Backtracking Example & Explanation:

Prolog Program

/* Facts */
likes(mary, food). // mary likes food.
likes(mary, wine). // mary likes wine

/* Rule */
likes(john, X):-likes(mary,X).
// john likes everything whatever mary like.

/* Goal */
?- likes(john, What).
// What does john like.

Query Prompt

Used 'trace' in SWI-Prolog to explain Backtracking Process.

?- trace.
ture.

?- likes(john, What).
   Call: (6) likes(john, _G7400) ? creep
   Call: (7) likes(mary, _G7400) ? creep
   Exit: (7) likes(mary, food) ? creep
   Exit: (6) likes(john, food) ? creep
What = food ;
   Redo: (7) likes(mary, _G7400) ? creep
   Exit: (7) likes(mary, wine) ? creep
   Exit: (6) likes(john, wine) ? creep
What = wine ;

?- nodebug.

Explanation:

- Here in query ?- likes(john, What). 'What' is a variable.
- Upon asking the query prolog first reads the query and find the match for it in the knowledge base in top-down manner.
- It finds the matching rule - likes(john, X). But from that rule we can infer likes(mary, X). Now prolog will go back (backtrack) and will again read the knowledge base from the first line and finds likes(mary, food) and it will return answer as What = food - variable 'What' is instantiated with value 'food'. Here Prolog will keep the mark in database that for this goal What = food has been answered and it satisfies.
• But there more than one solutions for the given query.
• Then again prolog will read the knowledge base from the first line (backtrack) but it already had a mark on 'food' so it will not answer 'food' again. Now it will try to match and satisfy the goal with second fact likes(mary, wine). and match occurs. So, it replies with What = wine.

[ In above 'trace' query, _G7400 is the default value assigned to the variable 'What' by Prolog. It is possible that different computers gives different values for this. ]

Try it yourself - How does prolog respond to these queries.

?- write(What).
?- write(X).
?- write(_What).