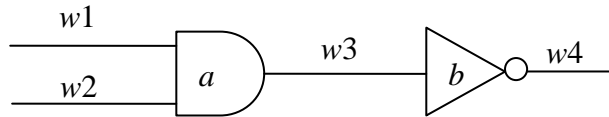


Esercizio 1 (punti 6)

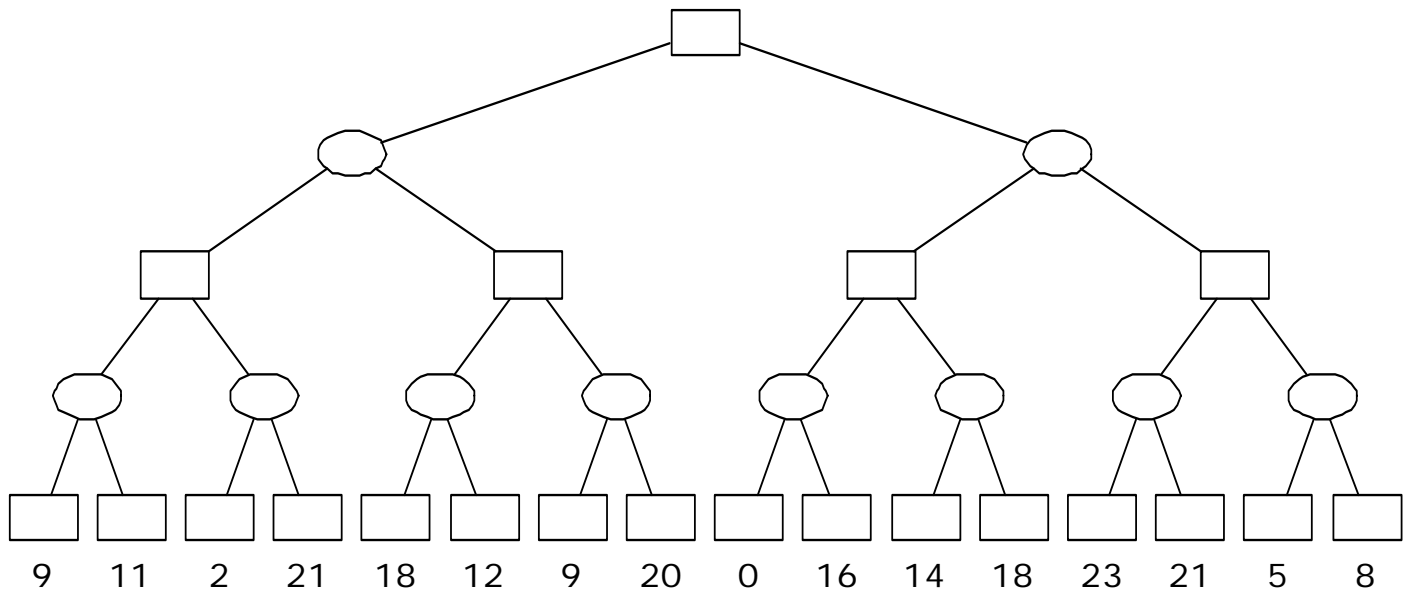
Il seguente circuito ha 4 segnali, $w1$, $w2$, $w3$ e $w4$. Ha una porta AND, la porta a , e un invertitore, la porta b . I segnali di ingresso, $w1$ e $w2$, possono essere o non essere *on*. Se la porta a funziona correttamente, il segnale $w3$ è *on* se e solo se i segnali $w1$ e $w2$ sono entrambi *on*. Se l'invertitore b funziona correttamente, il segnale $w4$ è *on* se e solo se il segnale $w3$ non è *on*.



- a. Si usino predicati del tipo $ok(a)$, $on(w1)$, $\neg on(w3)$ per descrivere nella logica dei predicati il funzionamento del circuito rappresentato in figura.
- b. Assumendo che gli ingressi $w1$ e $w2$ siano *on* e che l'uscita $w4$ sia anch'essa *on*, si usi il metodo di risoluzione per dimostrare che la porta a oppure l'invertitore b non funzionano correttamente. Si usi una strategia *lineare da input* o, se impossibile, *lineare*.

Esercizio 2 (punti 5)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore. Si assuma che il primo giocatore sia Max. Si mostri come gli algoritmi min-max e alfa-beta risolvono il problema



Esercizio 3 (punti 3)

Si scriva un predicato Prolog che date due liste in ingresso, è vero se la prima è sottoinsieme (proprio, o uguale) della seconda.

