Tale funzione deve essere totale, iniettiva e suriettiva.



Arietà

Esistono relazioni e funzioni con più argomenti o operandi, in cui l'insieme di partenza è definito dal prodotto cartesiano di più insiemi.

- funzione unaria $f: A \rightarrow B$
- funzione binaria $f: A_1 \times A_2 \rightarrow B$
- funzione ternaria $f: A_1 \times A_2 \times A_3 \rightarrow B$
- funzione n-aria $f: A_1 \times \dots \times A_n \rightarrow B$

L'arietà di una relazione è il numero dei domini nel prodotto cartesiano.

Punto fisso

Sia A un insieme e $f: A \rightarrow A$ una funzione. Un punto fisso di f è un elemento di A che coincide con la sua immagine $x = f(x)$

Esempi:

- la funzione op(opposto) ha un solo punto fisso $\rightarrow 0$
- la funzione identità $\text{id}: A \rightarrow A$ dove $\text{id}(x) = x$ per ogni $x \in A$ ha tutti gli elementi di A come punti fissi

Operazioni

Sia A un insieme; una **operazione n-aria** su A è una funzione $A^n \rightarrow A$ che prende n elementi di un insieme e ci restituisce un elemento dello stesso insieme. (es. $f(x,y,z) = x + y + z$). L'operazione è totale sse la funzione è totale. Qualche esempio è la **divisione intera** su Z, e l'addizione su N.

Immagine inversa

Sia $f: A \rightarrow B$ una funzione e $y \in B$, l'**immagine inversa** di f^{-1} di un elemento y è

$$f^{-1}(y) = \{ x \in A \mid f(x) = y \}$$

E' una relazione, ma in generale *non è una funzione*. Definisce una partizione del dominio. F è iniettiva sse per ogni $y \in B$, $f^{-1}(y)$ ha al più un elemento.

Funzione inversa

Una funzione $f : A \rightarrow B$ è **invertibile** se esiste una funzione $g : B \rightarrow A$ tale che per ogni $x \in A$ e ogni $y \in B$: $g(f(x)) = x$, $f(g(y)) = y$. In questo caso, g è l'inverso di f e si rappresenta come f^{-1} .

Una funzione f è **invertibile** sse è iniettiva, e f^{-1} è una funzione totale sse f è suriettiva.

Composizione di due funzioni

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

Siano $f : A \rightarrow B$ e $g : C \rightarrow D$ due funzioni; la funzione composta $g \circ f : A \rightarrow D$ è definita per ogni $x \in A$ da

$$(g \circ f)(x) = g(f(x)) \quad \begin{matrix} \text{prendo la } f(x) \text{ e la so } \\ \text{sostituisco } \end{matrix}$$

$(g \circ f)(x)$ è definita sse: $f(x)$ e $g(f(x))$ sono definite, e $\text{codom}(f) \subseteq \text{dom}(g)$. $\begin{matrix} \text{al posto della } x \text{ in } g(x) \end{matrix}$

La composizione è 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: $((g \circ f)^{-1} = f^{-1} \circ g^{-1})$

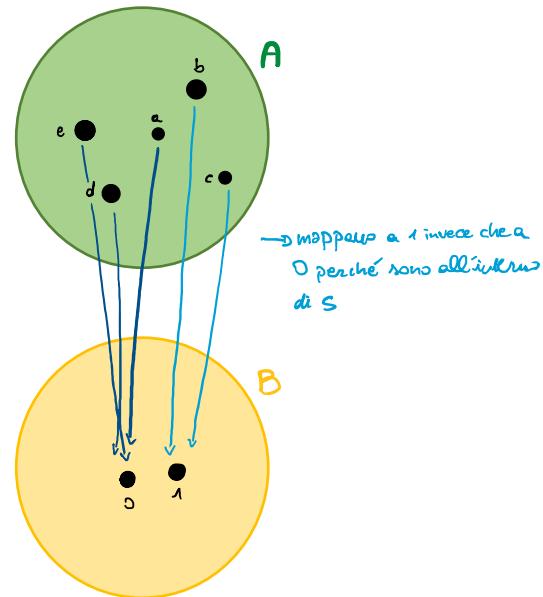
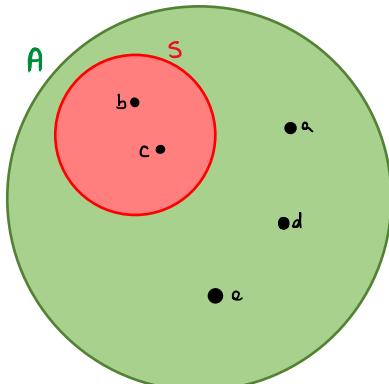
Funzione caratteristica

I sottoinsiemi di un insieme A si possono anche rappresentare tramite una funzione **caratteristica**: dato un insieme $S \subseteq A$, è la funzione:

$$f_S : A \longrightarrow \{0,1\} \text{ dove: } \begin{cases} 0 & \text{se } x \notin S \\ 1 & \text{se } x \in S \end{cases}$$

Un esempio: $A = \{a,b,c,d,e\}$; $S = \{b,c\}$; $S \subseteq A$; $B = \{0,1\}$

X	$f_S(x)$
A	0
B	1
C	1
D	0
E	0



La funzione caratteristica gode di tre proprietà: per ogni $x \in A$:

$$f_{S \cap T}(x) = f_S(x) \cdot f_T(x)$$

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

$$f(x) = 1 * 1 = 0; 1 * 0 = 0$$

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

$$f(x) = 1+1-1*1 = 1; 1+0-0*1 = 1$$

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

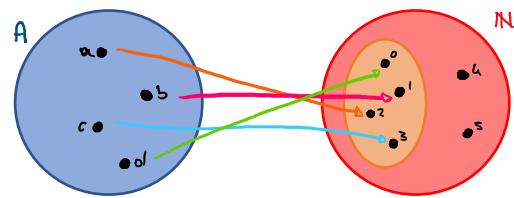
$$1+1-2*1*1 = 0; 1+0-2*0*1 = 1$$

Multinsiemi

Un multinsieme è una variante di insieme dove gli elementi si possono ripetere; è una funzione da un insieme a \mathbb{N} , che esprime quante volte si ripete un elemento nel multinsieme.

$A = \{a, b, c, d\}$; insieme $\{a, a, b, c, c, c\}$
 $f: A \rightarrow \mathbb{N}, f(x) = \{<a, 2>, <b, 1>, <c, 3>, <d, 0>\}$

Elemento	Ripetizione
a	2
b	1
c	3
d	0



Cardinalità

Numeri cardinali

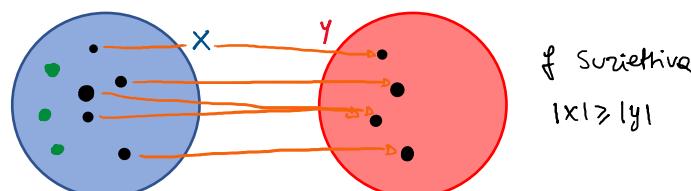
I numeri cardinali sono un tipo generalizzato di numeri utilizzati per misurare gli insiemi, ovvero la loro **grandezza**.

Se un insieme è finito, la sua cardinalità è un numero naturale, ovvero *il numero di elementi dell'insieme*. Con i numeri cardinali, possiamo anche misurare e classificare insiemi infiniti.

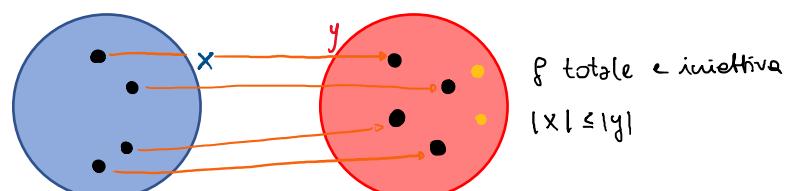
Funzioni e cardinalità

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

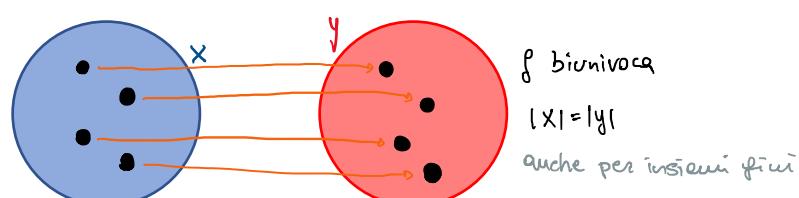
- se f è **suriettiva** allora B non è più grande di A



- se f è **totale e iniettiva** allora A non è più grande di B



- se f è **biunivoca** allora A e B hanno la stessa cardinalità → $A \sim B$



Cardinalità finite

Se A ha k elementi, allora si indica come $A \sim (1, \dots, k)$; in questo caso, si dice che A è finito e ha **cardinalità k** , usando la notazione $|A| = k$.

Si può definire anche la cardinalità dei numeri naturali (Detti cardinali finiti), come: $|N| = \aleph_0$. È il più piccolo dei "cardinali transfiniti", i cardinali per misurare insiemi infiniti, e per ciò la cardinalità di $P(N)$ è: $|P(N)| = 2^{\aleph_0} (= \aleph_1)$.

Un insieme A è numerabile se è equipotenti ad N, e ha cardinalità aleph zero; i suoi elementi possono essere posti in corrispondenza biunivoca con i naturali.

Il continuo

$[0,1] = \{x \in R \mid 0 \leq x \leq 1\}$ non è numerabile; lo possiamo dimostrare attraverso la dimostrazione dell'**argomento diagonale di Cantor**:

- Se fosse numerabile, potremmo elencare tutti i suoi elementi, scrivendoli in forma decimale
 - o $x_1 = 0, a_1 a_2 a_3 \dots$
 - o $x_2 = 0, a_2 a_3 a_4 \dots$
 - o $x_3 = 0, a_3 a_4 a_5 \dots$
 - o Con $a_{ij} \in A$, $A = \{y \in N \mid y < 10\}$ (*tutte le cifre da 0 a 9*)
- Definiamo il numero $z = 0, a_1 a_2 a_3 \dots$ con $a_1 \neq a_{11}, a_2 \neq a_{22}, a_3 \neq a_{33} \dots$; z è *per costruzione* ≠ da ogni numero presente nella lista, e non fa parte di essa

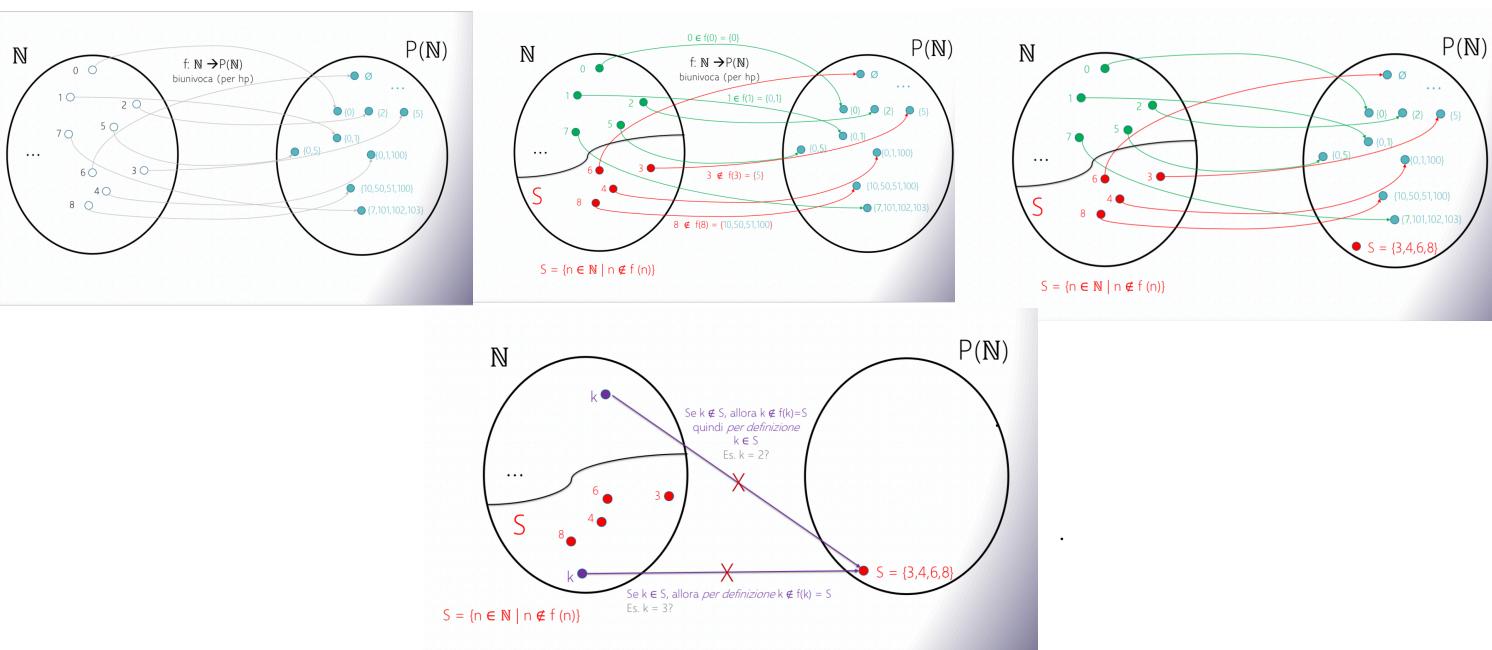
Però, se prendiamo l'insieme iniziale $[0,1]$ e $P(N)$, sono equipotenti (insieme reale vs insieme potenza); siccome $|P(N)| = 2^{\aleph_0}$, allora $|R| \geq 2^{\aleph_0}$; Cantor dimostrò che $\aleph_0 < 2^{\aleph_0}$ ($|A| < |P(A)|$)

La cardinalità dei numeri naturali è strettamente più piccola della cardinalità del suo insieme potenza, e quindi di $|R|$. L'ipotesi del continuo dice che non ci sono transfiniti intermedi tra \aleph_0 e \aleph_1 .

Teorema di Cantor

Vogliamo dimostrare che l'insieme potenza di un insieme ha sempre cardinalità strettamente maggiore di quella di tale insieme, cioè che **non può esistere una funzione biunivoca tra N e P(N)**.

- Supponiamo che esista questa funzione biunivoca (*mappa ogni numero naturale ad un insieme*)
- Definiamo un insieme $S = \{n \in N \mid n \notin f(n)\}$, $f(k) = S$
 - o Se $k \in S$, allora per definizione $k \notin f(k) = S$
 - o Se $k \notin S$, allora $k \notin f(k)$, e quindi per definizione $k \in S$
- La funzione, quindi, **non può esistere**.



Le matrici booleane

Una relazione binaria può essere rappresentata mediante una matrice booleana; la matrice booleana MR associata a una relazione ha n righe ed m colonne, che corrispondono agli elementi del primo e secondo insieme. Si inserisce un **uno** se l'elemento proposto è all'interno della relazione, e **zero** se non lo è.

Relazione

S ₁	Q ₁
S ₁	Q ₂
S ₂	Q ₂
S ₃	Q ₄
S ₃	Q ₅
S ₄	Q ₅

$$R = \{ \langle S_1, Q_2 \rangle, \langle S_1, Q_4 \rangle, \langle S_3, Q_2 \rangle, \langle S_3, Q_5 \rangle, \langle S_3, Q_4 \rangle, \langle S_4, Q_5 \rangle \}$$

Matrice

	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	B
S ₁	0	1	0	1	0	
S ₂	0	0	0	0	0	
S ₃	1	1	0	1	0	
S ₄	0	0	0	0	1	

$$A = \{S_1, S_2, S_3, S_4\}$$

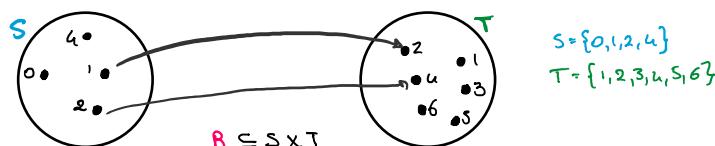
$$B = \{Q_1, Q_2, Q_3, Q_4, Q_5\}$$

RAPPRESENTAZIONI

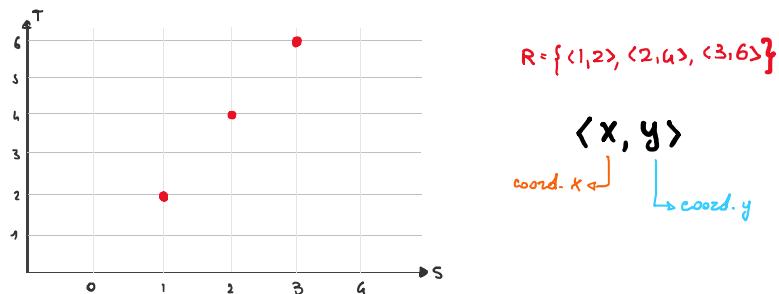
Relazioni

Le relazioni possono essere rappresentate come:

- **Estensionale:** descrive l'insieme di coppie ordinate
- **R = {<1,2>, <2,4>, <3,6>}**
- **Sagittale:** collega con delle frecce gli elementi che verificano la relazione



- **Diagramma cartesiano:** se S e T sono sottoinsiemi di R, rappresenta coppie ordinate come coordinate sul piano cartesiano.



- **Tabella:** nelle colonne si mettono gli elementi dell'insieme di arrivo, nelle righe l'insieme di partenza. In una matrice booleana si usano 0 e 1.

S \ T	1	2	3	4	5	6
0						
1		X				
2				X		
3						X
4						

$$R = \{ \langle 1,2 \rangle, \langle 2,4 \rangle, \langle 3,6 \rangle \}$$

⇒ X in corrispondenza delle coordinate date

- **Grafo** (paragrafo dopo): se una relazione è in S, è *il collasso della rappresentazione sagittale*.

