Practical Assignment – Artificial Intelligence

By – Anshul Verma, 78065
BSc (Hons) Computer Science
Semester – 6, Part – III

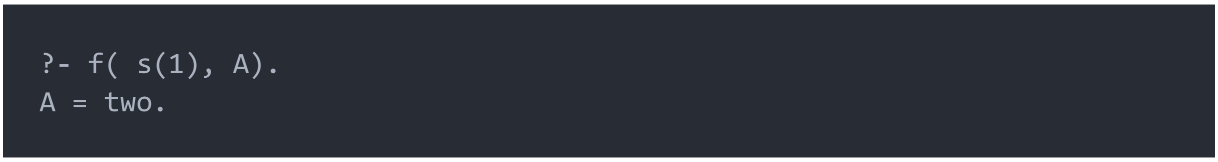
Ques 1. (2.6) Consider the following program:

```
f( 1, one).
f( s(1), two).
f( s(s(1)), three).
f( s(s(s(X))), N) :-
   f( X, N).
```

How will Prolog answer the following questions? Whenever several answers are possible, give at least two.
(a) ?- f(s(1), A).
(b) ?- f(s(s(1)), two).
(c) ?- f(s(s(s(s(s(s(1)))))), C).
(d) ?- f(D, three).

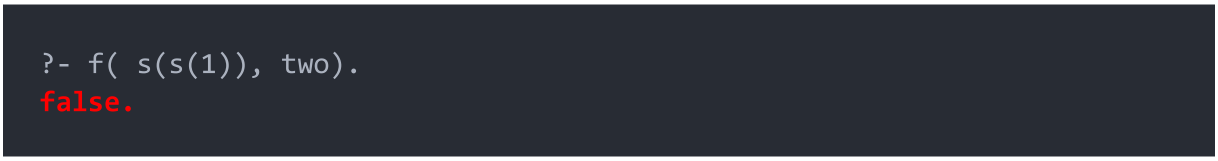
Ans:

(a)



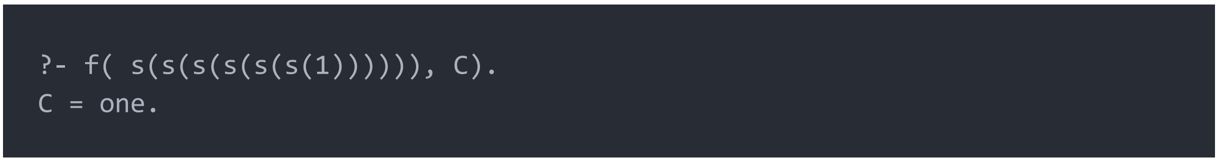
```
?- f( s(1), A).
A = two.
```

(b)



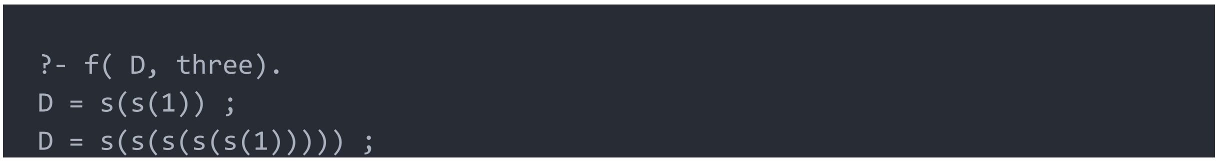
```
?- f( s(s(1)), two).
false.
```

(c)



```
?- f( s(s(s(s(s(s(1)))))), C).
C = one.
```

(d)



```
?- f( D, three).
D = s(s(1)) ;
D = s(s(s(s(s(1))))) ;
```

Ques 2. (2.7) The following program says that two people are relatives if

- one is a predecessor of the other, or
- they have a common predecessor, or
- they have a common successor:

```
relatives( X, Y) :-
  predecesso( X, Y).

relatives( X, Y) :-
  predecessor( Y, X).

relatives( X, Y) :-  % X and Y have a common predecessor
  predecessor( Z, X),
  predecessor(Z, Y).

relatives( X, Y) :-  % X and Y have a common successor
  predecessor( X, Z),
  predecessor(Y, Z).
```

Can you shorten this program by using the semicolon notation?

