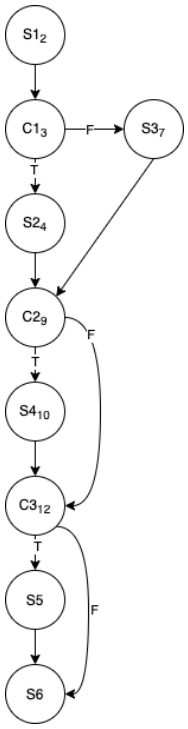
SQA Assignment 2
Sarah Pham

Quick Notes:

- Cyclomatic number tells us the upper bound (max number of ip paths to cover all branches/paths) of the number of paths in basis set
- P* is the total number of syntactic paths
- If there are parallels, you add them instead of multiply
- DD path is from decision to decision and are partial paths
- Statement coverage is not enough, it's the bare minimum
- Path predicates are rules we need to satisfy in order to execute the path itself

Problem 1:

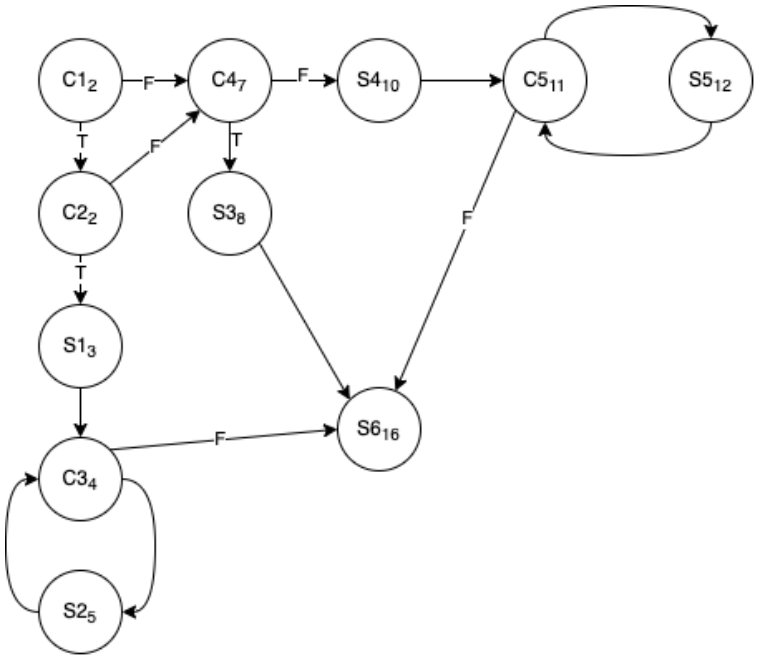
```
1     void Q1(){
2           S1;
3           if(C1){
4                 S2;
5           }
6           else{
7                 S3;
8           }
9           if(C2){
10                S4;
11          }
12          if(C3){
13                S5;
14          }
15          S6;
16    }
```



Path #	Path	C1	C2	C3
1	1-2-3-7-9-12-15	F	F	F
2	1-2-3-7-9-12-13-15	F	F	T
3	1-2-3-7-9-10-12-13-15	F	T	T
4	1-2-3-4-9-10-12-13-15	T	T	T

Problem 2:

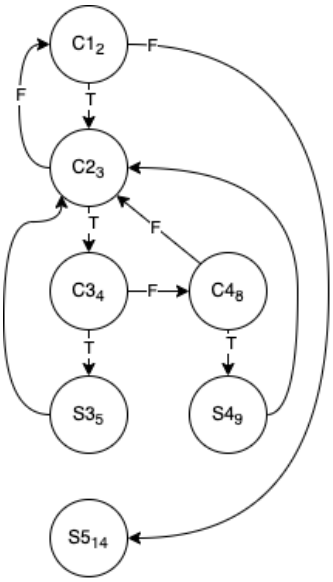
```
1    void Q2(){
2          if(C1&&C2){
3                S1;
4                while(C3){
5                      S2;}
6          }else{
7                if(C4){
8                      S3;
9                }else{
10                     S4;
11                     While(C5){
12                           S5;
13                     }
14               }
15         }
16         S6;
17   }
```