Grafo

Una relazione $R \subseteq S \times S$ è detta **relazione in S**, la cui rappresentazione sagittale collassa ad un grafo. Usiamo lo stesso insieme per l'origine e la destinazione di ogni freccia. Formalmente, un grafo è costituito da **nodi** collegati tra loro da **archi orientati**.

Un grafo è un oggetto definito da un insieme di **nodi** (vertici), collegamenti tra vertici che possono essere:

- Orientati (**archi**, creano **grafi orientati**)
- Non orientati (**spigoli**, creano **grafi non orientati**)

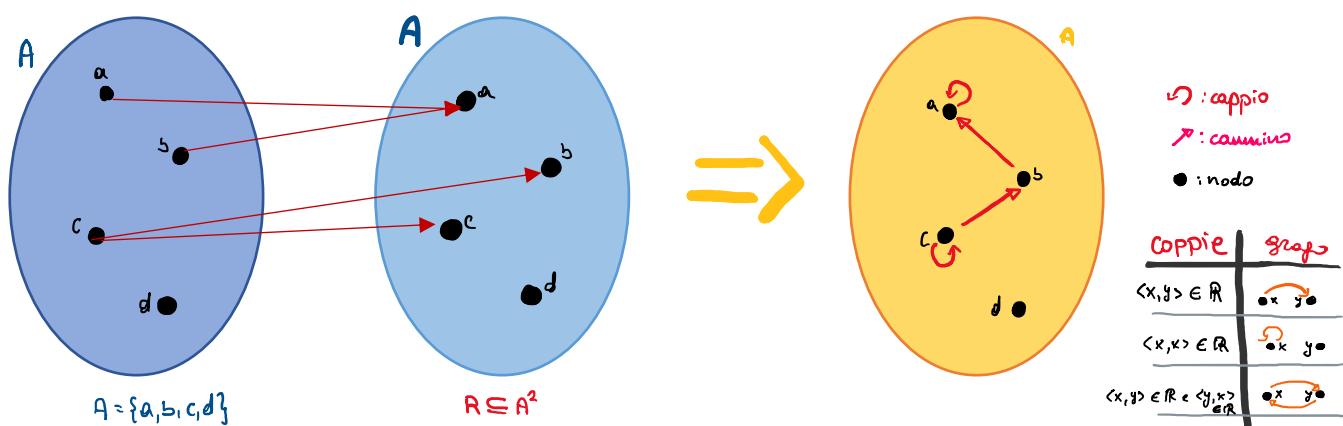
E successivamente anche etichette. Viene rappresentato disegnando **punti** per i nodi, e **segmenti/curve** per i collegamenti tra i nodi. La posizione o forma dei nodi e dei collegamenti è irrilevante.

Formalmente, 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:

$$(v, w) \text{ dove } (v, w) = (w, v) \text{ per ogni } v, w \in V$$

Graficamente, togliamo le frecce agli archi.



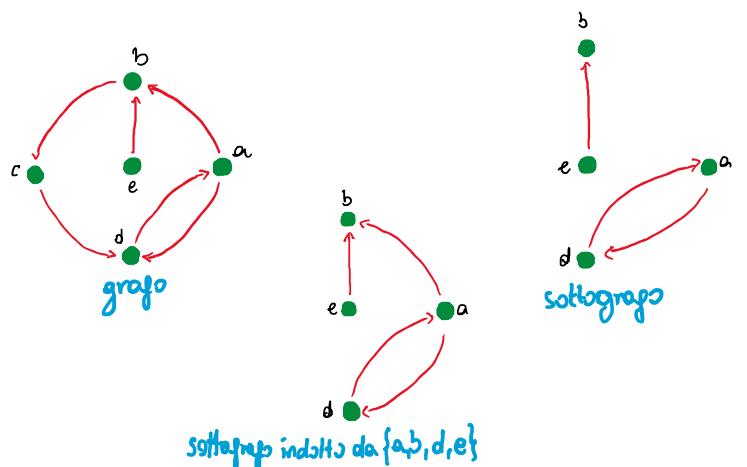
I grafi sono oggetti che descrivono situazioni, collegamenti e relazioni in modo astratto.

$$R = \{\langle a, a \rangle, \langle b, a \rangle, \langle c, b \rangle, \langle c, c \rangle\}$$

Sottografo

Il grafo $G_1 = (V_1, E_1)$ è un sottografo di $G_2 = (V_2, E_2)$ sse: $V_1 \subseteq V_2$ ed $E_1 \subseteq E_2$. Un sottografo si ottiene togliendo nodi e/o archi dal grafo (togliendo un nodo, devo togliere tutti gli archi incidenti).

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



Il problema dei sette ponti di Königsberg

Euler, per dimostrare l'uso dei grafi nella vita reale, crea questo problema, ovvero: è possibile con una passeggiata seguire un percorso che attraversi ogni ponte una e una volta soltanto? Grazie ai grafi, dimostra che non è possibile.

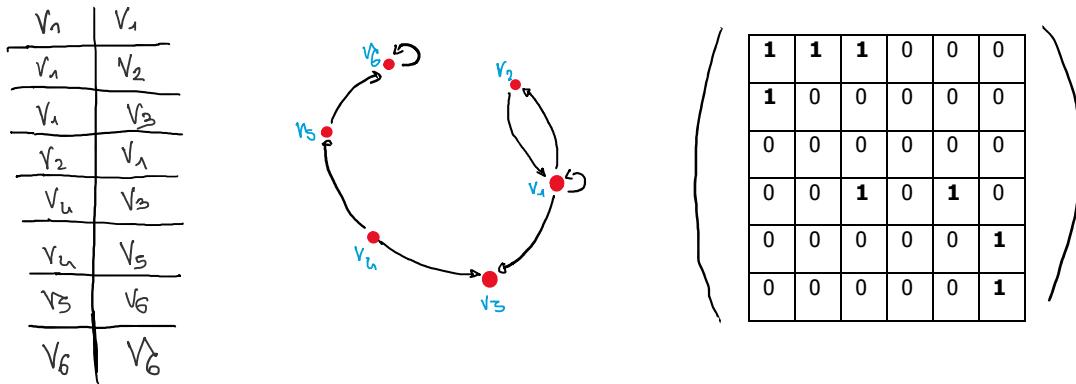
Relazioni binarie

I grafi possono rappresentare, o essere rappresentati da relazioni binarie.

Esempio:

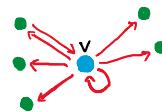
$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\} \subseteq V \times V$$

$$E = \{\langle v_1, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_1, v_3 \rangle, \langle v_2, v_1 \rangle, \langle v_4, v_3 \rangle, \langle v_4, v_5 \rangle, \langle v_5, v_6 \rangle, \langle v_6, v_6 \rangle\}$$



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

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



Un nodo v è chiamato:

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



Un **cammino** è una sequenza finita di nodi, 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 $v_n=w$

Un grafo è fortemente connesso se per ogni coppia di nodi $v, w \in V$ esiste un cammino da v a w .
Esiste sempre un ciclo che visita ogni nodo, e non ci sono né sorgenti né pozzi.

Un **semicammino** è una sequenza finita di nodi, tali che per ogni i , $1 \leq i < n$, esiste un arco che collega v_i e v_{i+1} in direzione arbitraria.

Un grafo è connesso se per ogni $v, w \in V$ esiste un semicammino da v a w .

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

$$v_1 = v \quad \text{O}$$

$$v_3$$



Un (semi)cammino è **semplice** se tutti i nodi nella sequenza sono diversi; un grafo è connesso se esiste sempre un semicammino tra due nodi qualsiasi. Un grafo è connesso se esiste sempre un semicammino tra due nodi qualsiasi.

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

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 .

Algoritmo per determinare le distanze

Serve per ricercare in ampiezza delle distanze da v ad ogni nodo.

Inizializzazione:

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

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 assegna $d(w'') = \infty$

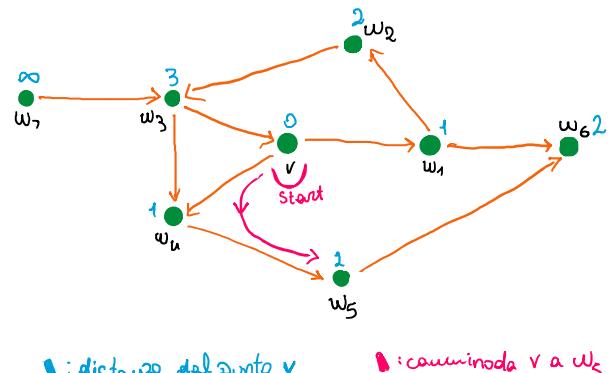
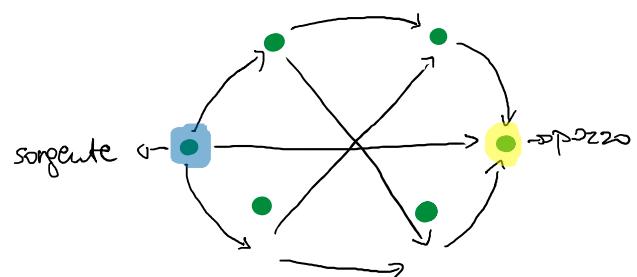


Grafico aciclico orientato (DAG)

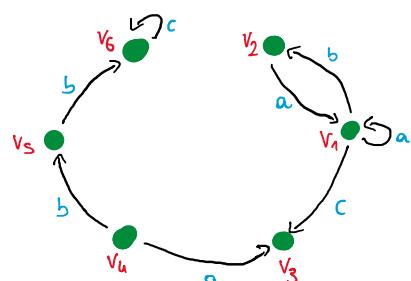
Un grafo orientato senza cicli si chiama *grafo aciclico orientato* (DAG); in un DAG **non esiste** nessun cammino da un nodo a se stesso. Ci devono essere almeno un nodo sorgente, e almeno un nodo pozzo.



Grafi etichettati

Un grafo etichettato è una tripla $G = (V, E, f)$ dove:

- (V, E) è un grafo e
- $f : E \rightarrow L$ è una funzione totale che associa ad ogni arco $e \in E$ un'etichetta da un insieme L . Diamo un'etichetta ad ogni arco della grafo, e sul disegno, posizioniamo l'etichetta accanto ad ogni arco. Un grafo etichettato può rappresentare una **relazione ternaria**, e viceversa.



Matrice di adiacenza

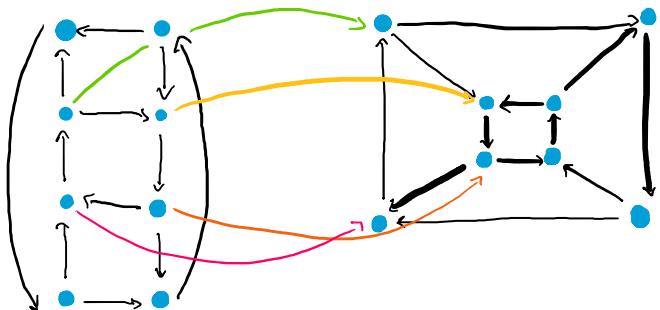
La matrice di adiacenza di un grafo G è la matrice booleana della relazione E ; la matrice di adiacenza di grafi non orientati è **sempre simmetrica**.

Grafo completo

Un grafo completo collega ogni nodo con tutti gli altri nodi, ma non con se stesso. La sua matrice di adiacenza ha zero su tutta la diagonale, ed uno sulle altre posizioni.

Isomorfismi tra grafi

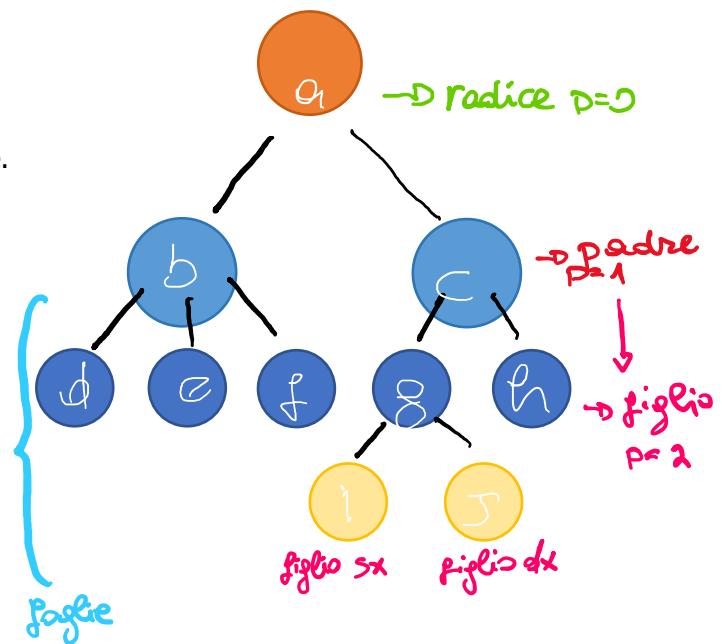
Due grafi $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ sono isomorfi se esiste una **funzione biunivoca** $f: V_1 \rightarrow V_2$ tale che: $\langle v, w \rangle \in E_1 \iff \langle f(v), f(w) \rangle \in E_2$. L'isomorfismo f mantiene al struttura del grafo G_1 , ma sostituisce i nomi dei vertici per quelli di G_2 . Due grafi isomorfi sono in realtà **lo stesso grafo con i nodi rinominati**.



Alberi

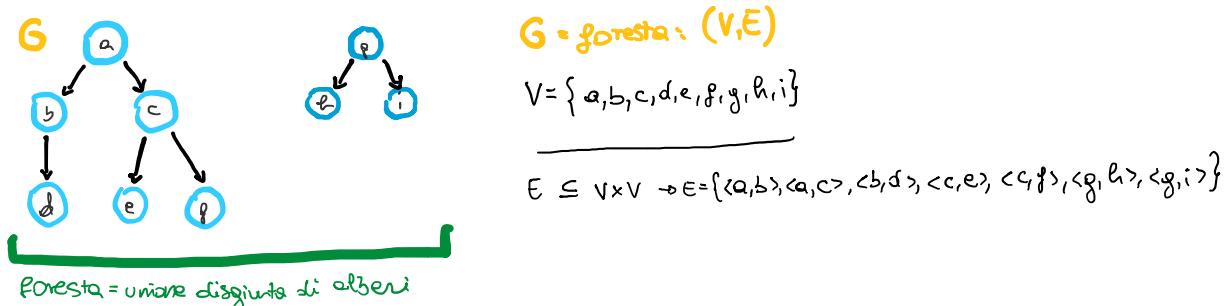
Un albero è un DAG connesso tale che ha esattamente un **nodo sorgente** (chiamato radice dell'albero), e ogni nodo diverso dalla radice ha un solo arco entrante. I nodi pozzi di un albero sono chiamati **foglie**, oppure nodi esterni, mentre gli altri sono interni. Le relazioni tra i nodi usano i nomi di **relazioni familiari**.

- 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
- I nodi intermedi, o interni, sono contemporaneamente padre e figlio
- Se l'albero è rappresentato come una struttura gerarchica, l'ordine è implicito e gli archi si disegnano senza frecce
- Viene chiamato **grado di un nodo** il numero di figli, o di sottoalberi di un nodo.



Foreste e path

Le foreste sono unioni disgiunte di alberi. Ogni albero che compone una foresta ha una propria radice. Si propone un esempio qua sotto:



Un **path graph**, o **linear graph** è un albero in cui tutti gli elementi hanno grado in un'uscita uguale ad **1**, tranne l'unica foglia.

Come rappresentare gli alberi

In un albero c'è esattamente un cammino dalla radice a qualunque nodo v diverso dalla radice. In tale cammino, **ogni nodo w è un ascendente di v**, e **V è discendente di W**. La radice è l'unico nodo **senza ascendenti**.

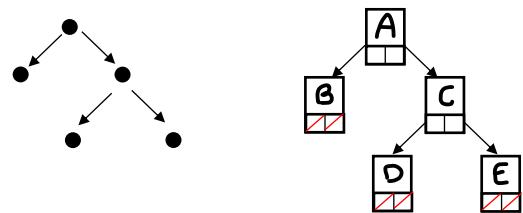
Se il cammino da w a v ha lunghezza 1, allora **w è padre di v**, e **v è figlio di w**.

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.

Alberi binari

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

Un albero binario ha al massimo **2^p nodi** di profondità p; un albero di altezza n ha al massimo $\sum(i=0 \rightarrow n) 2^i = 2^{(n-1)} + 1$ nodi. Un albero binario ha al massimo **2^p nodi** di profondità p; un albero di altezza n ha al massimo $\sum(i=0 \rightarrow n) 2^i = 2^{(n-1)} + 1$ nodi.



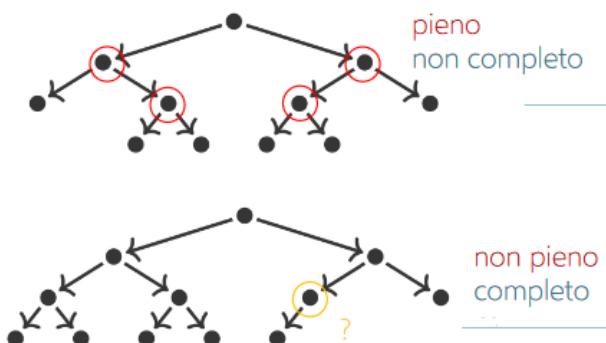
Un albero binario può essere rappresentato come:

- 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, con **n** è l'altezza dell'albero

1	A	1
2	B	1
3	C	1
4	0	0
5	0	0
6	D	1
7	E	1

Un albero binario è:

- **Pieno** (o strettamente binario) se ogni nodo interno ha due figli
- **Completo** se ha altezza n , ad ogni profondità i , $0 \leq i < n$ ci sono 2^i nodi, e l'ultimo livello è riempito da sinistra a destra



R	1	1
Liv 1	2	1
	3	1
Liv 2	4	1
	5	1
	6	1
	7	1
Liv 3	8	0
	9	0
	10	1
	11	1
	12	1
	13	1
	14	0
	15	0

R	1	1
Liv 1	2	1
	3	1
Liv 2	4	1
	5	1
	6	1
	7	1
Liv 3	8	1
	9	1
	10	1
	11	1
	12	1
	13	0
	14	0
	15	0

Possiamo vedere un albero binario come una struttura ricorsiva composta da *un nodo*, un albero binario sinistro e uno destro.