Ans:

The above given code can be shortened and rewritten using the semicolon notation as follows,

```
relatives( X, Y) :-
  predecesso( X, Y);
  predecessor( Y, X);
  predecessor( Z, X), predecessor(Z, Y);
  predecessor( X, Z), predecessor(Y, Z).
```

Ques 3. (2.8) Rewrite the following program without using the semicolon notation.

```
translate( Number, Word) :-
  Number = 1, Word = one;
  Number = 2, Word = two;
  Number = 3, Word = three.
```

Ans:

The above given code can be rewritten without using the semicolon notation as follows,

```
translate( 1, one).
translate( 2, two).
translate( 3, three).
```

Ques 4. Write five possible queries with output and explanation.

```
% Scheduling a meeting

% schedule(TimeA, A1, A2, TimeB, B1, B2, TimeD, D1, D2):
%   TimeA and experts A1, A2 assigned
%     to session on Artificial Intelligence,
%   TimeB, B1, B2 assigned to session on
%     bioinformatics, and similar for databases

schedule( Ta, A1, A2, Tb, B1, B2, Td, D1, D2) :-
  % Session AI at time Ta, with experts A1 and A2
  session( Ta, artificial intelligence, A1, A2),
  % Bioinformatics at Tb, with experts B1, B2
  session( Tb, bioinformatics, B1, B2),
  % Databases at Td, with experts DI and D2
  session( Td, databases, D1, D2),
  % No conflict between AI and Bioinfo
  no_conflict( Ta., A1, A2, Tb, B1, B2),
  % No conflict between Databases and AI
  no_conflict( Ta, A1, A2, Td, D1, D2),
  % No conflict between Bioinfo and Databases
  no_conflict(Tb, B1, B2, Td, D1, D2).

% session( Time, Topic, P1, P2):
%   session at Time on Topic attended
%     by responsible experts P1, P2

session( Time, Topic, P1, P2):-
  time( Time),         % Time is morning or afternoon
  expert( Topic, P1),  % Person P1 is expert on Topic
  expert( Topic, P2),  % P2 is also expert on Topic
  P1 \= P2.            % P1, P2 different persons

% no_conflict( Time1, P1, P2, Time2, Q1, Q2):
%   There is no time conflict between two sessions
%     at Time1 and Time2
%   and experts P1, P2, and Q1, Q2, respectively

no_conflict( Time1, _, _, Time2, _,_):-
  % Two sessions at different times- not conflict
  Time1 \= Time2.

% Two sessions at the same time
no_conflict( Time, P1,P2, Time, Q1,Q2):-
  P1 \= Q1 , P1 \= Q2, % No overlap between experts
  P2 \= Q1 , P2 \= Q2.

% Possible times of sessions
time( morning).
time( afternoon).

% Experts for topics
expert( bioinformatics, barbara).
expert( artificial intelligence, adam).
expert( artificial intelligence, barbara).
expert( databases, danny).
expert( bioinformatics, ben).
expert( artificial intelligence, ann).
expert( databases, adam).
```

Prolog program for Scheduling a project meeting and assigning experts to sessions.

