

COMPUTER SCIENCE




Concurrency Mechanisms 09



Dr. KHALEEL KHAN SIR



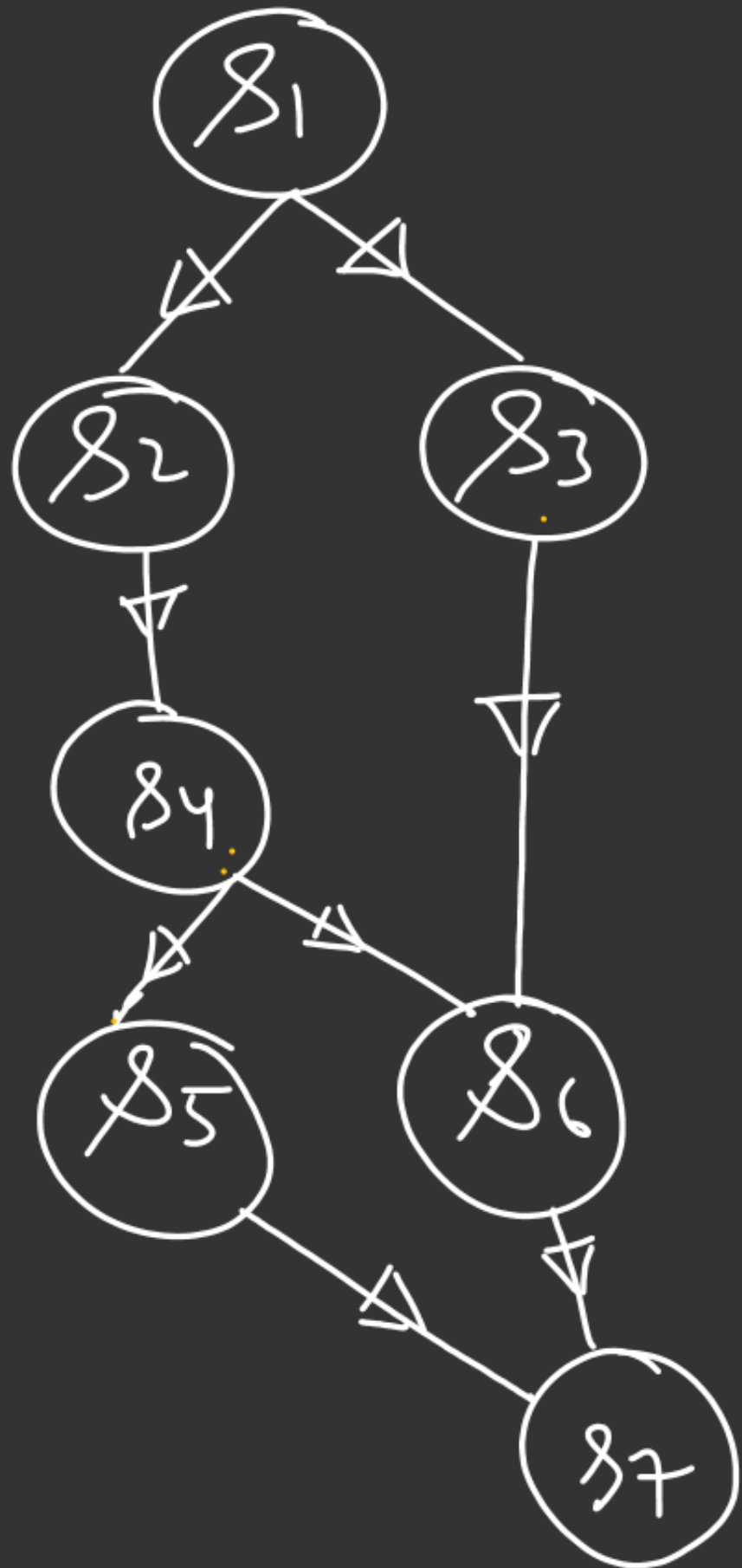
TOPICS TO BE COVERED



Concurrency mechanisms

Session - II: 19/11/2022 Parbegin-Parend with Semaphores

13 SEM $a, b, c, d, e, f, g = \{\emptyset\}$



Parbegin

begin $S_1; v(a); v(b); \text{end}$

begin $P(a); S_2; S_4; v(c); v(d); \text{end};$

begin $P(b); S_3; v(e); \text{end}$

begin $P(c); S_5; v(f); \text{end}$

begin $P(d); P(e); S_6; v(g); \text{end};$

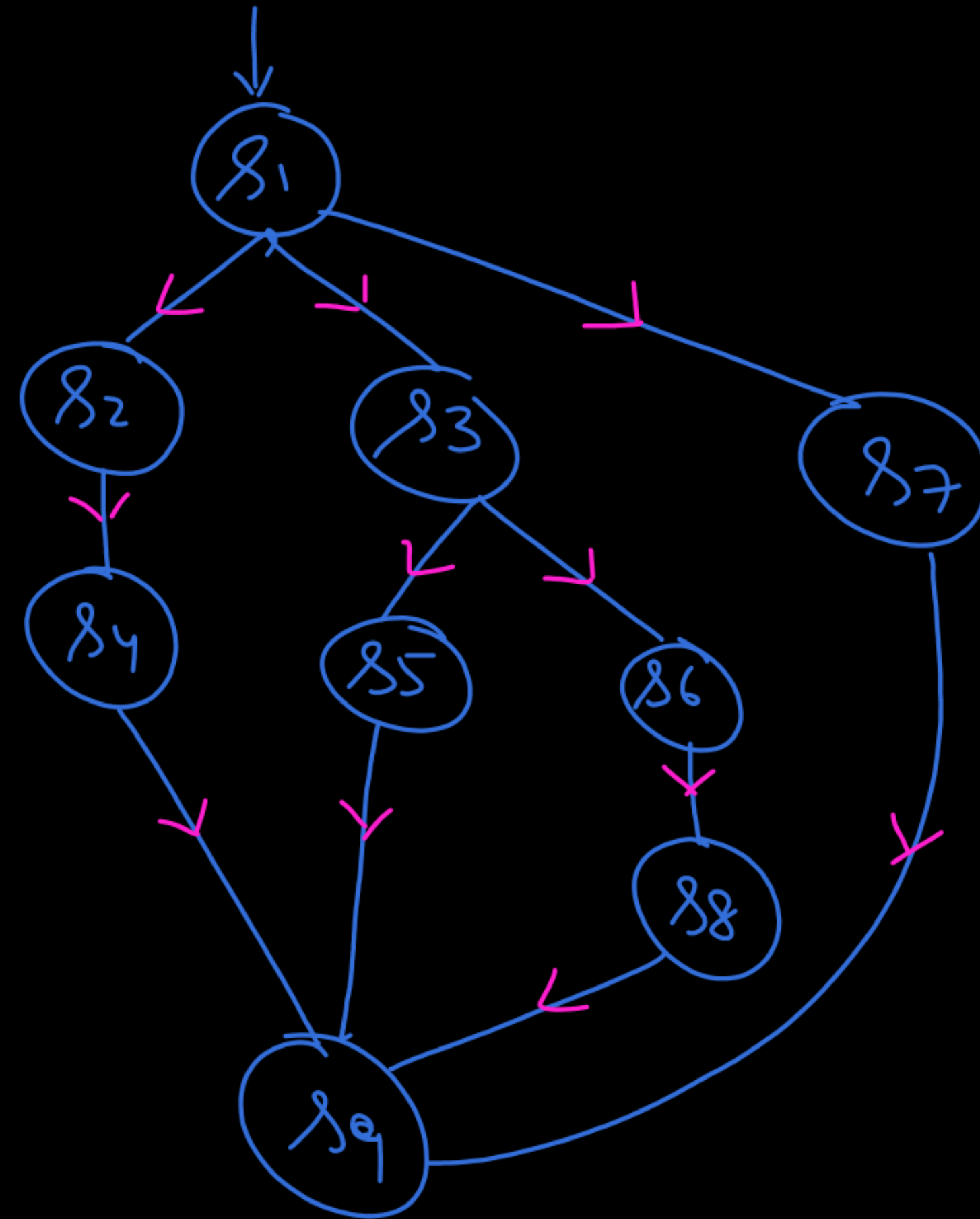
begin $P(f); P(g); S_7; \text{end}$

Parend

16. Draw the precedence graph for the concurrent program given below: 5M

```

S1;
Parbegin
  ① begin
    S2; S4;
  end;
  begin
    S3;
    Parbegin
      S5;
      begin
        S6; S8
      end
    Parend
  end;
end;
S7;
Parend;
S9;
  
```



Q.1

```
void P(void)