Può essere anche rappresentato con una tabella con $2^{n+1}-1$ righe, dove n è l'altezza dell'albero.

Un albero binario è **bilanciato** se, per ogni nodo v , la differenza fra i numeri di nodi nell'albero sinistro di v e nell'albero destro di v è al massimo 1.

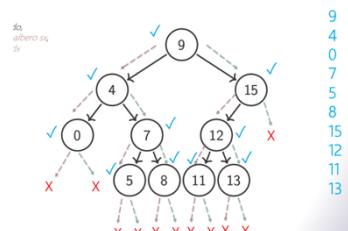
Albero binario di ricerca

Un albero di ricerca è un albero binario $G = (V, E)$ con $V \subseteq Z$ (ovvero qualunque insieme ordinabile) tale che per ogni nodo z :

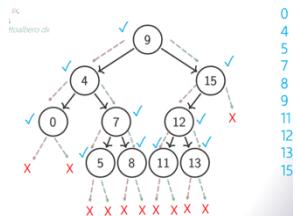
Attraversamento

Un **attraversamento** è un processo che visita tutti i nodi di un albero, di solito in un ordine particolare; un attraversamento che elenca ogni nodo esattamente ogni volta si chiama **enumerazione**. Abbiamo tre tipi:

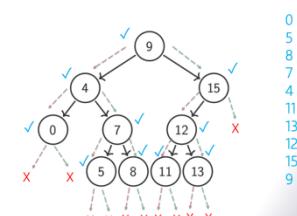
1. **In profondità**: esplora ogni ramo dell'albero fino in fondo (figli prima di fratelli)
 - a. Tre tipi diversi di enumerazione in profondità, basati su quando si enumera un nodo
 - i. V: visita il nodo
 - ii. L: attraversa ricorsivamente il sottoalbero sinistro
 - iii. R: attraversa ricorsivamente il sottoalbero destro
 - b. Ci sono tre tipi di enumerazione:
 - i. **Preordine** (VLR): si enumera un nodo, poi si visita il sottoalbero sinistro e poi il sottoalbero destro



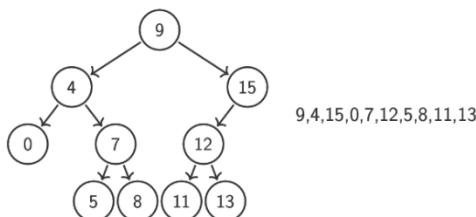
- ii. **Ordine** (LVR): si visita il sottoalbero sinistro poi si enumera il nodo, prima di visitare il sottoalbero destro



- iii. **Postordine (LRV)**: si visita il sottoalbero sinistro, poi il sottoalbero destro, poi si enumera il nodo



2. **In larghezza**: esplora prima i nodi più vicini alla radice (fratelli prima di figli)
3. **Enumerazione in ampiezza**: visita tutti i nodi ad una profondità prima di esplorare altri livelli dell'albero.



Foglie di un albero

Un albero finito ha sempre **almeno una foglia**; per massimizzare il numero di foglie, dobbiamo avere un albero pieno.

Il numero di foglie di un albero pieno con n nodi interni è $n+1$

Th: *Se un albero pieno ha n nodi interni, allora ha n+1 foglie*

1. Caso base ($n=0$)

Un albero non è mai vuoto, quindi un albero senza nodi interni deve avere soltanto foglie. L'albero dev'essere composto di un solo nodo, che è la radice e la foglia allo stesso tempo $\rightarrow \mathbf{0 nodi interni e 1 foglia}$

2. Ipotesi di induzione basata su un numero n

Un albero con n nodi interni ha n+1 foglie

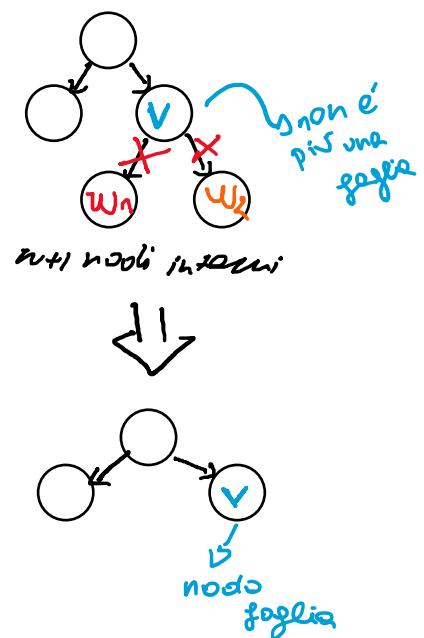
3. Dimostrare il **passo di induzione**: mostrare che è vero anche per $n+1$

Sia $G = (V, E)$ un albero con $n+1$ nodi interni, e $v \in V$ un nodo interno con due figli foglia w_1, w_2 . Il sottografo G' indotto da $V \setminus \{w_1, w_2\}$ è un albero dove v è un nodo foglia (*tolto i figli*), e tutti gli altri nodi interni e foglie rimangono tali.

Il sottografo ha n nodi interni, e per **ipotesi di induzione** ha $n+1$ foglie.

Il numero di foglie di G è esattamente il numero di foglie di G' meno una (v non è più una foglia) più due (i nuovi nodi w_1, w_2).

Generalizzandola:



$n+1 - 1 + 2$ foglie

- Ipotesi di induzione: *il risultato è vero per alberi con massimo n nodi interni. Se un albero pieno ha $m \leq n$ nodi interni, allora ha $m+1$ foglie*
- Passo di induzione: *dimostrare il caso $n+1$, cioè se un albero pieno ha $n+1$ nodi interni, allora ha $n+2$ foglie*
 - o Sia G un albero con $n+1$ nodi interni, almeno un nodo interno e quindi la radice ha due successori
 - o Siano \mathbf{G}_L e \mathbf{G}_R gli alberi sinistro e destro della radice di G , e invece m_L e m_R i numeri di nodi interni
 - Allora $m_L + m_R = n$, e quindi $m_L \leq n$ e $m_R \leq n$.
 - Per l'ipotesi di induzione, \mathbf{G}_L ha $m_L + 1$ foglie, e \mathbf{G}_R ha $m_R + 1$ foglie
 - Se noi sommiamo le foglie dei due, troviamo che ha $n+2$ foglie $\rightarrow G = m_L + m_R + 2$ foglie = $n+2$ foglie

Reticoli

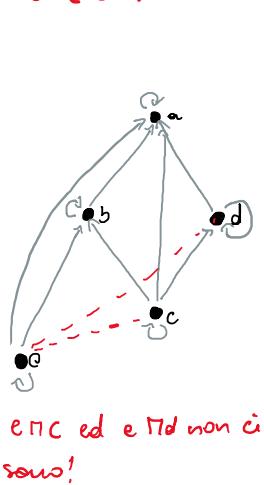
Un reticolo è un tipo particolare di poset (S, \leq) tale che per **ogni coppia $x,y \in S$:**

- Esiste un **minimo maggiorante** $\{x,y\}$
= $x \sqcup y$ (join)
- Esiste un **massimo maggiorante** $\{x,y\}$
= $x \sqcap y$ (meet)

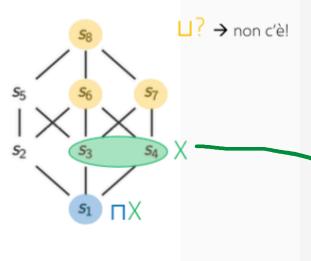
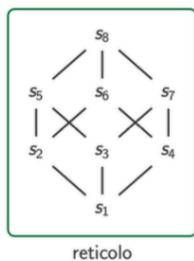
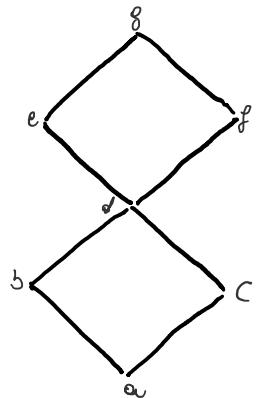
Se nel grafo esiste un cammino tra i nodi x e y ,

- Quello che segue (**in alto nel diagramma di Hasse**) è il join
- Quello che precede (**in basso nel diagramma di Hasse**) è il meet
- Per tutte le coppie $\{x,x\}$, x è sia join sia meet

NON È UN RETICOLO



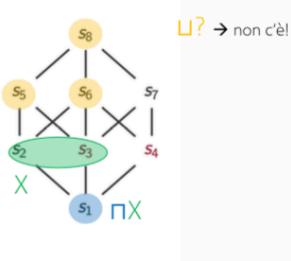
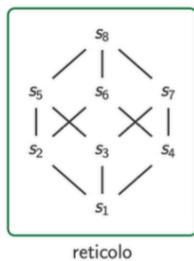
E' UN RETICOLO



risp

maggioranti

minoranti



Se $(L_1, \leq L_1)$ e $(L_2, \leq L_2)$ sono reticoli, allora anche:

$(L_1 \times L_2, \leq_{L_1 \times L_2})$ è un reticolo.

Proprietà

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

- $a \leq a \sqcup b, b \leq a \sqcup b$ ($a \sqcup b$ è maggiorante di a e b)
- $a \sqcap b \leq a, a \sqcap b \leq b$ ($a \sqcap b$ è un minorante di a e b)
- Se $a \leq c, b \leq c$ allora $a \sqcup b \leq c$ ($a \sqcup b$ è il minimo maggiorante di a e b)
- Se $c \leq a, c \leq b$ allora $c \leq a \sqcap b$ ($a \sqcap b$ è il massimo maggiorante di a e b)
- $a \sqcup b = b$ sse $a \leq b$ (se c'è cammino da a a b)
- $a \sqcap b = a$ sse $a \leq b$ (se c'è cammino da a a b)
- **Idempotenza:** $a \sqcup a = a \sqcap a$
- **Commutatività:** $a \sqcup b = b \sqcup a$
- **Associatività:** $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$
- **Assorbimento:** $a \sqcup (a \sqcap b) = a = a \sqcap (a \sqcup b)$

$$a \sqcap b = b \sqcap a$$

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

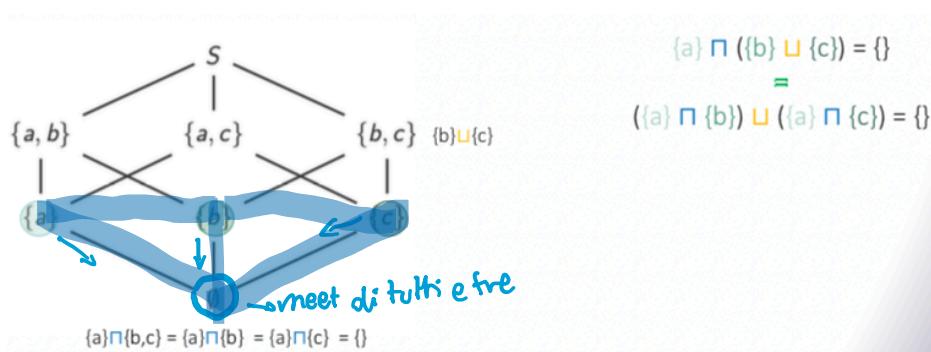
Monotonicità

Il join e il meet sono monotoni, cioè se $a \leq c$ e $b \leq d$, allora $a \sqcup b \leq c \sqcup d$, e lo stesso vale per il meet.

Tipi di reticololi

Un reticolo è:

- **Completo** sse per ogni $M \subseteq L$, M e M esistono.
 - o L'insieme dei sottoinsiemi di un insieme, ordinato secondo la relazione di inclusione.
 - Il **join** è l'unione dei sottoinsiemi, il **meet** l'intersezione dei sottoinsiemi
 - o L'intervallo unitario $[0,1]$ e la retta reale, con la relazione d'ordine totale
 - **Join** e **meet** sono dati dagli estremi
 -
- **Limitato** sse $\underline{1} = L$ e $\underline{0} = L$ esistono, ovvero sse esistono minimo e massimo
 - o Ogni reticolo completo è limitato, e ogni reticolo finito è completo e limitato
- **Distributivo** sse meet e join distribuiscono tra di loro:
 - o $a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$
 - o $a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$



Complemento

Siano (L, \leq) un reticolo distributivo limitato, e $a \in L$;

Un elemento $b \in L$ è il **complemento di a** sse:

$$a \sqcap b = \underline{0} \text{ (minimo)} \quad | \quad a \sqcup b = \underline{1} \text{ (massimo)}$$

Algebra booleana

Boole provò a formalizzare le regole di ragionamento combinando proposizioni in base al loro **valore di verità**, e corrisponde alla prima formalizzazione delle operazioni logiche:

1. **Congiunzione**: 'e', and, \wedge , $\&&$, ...
2. **Disgiunzione**: 'oppure', or, \vee , $\|$, ...
3. **Negazione**: 'non', \neg , ...

Operatori logici vero-funzionali

And, \wedge
Or, \vee
Not, \neg

Boole prima considerò due valori di verità: **vero** (1) o *falso* (0).

Reticolo booleano

È un reticolo (L, \leq) che può essere:

1. **Limitato** sse $\underline{1} = \sqcup L$ e $\underline{0} = \sqcap L$ esistono
2. **Distributivo** sse meet \sqcap e join \sqcup distribuiscono fra loro
3. **Complementato** sse ogni a L ha un complemento a

Un reticolo booleano (L, \leq) definisce l'algebra di Boole: $(L, \sqcup, \sqcap, \neg, \underline{0}, \underline{1})$:

Operazioni

- La disgiunzione è data dal join: operazione binaria
- La congiunzione è data dal meet: operazione binaria
- La negazione è data dal complemento: operazione unaria

Elementi neutri

- $\underline{0}$ è l'elemento neutro per la disgiunzione
- $\underline{1}$ è l'elemento neutro per la congiunzione

Esempio:

Per ogni insieme S, il reticolo $(P(S), \subseteq)$ è un reticolo booleano?

Sappiamo che è un reticolo, è limitato (minimo $\underline{0} = \emptyset$, massimo $\underline{1} = S$), è complementato (il complemento del reticolo equivale al complemento insiemistico).

Per ogni $X, Y \subseteq S$:

1. $X \sqcup Y = X \cup Y$ (elemento neutro O , $T \cup \emptyset = \emptyset \cup T = T$)
2. $X \sqcap Y = X \cap Y$ (elemento neutro S , $T \cap S = S \cap T = T$)
3. $\bar{X} = S \setminus X$

La struttura dipende soltanto da
caratteristiche di S

Algebra di Boole

L'algebra di Boole "tradizionale" è quella definita dal reticolo $(P\{a\}, \subseteq)$, dove gli elementi \emptyset e $\{a\}$ diventano **0** (falso) e **1** (vero).

Reticoli da insiemi potenza

Operazioni logiche

Disgiunzione: join (il massimo fra valori logici) \sqcup
 Interpretazione calcola il **minimo** fra i valori di verità

Congiunzione: meet (il minimo fra valori logici) \sqcap
 Interpretazione calcola il **massimo** fra i valori di verità

Negazione: complemento (l'opposto del valore logico)
 L'interpretazione di \neg complementa il valore di verità

	0	1
0	0	1
1	1	1
\sqcup	0	1
0	0	0
1	0	1

	0	1
0	0	1
1	1	1
\sqcap	0	1
0	0	0
1	0	1

	0	1
0	0	1
1	1	0
\neg	1	0
0	1	0
1	0	1

Implicazione: operazione logica per descrivere situazioni condizionali

Nel caso la seconda sia vera, allora la formula è vera;
 è vera anche nel caso ci siano due zeri o due uno.

Doppia implicazione: $F \leftrightarrow G$ è vera sse i valori di verità **coincidono**

	0	1
0	0	1
1	0	1
\rightarrow	1	1

	0	1
0	0	1
1	0	0
\rightarrow	0	1
1	0	1

Collegamento con la teoria degli insiemi

Non è casuale che le operazioni su insiemi e su valori logici si somiglino; infatti, come abbiamo visto, le **operazioni su insiemi** definiscono un reticolo di Boole. Di conseguenza, le proprietà delle operazioni si mantengono.

- \wedge e \vee sono **idempotenti, commutative e associative**.
- \neg è involutivo: $\neg\neg x = x$
- \wedge e \vee distribuiscono fra di loro (siccome il reticolo è distributivo):
 - o $X \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
 - o $X \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
- Si soddisfano le **leggi di De Morgan**:
 - o $\neg(x \wedge y) = \neg x \vee \neg y$
 - o $\neg(x \vee y) = \neg x \wedge \neg y$

Circuiti logici

Le operazioni booleane si possono implementare fisicamente su **porte logiche**; queste ci permettono di manipolare valori logici in applicazioni molto complesse.

Shannon ha riconosciuto la coincidenza tra il funzionamento dei circuiti commutatori e la logica proposizionale, creando 2 stati di funzionamento: **aperto** e **chiuso**.

AND



OR



NOT



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 deve essere ben fondata.

