

fac simile

es 1 - logica

1. se la macchina è rotta allora si fa aggiustare
2. se la macchina è rotta o non è rotta allora si va dal meccanico
3. non si va dal meccanico oppure il conto è salato
4. se si fa aggiustare la macchina si paga un conto non salato

dimostrare se si paga un conto salato o non salato indicando la sequenza di formule per arrivare alla formula finale

risoluzione

MR = rotta | MA = aggiustare | MM = meccanico | CS = conto salato |

S = {

1. $MR \rightarrow MA$
2. $MR \vee \text{not}(MR) \rightarrow MM$
3. $\text{not}(MM) \vee CS$
4. $MA \rightarrow \text{not}(CS)$

}

goal $\rightarrow CS \vee \text{not}(CS)$

passaggi

1. da 2 con tautologia ($P \vee \text{not}(P) = \text{vero}$)
 $MR \vee \text{not}(MR) = \text{vero}$
2. da S
 $MR \vee \text{not}(MR) \rightarrow MM$
3. da P1, P2 con modus ponens
MM

4. da S
not(MM) \vee CS = MM \rightarrow CS
5. da P3, P4 e modus ponens
CS

es 2 - codice prolog

data una lista l1 di interi e un intero n, scrivere un predicato prolog "domanda 1" di l1, n, l2 che restituisca in l2 la lista degli elementi di l1 che sono liste contenenti solo due valori interi la cui somma valga n

esempio: domanda([3,1], 5, [2,1,1], [3], [1,1,1], e, [2,2]), 4, l2)

risoluzione

```
domanda1([], N, []). //se la l1 è vuota -> l2 sarà vuota

domanda1([[A,B]|Xs], N, [[A,B]|Ys])
    :- (A+B) is N, domanda1(Xs, N, Ys)
//se la testa di l1 è da due elementi la metto
//dentro a l2 MA solo se la somma fa N
//e dopo richiamo la stessa funzione ma sulle code
//(quindi procedo sulle liste)

domanda1(_|Xs, N, Y) :- domanda1(Xs, N, Y)
//se la testa NON è composta da due elementi
//(oppure il caso prima sulla somma non è
//verificato) allora richiami la stessa funzione ma sulle code
//(quindi procedo sulle liste)
```

es 3 - albero sld right-most

```
cl1: classe(mammiferi, cani)
cl2: classe(animali, mammiferi)
```

```

c13: super(X,Y) :- classe(X,Y)
c14: super(X,Z) :- classe(W,Z), super(X,W)

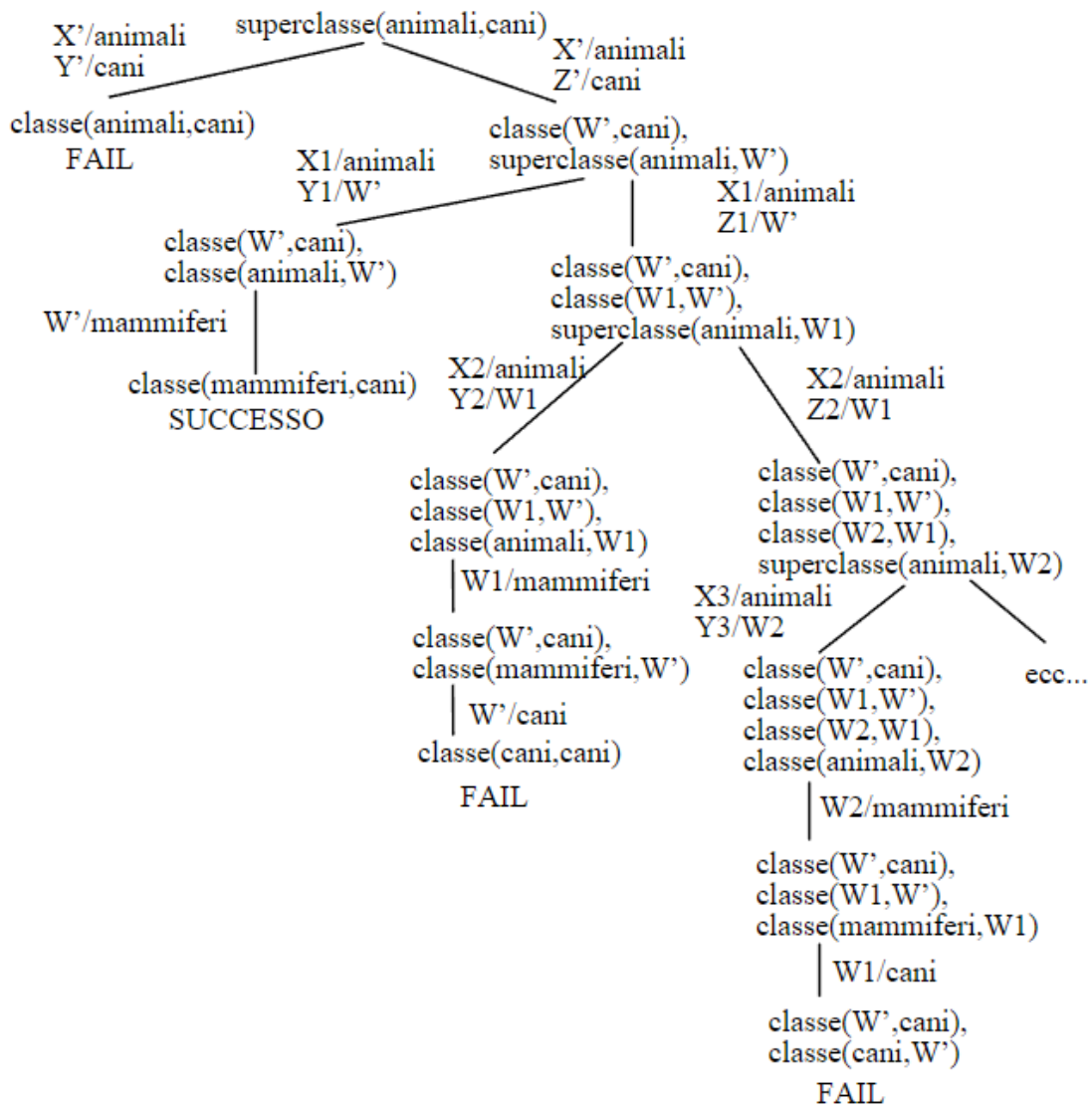
```

```

super(animali,cani)