Ans:

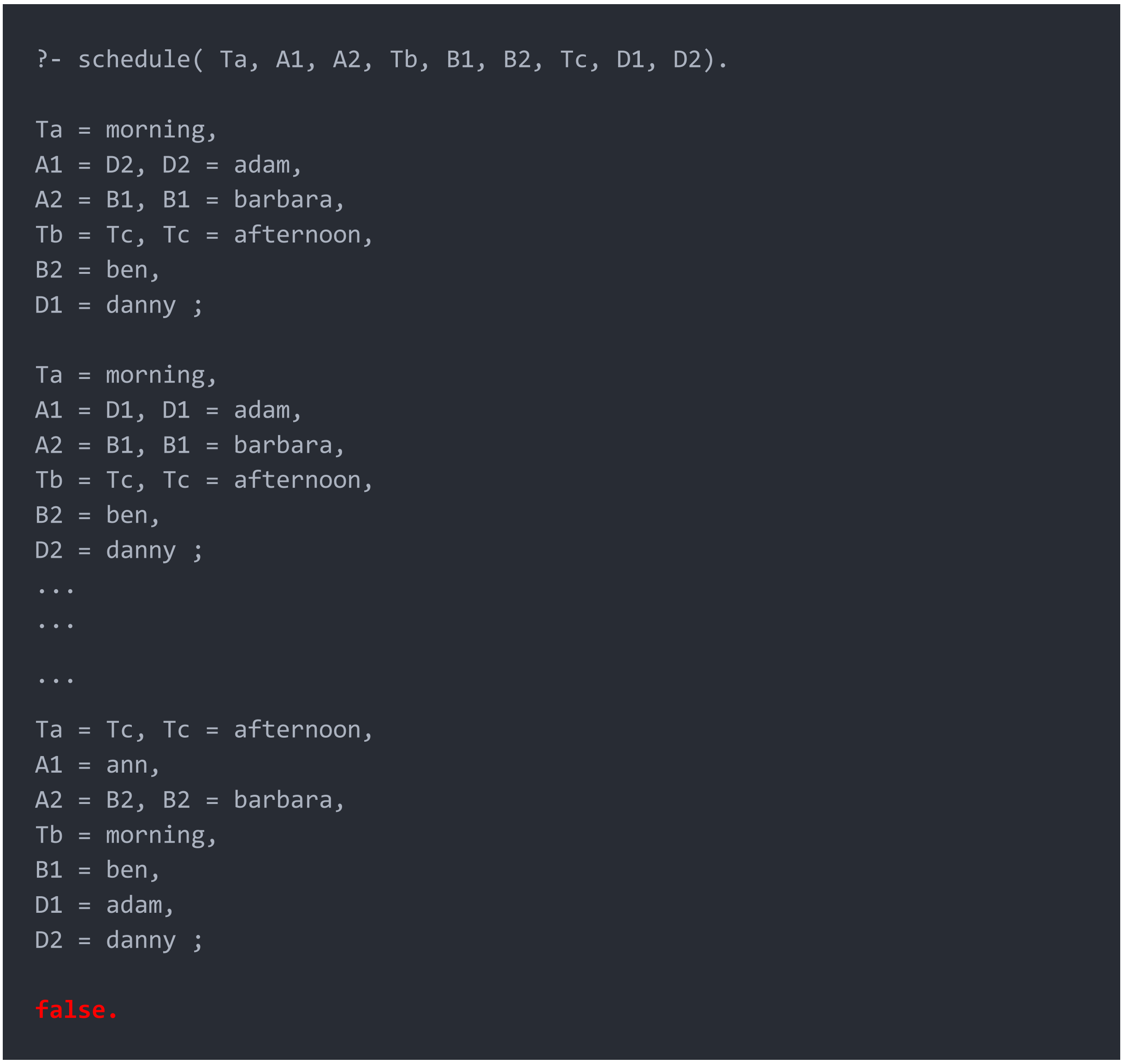
Query 1.