L'uso autoreferenziale in ricorsione e induzione è utile per:

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

Ricorsione

"Ricorsione" deriva da *ricorrere*; in matematica e informatica, si riferisce alla **definizione di strutture basate in sé stesse**. Come esempi:

- N: $0 \in N$; $\forall n \in N, s(n) \in N$
- Antenato: un genitore è antenato; l'antenato di un antenato è un antenato

La ricorsione ci permette di costruire insiemi basati su qualche proprietà specifica. La definizione richiede i **casi base** e il **ricorsivo** che costruisce nuovi elementi da quelli già presenti. Tipicamente, è basato su una funzione.

Induzione

"Induzione" si riferisce 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**. Come esempi:

- Per tutti $n \in N$, $2n \in N$
- Se il genitore di una persona è una persona, allora tutti gli antenati di una persona sono persone

Dimostrazione per induzione in N (esempio)

Per dimostrare che una proprietà P è vera per ogni $n \in N$, sfruttiamo la definizione ricorsiva di N .

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

Principio di induzione

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**.

Definizione

Una definizione caratterizza e descrive le proprietà che distinguono un oggetto di interesse dagli altri oggetti. Consiste nel verificare, nel senso di mostrarne la ragionevole verità rendendo innegabile un'affermazione.

Come esempi:

- **Pari**: numeri interi divisibili per 2
- **Primi**: numeri divisibili soltanto per 1 e sé stessi

Dimostrazione in logica

Dato un sistema deduttivo (insieme di regole di inferenza), una dimostrazione è una sequenza F_1, F_2, \dots, F_n di formule tale che ogni formula è la conclusione di una regola applicata a formule precedenti.

Per ogni $1 \leq i \leq n$ esiste una **regola di inferenza**

Tale che $\{G_1, \dots, G_m\} \subseteq \{F_1, \dots, F_{i-1}\}$

Assioma

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

La scelta degli assiomi può avere ripercussioni importanti. Come esempio:

La **geometria euclidea** si basa su cinque assiomi:

1. Fra due punti esiste un segmento di retta
2. Ogni segmento di retta si può estendere
3. Per ogni punto e segmento esiste una circonferenza
4. Tutti gli angoli retti sono uguali
5. Data una retta e un punto fuori di essa, esiste soltanto una parallela

Variando l'ultimo assioma, si sono create diverse geometrie.

Ipotesi

Un'ipotesi è una proposizione considerata **temporalmente vera** durante il processo di dimostrazione. È fondamentale per l'induzione, dove ci sono diversi casi da analizzare.

Teorema

Un **teorema** è una conseguenza logica degli assiomi. È una proposizione che è **sempre vera** nella teoria definita da essi. Per essere sicuri che siano teoremi, abbiamo bisogno di una **dimostrazione**.

Teorema in logica

Un teorema è una formula che ha una dimostrazione. La notazione $\vdash F$ esprime che F è un teorema; cioè, F è derivabile nel sistema deduttivo.

Ogni formula appare nella dimostrazione di un teorema è anch'essa un teorema.

Definizioni ricorsive

In generale, una definizione ricorsiva ha bisogno di uno o più **casi base** (base della ricorsione), e una **funzione** per costruire nuovi casi da quelli esistenti. Come esempio: se n è un numero naturale, allora $s(n)$ è un numero naturale.
funzione base

Ordine naturale

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

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

Buon ordinamento

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

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

Funzioni e procedure

Spesso serve la ricorsione per definire funzioni su \mathbb{N} e procedure di calcolo con una struttura ricorrente.

Qualche esempio:

$$- \quad \sum_{k=0}^n k = 0 \rightarrow = k = 0 \quad \sum_{k=0}^n k = \sum_{k=0}^n k + (n+1)$$

- Il fattoriale di un numero $n \in \mathbb{N}^+$ è il prodotto di tutti i naturali positivi fino a n ($n! = 1 * 2 * * * (n-1) * n$) - questa è una definizione ricorsiva.

Codice ricorsivo

Usando pseudo-codice, possiamo descrivere un algoritmo che calcola il fattoriale:

```
function fattoriale(n)
    if n = 0 then return 1
    else
        return n*fattoriale(n-1)
```

La funzione chiama sé stessa, ma sempre con un valore più piccolo. Qui si vede l'importanza del buon ordine della ricorsione.

Implementando funzioni su parole

Possiamo implementare funzioni ricorsive su parole. Immaginiamo che abbiamo una funzione **coda** che data una parola non vuota w ci restituisce la stessa parola senza il primo simbolo.

Alberi binari

Un albero binario è un grafo dove ogni nodo ha al massimo due successori e un predecessore. Spesso, questi alberi sono usati come **strutture dati**, ed è utile pensare ad una definizione ricorsiva.
L'albero vuoto ϵ è un albero binario.

Funzioni su alberi binari

Altezza di un albero:

$$\begin{aligned}\text{alt}(\epsilon) &= 0 \\ \text{alt}((n, t_1, t_2)) &= 1 + \max\{\text{alt}(t_1), \text{alt}(t_2)\}\end{aligned}$$

Numero di nodi di un albero:

$$\begin{aligned}\text{nodi}(\epsilon) &= 0 \\ \text{nodi}((n, t_1, t_2)) &= 1 + \text{nodi}(t_1) + \text{nodi}(t_2)\end{aligned}$$

Induzione:

Se $\text{alt}(t) = n$ allora $\text{nodi}(t) \leq 2^n - 1$

Base: $\text{nodi}(\epsilon) = 0 = 2^0 - 1$

IIG: Supponiamo vero per ogni k , $0 \leq k \leq n$

Passo: dimostriamo che se $\text{alt}(t) = n + 1$ allora $\text{nodi}(t) \leq 2^{n+1} - 1$

Se $\text{alt}(t) = n + 1$, allora $t = (z, t_1, t_2)$ e $\text{alt}(t_1), \text{alt}(t_2) \leq n$.

Per IIG, $\text{nodi}(t_1), \text{nodi}(t_2) \leq 2^n - 1$, quindi

$$\text{Nodi}(t) = 1 + \text{nodi}(t_1) + \text{nodi}(t_2) \leq 1 + 2^n - 1 + 2^n - 1 = 2 * 2^n - 1 = 2^{n+1} - 1$$

Definizioni di sottoinsiemi

Albero bilanciato

- ϵ è bilanciato
- Se t_1, t_2 sono bilanciati e $|\text{nodi}(t_1) - \text{nodi}(t_2)| \leq 1$, allora (n, t_1, t_2) è bilanciato.

Albero binario di ricerca (BST)

- $\epsilon \in \text{BST}$
- Se $t_1, t_2 \in \text{BST}$ e $\max(t_1) < n < \min(t_2)$ allora $(n, t_1, t_2) \in \text{BST}$

Inferenza e ragionamento

L'inferenza è quel meccanismo per cui deduciamo delle conclusioni a partire da un insieme di premesse; è parte essenziale del ragionamento. Alcuni esempi:

1. Mario è un architetto oppure un geometra
Se Mario fosse architetto, allora Mario sarebbe laureato. → *Mario e' un geometra*
Mario non è laureato.
 2. L'assassino ha sporcato di fango il tappeto
 3. Chiunque fosse entrato dal giardino avrebbe sporcato di fango il tappeto → *L'assassino e' entrato dal giardino*

La **logica** è lo studio delle forme del ragionamento, con l'obiettivo principale di riconoscere e individuare le forme dei ragionamenti corretti, e distinguerle da quelli scorretti. Per questo motivo essa viene a volte chiamata logica "formale", o simbolica, perché le *forme del ragionamento* vengono espresse mediante l'uso di simboli, che consentono di astrarre dal contenuto dei ragionamenti stessi.

Astrarre significa "semplificare": sbarazzandosi dei dettagli, si guardano oggetti diversi da uno stesso punto di vista, e gli si può dare un trattamento uniforme.

È uno strumento base per la **meta-informatica**: ha portato a definire la nozione di calcolabilità, ai risultati di indecibilità, alla classificazione dei problemi secondo la loro complessità. Ha fornito all'informatica i concetti fondamentali di **semantica formale** (alla base della semantica denotazione dei linguaggi di programmazione), e metodologie generali come il **trattamento assiomatico di informazioni**.

È inoltre fondamentale per la rappresentazione della conoscenza in **intelligenza artificiale**, dove trovano applicazione i metodi automatici per la dimostrazione di teoremi.

I linguaggi di specifica formale e la specifica algebra di tipi astratti di dati si fondano su metodologie logiche. Altri due campi sono quelli della **risoluzione** e **riduzione**, associati alla **programmazione logica** e alla **programmazione funzionale**.

Sintassi logica

Per iniziare logica, dobbiamo considerare l'algebra di Boole più semplice:

$$B = (\{0,1\}, \leq).$$

Questo è un **reticolo complementato**, dove valgono alcune regole.

$\neg 0=1$	$\neg 1=0$
$1 \sqcap 0 = 0 \sqcap 0 = 0$	$1 \sqcap 1 = 1$
$0 \sqcup 1 = 1 \sqcup 1 = 1$	$0 \sqcup 0 = 0$

Per rappresentare i valori di verità, usiamo **0** (falso) e **1** (vero), e delle operazioni del reticolo per manipolarle:

- \sqcap , "e", \wedge il \sqcap di due valori è vero sse entrambi sono veri
 - \sqcup , "oppure", \vee il \sqcup di due valori è vero sse almeno uno è vero
 - \neg , "no", negazione il \neg di un valore è vero sse il valore è falso

Usiamo delle variabili in B, che si chiamano **proposizioni atomiche**, che non sono altro che **affermazioni** che possono essere vere o false.

- **Atomi**, variabili proposizionali, o variabili agli elementi di A
 - o Per ogni formula ben formata **non atomica** F esiste esattamente un connettivo principale \odot
 - o F + formato dalla applicazione di \odot a una o due fbf, quindi è possibile descrivere ogni fbf come un **albero sintattico**, e viceversa
 - **Letterali** alle formule A e \bar{A} , dove $A \in A$
 - o A è un letterale positivo, $\neg A$ è un letterale negativo
 - **Formule** agli elementi di F

Formule

Una **formula** è un'espressione complessa su B che **combina** diverse proposizioni; anche le formule possono essere vere o false, e questo dipende dalle proposizioni atomiche che la compongono.

Siano A un insieme non vuoto di proposizioni atomiche,

- Op1 un insieme di operatori connettivi unari
- Op2 un insieme di operatori binari

\neg , \wedge , \vee

L'insieme di **formule ben formate** su $(A, \text{Op1}, \text{Op2})$ è definito ricorsivamente da:

- o $A \subseteq F$ Ogni proposizione atomica è una formula ben formata
- o Se $F \in F$, $*$ E Op1, allora $(*F) \in F$ Se $*$ è un operatore unario, allora $*F$ deve appartenere alle formule
- o Se $F_1, F_2 \in F$, \circ E Op2, allora $(F_1 \circ F_2) \in F$ Se \circ è un operatore binario, allora la composizione deve appartenere alle formule
- o Se $F \in F$ allora $(\neg F) \in F$
- o Se $F \in F$ allora $\{(F \wedge G), (F \vee G), (F \rightarrow G), (F \leftrightarrow G)\} \subseteq F$ L'insieme delle operazioni appartiene alle formule

Le formule possono essere circondate da **parentesi**, che esistono per evitare ambiguità. Esiste comunque, in logica proposizionale, una convenzione di ordine di precedenza tra i connettivi (se ci sono gli stessi connettivi, si procede in ordine di scrittura):

\neg \wedge \vee \rightarrow \leftrightarrow

Albero sintattico

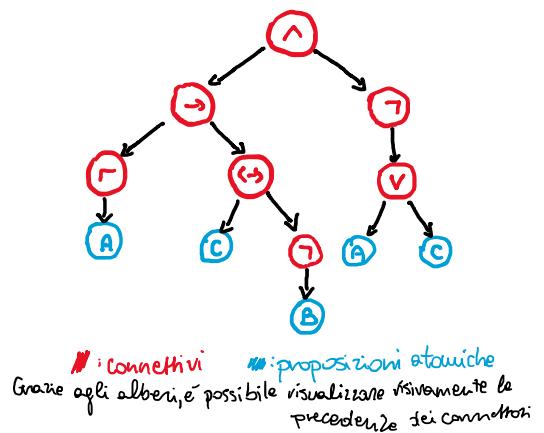
Per ogni formula $F \in F$, l'albero sintattico T_F di F è un albero binario definito da:

- Se $F \in A$, allora T_F è un albero con un solo nodo etichettato: F
- Se $F = (*G)$ con $*$ E Op₁, allora T_F ha un nodo radice etichettato $*$ e un unico successore T_G
- Se $F = (G_1 \circ G_2)$ con \circ E Op₂, allora T_F ha la radice etichettata \circ , e successore sinistro T_{G_1} e successore destro T_{G_2}

Abbiamo le seguenti proprietà:

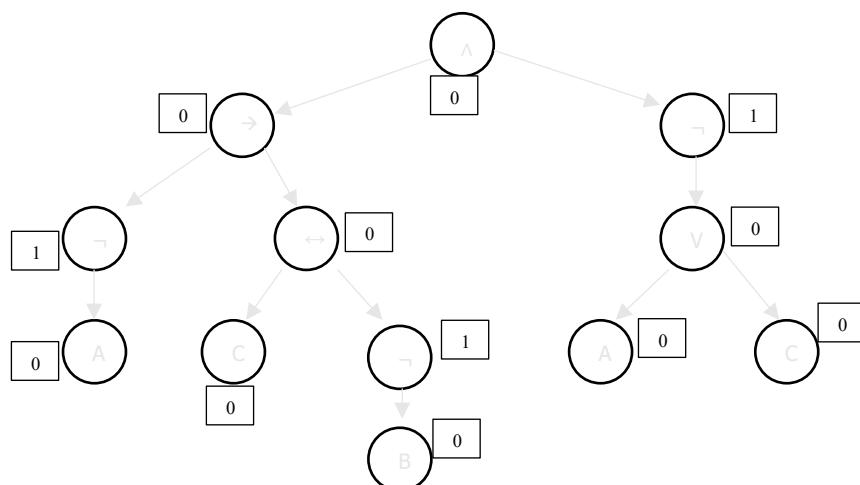
- L'albero sintattico è sempre un **albero non vuoto**
- Le foglie corrispondono sempre a **proposizioni atomiche**
- Tutti gli altri nodi corrispondono a **connettivi**
- Stessa sottoformula può apparire più volte in un albero.

Esempio: $(\neg A \rightarrow (C \leftrightarrow \neg B)) \wedge \neg (A \vee C)$



Propagazione in albero sintattico

La costruzione della valutazione di una formula si può anche capire dalla propagazione sull'albero sintattico. Cominciando dalle foglie assegnate da V, si traversa l'albero verso la radice, applicando la definizione degli operatori.



Assegnazioni e valutazioni

Un'assegnazione booleana è una funzione totale $V: A \rightarrow \{0,1\}$.

V dice quali proposizioni atomiche sono vere, e quali sono false. Data questa interpretazione, possiamo calcolare il valore di verità di ogni fbf.

Data un'assegnazione booleana $V: A \rightarrow \{0,1\}$, la valutazione booleana $I_V : F \rightarrow \{0,1\}$ è l'estensione di V che soddisfa la semantica delle connettive. Come esempio:

$$V(A) = V(B) = V(C) = 0$$

$$I_V(\neg A) = I_V(\neg B) = 1$$

$$I_V(C \leftrightarrow \neg B) = 0$$

$$I_V(\neg A \rightarrow (C \leftrightarrow \neg B)) = 0$$

$$I_V(A \vee C) = 0$$

$$I_V(\neg(A \vee C)) = 1$$

$$I_V((\neg A \rightarrow (C \leftrightarrow \neg B)) \wedge \neg(A \vee C)) = 0$$

Tavole di verità

Quando abbiamo una formula F , dal solito non ci interessa conoscere $I_V(F)$, per una assegnazione V . Vogliamo, invece, sapere il suo "comportamento" tutte le assegnazioni possibili; un metodo per farlo è costruire la sua **tavola di verità**, con una riga per ogni assegnazione e una colonna per ogni sottoformula. Gli operatori si applicano ai valori di verità delle sottoformule; se queste sono atomiche o meno non è rilevante.

Come esempio: $F = (\neg A \rightarrow (C \leftrightarrow \neg B)) \wedge \neg(A \vee C) = G_1 \wedge G_2$

A	B	C	$\neg A$	$\neg B$	$C \leftrightarrow \neg B$	G_1	$A \vee C$	G_2	F
0	0	0	1	1	0	0	0	1	0
0	0	1	1	1	1	1	0	0	0
0	1	0	1	0	1	1	1	1	1
0	1	1	1	0	0	0	1	0	0
1	0	0	0	1	0	1	1	0	0
1	0	1	0	1	1	1	0	0	0
1	1	0	0	0	1	1	1	0	0
1	1	1	0	0	0	1	1	0	0

Equivalenti

Due formule sono **equivalenti** se non sono distinguibili tramite assegnazioni. F e G sono equivalenti, sse per ogni assegnazione V , $I_V(F) = I_V(G)$, ovvero se definiscono la stessa tavola di verità.

Composizionalità

Logica proposizionale

<https://elearning.unimib.it/mod/resource/view.php?id=1020226> È una delle logiche più semplici, e studia formule e valori di verità.