{
    A;
    B;
    C;
}
```

```
void Q(void)
{
    D;
    E;
}
```


```
main()
{
    Parbegin
        PO;
        QO;
    Parend
}
```

Indicate the Valid Output Sequences

after the completion of
both P() & Q()

Foundation Problem

- 1. P ; Q ; ✓
 - 2. Q ; P ; ✓
 - 3. P ; Q ; P ; Q ; P ; ✓
 - 4. x
 - 5. x
- without Pre

→ Any Process  Can Start first
→ Anyone Can Complete first
→ Pre Emption is always Possible

- ✓ 1. A B C D E
- 3. A D B E C
- 5. D C E B A

- ✓ 2. D E A B C
- ✗ 4. A E B D C

Q.2

int x = 0, y = 0;

Cobegin

begin

1: x = 1;

* 2: y = y + x;

end

begin

3: y = 2;

* 4: x = x + 3;

end

Coend

Final values of x & y

I) x = 1; y = 2 ✗

II) x = 1; y = 3 ✓

III) x = 4; y = 6 ✓

a)

b)

c)

d)

II & III

Sequential

Sequential

orderings:

- ✓ 1) 1, 2, 3, 4
- ✓ 2) 3, 4, 1, 2
- ✓ 3) 1, 3, 2, 4