A query to produce schedule according to the given program.

```
?- schedule( Ta, A1, A2, Tb, B1, B2, Tc, D1, D2).
```

Output:

There will be 80 such schedules that can be obtained from above query, few of which are:



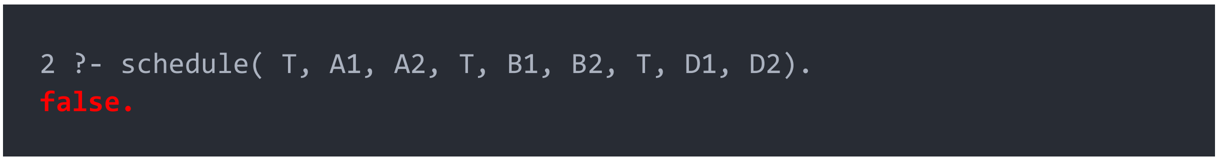
```
?- schedule( Ta, A1, A2, Tb, B1, B2, Tc, D1, D2).

Ta = morning,
A1 = D2, D2 = adam,
A2 = B1, B1 = barbara,
Tb = Tc, Tc = afternoon,
B2 = ben,
D1 = danny ;

Ta = morning,
A1 = D1, D1 = adam,
A2 = B1, B1 = barbara,
Tb = Tc, Tc = afternoon,
B2 = ben,
D2 = danny ;
...
...

...

Ta = Tc, Tc = afternoon,
A1 = ann,
A2 = B2, B2 = barbara,
Tb = morning,
B1 = ben,
D1 = adam,
D2 = danny ;

false.
```

Query 2.

A query to produce a schedule (if possible), such that all the sessions can be completed in half day, i.e. all the three sessions have to be scheduled at the same time.

```
?- schedule( T, A1, A2, T, B1, B2, T, D1, D2).
```

Output:



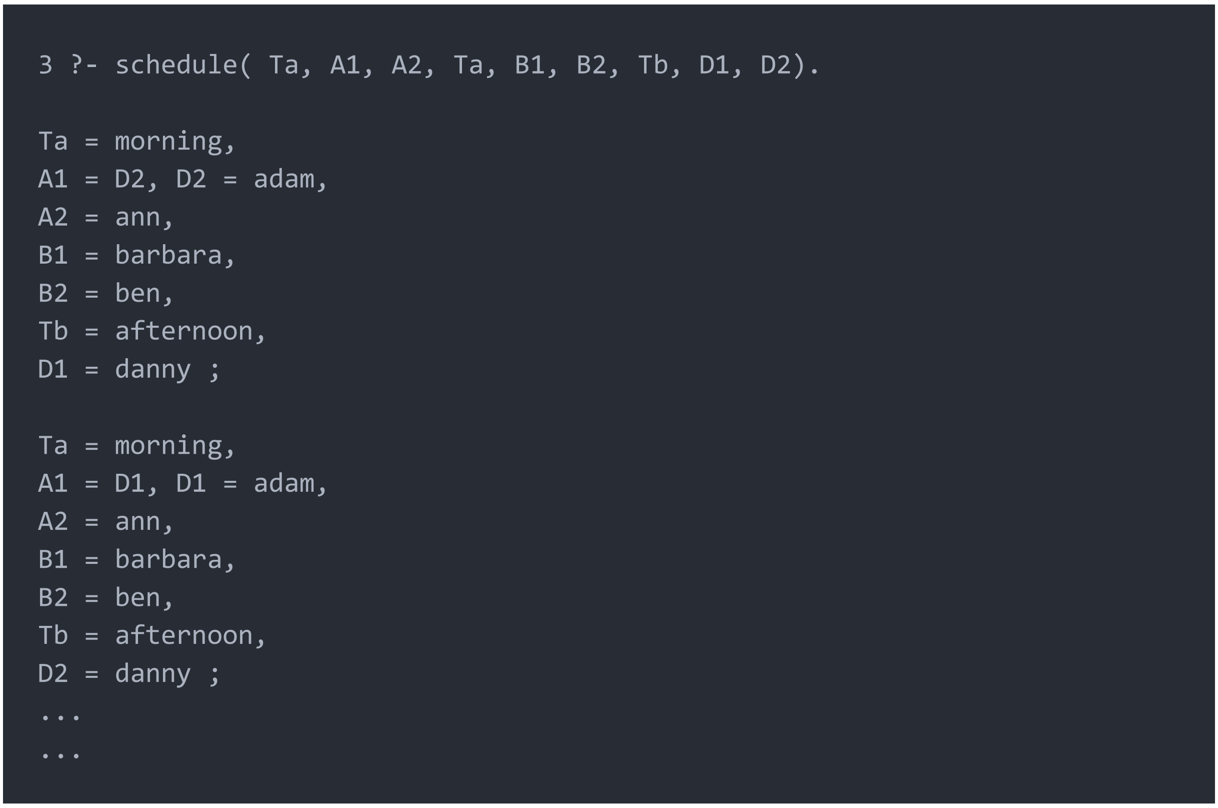
```
2 ?- schedule( T, A1, A2, T, B1, B2, T, D1, D2).
false.
```

