

1

SISTEMA FORMALE

↓

APPARATO che vive sulle basi della forma → Non ci sono elementi che non siano presenti qui, ^{nelle forme} qui c'è tutto

- c'è della conoscenza, inguro, che va formalizzata

- dopo che ~~si~~ formalizza, va anche usata

- CONOSCENZA → ULENTE FORMALIZZATA

↓

→ DIRETTA DINAMICA → BASSO TRAPIANCO

↳ ricomporre nuove info da quelle che poi ho

= costruire conoscenza / sistema decidendo nuovi elementi da elementi già formalizzati di più sottimo

- RAPPRESENTAZIONI

↓ rappresentare

elementi di conoscenza

in un sistema di rappresentazione

- ALGORITMI

conventioni di

eseguire quel calcolo

→ calcolare nuovi elementi a partire da questi dati

RAPPRESENTAZIONI "PRIVILEGIATE" → nel corso del ^{secoli} tempo sono diventate importanti e fondamentali di ciò che sostiene un po' tutti i sistemi di oggi: → LA CODICE → linguaggio di rappresentazione di CACCOCI

- CACCOCIARE nuovi elementi e pochi da quelli non
↓ tramite

PROCESSI DI INFERENZA
cioè di RAGIONAMENTO

testo = sistema di rappresentazione

↓
trasforma testo testuale in una rappresentazione testuale

rimozione video di uno spettacolo teatrale diventa una nuova rappresentazione

↓
direttamente
il video
LEAVE

↓
diventa la rappresentazione testuale

Son tutti sistemi di rappresentazione quando faccio un paragrafo da una
cosa o un'alta es: opera teatrale suita → opera messa in scena

MONDO

LA RAPPRESENTAZIONE

da qui posso ricostruire una parte di ^{cos}

annotazioni nomi
~~qualche~~ ~~sono~~ studenti →

rappresentazione classe

territorio →

invento grammatica, convenzione
simboli, colori (blu → acqua)

scantino →

pese

infografiche →

forme, colori, distese linee,
dimensioni em

- alcuni elementi spariscono dalla rappresentazione

↓
il mondo è stato ricco

↓
es: font di come è scritto dramma → a testo sparsa

↓
Si reagiscono gli elementi da trasferire nella rappresentazione

- qualcosa si perde e così si guadagna perché potrai anche arricchirlo

es: riepilogo + speso del testo

↓
u sono tante cose info

la scelta si fa in base a

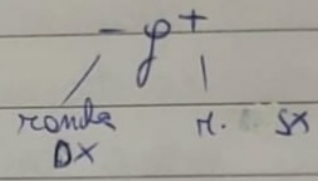
obiettivi nel costruire il modello

es. nel DNSTA: mettere in scena una storia

- posso anche cambiare l'inguegno, e non usare + per ~~le~~ stringe

- grafimi dei resti

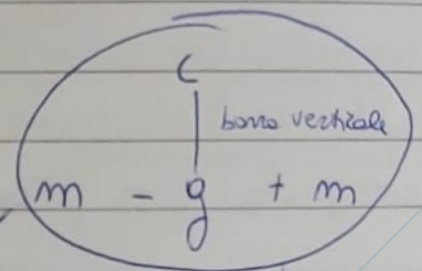
- pilot maniche appennibili:



m - → manica DX

m + → manica SX

m - g + m → pilot con maniche appennate
C - D collettore



SISTEMA DI SIMBOLI DI INCASTRO

→ rappresentazione bidimensionale, non stringe, unica come problema delle caratteri

↓ algoritmo combinatorio C in A

si uniscono simboli uguali:

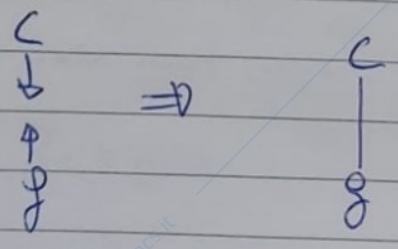
la rappresentazione rappresenta con:

lettere → oggetti

elementi di appoggio → condivisione di un certo elemento (es: - - -)

un incastro, una manica

- - con -
+ con +



(CPU) Central processing unit, unità elaborazione centrale, fa conti prende un linguaggio L e lo esegue

- DISPOSITIVO CENTRALE DI CACCIO - prende dati della memoria e
- memoria centrale - di MASSA