```

risoluzione

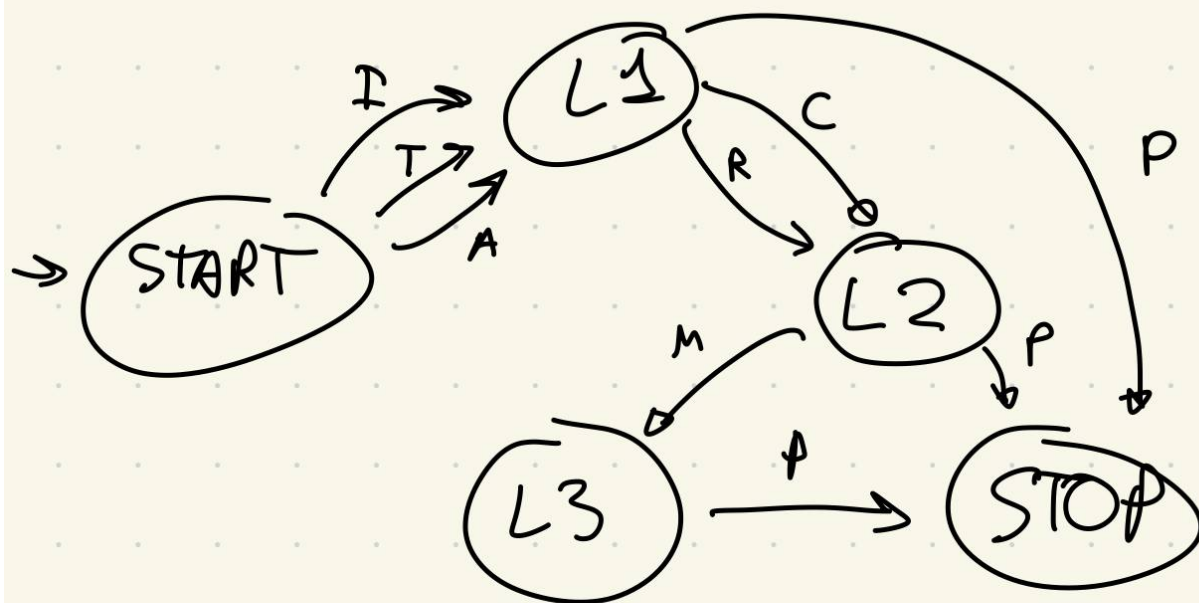


es 4 - automi a stati finiti

initial (start)
final (stop)
delta (start, 'I', level1)
delta (start, 'A', level1)
delta (start, 'T', level1)
delta (level1, 'C', level2)
delta (level1, 'R', level2)
delta (level2, 'M', level3)
delta (level1, 'P', stop)
delta (level2, 'P', stop)
delta (level3, 'P', stop)

risoluzione

cosa fa? riconosce i nomi di alcuni protocolli di rete



$$\left((I + A + T) \cdot \left((R + C)M + (R + C) \right) + (I + A + T) \right) \cdot P$$

oppure

$$(I + A + T) \left((R + C) + (R + C)M + \epsilon \right) P$$

((probabilmente l'esercizio consisterà nel descrivere cosa fa e/o se delle stringhe date sono o meno riconosciute))