Question and Answers Document

1. What are the error recovery techniques in top-down parsing?  
  
Top-down parsing is a recursive descent parsing technique. It starts with the goal symbol and tries to match it with the input string. If it finds a match, it continues parsing the rest of the string. If it does not find a match, it tries to recover from the error by backtracking.  
  
There are two main error recovery techniques in top-down parsing:  
  
\* \*\*Error recovery by backtracking:\*\* This technique involves trying all possible alternatives until a match is found. If no match is found, the parser reports an error.  
\* \*\*Error recovery by rescanning:\*\* This technique involves rescanning the input string from the point where the error occurred. If the error is found to be a syntax error, the parser reports an error. If the error is found to be a semantic error, the parser tries to recover from the error by making a different choice.  
  
2. What is Boot ‘Strapping?  
  
Boot ‘strapping is a technique used to initialize a compiler. It involves writing a small program in the language that the compiler is being written for. This program is called the bootstrap program. The bootstrap program is used to compile the rest of the compiler.  
  
3. What is the difference between top-down parsing and bottom-up parsing?  
  
Top-down parsing is a recursive descent parsing technique. It starts with the goal symbol and tries to match it with the input string. If it finds a match, it continues parsing the rest of the string. If it does not find a match, it tries to recover from the error by backtracking.  
  
Bottom-up parsing is a non-recursive parsing technique. It starts with the input string and tries to match it with the goal symbol. If it finds a match, it continues parsing the rest of the string. If it does not find a match, it tries to recover from the error by backtracking.  
  
4. What are ambiguous grammars? Give an example.  
  
An ambiguous grammar is a grammar that has more than one parse tree for a given input string. For example, the following grammar is ambiguous:  
  
```  
S -> NP VP  
NP -> Det N  
VP -> V NP | V  
Det -> the | a  
N -> man | woman  
V -> saw | ate  
```  
  
The sentence "The man saw the woman" can be parsed in two ways:  
  
\* S -> NP VP  
\* NP -> Det N  
\* N -> man  
\* VP -> V NP  
\* V -> saw  
\* NP -> Det N  
\* N -> woman  
  
\* S -> NP VP  
\* NP -> Det N  
\* N -> woman  
\* VP -> V NP  
\* V -> saw  
\* NP -> Det N  
\* N -> man  
  
5. What is Yacc? Explain the Syntax.  
  
Yacc is a parser generator. It is a program that takes a grammar as input and generates a parser as output. The parser can be used to parse any input that conforms to the grammar.  
  
The syntax of Yacc is as follows:  
  
```  
%{  
...  
%}  
  
%token <token-name>  
  
%left <production-name>  
  
%right <production-name>  
  
<declaration>  
<rule>  
<start>  
```  
  
The `%{` and `%}` lines are comment lines. The `%token` line defines the tokens that the parser will recognize. The `%left` and `%right` lines define the associativity and precedence of the operators. The `<declaration>` line defines variables and functions that are used by the parser. The `<rule>` line defines a production rule. The `<start>` line defines the start symbol of the grammar.  
  
6. Define shift-reduce conflict and reduce-reduce conflict.  
  
A shift-reduce conflict is a situation where the parser can either shift the next token onto the stack or reduce the current state. A reduce-reduce conflict is a situation where the parser can either reduce the current state to one of two different states.  
  
Shift-reduce conflicts can be resolved by using a conflict resolution algorithm. Reduce-reduce conflicts can be resolved by using a precedence grammar or by using a left-corner grammar.