Query 3.

A query to produce a schedule, such that the *Artificial Intelligence* session and the *Bioinformatics* session can be completed in half day, i.e. both sessions have to be scheduled at the same time.

```
?- schedule( Ta, A1, A2, Ta, B1, B2, Tb, D1, D2).
```

Output:
There will be 16 such schedules that can be obtained from above query, few of which are:



```
3 ?- schedule( Ta, A1, A2, Ta, B1, B2, Tb, D1, D2).

Ta = morning,
A1 = D2, D2 = adam,
A2 = ann,
B1 = barbara,
B2 = ben,
Tb = afternoon,
D1 = danny ;

Ta = morning,
A1 = D1, D1 = adam,
A2 = ann,
B1 = barbara,
B2 = ben,
Tb = afternoon,
D2 = danny ;
...
...
```

```
Ta = afternoon,
A1 = ann,
A2 = D1, D1 = adam,
B1 = ben,
B2 = barbara,
Tb = morning,
D2 = danny ;
false.
```

Query 4.

A query to produce a schedule, such that the *Artificial Intelligence* session and the *Databases* session can be completed in half day, i.e. both sessions have to be scheduled at the same time.
`?- schedule( Ta, A1, A2, Tb, B1, B2, Ta, D1, D2).`

Output:
There will be 16 such schedules that can be obtained from above query, few of which are:

```
4 ?- schedule( Ta, A1, A2, Tb, B1, B2, Ta, D1, D2).

Ta = morning,
A1 = B1, B1 = barbara,
A2 = ann,
Tb = afternoon,
B2 = ben,
D1 = danny,
D2 = adam ;

Ta = morning,
A1 = B1, B1 = barbara,
A2 = ann,
Tb = afternoon,
B2 = ben,
D1 = adam,
D2 = danny ;
...
...
...
Ta = afternoon,
A1 = ann,
A2 = B2, B2 = barbara,
Tb = morning,
B1 = ben,
D1 = adam,
D2 = danny ;
false.
```

Query 5.

A query to produce a schedule, such that the *Bioinformatics* session and the *Databases* session can be completed in half day, i.e. both sessions have to be scheduled at the same time.

```
?- schedule( Ta, A1, A2, Tb, B1, B2, Tb, D1, D2).
```

Output:

There will be 48 such schedules that can be obtained from above query, few of which are:

```
5 ?- schedule( Ta, A1, A2, Tb, B1, B2, Tb, D1, D2).

Ta = morning,
A1 = D2, D2 = adam,
A2 = B1, B1 = barbara,
Tb = afternoon,
B2 = ben,
D1 = danny ;

Ta = morning,
A1 = D1, D1 = adam,
A2 = B1, B1 = barbara,
Tb = afternoon,
B2 = ben,
D2 = danny ;
...
...
...
Ta = afternoon,
A1 = ann,
A2 = B2, B2 = barbara,
Tb = morning,
B1 = ben,
D1 = adam,
D2 = danny ;

false.
```