- Unità di analisi simbolica: proposizioni/enunciati
 - o Asserzioni dotate di un valore di verità (vero vs falso)

Mario è un architetto oppure Mario è un geometra.
 Se Mario è un architetto, allora Mario è laureato.
 Mario non è laureato
 quindi Mario è un geometra

$$\begin{array}{l}
 P \vee Q \quad | (P \vee Q) \wedge \neg(P \wedge Q) \\
 P \rightarrow Z \quad | \\
 \neg Z \quad | \\
 Q \quad |
 \end{array}$$

- Il cane abbaia e il gatto miagola
 - Il cane abbaia \wedge il gatto miagola
 - $p \wedge q$
- Se piove, la temperatura si abbassa e diventa umido
 - piove \rightarrow (la temperatura si abbassa \wedge diventa umido)
 - $p \rightarrow (q \wedge r)$
- $\neg p \rightarrow (q \vee (p \rightarrow r))$

Mario è un architetto oppure Mario è un geometra.
 Se Mario è architetto, allora Mario è laureato.
 Mario non è laureato.

→ $p \vee q$
 → $p \rightarrow z$
 → $\neg z$

 → q

Quindi: Mario è un geometra.

Un'assegnazione è una funzione $V \rightarrow \{0,1\}$ che definisce un valore di verità per ogni proposizione atomica. Per rappresentarle, spesso utilizziamo un insieme con tutti gli **atomi veri**.

- $V(A) = V(C) = 1, V(B) = V(D) = 0$ $\{A, C\}$
- $V(A) = V(B) = V(C) = V(D) = 0$ \emptyset *v: funzione caratteristica dell'insieme*

Una formula è **soddisfacibile** se esiste almeno una forma che la può rendere vera. Chiamiamo **modello** un'assegnazione che rende vera la formula, e **contro modello** quello che rende falsa la formula. Si può rappresentare un'assegnazione attraverso gli insiemi.

La logica proposizionale parla di proprietà generali, ma è incapace di parlare di oggetti specifici. Quindi, è anche impossibile sviluppare argomenti logici che dipendono di questi oggetti. Per esempio, non possiamo fare affermazioni esistenziali

Tautologie

In generale, abbiamo che, $\Gamma F \models G$ sse $\Gamma \models F \rightarrow G$, ovvero possiamo *spostare una formula dall'insieme di assiomi alla conseguenza*.

Ripetendo questo argomento per ogni assioma, possiamo arrivare a: $\Gamma \models G$ sse $(\wedge_{F \in \Gamma} F) \rightarrow G$

Per capire se una forma è tautologica, dobbiamo chiederci se ogni assegnazione è un **modello**. F è dunque una tautologia sse $\neg F$ è una contraddizione, e se $\neg F$ non ha modelli.

Per decidere se F è una tautologia:

1. Negare F
2. Provare a costruire un modello di $\neg F$
 - a. Se esiste un modello, F non è tautologica, altrimenti lo è.

Per decidere se F è una contraddizione:

1. Provare a costruire un modello di F
2. Se esiste un modello, F non è una contraddizione, altrimenti lo è.

Relazione di conseguenza

Sia F l'insieme di tutte le forme di una logica; una **relazione di conseguenza** è una relazione che va da sottoinsiemi dell'insieme di formule, $| = \subseteq P(F) \times F$

- $| = \rightarrow$ relazione a che fare con la semantica, e vuol dire che un elemento è una "conseguenza logica" dell'altro
- $\Gamma | = F$ dove $\Gamma \subseteq F, F \in F \rightarrow$ si legge che "F è una conseguenza di Γ "

$M =$ insieme di modelli
 $I_v =$ valore di verità

In logica classica, la relazione di conseguenza è definita da una **classe M di interpretazioni** e una **relazione di soddisfibilità** $| = \subseteq M \times F$.

Dato $M \in M, F \in F$, se $M | = F$ si può dire che **M soddisfa F**, o che **M è un modello di F**.

Nella logica proposizionale, M è l'insieme di **tutte le assegnazioni**. Per $\forall M, V | = F$ sse $I_v(F) = 1$. Quindi, F è una conseguenza di Γ se per ogni assegnazione V che soddisfa tutte le formule in Γ — $I_v(F) = 1$

Come esempio:

- Zeus è — p: umano; q: mortale, r:divinità
 - o $P \rightarrow q$
 - o $\neg(r \wedge q)$
 - o R

P	Q	R	$P \rightarrow q$	$\neg(R \wedge q)$	R
0	0	0	1	1	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	0	1

Una formula è **valida** sse $\emptyset | = F$, cioè se è una conseguenza dall'insieme vuoto di assiomi.

F è valida sse ogni interpretazione soddisfa F, quindi sse **F è una tautologia**

Spostamenti

Cosa vuol dire $F | = G$? Formalmente, per ogni interpretazione M , se $M | = F$ allora $M | = G$. Ovvero, *sempre che F sia vera, G dovrà necessariamente essere vera*.

Dimostrazione di $F | = G$ sse $| = F \rightarrow G$

- Sia V un'assegnazione
 - o Se $I_v(F) = 0$, allora per la semantica $I_v(F \rightarrow G) = 1$ Se antecedente F, implicazione vera.
 - o Se $I_v(F) = 1$, allora $V | = F$ e quindi $V | = G$, cioè $I_v(G) = 1$. Questo significa che $I_v(F \rightarrow G) = 1$
 - o Per il converso, se $V | = F$, come $I_v(F \rightarrow G) = 1$, sappiamo che $I_v(G) = 1$, e quindi $V | = G$

Sistemi deduttivi

Un sistema deduttivo è un insieme di regole per manipolare formule. Le **regole di inferenza**... hanno la formula:

$$\frac{F_1, \dots, F_n}{F}$$

F_1, \dots, F_n sono le premesse della regola e
 F è la sua conclusione
 Se tutte le premesse sono vere, anche la conclusione è vera.

In ogni sistema deduttivo ci sono **assiomi base** che descrivono la semantica e altre proprietà della logica. Essendo validi, non hanno bisogno di premesse per derivarli.

Alcuni esempi:

$$\frac{F \wedge G}{F}$$

$$\frac{F \rightarrow G, F}{G}$$

modus ponens

$$\frac{F \rightarrow G, \neg G}{\neg F}$$

modus tollens

Sia Γ un insieme di formule, e F una formula. F si deriva da sse F p u nteorema del sistema deduttivo ottenuto da aggiungere tutte le formule di come assiomi (in simboli, $\Gamma \vdash F$). Nella dimostrazione di F , quindi, possiamo sempre aggiungere formule in Γ .

Formalizzazione:

- $\Gamma \vdash F$ sse esiste una sequenza F_1, \dots, F_n di formule tale che:
 - o $F_n = F$
 - o Per ogni $1 \leq i \leq n$:
 - $F_i \in \Gamma$
 - Esiste una regola $\frac{\Delta}{F}$ Con $\Delta \subseteq \{F_1, \dots, F_{i-1}\}$

Dimostrazione di F da Γ

Elementi di Γ : premesse ipotesi

Un sistema deduttivo è sempre:

- **Inclusivo:** $C_n(\Gamma) \subseteq F$
 - o posso aggiungere un'altra premessa e possiamo arrivare alla stessa deduzione
- **Monotono:** se $\Gamma \subseteq \Delta$ allora $C_n(\Gamma) \subseteq C_n(\Delta)$
 - o Se noi aggiungiamo una premessa, allora dobbiamo rimuovere dalle deduzioni quello che abbiamo detto prima

Può essere inoltre:

- **Compattezza:** $\Gamma \vdash F$ sse esiste $\Delta \subseteq \Gamma$ finito tale che $\Delta \vdash F$
 - o Se derivo qualcosa, lo posso derivare da un insieme finito di premesse
- **Taglio di premesse:** se $\Delta \vdash F$ e $\Gamma \vdash G$ per ogni $G \in \Delta$, allora $\Gamma \vdash F$
- **Deduzione:** $\Gamma, F \vdash G$ sse $\Gamma \vdash F \rightarrow G$
- **Corretto:** sse per ogni $F \in F, \vdash F \text{ implica } \perp = F$
 - o Un sistema corretto è capace di dimostrare unicamente formule valide, quindi ogni teorema è vero nel sistema logico.
- **Completo** sse per ogni $F \in F, \perp = \text{implica } \vdash F$
 - o Un sistema completo è capace di dimostrare ogni formula valida.

Decidibilità

Quando dimostriamo una correttezza, sappiamo che ogni tautologia può essere dimostrata, ma non sappiamo come trovare la dimostrazione, e quanto lunga sarà.

Sarà importante costruire un **algoritmo** per sviluppare dimostrazioni; se tale algoritmo sempre termina, allora si dice che **la logica è decidibile**.

LA LOGICA DEL PRIMO ORDINE

Algoritmi

Esistono diversi algoritmi per decidere la validità di una formula.

Tableaux

È un metodo di prendere le idee dalle regole di inferenza per decomporre una formula in pezzi più semplici. Si riferisce a una **classe di metodi di decisione** sviluppati per diverse logiche.

La caratteristica principale è che prova a **costruire modelli** di una o più formule. Decomponi formule in sottoformule fino a quando trova un modello, oppure capisce che non ci possono essere modelli.

Per costruire un modello, un tableaux assegna valori di verità alle **sottoformule**, mantenendo la semantica. Dal valore di una formula, deduce quello delle sottoformule, fino ad arrivare ad una assegnazione o una contraddizione. Per rappresentare i valori di verità, si usa $T: \varphi$ e $F: \varphi$, dove φ è una formula.

Esempio:

- Vogliamo un modello per $\neg(p \rightarrow q)$ $T: \neg(p \rightarrow q) \quad | \quad F: p \rightarrow q \quad | \quad T: p, F: q$

- Vogliamo un modello per $p \wedge \neg p$ $T : p \wedge \neg p \mid T : p, T : \neg p \quad [T : p, F : p]$

Le regole del tableaux

Negazione

$$\frac{T : \neg\varphi}{F : \varphi} \qquad \frac{F : \neg\varphi}{T : \varphi}$$

Congiunzione e disgiunzione

$$\frac{T : \varphi \wedge \psi}{T : \varphi, T : \psi} \qquad \frac{F : \varphi \wedge \psi}{F : \varphi \mid F : \psi}$$

$$\frac{T : \varphi \vee \psi}{T : \varphi \mid T : \psi} \qquad \frac{F : \varphi \vee \psi}{F : \varphi, F : \psi}$$

Implicazioni

$$\frac{T : \varphi \rightarrow \psi}{F : \varphi \mid T : \psi} \qquad \frac{F : \varphi \rightarrow \psi}{T : \varphi, F : \psi}$$

$$\frac{T : \varphi \leftrightarrow \psi}{T : \varphi, T : \psi \mid F : \varphi, F : \psi} \qquad \frac{F : \varphi \leftrightarrow \psi}{T : \varphi, F : \psi \mid F : \varphi, T : \psi}$$

Il metodo di tableau:

- Comincia con una formula e un valore di verità associato
- Ad ogni passo, sostituisce una formula con una o due formule, formando uno o due rami
- Si ferma quando tutte le formule sono atomiche.

Un ramo del tableau è aperto sse non ha una contraddizione atomica. I rami aperti rappresentano assegnazioni che garantiscono il valore di verità richiesto.

Il metodo del tableau **termina dopo un numero finito di passi**; la profondità della struttura è limitata dal numero di connettive nella formula. Ad ogni livello, al massimo si duplica il numero di rami; quindi, il tableau è un algoritmo di decisione.

È una tautologia?

1. Si segna la formula come falsa
 - a. Se il tableau è chiuso allora è una tautologia
 - b. Se il tableau non è chiuso allora si chiede f è una contraddizione?

- i. Se il tableau è chiuso allora è una contraddizione
- ii. Se il tableau è aperto allora f_i è soddisfacibile non tautologica.

Descrizione atomica

Bruh

Decomposizione

Per costruire un modello, un tableau assegna valori di verità alle sottoformule mantenendo la semantica. Dal valore di una formula, deduce quello delle sottoformule fino ad arrivare ad una assegnazione o una contraddizione. Per rappresentare i valori di verità, utilizziamo $T : f_i$ e $F : f_i$, dove f_i è una formula.

Alcuni operatori richiedono una **scelta**, e in questo caso basta che una possibilità produca un modello. Dobbiamo cercare dei casi comuni tra tutte le scelte che possiamo fare, e raggrupparle.

V

Vogliamo un modello per $\neg(p \rightarrow q)$

$T : \neg(p \rightarrow q)$
 $F : p \rightarrow q$ deve essere falsa
 $T : p, F : q$ p vera, a falso

$p \wedge (\neg p \vee q)$ $T : p \wedge (\neg p \vee q)$
 $\frac{}{T : p, T : \neg p \vee q}$
 $T : p$ $T : \neg p$ $T : q$ sottopruning
 $T : p, F : p$ |

Chiusura e consistenza

La chiusura deduttiva di un insieme Γ di formule è l'insieme $Cn(\Gamma) := \{F \in F \mid \Gamma \vdash F\}$ di tutte le formule derivabili da Γ .

L'insieme Γ è consistente sse $Cn(\Gamma) \subseteq F$; in un sistema classico, questo equivale alla impossibilità di derivare una formula F e la sua negazione.

Logica dei predicati

Vogliamo un linguaggio logico capace di riferirsi ad oggetti, concetti, proprietà e relazioni, fare affermazioni particolari o universali, potenzialmente con pronomi e aggettivi **indefiniti**. Utilizziamo **costanti**, **variabili** e **quantificatori**.

È una logica di **primo ordine** dove le variabili si riferiscono a individui, e come quantificatori si usano **esiste** e **"per ogni"**, che fanno riferimento alle variabili.

Nella logica dei predicati, si riferisce a degli oggetti, alle loro proprietà e alle loro relazioni.

- **Termini**: nodi per individui di un dominio
- **Predicati**: si riferisce a insiemi di individui e relazioni tra individui

Inoltre, ci si riferisce all'affermazione di proprietà, e a relazioni tra oggetti.

- **Formule atomiche**, per affermare che individui appartengono a insiemi o che le relazioni esistono
 - o $PresidenteDi(b_obama, usa, 2009, 2017) \rightarrow PresidenteDi(x, y, 2009, t)$
- **Formule complesse** via connettivi e quantificatori
 - o $\forall x (\exists y \exists z, PresidenteDi(x, usa, y, z) \rightarrow PresidenteUsa(x))$

La base della conoscenza

KB è un insieme di predicati, predicativi esplicativi di frasi che sono vere (comprese quelle presunte connessioni tra simboli non logici).

KB $\models a - a$ è un'ulteriore conseguenza di quanto si crede.

- Conoscenza esplicita: KB

- Conoscenza implicita: $\{a \mid KB \models a\}$

$$S = \{\text{On}(a,b), \text{ On}(b,c), \text{ Green}(a), \neg \text{Green}(c)\}$$

all that is required

$$\alpha = \exists x \exists y [\text{Green}(x) \wedge \neg \text{Green}(y) \wedge \text{On}(x,y)]$$

Claim: $S \models \alpha$

Proof:

Let \mathfrak{I} be any interpretation such that $\mathfrak{I} \models S$.

Case 1: $\mathfrak{I} \models \text{Green}(b)$.

$$\therefore \mathfrak{I} \models \text{Green}(b) \wedge \neg \text{Green}(c) \wedge \text{On}(b,c).$$

$$\therefore \mathfrak{I} \models \alpha$$

Case 2: $\mathfrak{I} \not\models \text{Green}(b)$.

$$\therefore \mathfrak{I} \models \neg \text{Green}(b)$$

$$\therefore \mathfrak{I} \models \text{Green}(a) \wedge \neg \text{Green}(b) \wedge \text{On}(a,b).$$

$$\therefore \mathfrak{I} \models \alpha$$

Either way, for any \mathfrak{I} , if $\mathfrak{I} \models S$ then $\mathfrak{I} \models \alpha$.

So $S \models \alpha$. QED

Semantica intuitiva

Via **funzione di interpretazione I** su un dominio D (più interpretazioni, ciascuna eventualmente associata a un dominio diverso, sono possibili)

- Esempio interpretazione I_1 su dominio D_1 : l'insieme delle cose che esistono al mondo

Termini: interpretati come **nomi per gli elementi di D**, ovvero denotano elementi in D:

- Esempio: $I_1[b_{\text{obama}}] = \text{Barack Obama}$, $[\text{annoDiNascita}(b_{\text{obama}})] = \text{la data di nascita di Obama}$

Predicati: interpretati come insiemi e relazioni nel dominio D

- Esempio: $I_1[\text{PresidenteDi}(x,y,s,t)] = \text{la relazione che vale in } D \text{ tra individui che sono presidenti di qualche paese, il paese di cui sono stati presidenti, e l'intervallo in cui sono stati presidenti}$

Formule atomiche: interpretate come vere o false in una data interpretazione

- $I_1[\text{PresidenteDi}(b_{\text{obama}}, \text{usa}, 2009, 2017)] = \text{vera se e solo se tra gli oggetti denotati dai simboli "b_{\text{obama}}", "usa", "2009" "2017", vale, in } D_1, \text{ la relazione usata in } I_1 \text{ per interpretare il predicato "presidenteDi"}$

Formule complesse via connettivi e quantificatori

- $I_1[\forall x (\exists y \exists z, \text{PresidenteDi}(x, \text{usa}, y, z) \rightarrow \text{Presidente}(x))] = \text{vera sse dipende dall'interpretazione di PresidenteDi}(x, \text{usa}, y, z)$

Simboli

Il linguaggio della logica dei prediciato è definito da insiemi potenzialmente infiniti di:

- **Variabili** $V = x_1, x_2, \dots, y_1, y_2, \dots$
- **Simboli di costanti** $C = a_1, a_2, \dots, b, \dots$
- **Simboli predicativi** $P = P_1, Q, \dots$ associati ad un'arietà
- **Simboli funzionali** $F = f_1, g, \dots$, associati ad un'arietà

Questi insiemi formano la **segnatura** del linguaggio, e le formule sono costruite tramite i costruttori logici (\neg, \vee) e i quantificatori (\exists, \forall).

