

COMP360 Exam 1 Solutions

You are allowed to have one 8 ½ by 11" page of notes for this exam.

Consider a language to describe a hierarchical organization. Each committee has a leader and possibly other people on the committee. A committee may have subcommittees. Just like a committee, each subcommittee has a leader, possible other people and possible subcommittees.

A committee is represented as by square brackets with the name of the leader inside the left bracket. After the leader can come the names of members or subcommittees, in any order. There may be none or many other members and subcommittees. All names are a single word of letters. There can multiple spaces between brackets and words.

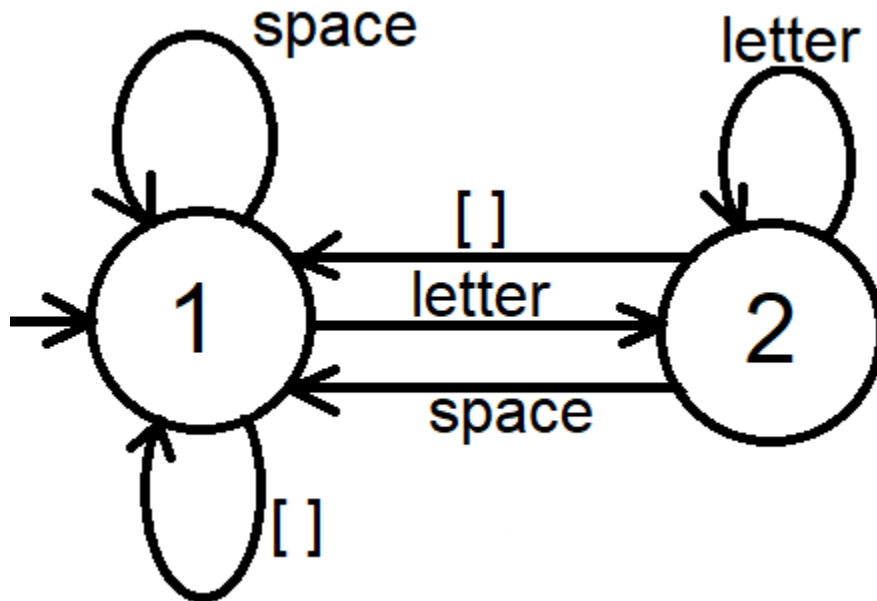
Valid strings in the committee language are:

[Fred] or [Sadie Betsy [Mark [Cindy Jake]] Joe] or [Dorothy Bill]

Invalid strings are:

[Fred [Mary]	unbalanced brackets
[[Joe Cindy] Mark]	no leader
[]	no anything

1. [20 points] Draw circle and arrow graph showing a deterministic Finite State Automata (FSA) for a scanner that creates tokens for the committee language. Transitions between states are based on individual characters. This should be a Mealy machine that takes actions at the state transitions. Number the states from 1 to as large as needed.



COMP360 Exam 1 Solutions

2. [20 points] Create the state table and action table for the lexical analyzer in question 1. In the table below, write the new state, a slash and the action to take at this transition. Use state zero to represent a terminal failing state. The available actions are:

- 0 do nothing
- 1 Create a token containing the input symbol.
- 2 Add the input symbol to the end of the current token string.

Write the input symbols (or groups of symbols) in the left column.

input symbol	state 1	state 2	state 3	state 4	state 5
letter	2/1	2/2	/	/	/
[]	1/1	1/1	/	/	/
space	1/0	1/0	/	/	/
other	0/0	0/0	/	/	/
	/	/	/	/	/

There may or may not be more columns and rows than necessary.

3. [20 points] Write a Backus–Naur Form (BNF) description of the committee language. The terminal symbols are **name**, [and]

```

cmte    → [ name ]
           | [ name memberlist ]
memberlist → member
           | member memberlist
member    → name
           | cmte
    
```

COMP360 Exam 1 Solutions

4. [20 points] Write a method of a recursive descent parser for **ONE** of the following BNF productions. You may write the method in Java, C++ or pseudo code. Assume there is a global list or array of Token objects called `tokens` and an integer `curtok` that is an index into `tokens` for the next token.

1. `gerbil` → `mouse` | `mouse $ gerbil`
2. `mouse` → `hamster` | `hamster * mouse`
3. `hamster` → *name* | `# gerbil`

where `gerbil` is the root non-terminal symbol. "\$", "#", "*" and "name" are terminal symbols.

What is the number of the one BNF production you are implementing? _____

```
boolean gerbil() {
    if (!mouse() ) return false;
    if (tokens[curtok] == "$") {
        curtok++;
        if ( !gerbil() ) throw parse error;
    }
    return true;
}
```

```
boolean mouse() {
    if (!hamster() ) return false;
    if (tokens[curtok] == "*") {
        curtok++;
        if ( !mouse() ) throw parse error;
    }
    return true;
}
```

```
boolean hamster() {
    if (tokens[curtok].isName() ) {
        curtok++;
        return true;
    }
    if (tokens[curtok] != "#") return false;
    curtok++;
    if ( !gerbil() ) throw parse error;
    return true;
}
```

5. [10 points] Give **TWO** strings of three or more symbols that are in the language above in question 4. Use `trout`, `cod`, `bass` and `shark` for names as needed.

`bass * trout` or `shark $ cod $ trout` or many more

COMP360 Exam 1 Solutions

6. [10 points] In a two-pass assembler, what is the goal of:

Pass 1

Build the symbol table mapping names to numerical values, such as labels to addresses.

Pass 2

Generate machine language. The addresses for all names should be in the symbol table.