Path #	Path	C1	C2	C3	C4	C5
1	1-2_1-7-8-16	F	X	X	T	X
2	1-2_1-2_2-7-8-16	T	F	X	T	X
3	1-2_1-7-10-11-16	F	X	X	F	F
4	1-2_1-2_2-3-4-5-4-16	T	T	T/F	X	X
5	1-2_1-7-10-11-12-11-16	F	X	X	F	T/F
6	1-2_1-2_2-7-10-11-12-11-16	T	F	X	F	T/F

Problem 3:

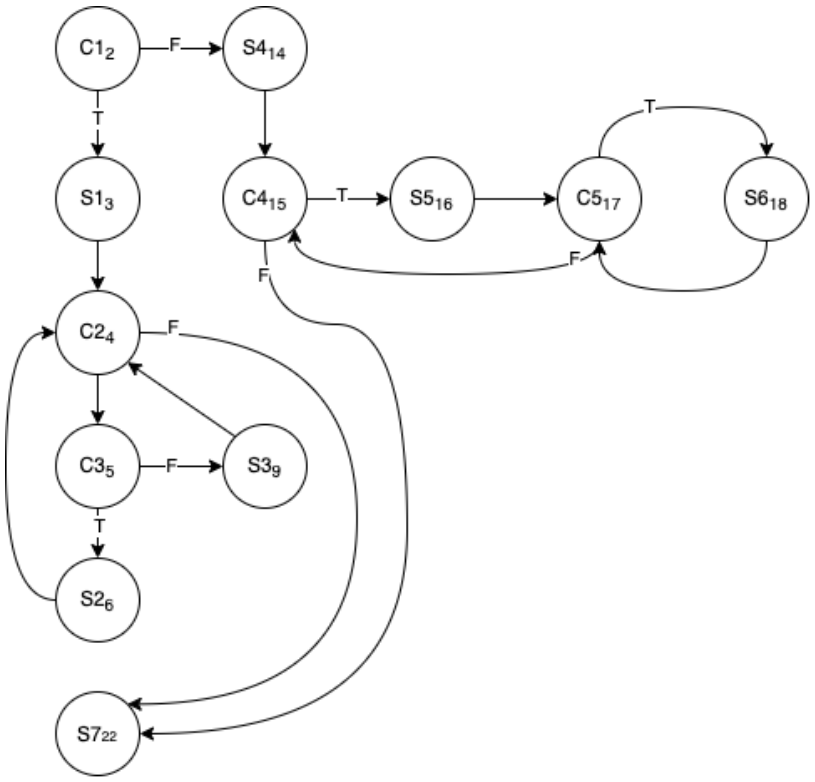
```
1     void Q1(){
2           for(S1;C1;S2){
3                 while(C2){
4                       if(C3){
5                             S3;
6                       }
7                       else{
8                             if(C4){
9                                   S4;
10                            }
11                      }
12                }
13          }
14          S5;
15    }
```



Path #	Path	C1	C2	C3	C4
1	1-2-14	F	X	X	X
2	1-2-3-4-8-9-3-2-14	T/F	T/F	F	T
3	1-2-3-4-8-3-2-14	T/F	T/F	F	F
4	1-2-3-4-5-3-2-14	T/F	T/F	T	T
5	1-2-3-2-14	T/F	T/F	X	X

Problem 4:

```
1     void Q1(){
2           if(C1){
3                 S1;
4                 while(C2){
5                       if(C3){
6                             S2;
7                       }
8                       else{
9                             S3;
10                      }
11                }
12          }
13          else{
14                S4;
15                while(C4){
16                      S5;
17                      while(C5){
18                            S6;
19                      }
20                }
21          }
22          S7;
23    }
```



Path #	Path	C1	C2	C3	C4	C5
1	1-2-14-15-22	F	X	X	F	X
2	1-2-3-4-22	T	F	X	X	X
3	1-2-3-4-5-6-4-22	T	T/F	T	X	X
4	1-2-3-4-5-9-4-22	T	T/F	F	X	X
5	1-2-14-15-16-17-15-22	F	X	X	T/F	F
6	1-2-14-15-16-17-18-17-15-22	F	X	X	T/F	T/F