Gli insiemi C e F possono essere **vuoti**; V e P non sono mai vuoti; V è **sempre infinito**. Spesso, ma non sempre, pensiamo che abbiamo il predicato di uguaglianza in P; in realtà, le costanti sono un tipo particolare di funzioni.

Arietà

I simboli predicativi e funzionali sono associati ad un'arietà; il numero di parametri, oppure oggetti che manipolano.

L'arietà si rappresenta con un **apice** (P^n – P è un predicato n-ario), oppure un quoiente (p/n). UN esempio è:

- Il predicato di uguaglianza = è binario
- Il predicato genitore_di è binario (genitore_di(ana,bob))
- La funzione matricola_di è unaria: ritorna la matricola di uno studente

Quantificazione

I quantificatori si riferiscono a un **oggetto potenzialmente sconosciuto** (\exists), o a ogni oggetto di dominio (\forall). Le quantificazioni possono essere:

Quantificazione esistenziale	Quantificazione universale
Qualche messicano vive in Italia	Tutti i pianeti orbitano intorno al Sole
Esiste una stella più grande del Sole	Ogni animale è mortale
Ci sono pianeti abitati	Non esistono creature aliene
Lorenzo viaggia con qualcuno	Tutti gli studenti hanno una matricola

Questi ingredienti, insieme, producono espressioni formali che ci aiutano a rappresentare la conoscenza.

- Esiste un numero maggiore di 0 –
- Il successore di qualunque numero naturale è maggiore di 0 –
- La somma di due numeri è sempre maggiore o uguale agli addendi –
- La somma è commutativa –
- La somma di un numero e il successore di un altro è uguale al successore della somma dei due numeri –
- Ogni numero diverso da 0 è il successore di qualche numero –
- Non esiste un numero più grande di tutti gli altri –

esiste un numero maggiore di 0

$$\exists x.(0 \leq x \wedge \neg(0 = x))$$

$$\forall x.\forall y.(x + y = y + x)$$

il successore di qualunque numero naturale è

maggiore di 0

in numero e il successore di un'ala
uale al successore della somma d

$$\forall x.(0 \leq s(x) \wedge \neg(0 = s(x)))$$

$$\forall x.\forall y.(x + s(y) = s(x + y))$$

la somma di due numeri è sempre

maggiore o uguale agli addendi

$$\forall x.\forall y.(x \leq x + y \wedge y \leq x + y)$$

diverso da 0 è il successore di qua

$$\forall x.(\neg(x = 0) \rightarrow \exists y.(x = s(y)))$$

Termini

Data una segnatura L, l'insieme dei **termini** di L, che si riferisce sempre a oggetti, è definito induttivamente da:

- Ogni simbolo di costante e ogni variabile è un termine
- Se t_1, \dots, t_n E Term e f è un simbolo di funzione n-ario, allora $f(t_1, \dots, t_n)$ è un termine – **termine funzionale**

Atomi

L'insieme Atom degli **atomi** è definito induttivamente da:

- T e T rovesciata sono **atomi**
- Se t_1, \dots, t_n E term, e P è un simbolo di predicato n-ario, allora $P(t_1, \dots, t_n)$ è un atomo

Gli atomi parlano di **proprietà**, e possono essere veri o falsi. Quindi, si possono combinare in formule complesse, con attenzione alle variabili.

Formule

Le formule sulla segnatura L sono definite da:

- Ogni atomo è una formula
- Se ϕ è una formula, allora **non ϕ** non è anche
- Se **ϕ e ψ** sono formule, allora lo sono anche: $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$ e $\phi \leftrightarrow \psi$;
- Se **ϕ** è una formula, e $X \in V$, allora (*per ogni* x). ϕ e (*esiste* x). ϕ sono formule

$$\forall y \forall z (\forall x (x \in y \leftrightarrow x \in z) \rightarrow y = z)$$

Qualunque oggetto appartiene a un insieme di oggetti.

Se due insiemi di oggetti hanno gli stessi elementi, allora sono uguali.

presentare in un linguaggio del primo ordine, notiamo che le frasi parlano di "elementi di insiemi" e di "appartenenza":

$$\begin{aligned} \forall x \exists y (x \in y) \\ \forall y \forall z (\forall x (x \in y \leftrightarrow x \in z) \rightarrow y = z) \end{aligned}$$

Precedenza tra gli operatori

Consideriamo la precedenza tra gli operatori stabilita da: $\forall > \exists > \neg > \vee > \wedge > \rightarrow > \leftrightarrow$

Gli operatori, come nel caso proposizionale, associano a destra.

Campo di azione

Il campo d'azione:

- del quantificatore $\forall x$ nella formula $\forall x. \varphi$ è φ
- del quantificatore $\exists x$ nella formula $\exists x. \varphi$ è φ

Esempio

$$\forall x. (\forall y. \neg P(x, y) \rightarrow \exists z. Q(z, w)) \vee \exists w. Q(x, w)$$

Il campo d'azione di $\exists w$ è $Q(x, w)$

Il campo d'azione di $\forall y$ è $\neg P(x, y)$

Il campo d'azione di $\forall x$ è $\forall y. \neg P(x, y) \rightarrow \exists z. Q(z, w)$

Variabili e quantificatori

Una formula può avere variabili senza quantificatori, dove la variabile x è libera, e non possiamo assegnare un valore di verità alla formula.

Se invece ha prima un quantificatore, la variabile è quantificata, e la formula può acquisire un valore di verità.

Variabile di un termine

L'insieme $\text{var}(t)$ delle variabili di un termine t è definito da:

$$\begin{aligned}\text{var}(t) &= \{t\} \text{ se } t \in \mathcal{V} & (t \text{ è una variabile}) \\ \text{var}(t) &= \emptyset \text{ se } t \in \mathcal{C} & (t \text{ è una costante})\end{aligned}$$

$$\text{var}(f(t_1, \dots, t_n)) = \bigcup_{i=1}^n \text{var}(t_i)$$

invece, un termine t è chiuso se $\text{var}(t)$ è l'insieme vuoto, ovvero **se non contiene variabili.**

Variabili libere

Le variabili di un atomo sono:

- $\text{var}(\top) = \text{var}(\perp) = \emptyset$;
- $\text{var}(P(t_1, \dots, t_n)) = \bigcup_{i=1}^n \text{var}(t_i)$

L'**occorrenza libera** di x in una formula è definita da:

- x occorre libera in un atomo φ sse x occorre in φ ;
- x occorre libera in $\neg\varphi$ sse x occorre libera in φ ;
- x occorre libera in $(\varphi \circ \psi)$ sse
 x occorre libera in φ o x occorre libera in ψ ;
- x occorre libera in $\forall z.\varphi$ sse
 $x \neq z$ e x occorre libera in φ ; e
- x occorre libera in $\exists z.\varphi$ sse $x \neq z$ e x occorre libera in φ

Variabili legate

I quantificatori "esiste" e "per ogni" **legano** le occorrenze libere di x nel proprio campo d'azione; un'occorrenza di una variabile x è legata, se non libera.

Le occorrenze legate sono esattamente quelle nel campo d'azione di un quantificatore.

In questo esempio, la variabile x ha due occorrenze legate (dato il quantificatore prima della parentesi), mentre la variabile y ha una occorrenza y libera (quella in Q) e una legata (quella in P)

$$\forall x.(\exists y.P(x, y) \rightarrow Q(x, y))$$

il campo di azione del quantificatore $\exists y$ è $P(x, y)$
(non include $Q(x, y)$)

Una formula phi è **chiusa** sse nessuna variabile occorre libera in phi; le formule chiuse si chiamano anche **enunciati**. La semantica è propriamente definita unicamente per formule chiuse.
Ogni formula chiusa riceve un valore di verità che dipende dagli elementi che la compongono.

Nella **logica proposizionale** si assegna un valore di verità ad ogni variabile proposizionale. Il valore di una formula dipende soltanto da questa assegnazione.

Vogliamo estendere questa idea alla logica dei predicati, ovvero assegnare un valore di verità a ogni atomo, e propagarlo a formule complesse: dobbiamo specificare anche gli **oggetti di interesse**.

Gli atomi **chiusi** ricevono un valore di verità; se il valore di verità include delle variabili, dipende allora dall'**interpretazione** di questi oggetti.

Interpretazione

Un'interpretazione, o struttura del primo ordine, è una coppia formata dal dominio e quantificatore, tale che:

$$\Delta^{\mathcal{I}} := \{\alpha, \beta, \gamma, \delta\};$$

$$\text{ana}^{\mathcal{I}} := \alpha, \quad \text{bob}^{\mathcal{I}} := \beta;$$

$$\text{matricola}^{\mathcal{I}} := \{\langle \beta, \delta \rangle\};$$

$$\text{Professore}^{\mathcal{I}} := \{\gamma\}; \quad \text{Pari}^{\mathcal{I}} := \{\delta\};$$

$$\text{Amico}^{\mathcal{I}} := \{\langle \alpha, \beta \rangle, \langle \beta, \gamma \rangle, \langle \gamma, \beta \rangle\}.$$

$$\text{Professore}(\text{ana}) \quad \text{Amico}(\text{ana}, \text{bob}) \quad \text{Pari}(\text{matricola}(\text{bob}))$$

$\Delta^{\mathcal{I}}$ è un insieme non vuoto chiamato il **dominio** di \mathcal{I} ; e

$\cdot^{\mathcal{I}}$ è una **funzione di interpretazione** che associa:

– a ogni $c \in \mathcal{C}$ un **elemento** $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$;

– a ogni $f/n \in \mathcal{F}$ una **funzione** n-aria $f^{\mathcal{I}} : (\Delta^{\mathcal{I}})^n \rightarrow \Delta^{\mathcal{I}}$;

– a ogni $P/n \in \mathcal{P}$ una **relazione** n-aria $P^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$.

Assegnazione

Data un'interpretazione \mathcal{I} , un'assegnazione (in \mathcal{I}) è una funzione $\eta : \mathcal{V} \rightarrow \Delta^{\mathcal{I}}$

L'assegnazione η associa un elemento del dominio alle variabili in \mathcal{V} ; consideriamo η come una funzione totale.

Data una interpretazione \mathcal{I} e un'assegnazione $\eta : \mathcal{V} \rightarrow \Delta^{\mathcal{I}}$,

l'assegnazione sui termini $\bar{\eta}$ è definita da:

- per $x \in \mathcal{V}$, $\bar{\eta}(x) = \eta(x)$; variabili come in η
- per $c \in \mathcal{C}$, $\bar{\eta}(x) = c^{\mathcal{I}}$; costanti come in \mathcal{I}
- se $f/n \in \mathcal{F}$ e t_1, \dots, t_n sono termini, allora

$$\bar{\eta}(f(t_1, \dots, t_n)) = f^{\mathcal{I}}(\bar{\eta}(t_1), \dots, \bar{\eta}(t_n))$$

Sostituiamo ogni variabile x nel termine per $\eta(x)$

Per abbreviare, scriviamo $t^{\mathcal{I}, \eta}$ invece di $\bar{\eta}(t)$

Soddisfacibilità atomica

Un'interpretazione \mathcal{I} e un'assegnazione η insieme

- associano ogni termine ad un elemento del dominio
 $(t^{\mathcal{I}, \eta} \in \Delta^{\mathcal{I}})$
- determinano un valore di verità per ogni atomo

$$\mathcal{I}, \eta \models P(t_1, \dots, t_n) \text{ sse } \langle t_1^{\mathcal{I}, \eta}, \dots, t_n^{\mathcal{I}, \eta} \rangle \in P^{\mathcal{I}}$$

$\mathcal{I}, \eta \models P(t_1, \dots, t_n)$ si legge
“ \mathcal{I}, η soddisfanno $P(t_1, \dots, t_n)$ ”

L'atomo è vero nell'interpretazione \mathcal{I} sotto l'assegnazione η

Soddisfacibilità delle formule

La soddisfacibilità di una formula si definisce ricorsivamente, nell'interpretazione \mathcal{I} rispetto all'assegnazione

Atomi

- $\mathcal{I}, \eta \models \top; \quad \mathcal{I}, \eta \not\models \perp;$
- $\mathcal{I}, \eta \models P(t_1, \dots, t_n) \text{ sse } \langle t_1^{\mathcal{I}, \eta}, \dots, t_n^{\mathcal{I}, \eta} \rangle \in P^{\mathcal{I}}$

Operatori booleani

- $\mathcal{I}, \eta \models \neg \varphi \text{ sse } \mathcal{I}, \eta \not\models \varphi;$
- $\mathcal{I}, \eta \models \varphi \wedge \psi \text{ sse } \mathcal{I}, \eta \models \varphi \text{ e } \mathcal{I}, \eta \models \psi;$
- $\mathcal{I}, \eta \models \varphi \vee \psi \text{ sse } \mathcal{I}, \eta \models \varphi \text{ oppure } \mathcal{I}, \eta \models \psi;$
- $\mathcal{I}, \eta \models \varphi \rightarrow \psi \text{ sse } \mathcal{I}, \eta \models \neg \varphi \vee \psi;$
- $\mathcal{I}, \eta \models \varphi \leftrightarrow \psi \text{ sse } \mathcal{I}, \eta \models (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi).$

Quantificatori:

- $\mathcal{I}, \eta \models \exists x. \varphi$ sse esiste un $d \in \Delta^{\mathcal{I}}$ tale che
 $\mathcal{I}, \eta[x/d] \models \varphi$;
- $\mathcal{I}, \eta \models \forall x. \varphi$ sse per ogni $d \in \Delta^{\mathcal{I}}$ si verifica
 $\mathcal{I}, \eta[x/d] \models \varphi$.

ome intuizione, pensate che $\mathcal{I}, \eta[x/d] \models \varphi$
sostituisce tutte le occorrenze libere di x in φ per d

Sostituzioni

Sostituzioni

Siano \mathcal{I} un'interpretazione e $\eta : \mathcal{V} \rightarrow \Delta^{\mathcal{I}}$ un'assegnazione

Date $x \in \mathcal{V}$ e $d \in \Delta^{\mathcal{I}}$, l'assegnazione $\eta[x/d]$ è la funzione

$$\eta[x/d](y) := \begin{cases} \eta(y) & \text{se } x \neq y \\ d & \text{se } x = y \end{cases}$$

Modifica l'assegnazione per la variabile x

Interpretazioni

Supponiamo di avere il linguaggio puro dei predicati e il seguente enunciato: $\forall x \exists y P(x, y)$.

Sia D l'insieme degli esseri umani, e P^I l'insieme delle coppie $\langle A, B \rangle$ tali che B è padre di A . Allora, l'enunciato "Tutti gli esseri umani hanno un padre" è la corretta interpretazione dell'enunciato.

Se consideriamo l'interpretazione I' , tale per cui $P^{I'}$ è l'insieme delle coppie $\langle A, B \rangle$, tale che B è madre di A , abbiamo un enunciato analogo: "Tutti gli esseri umani hanno una madre".

Teoria dei grafi

L'insieme di tutti i grafi è trivialmente definito da una segnatura con un solo simbolo di relazione binaria "Arco"; ogni interpretazione deve definire:

- un dominio, ovvero i nodi del grafo
- Una relazione binaria che interpreta Arco, ovvero gli archi
- $\forall x . \exists y. \text{Arco}(x, y) \rightarrow \text{grafo inflessivo}$
- $\forall x y. (\text{Arco}(x, y) \rightarrow \text{Arco}(y, x)) \rightarrow \text{grafo non orientato}$

Teoria delle liste

Sia A un insieme non vuoto di simboli di costanti e sia LA il linguaggio costruito su A e sul seguente alfabeto per le funzioni e i predicati: $cons$ simbolo di funzione binaria, car , cdr simboli di funzioni binarie. $Atom$ e $List$ due simboli di predicato monadici, e nil è un simbolo di costante.

Modelli

L'interpretazione I è un **modello** della formula ϕ , sse per ogni assegnazione η si verifica che $I, \eta \models \phi$. In questo caso, scriviamo $I \models \phi$ e diciamo che **ϕ è vera in I** .

La formula ϕ è **valida**, o tautologica, sse ϕ è vera in ogni interpretazione I . In questo caso, scriviamo $\models \phi$.

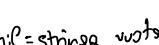
Formule chiuse

Una formula con variabili libere non sarà quasi mai valida. Sarà vera soltanto in condizioni molto particolari, e perciò utilizziamo soltanto formule chiuse. In loro, le variabili hanno un significato nella formula, indipendente dall'interpretazione.

$$I, \eta \models \exists x. \varphi \quad \rightsquigarrow \quad I, \eta[x/d] \models \varphi$$

L'assegnazione originale di x in η diventa **irrilevante** - in una formula chiusa, tutte le variabili sono legate da quantificatori. Possiamo costruire le assegnazioni finali procedendo dall'esterno verso l'interno della formula.

$$\begin{aligned} \forall x (\text{list}(x) \rightarrow \text{cons}(\text{car}(x), \text{cdr}(x)) = x) \\ \forall x (\text{Atom}(x) \rightarrow \text{list}(\text{cons}(x, \text{nil}))) \\ \forall x y (\text{Atom}(x) \wedge \text{list}(y) \rightarrow \text{car}(\text{cons}(x, y)) = x) \\ \forall x y (\text{Atom}(x) \wedge \text{list}(y) \rightarrow \text{cdr}(\text{cons}(x, y)) = y) \\ \text{Atom}(\text{nil}) \wedge \text{list}(\text{nil}) \rightarrow [\text{Atom}(\text{nil}) \in \emptyset, \text{list}(\text{nil}) \in \emptyset] \end{aligned}$$

cost. 
nil  **car(x)**  **cdr(x)**  **list** 
cons(x,y)  **Atom(x)**  **Atom(y)** 
list(cons(x,y)) 
empty 
nil = stringa vuota 
poi ci aggiungerà alpha 

Quantificatori

Quantificatore esistenziale

“Qualche messicano vive in Italia”

“Esiste un individuo che è messicano e vive in Italia”

$$\exists x. (\text{messicano}(x) \wedge \text{vive_in}(x, \text{Italia}))$$

Paragonare con l'espressione

$$\exists x. (\text{messicano}(x) \rightarrow \text{vive_in}(x, \text{Italia}))$$

ha un significato **totalmente diverso**

Quantificatore universale

“Ogni (oggetto che è un) animale è mortale”

“Per ogni oggetto, se è un animale, allora è mortale”

$$\forall x. (animale(x) \rightarrow mortale(x))$$

Invece, l'espressione

$$\forall x. (animale(x) \wedge mortale(x))$$

richiede che nel dominio ci siano soltanto animali

Equivalenza semantica

Due formule phi e gamma sono **equivalenti** sse per ogni interpretazione I: $\mathcal{I} \models \varphi$ sse $\mathcal{I} \models \psi$
Le formule hanno lo stesso valore di verità in qualunque interpretazione.