Esempio:

```
?-subset([a,1,b(X,Y)],[1,a,k(f(3)),b(X,Y)]).
```

yes

Esercizio 4 (punti 7)

Si consideri il seguente programma Prolog:

```
findexpr(R,E,L):- espress(E,L), R is E,!.  
espress(E,[E]):- !.  
espress(E,[X|T]):- operazione(E,X,B), espress(B,T).  
operazione(A+B,A,B).  
operazione(A*B,A,B).
```

Si mostri l'albero di derivazione SLD relativo alla query `findexpr(5,E,[1,2,3])` e si dica qual è la risposta calcolata.

Esercizio 5 (punti 6)

In un affollato laboratorio, con un solo banco prova, gravitano su 5 giorni della settimana ben cinque tecnici, nell'ordine Aldo, Brando, Carlo, Dado e Elena. Il banco prova può essere usato anche da due persone nella stessa giornata.

Per riuscire a lavorare senza disturbarsi troppo, i tecnici decidono di darsi la regola che coloro che lavorano insieme sulle stesse tematiche dovranno condividere la giornata. Inoltre, si ha che:

Brando e Dado lavorano su tematiche comuni.

I risultati di Dado servono a Carlo, che quindi deve fare le sue prove successivamente.

Carlo lavora con Aldo.

I risultati di Carlo servono a Elena, che quindi deve fare le sue prove successivamente.

Aldo viene da lontano e ha chiesto di non usare il laboratorio né il lunedì né il venerdì.

Si modelli il problema come un CSP. Si disegni il grafo di vincoli e si applichi alla rete iniziale la tecnica dell'arc-consistenza (si considerino a tale scopo solo i vincoli unari e binari nell'applicazione della arc-consistenza).

Si proceda poi alla ricerca di una soluzione applicando il forward checking (e propagando tutti i vincoli). Come euristica di selezione delle variabili si usi il first fail.

A parità di cardinalità dei domini, si istanzino le variabili secondo l'ordine dato ai tecnici. Si considerino i valori dei domini secondo il loro ordine crescente.

Esercizio 6 (punti 3)

Con riferimento alla ricerca euristica, si pensi ad un gioco dell'otto in cui A può essere spostata in B, se B è vuota. Sia H_g una stima esatta della distanza della soluzione minima per questo gioco "meno vincolato".

- H_g (detta l'euristica di Gasching) è ammissibile per il gioco dell'otto tradizionale?
- H_g è un'euristica migliore (più informata) rispetto all'euristica che conta il numero di caselle fuori posto per il gioco dell'otto tradizionale?
- Esistono casi in cui H_g è più accurata dell'euristica che somma le distanze Manhattan delle caselle dalla loro collocazione finale (anche se questa è occupata)? (FACOLTATIVO)

Esercizio 7 (punti 2)

Dare la definizione di correttezza di un sistema assiomatico deduttivo. Il calcolo dei predicati del I ordine è corretto?

VOTO:

Esame da 6 CF, il voto è determinato da questa prima parte

Esame da 9 CFU, è la media pesata della I parte (che vale 2/3) e della II (che vale 1/3) ovvero il voto finale è dato da: $((\text{voto_Iparte} + \text{voto_IIparte})/3)*2$ e varia quindi da 0 a massimo 32 (equivalente a lode).

Logiche descrittive (punti 5)

Cosa rappresentano concetti e ruoli nelle Logiche Descrittive?

Come potrebbero essere modellati concetti e ruoli in logica dei predicati del primo ordine?

Qual è il significato del seguente esempio?

```
(and man
  (atmost 2 parent)
  (atleast 2 parent)) [george]
```

Scrivere la formula logica equivalente al costrutto precedente.