RAM
random access memory

- contiene dati di utilizzo pronto,
- è molto veloce da utilizzare

+ solo => > velocità macchina

disco
dati utilizzati poco frequentemente
sono sul disco fisso

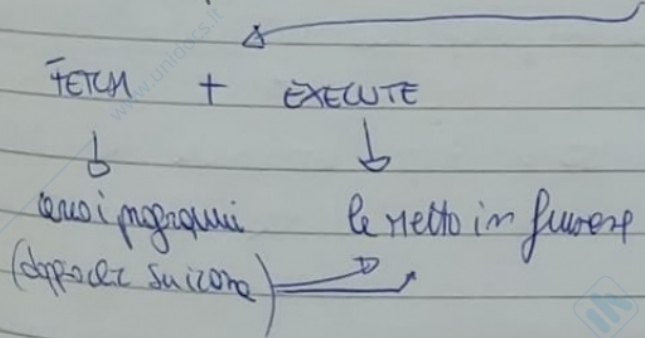
PROMISCUITA -> RAPPRESENTAZIONI E ALGORITMI STANNO NELLA STESSA MEMORIA

- ISTRUZIONI + DATI (cio' che deve essere compilato) // solo che la macchina sa che alcuni sono dati PASSIVI, su cui deve lavorare, e altri DATIVI, come gli algoritmi, che manipolano quella struttura

-> la macchina sa distinguere che sono dati diversi

- l'operazione si chiama FETCH EXECUTE

1.20



- INDIRIZZI -> numeri di celle -> perché le trovare è suddiviso in celle

- ogni cella contiene un dato

- se so l'indirizzo, posso andare e prendere il contenuto

(1) -> celle di memoria, corrisponde a un indirizzo variabile

→ NOMINAZIONE -> diamo nome alle variabile
→ così possiamo mantenerle della memoria

4

con l'algoritmo REACIZO le POSSIBILITA' di CAMBIO

- es: cartina geografica \rightarrow ruota e contiene percorsi

contiene \downarrow
contiene M, nodi, strade A e B \rightarrow e questi oggetti quasi

\downarrow
di cambiamenti del mondo S

\downarrow
desto la sua realizzazione
in un mondo formale che implementa

es. algoritmi: - programmazione logica
- addizione (sue regole)

- Ho tanti ~~algoritmi~~ ALGORITMI per ^{REALIZZARE} lo stesso OGGETTO

□ FORMULAZIONE IN FORMA DI STRUTTURA

- NOTIZIA: ^{MENTI} ISTANTI SIGNIFICATIVI

1^o e 15^o istantanea che ha fatto del mondo
es: P, G, O, A

- ELEMENTI SIGNIFICATIVI

\rightarrow lettere STATUS = COSE IMPORANTI (PRODOTTO, GIUSTO, QUALITÀ)

- ELEMENTI ATTRIBUTI, AGGIUNTI, APPOSITIVI

\rightarrow lettere MINUSCOLE \rightarrow chiavi

aggiungo l'attributo dell'oggetto

IS: P, G, O, A \rightarrow ^{rette opp. esprimo} significativi insieme

es: Giorgio ha gli occhi \rightarrow IS: P, G, O, A

- elementi significativi sono separati da virgola

□ ALGORITMI

- procedure, procedimenti per il CAMBIAZIENTO
 - elementi che ci permettono di cambiare, evolvere
 - il CAMBIAZIENTO può avvenire anche nel mondo
 - es: apertura del ombrello da questo mondo e questo mondo
 - IL CONCETTIVO nei SISTEMI è l'ALGORITMO → l'ESECUZIONE dello ALGORITMO
- ▷ "SERIE DI ISTRUZIONI" CHE REALIZZANO CAMBIAZIENTI

representazione R



algoritmo



representazione R'

ci vogliono dei comandi che vanno eseguiti
 - controllare rapporto da considerare dell'ambiente
 - disegnarne le istruzioni
 ALGORITMO: SPOSTARE/UNO STUDENTE
 CAMBIAZIENTO → simulare azione

es. algoritmo che cambia di posto gli studenti

questo rappresentando il mondo, per rappresentare quel cambiamento che sta avvenendo lì

- con l'algoritmo sto manipolando la parte di (IT) della representazione
 arco di simulare la parte di (IT)

- ALGORITMO è IN POTENZA e CAMBIAZIENTO,
- ci vuole un agente, computer, che lo mette in pratica
 umano, macchina

- quando vedo e costruire una rappresentazione, mi baso su una struttura,
 elementi di rappresentazione, che rappresentano qualcosa
- l'algoritmo cambia manipola quegli elementi (non elti) e se manipolerei

- Il linguaggio deve essere leggibile dalle macchine

PCU \rightarrow ^{metodi} colui:

ingreso - (memorie input da esterno (dati da tastiera, microfono etc))
 uscita - output \rightarrow autoportanti, schermi, cuffie

PCU \rightarrow memorie dati da

- MEMORIA
 - ESTERNA (RAM)

$\uparrow \rightarrow$

CI ERABORA,
 CI SCRIVE IN MEMORIA,

FA USARE I DATI
 SU OUTPUT

Lo legge la mente una PCU (quando scriviamo un algoritmo o operando facciamo una trasposizione), quindi dobbiamo fare un agc che è compatibile con la PCU