- $\forall x.\varphi \equiv \neg \exists x.\neg\varphi$; $\exists x.\varphi \equiv \neg \forall x.\neg\varphi$;
- $\exists x.\exists y.\varphi \equiv \exists y.\exists x.\varphi$; $\forall x.\forall y.\varphi \equiv \forall y.\forall x.\varphi$;
- se x non compare in φ , allora $\varphi \equiv \exists x.\varphi \equiv \forall x.\varphi$
- $\forall x.(\varphi \wedge \psi) \equiv \forall x.\varphi \wedge \forall x.\psi$
- $\exists x.(\varphi \vee \psi) \equiv \exists x.\varphi \vee \exists x.\psi$
(i **duali** non si soddisfanno)

Se x non compare libera in ψ , allora:

- $\forall x.(\varphi \vee \psi) \equiv \forall x.\varphi \vee \psi$
- $\exists x.(\varphi \wedge \psi) \equiv \exists x.\varphi \wedge \psi$
- $\forall x.\varphi \rightarrow \psi \equiv \exists x.(\varphi \rightarrow \psi)$
- $\exists x.\varphi \rightarrow \psi \equiv \forall x.(\varphi \rightarrow \psi)$
- $\psi \rightarrow \forall x.\varphi \equiv \forall x.(\psi \rightarrow \varphi)$
- $\psi \rightarrow \exists x.\varphi \equiv \exists x.(\psi \rightarrow \varphi)$

Ordinamenti

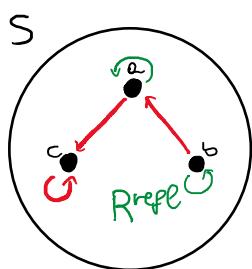
Chiusura riflessiva

Data una relazione R su S, la **chiusura riflessiva** di R è la più piccola relazione riflessiva su S che contiene R:

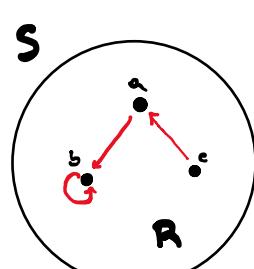
$$R^{\text{refl}} = R \cup I_S$$

$$R \subseteq R^{\text{refl}}$$

$$I_S = \{(x, x) \mid x \in S\}$$



R ^{refl}		
a	b	c
a	1 1	0
b	0 1	0
c	1 0	1



R		
a	b	c
a	0 1	0
b	0 1	0
c	1 0	0

$$S = \{a, b, c\}, R \subseteq S \times S$$

$$R^{\text{refl}} = \{ \langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle, \langle c, a \rangle, \langle c, c \rangle \}$$

$$R \subseteq R^{\text{refl}}$$

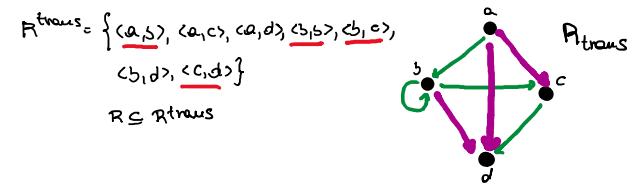
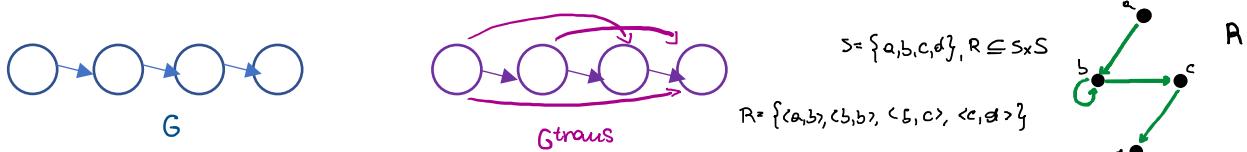
il collegamento è effettuato da riga a colonna; se c'è uno vuoto deve che c'è un cammino da un elemento all'altro

Chiusura transitiva

Data una relazione R su S, la **chiusura transitiva** di R è la più piccola relazione transitiva su S che contiene R:

$$R^{\text{trans}} = \{ \langle x, y \rangle \mid \exists y_1, \dots, y_n \in S \text{ con } n \geq 1, y_1 = x, y_n = y, \langle y_i, y_{i+1} \rangle \in R, i = 1, \dots, n-1 \} \quad R \subseteq R^{\text{trans}}$$

Nella rappresentazione a grafo, dato un grafo G la chiusura transitiva è quel grafo G^{trans} in cui esiste un arco tra i nodi x e y, se esiste un cammino tra x e y in G.



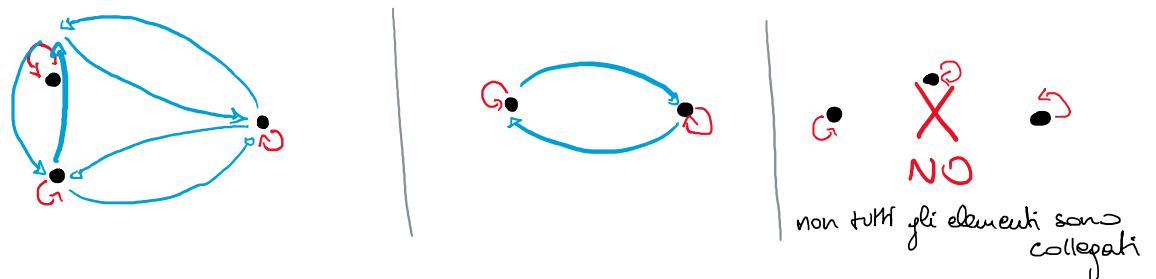
Chiusura riflessiva e transitiva

Data una relazione R su S, la chiusura riflessiva e transitiva di R è la più piccola relazione riflessiva e transitiva refl-trans che contiene R. *Per il grafo, è lo stesso della chiusura transitiva ma con cappi intorno ad ogni elemento.*

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 la uguaglianza tra oggetti; **se una relazione è riflessiva, simmetrica e transitiva**, allora è detta relazione di equivalenza. Dal punto di vista di una proprietà, non esistono differenze tra due elementi.

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



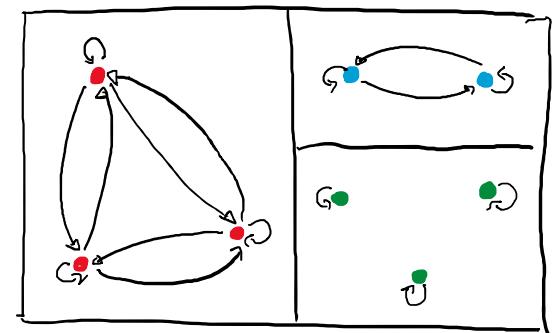
Dividendo S in gruppi i cui elementi sono uguali rispetto ad una proprietà, possiamo studiare insiemi grandi, osservando soltanto pochi elementi. Questi gruppi sono chiamati **classi di equivalenza**.

- Sia S un insieme
- Una **partizione** di S è una famiglia di insiemi $P = \{T_1, \dots, T_n\}$, $T_i \subseteq S$, $1 \leq i \leq n$ tali che:
 - o $T_i \neq \emptyset$ per ogni i , $1 \leq i \leq n$
 - o $T_i \cap T_j = \emptyset$ per ogni $1 \leq i < j \leq n$;
 - o $\bigcup P = S$
- Se R è una relazione di equivalenza su S, allora **T \subseteq S** è una classe di equivalenza se:
 - o $T \neq \emptyset$
 - o Per ogni $x \in S$, $x \in T$ sse $\{y \in S \mid (x, y) \in R\} = T$
- Se R è un insieme e R una relazione di equivalenza su S,
 - o Ogni elemento $x \in S$ definisce una classe di equivalenza: $[x]R = \{y \in S \mid (x, y) \in R\}$
- La famiglia di insiemi $\{[x]R \mid x \in S\}$ che ha per elementi le classi di equivalenza di S, è chiamato **insieme quoziente** di S rispetto a R, e si indica con S/R
- L'insieme quoziente è una **partizione** di S.

Esempio:

Ogni componente连通的 è una classe di equivalenza.

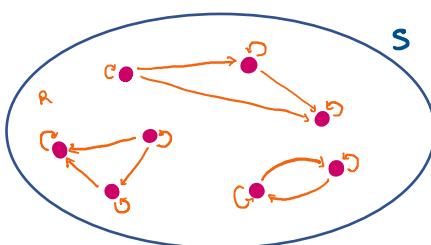
L'insieme quoziente ha 5 insiemi: uno con tre elementi, uno con due elementi, e tre singoletti. La relazione R partizione o divide l'insieme in *classi di equivalenza*.



Ordinamenti

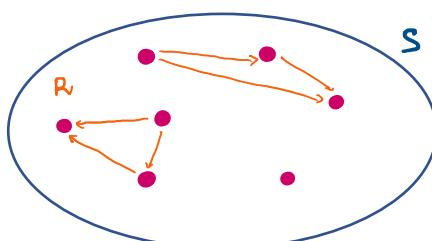
Molto spesso, gli elementi in un insieme hanno una struttura d'ordine; per esempio, N e R. Studiare le proprietà degli ordinamenti è utile per la gestione di dati, ottimizzazione di processi, sviluppo di strutture di dati, studiare/inferire modelli di fenomeni reali. Un ordinamento è un **tipo particolare di relazione** fra elementi che definisce la precedenza.

Una relazione binaria R su un insieme S è un **preordine** sse R è *riflessiva e transitiva* su S.



Ordine stretto

Una relazione binaria R su un insieme S è un **ordine stretto** sse R è *irriflessiva, transitiva (+ asimmetrica)* su S, e si può rappresentare con " $<$ ".

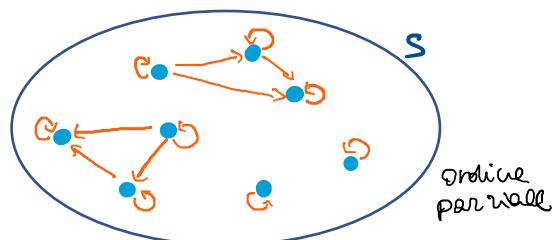


Ordine parziale

Una relazione binaria R su un insieme S è un **ordine parziale** sse R è un preordine antisimmetrico, cioè è *riflessiva, antisimmetrica e transitiva su S* (rappresentabile con \leq).

La coppia (S, \leq) si chiama **insieme parzialmente ordinato**, o **poset**.

- Se $x \leq y$ o $y \leq x$, allora x e y sono *comparabili* (precedenza); altrimenti, sono *incomparabili* ($x \parallel y$).
- Qualsiasi relazione di ordine parziale \leq è la **chiusura riflessiva** dell'ordine stretto associato $<$.

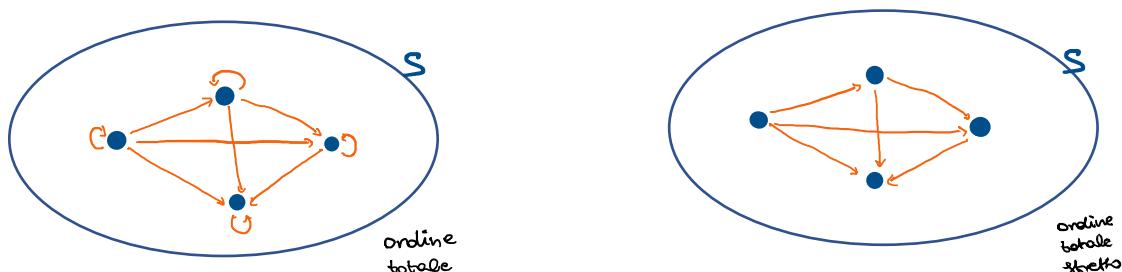


Ordine totale

Un ordine totale è un ordine parziale in cui ogni coppia di elementi è confrontabile: *per ogni $x, y \in S$: $x \leq y$ oppure $y \leq x$* .

Un ordine totale **stretto** è un ordine stretto in cui ogni coppia di elementi è confrontabile: *per ogni $x, y \in S$ tali che $x \neq y$: $x < y$ oppure $y < x$* .

In entrambi i casi la relazione è **connessa**.



Qualche esempio può essere utile per capire meglio il funzionamento di questi:

Esempio	Tipologia	Spiegazione
"Essere nato prima di"	Ordine stretto	Io non sono nato prima di qualsiasi persona
\leq su N	Ordine totale	L'insieme N è completamente ordinabile
\subseteq su insiemi	Ordine parziale	
$<$ su R	Ordine totale stretto	
Chiusura transitiva di un DAG	Ordine parziale	

Relazione fra ordini totali

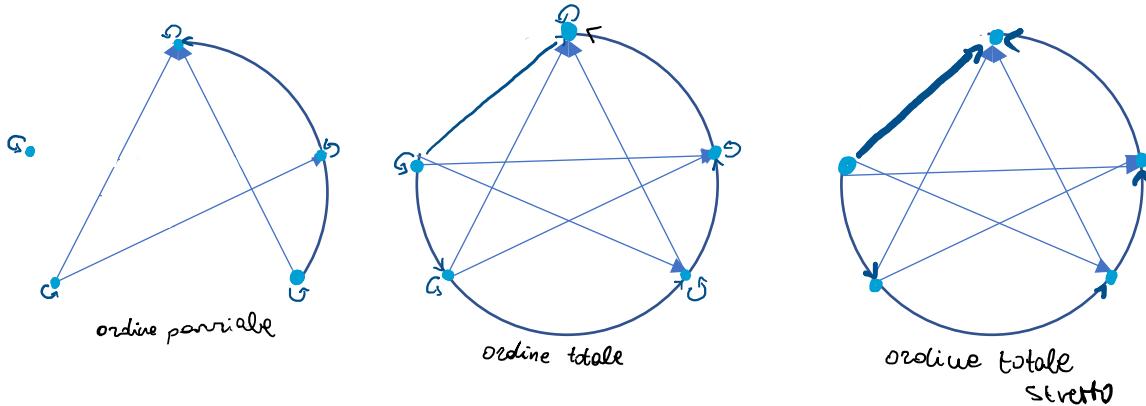
Ordini totali stretti e non-stretti sono ovviamente molto affini.

- Se R è un ordine totale (non-stretto), allora: $R \setminus Is$ è un **ordine totale stretto**
- Se R è un ordine totale stretto, allora: $R \cup Is$ è un **ordine totale**.

Tricotomia

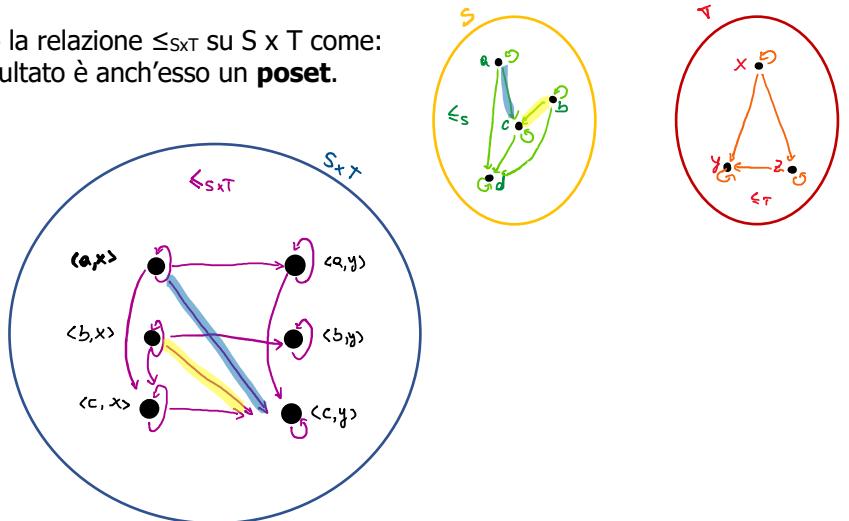
In un ordine totale stretto R per ogni $x, y \in S$ si soddisfa esattamente una condizione:

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



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: $\langle s, t \rangle \leq_{S \times T} \langle s', t' \rangle$ sse $S, \leq_s s', T, \leq_T t'$; il risultato è anch'esso un **poset**.



Ordine lessicografico

L'ordine lessicografico paragona **tuple** di elementi posizione per posizione.

Tupla: generico elemento di una relazione con attributi in un database relazionale
Funzione che a ciascun attributo appartiene ad X associa un valore appartiene al dominio dell' attributo —> **riga di una tabella**

La relazione \leq_{lex} su $S \times T$ è definita da:

$$\begin{aligned} \langle s, t \rangle \leq_{lex} \langle s', t' \rangle \text{ sse: } & s \leq_s s' \\ & s = s', t \leq_T t' \end{aligned}$$

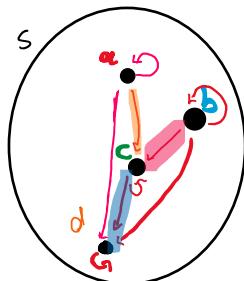
$S \times T, \leq_{lex}$ è un poset, ovvero preserva gli **ordini totali**. Generalizza l'ordine alfabetico usuale e si può estendere a tuple di lunghezza arbitraria.