Metainterpreti (punti 8)

Dato un linguaggio proposizionale, con regole “di produzione” del tipo:

```
imply([a,b],d) .           % a,b -> d
imply([a],e) .             % a -> e
imply([d],f) .             % d -> f
imply([a,f],g) .           % a,f -> g
```

Si realizzi, come programma Prolog, una parte di un motore inferenziale forward, che - data una lista di fatti in ingresso - è in grado di_

- identificare le regole il cui antecedente è vero (fase di *match*) e

- selezionarne una (*select*), chiedendo all'utente, per poi

- produrre in uscita la memoria di lavoro (lista di fatti veri, a partire da quelli dati):

Ad esempio:

```
?-match([a,b],Lout) .
% il programma deve scrivere a video:
[imply([a,b],d), imply([a],e)]
% chiede all'utente quale regola attivare, stampando:
quale?
% viene letto un intero, ad esempio 2 e selezionata la regola 2
% e quindi il predicato match restituisce:
Lout=[e,a,b]
```

Esercizio DCG (punti 3) (solo per chi non ha svolto l'esercitazione in itinere)

Data la seguente grammatica:

```
P = {
  <E> ::= <T> + <E> / <T>
  <T> ::= <F> * <T> / <T>
  <F> ::= <intero> / (<E>)
  <intero> ::= ...
}
```

Scrivere il programma DCG per il parsing di espressioni aritmetiche generabili dalla grammatica.

Si utilizzi il predicato SICStus **integer(X)** che è vero se **X** è un intero

Esempio:

```
?-expr([2+34*(25*27)], []).
Yes
?-expr([2++(25*27)], []).
No
```

SOLUZIONE – PARTE I

Esercizio 1

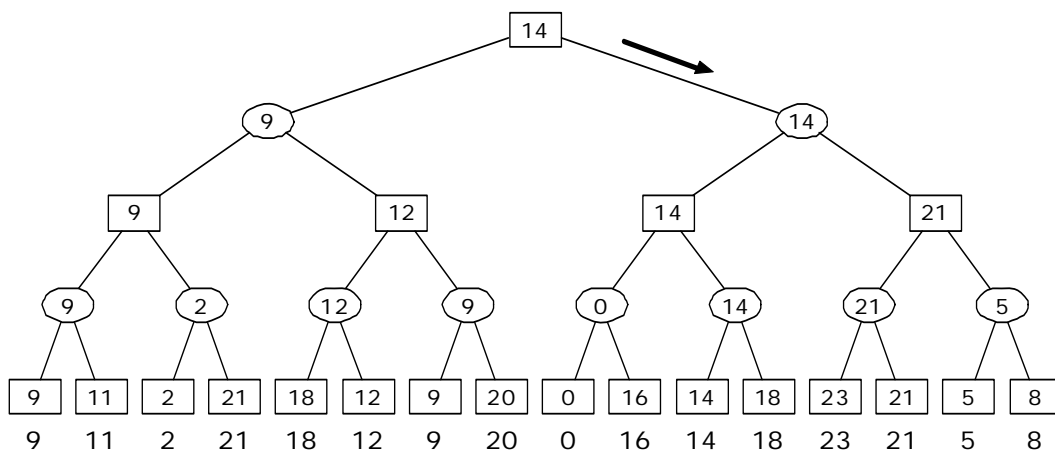
1. $Ok(A) \Rightarrow (On(W3) \Leftrightarrow (On(W1) \wedge On(W2)))$
2. $Ok(B) \Rightarrow (On(W4) \Leftrightarrow \neg On(W3))$
 - 1.1 $Ok(A) \Rightarrow (On(W3) \Rightarrow (On(W1) \wedge On(W2)))$
 - 1.1.1 $\neg Ok(A) \vee \neg On(W3) \vee On(W1)$
 - 1.1.2 $\neg Ok(A) \vee \neg On(W3) \vee On(W2)$
 - 1.2 $Ok(A) \Rightarrow (On(W1) \wedge On(W2)) \Rightarrow On(W3)$
 - 1.2.1 $\neg Ok(A) \vee \neg On(W1) \vee \neg On(W2) \vee On(W3)$
- 1.3 $Ok(B) \Rightarrow (On(W4) \Rightarrow \neg On(W3))$
 - 1.3.1 $\neg Ok(B) \vee \neg On(W4) \vee \neg On(W3)$
- 1.4 $Ok(B) \Rightarrow (\neg On(W3) \Rightarrow On(W4))$
 - 1.4.1 $\neg Ok(B) \vee On(W3) \vee On(W4)$

1. $\{\neg Ok(A), \neg On(W3), On(W1)\}$
2. $\{\neg Ok(A), \neg On(W3), On(W2)\}$
3. $\{\neg Ok(A), \neg On(W1), \neg On(W2), On(W3)\}$
4. $\{\neg Ok(B), \neg On(W4), \neg On(W3)\}$
5. $\{\neg Ok(B), On(W3), On(W4)\}$
6. $\{On(W1)\}$
7. $\{On(W2)\}$
8. $\{On(W4)\}$
9. $\{Ok(A)\}$
10. $\{Ok(B)\}$

