Backtracking Search Process Example 2:
Try to trace through the search process for Q1.

**Program:**
```
c1 : studies(charlie, csc135).
c2 : studies(charlie, csc131).
c3 : studies(jack, csc131).
c4 : teaches(collins, csc131).
c5 : professor(X, C) :- teaches(X, C)
```

**Question 1:**
What courses charlie and jack study?
(OR Which same courses charlie and jack study?)
(here 'and' in the question represents conjunction)

**Query Prompt:**
```
Q1 : ?- studies(charlie, C), studies(jack, C).
```

Here in above example query is studies(charlie, C), studies(jack, C). where C is a variable for question we asked about the courses.

So we will trace the whole backtracking search process for getting the solution of studies(charlie, C), studies(jack, C).

See the example-1 to understand 'trace' and step-by-step procedure of backtracking search process.

Below is the representation of backtracking process in prolog. In prolog you can see the whole backtracking process using ‘trace’.

```
?- trace. studies(charlie, C), studies(jack, C).
true.
   Call: (7) studies(charlie, _G868) ? [trace] 8 ?- creep
   Exit: (7) studies(charlie, csc135) ? creep
   Call: (7) studies(jack, csc135) ? creep
   Fail: (7) studies(jack, csc135) ? creep
   Redo: (7) studies(charlie, _G868) ? creep
   Exit: (7) studies(charlie, csc131) ? creep
   Call: (7) studies(jack, csc131) ? creep
   Exit: (7) studies(jack, csc131) ? creep
   C = csc131.
```

(1) So for query Q1 prolog will first read the query and in our example it is conjunction of subgoals. So, prolog will read left most goal first i.e. studies(charlie, C).
(2) Now for this goal prolog will try to find the same predicate in knowledge base with first argument 'charlie' and it can find clause c1 i.e. studies(charlie, csc135). Here after reading the second argument, variable C is instantiated to csc135. Hence, this subgoal is satisfied.

(3) Now, we move on to the other subgoal. We know value of C. So, prolog will search for studies(jack, csc135). But it will not succeed finding this fact in the program. So, this subgoal is not satisfied and prolog gives failure.

(4) Now, it backtracks to find another possible solution for given query and again starts with studies(charlie, C). Again it can find the clause c2 for this and thus it has studies(charlie, csc131) and C is instantiated to csc131. Again, this subgoal is satisfied.

(5) Move on to next subgoal with value C=csc131 -> studies(jack, csc131). But this time prolog will find the same fact in it's knowledge base. So, this subgoal is also satisfied.

(6) Query succeeds if all the subgoals succeeds, hence now prolog will give success.

The figure below shows the whole backtracking path for query studies(charlie, C), studies(jack, C) and is the final answer for this example.
Try to trace through the search process for Q1, but with subgoals reversed.

Q1 is \( \text{studies(charlie, C)}, \text{studies(jack, C)} \)
Q1 with subgoals reversed is \( \text{studies(jack, C)}, \text{studies(charlie, C)} \).

Program:
\[
\begin{align*}
\text{c1 : studies(charlie, csc135).} \\
\text{c2 : studies(charlie, csc131).} \\
\text{c3 : studies(jack, csc131).} \\
\text{c4 : teaches(collins, csc131).} \\
\text{c5 : professor(X, C) :- teaches(X, C)}
\end{align*}
\]

Prolog Query Prompt for studies(jack, C), studies(charlie, C) with 'trace':

\[
?\text{- trace. studies(jack, C)}, \text{studies(charlie, C)}.
\]

Call: (7) studies(jack, _G862) ? creep
Exit: (7) studies(jack, csc131) ? creep
Call: (7) studies(charlie, csc131) ? creep
Exit: (7) studies(charlie, csc131) ? creep
C = csc131.

We don't need to backtrack here because left most subgoal is studies(jack, C), and after unifying the same predicate from knowledge base, C is instantiated to csc131. We can also find another subgoal studies(charlie, csc131). in our knowledge base and query succeeds. Hence, we don't need to do backtracking process for this reversed query. This is how you can prevent prolog to backtrack and save time and improve the speed.

The figure below shows the whole backtracking path for query studies(jack, C), studies(charlie, C) and is the final answer for this example.