Copertura

Dato un poset $(S, <)$, e siano x, y e z elementi di S , si dice che **y è una copertura di x sse:**
 $x \leq y; x \neq y$; non esiste z , $z \neq y \neq x$ tale che: $x \leq z, z \leq y$.

Y è quindi una copertura di x sse: y è preceduto da x , e non esiste alcun elemento z che si frapponga nella relazione di ordinamento tra x e y . Nel grafo, per fare in modo che questo ci sia, ci deve essere un cammino di **lunghezza 1 da x a y** . Un esempio esemplificativo sotto forma di grafo:

B non è copertura di a	C è copertura di a	D non è copertura di a
A non è copertura di B	C è copertura di b	D non è copertura di b
A non è copertura di c	B non è copertura di c	D è copertura di c
A non è copertura di d	B non è copertura di d	B non è copertura di d



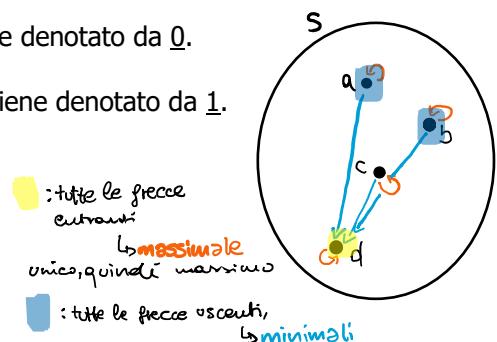
In un ordine totale, ogni elemento taglia al più una copertura, mentre gli altri hanno che ogni elemento può sr.

Elementi estremali

In un poset, un elemento $s \in S$ tale che:

- **Minimale** se non esiste un altro elemento $s' \neq s$ tale che $s' \leq s$, ovvero che nessun elemento lo precede
 - o Se il minimale è unico, allora è il **minimo** del poset e viene denotato da $\underline{0}$.
- **Massimale** se non esiste un elemento $s' \neq s$ tale che: $s \leq s'$
 - o Se il massimale è unico, allora è il **massimo** del poset e viene denotato da $\underline{1}$.

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

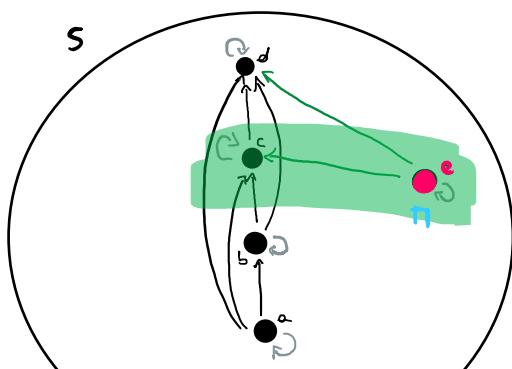


Minoranti e maggioranti

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

- **Minorante** di X sse $s \leq x$ per ogni $x \in X \rightarrow$ precede ogni elemento del sottoinsieme X , ha archi uscenti verso tutti gli elementi di X)
- **Massimo minorante** di X sse $s' \leq s$ per ogni minorante s' di $X \rightarrow$ precede tutti gli altri maggioranti di X , ha archi uscenti verso tutti i maggioranti
 - o Si scrive \sqcap ed è il simbolo di "meet"
- **Maggiorante** di X sse $x \leq s$ per ogni $x \in X \rightarrow$ è preceduto da ogni elemento del sottoinsieme X , ha archi entranti da tutti gli elementi di X
- **Minimo maggiorante** di X sse $s \leq s'$ per ogni maggiorante s' di $X \rightarrow$ precede tutti gli altri maggioranti di X , ha archi uscenti verso tutti i maggioranti
 - o Si scrive \sqcup ed è il simbolo di "join"

Minoranti



Maggioranti

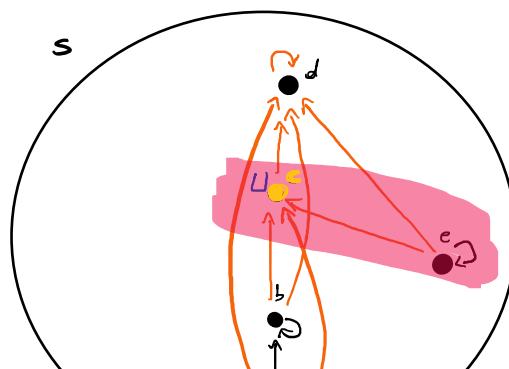


Diagramma di Hasse

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

Dato un poset (S, \leq) , è un **grafo non orientato** tale che per ogni x, y :

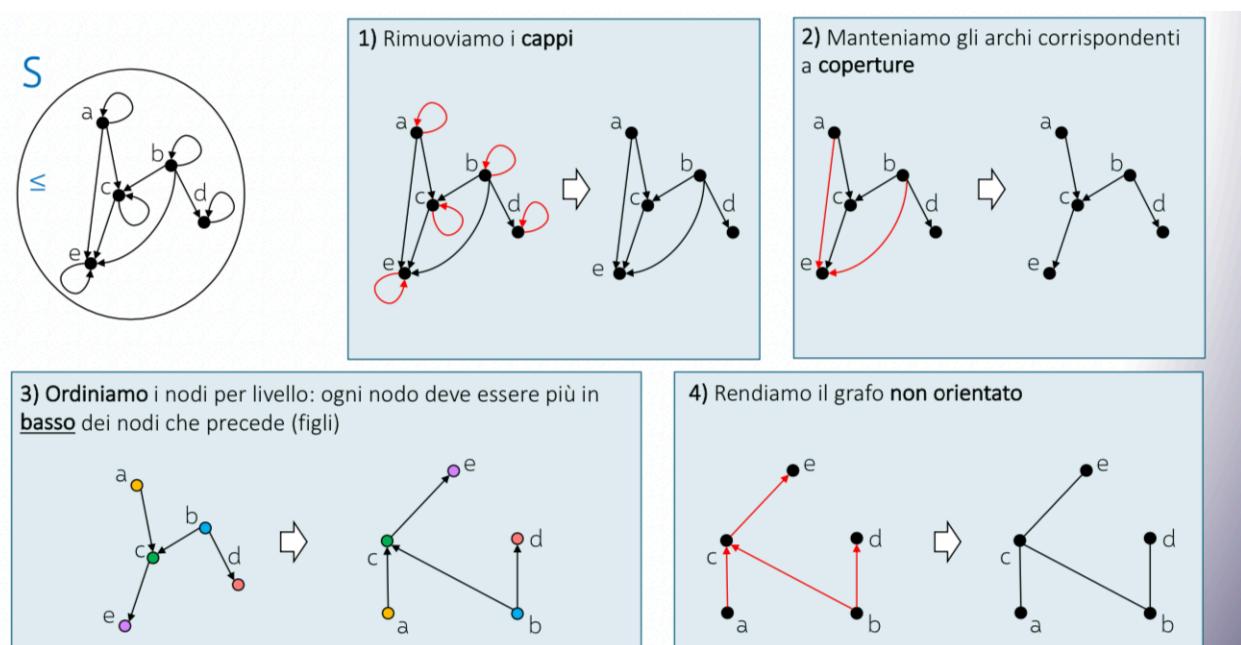
- x e y sono collegati sse y è una copertura di x
- Se $x \leq y$ allora x appare sotto di y (più in basso)

L'ordine corrispondente è la chiusura riflessiva e transitiva del grafo orientato dal basso verso l'alto

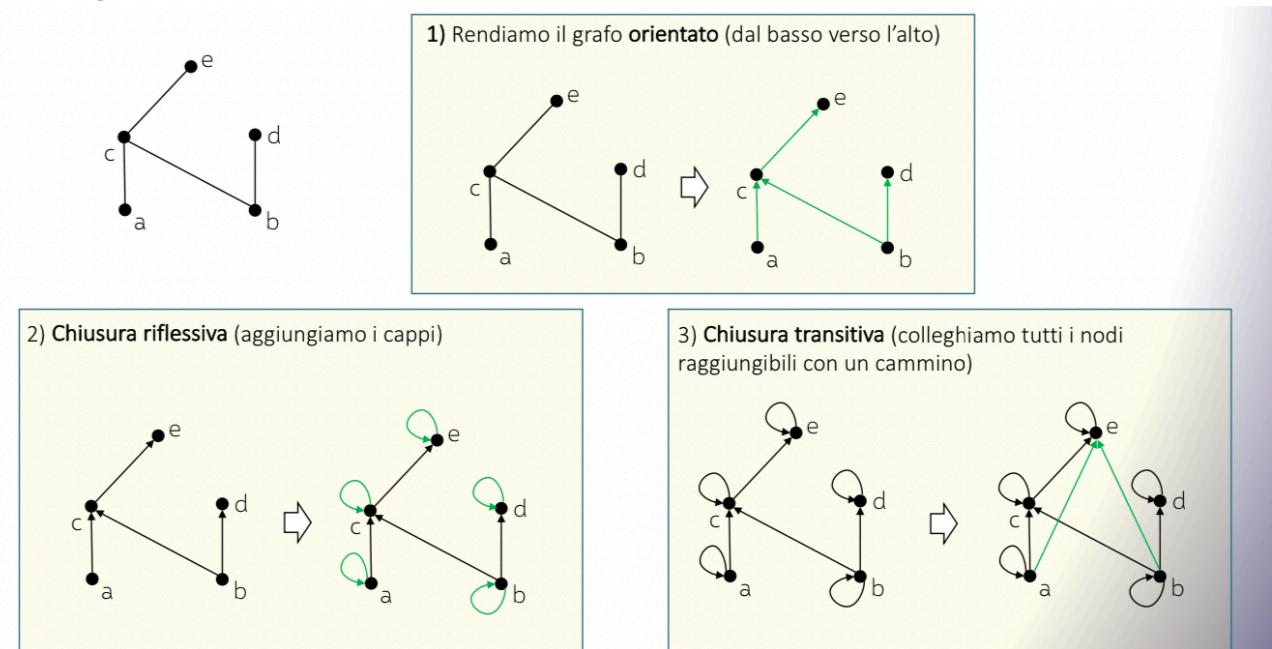
- Se c'è almeno un cammino verso l'alto che collega due nodi, allora quello in basso precede quello in alto
- Due elementi non confrontabili (\perp) non hanno un cammino verso l'alto che li colleghi
- Gli elementi **minimali** sono quelli che non hanno archi provenienti da nodi più in basso (se è unico, è il **minimo**)
- Gli elementi **massimali** sono quelli che non hanno archi verso nodi più in alto (se è unico, è il **massimo**)
- Un **minorante** è un nodo che raggiunge ogni elemento di $X \subseteq S$ con almeno un cammino verso l'alto
 - o Il **massimo minorante** è il minorante più vicino a X
- Un **maggiorante** è un nodo raggiungibile da tutti gli elementi di $X \subseteq S$ con almeno un cammino verso l'alto
 - o Il **minimo maggiorante** è il maggiorante più vicino a X

Il diagramma di Hasse di un **ordinamento totale** formerà sempre una catena; è per questo che gli ordini totali sono anche definiti come **lineari**.

Da poset a diagramma di Hasse



Da diagramma di Hasse a poset



Massimo $\textcolor{blue}{1} = S$ / Minimo $\textcolor{red}{0} = \emptyset$

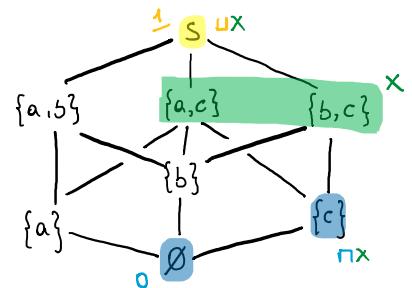
Maggioranti (\sqcup) = $S = \{a, b, c\}$

Minimo maggiorante $\sqcup \textcolor{blue}{X} = S$

Minoranti (\sqcap) = $\{\{c\}, \emptyset\}$

Massimo minorante $\sqcap \textcolor{blue}{X} = \{c\}$

$S = \{a, b, c\} = \{a, c\} \textcolor{blue}{\sqcup} \{b, c\}$ $\{c\} = \{a, c\} \textcolor{blue}{\sqcap} \{b, c\}$

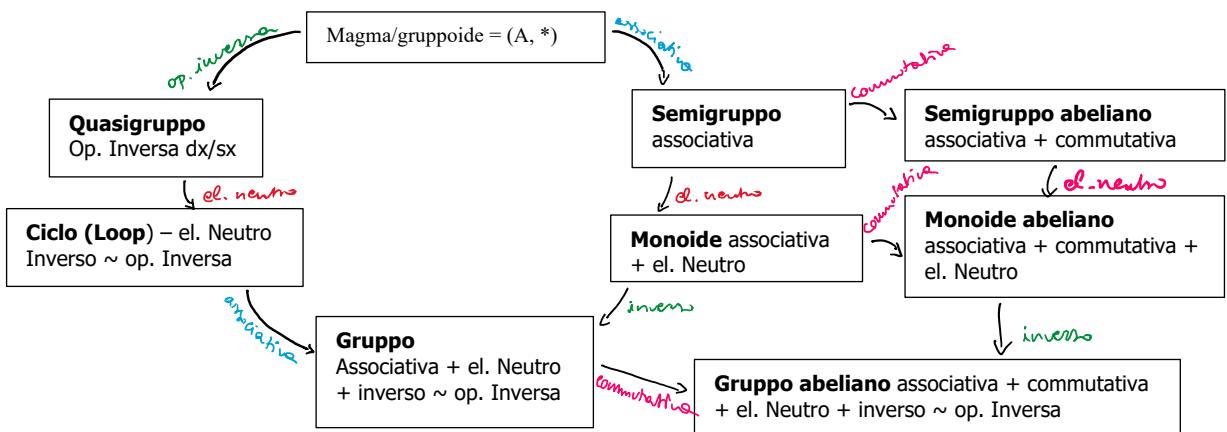


Strutture algebriche

Le strutture algebriche sono costituite da un insieme definito **insieme sostegno** (finito, numerabile, continuo, etc..) e un sistema di **operazioni** sugli elementi del sostegno.

Data un'operazione binaria \circ su un insieme S : $\circ : S \times S \rightarrow S$, tale operazione è:

- **Associativa** sse: $(x \circ y) \circ z = x \circ (y \circ z) = x \circ y \circ z, \forall x, y, z \in S$
- **Commutativa** sse: $x \circ y = y \circ x = \forall x, y \in S$
- **Elemento neutro** è quell'elemento $u \in S$ tale che: $x \circ u = u \circ x = x, \forall x \in S$



Semigruppo

Un Semigruppo è una struttura algebrica (S, \circ) costituita da:

- Un insieme $S \neq \emptyset$
- Un'operazione binaria \circ **associativa**
 - $\circ : S \times S \rightarrow S$, tale che: $x \circ (y \circ z) = (x \circ y) \circ z$
- Se \circ è **commutativa** (cioè $x \circ y = y \circ x$), allora si parla di **Semigruppo abeliano** (o commutativo)

Es. L'insieme dei numeri interi positivi munito dell'addizione; l'insieme numerabile di tutte le stringhe di lunghezza positiva sopra un dato alfabeto A formato da due o più caratteri.

Monoide

Un Monoide è un Semigruppo (S, \circ) dotato di:

- Un **elemento neutro** u
- Se \circ è **commutativa** si parla di *Monoide abeliano*.

Es. L'insieme $\{1, 2, 3, 4\}$ munito dell'operazione "scelta del massimo tra due numeri" (il minimo è l'elemento neutro).

Gruppi

Un elemento inverso x^{-1} è quell'elemento per cui vale: $x \circ x^{-1} = x^{-1} \circ x = u$.

Un **gruppo** è un Monoide in cui per ogni $x \in S$ esiste un elemento inverso; se \circ è commutativa, si parla di **gruppo abeliano**. È definito da:

$$G = (S, \circ, \cdot^{-1}, u)$$

Diagramma di spiegazione:

- Colore blu:
 - Operazione binaria \circ (associativa)
 - Elemento neutro u
- Colore verde:
 - Operazione unaria di inverso \cdot^{-1}

Chiusura

Sia \circ un'operazione sull'insieme S . Un sottoinsieme $X \subseteq S$ si dice **chiuso rispetto a \circ** se: per ogni $b, b' \in X \rightarrow b \circ b' \in X$.

Dato un Semigruppo (S, \circ) e $X \subseteq S$, se X è chiuso rispetto a \circ si definisce **sottosemigruppo** di S .

Congruenza

Dato un Semigruppo (S, \circ) o un gruppo, una relazione di congruenza \cong su S è una relazione di equivalenza che **preserva** \circ . Cioè, per ogni $x, y, z \in S$:

$x \cong y$ implica:

1. $x \circ z \cong y \circ z$
2. $z \circ x \cong z \circ y$

Omomorfismi ed isomorfismi

Siano (x, \circ_x) e (y, \circ_y) due semigruppi. Una funzione $h: X \rightarrow Y$ è detta **omomorfismo** di semigruppi sse:
 Per ogni $x_1, x_2 \in X$ si ha che: $h(x_1) \circ_y h(x_2) = h(x_1 \circ_x x_2) \rightarrow$ in tal caso (x, \circ_x) e (y, \circ_y) sono **omomorfi**.
 Se h è suriettiva, y è detto **immagine omomorfa** di X ; un omomorfismo biettivo tra x e y è un **isomorfismo** di semigruppi.