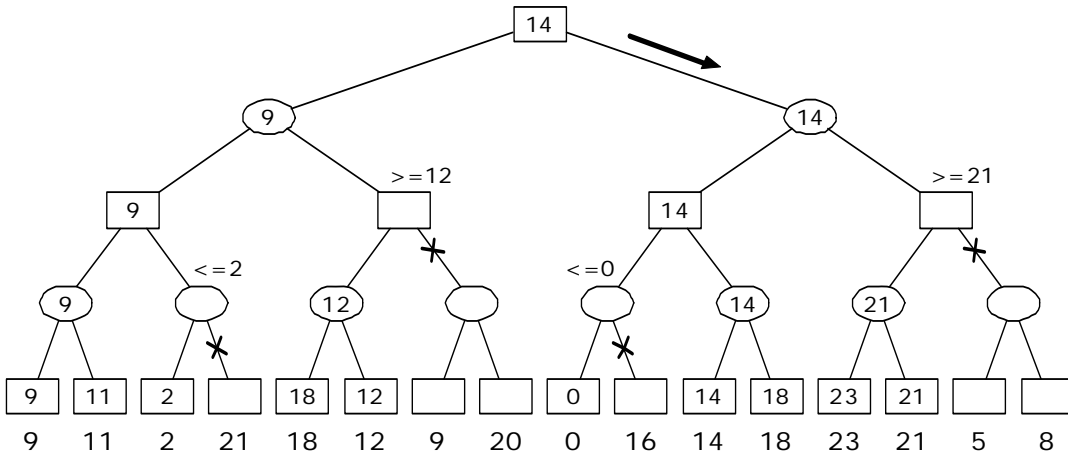
- | | |
|--|------------|
| $\{\neg On(W1), \neg On(W2), On(W3)\}$ | [da 3 e 9] |
| $\{\neg On(W2), On(W3)\}$ | [con 6] |
| $\{On(W3)\}$ | [con 7] |
| $\{\neg Ok(B), \neg On(W4)\}$ | [con 4] |
| $\{\neg On(W4)\}$ | [con 10] |
| $\{\}$ | [con 8] |

Esercizio 2

min-max:



Alfa-beta:

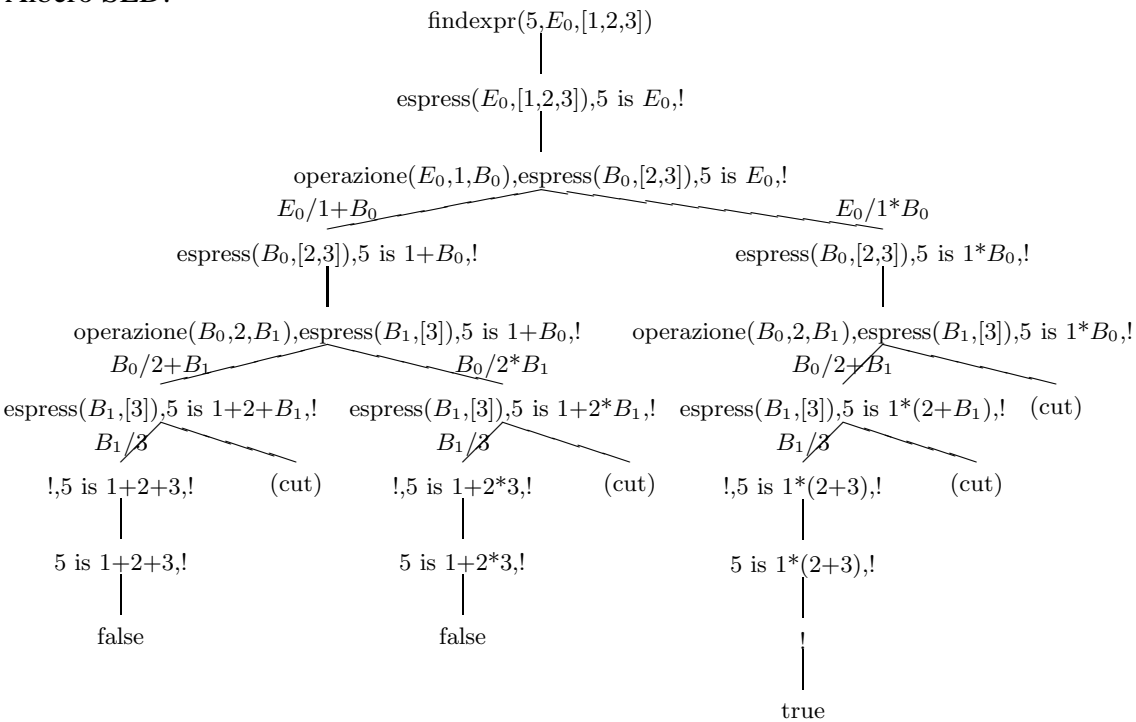


Esercizio 3

subset ([], _) :- ! .
 subset ([H|T] , L) :- member (H , L) , subset (T , L) .

Esercizio 4

Albero SLD:



La risposta calcolata è E/1*(2+3).

Esercizio 5

Variabili: i cinque tecnici Ci, i = 1.. 5 (il primo è Aldo, l'ultimo Elena)

Domini: 1..5

Vincoli:

- 2 variabili assegnate ogni giorno (2 persone):

$\forall i, j \forall k: C_i = C_j \Rightarrow \forall n \neq i, j C_n \neq C_i$ % non è un vincolo binario

Aldo viene da lontano e ha chiesto di non usare il laboratorio né il lunedì né il venerdì.

$C_1 \neq 1, C_1 \neq 5$

Brando e Dado lavorano su tematiche comuni.

$C_2 = C_4$

Carlo lavora con Aldo.

$C3=C1$

I risultati di Dado servono a Carlo, che quindi deve fare le sue prove successivamente.

$C4 < C3$

I risultati di Carlo servono a Elena, che quindi deve fare le sue prove successivamente.

$C3 < C5$

(il disegno della rete è banale e si omette qui)

	C1(aldo)	C2(brando)	C3(carlo)	C4(dado)	C5(elena)
	[1..5]	[1..5]	[1..5]	[1..5]	[1..5]
AC	[2..4]	[1..5]	[2..4]	[1..3]	[3..5]
AC	[2..4]	[1..3]	[2..4]	[1..3]	[3..5]
AC (quiescenza)	[2..4]	[1..3]	[2..4]	[1..3]	[3..5]
labeling	C1=2				
FC		[1..3]	[2]	[1..3]	[3..5]
labeling			C3=2		
FC		[1, 3]		[1]	[3..5]
labeling				C4=1	
FC		[1]			[3..5]
labeling		C2=1			
FC					[3..5]
labeling					C5=3

Esercizio 6

- Hg è ammissibile per il gioco dell'otto tradizionale in quanto tutte le mosse possibili nel gioco dell'otto tradizionale sono anche possibili nel gioco dell'otto meno vincolato. Quindi tipicamente ci vogliono più mosse e nel caso migliore ce ne vuole lo stesso numero.
- Per ogni n $H1(n) \leq Hg(n)$. Infatti le caselle fuori posto possono essere messe a posto con una o due mosse: una se la casella di destinazione è vuota, due se va svuotata prima di metterla a posto (non sempre la mossa di svuotamento può servire a mettere a posto un'altra casella).
- Si, esistono casi in cui Hg è più accurata di H2. Si consideri il seguente esempio:

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

$H1=2$ le due caselle fuori posto sono la 1 e la 2

$Hg=3$ per mettere le cose a posto nel gioco meno vincolato servono 3 mosse

SOLUZIONE – PARTE II

Metainterpreti

```
match(L, [A|L]):-
    findall(imply(B,H), (imply(B,H),subset(B,L)), HL),
% trova tutte le regole il cui antecedente B è sottoinsieme di L
% e le raccoglie con findall nella lista HL
    ask(HL,N),
% chiede all'utente quale regola N di HL vuole attivare
    select(N,HL,A).
% seleziona l'N-esima regola e ne restituisce il conseguente A

select(1,[imply(_,A)|_],A):-!.
select(N,[_|L],A):- N1 is N-1, select(N1,L,A).

subset([],_):-!.
subset([H|T],L):-member(H,L),subset(T,L).

ask(X,Risp):-
    write(X),
    write(" quale ? "),
    read(Risp).
```

SOLUZIONE – DCG

(esercizio 9.2 delle slide su Prolog e grammatiche)

```
expr --> term.
expr --> term, [+], expr.
term --> factor.
term --> factor, [*], term.
factor --> [I], {integer(I)}.
factor --> ['('], expr, [')'].
```