3.2.1 Backtracking Example

Excerpted from http://ww2.cs.mu.oz.au/482/lectures/Prolog_How_Works.pdf

```
member(X, [X|_]).
member(X, [_|L]) :-
    member(X, L).
```

Initial query: `member(Z, [a,b]), member(Z, [b,c])`

Both clauses match first goal; choose first and **push a choicepoint**

`Z = X = a`

New query: `member(a, [b,c])`

Clause 1 does not match; clause 2 matches with

`X1 = a,  L1 = [c]`

?- Backtracking Example (2)

New query: `member(a, [c])`

Clause 1 does not match; clause 2 matches with

`X2 = a,  L2 = []`

New query: `member(a, [])`

Neither clause matches: failure

Backtracking Example (3)

We are not done; we pop the choicepoint off the stack

This returns us to state at time the choicepoint was pushed, but now we go on to the next clause

New query: `member(Z, [a,b]), member(Z, [b,c])`

Now choose second clause, with

```
Z = X,    L = [b]
```

New query: `member(Z, [b]), member(Z, [b,c])`

Both clauses match first goal; choose first and **push a choicepoint**

```
Z = X = b
```

?- Backtracking Example (4)

New query: `member(b, [b,c])`

Both clauses match first goal; choose first and **push a choicepoint**

```
X2 = b
```

New query is empty: success, leaving

```
Z = b
```

Prolog prints this result; if we hit ; asking for more solutions, this forces an artificial failure, causing Prolog to backtrack, looking for more solutions

?- Backtracking Example (5)

New query: `member(b, [b,c])`

Clause 2 matches with `X2 = b,  L2 = [c]`

New query: `member(b, [c])`

Clause 2 matches with `X3 = b,  L3 = []`

New query: `member(b, [])`

Neither clause matches: pop last choicepoint

New query: `member(Z, [b]), member(Z, [b,c])`

Second clause matches, leaving

`X3 = Z,  L3 = []`

New query: `member(Z, []), member(Z, [b,c])`

Neither clause matches: final failure

3.2.2 Example: Search Tree