- ✓ 4) 3, 1, 2, 4
- ✓ 5) 3, 1, 4, 2
- ✓ 6) 1, 3, 4, 2

x
3 ✗ 1

y
3 ✗ 2 ✓

x
4 ✗

y
6 ✗ 2

(1, 2, 3, 4)

4 ✗
x

✗ ✗ 2
y

→ x = 4; y = 2 ✓

- 3, 1, 2, 4 ✓
- 4, 2, 3, 1 ✗
- 4, 2, 1, 3 ✗
- 3, 4, 2, 1 ✗



Q.3

int x = 0, y = 20;
Bsem mx = 1; my = 1;

Cobegin

begin

P(mx);

x = x + 1; L, Inc

V(mx);

end

begin

~~P(mx);~~ P(my);

x = y + 1;

~~V(mx);~~ V(my);

end

Coend

Final possible values of x 1, 21, 22

x [21] y [20]

i) P₁; P₂ : 21 ✓

x [22] y [20]

ii) P₂; P₁ : 22 ✓



iii : PreEmptive Exec.

x [1] y [20]

~~1~~ R

P₁: L, Inc;

P₂: ✓

P₁: S;

Q.4



5/3/11
HW

```
integer B = 2;  
P1()  
{  
  1. C = B - 1;  
  2. B = 2 * C;  
}  
P2()  
{  
  3. D = 2 * B;  
  4. B = D - 1;  
}  
main()  
{  
  Parbegin  
    P1()  
    P2()  
  Parend  
}
```

The number of distinct values of B is/are

Q.5



```
int count = 0; ✓
```

```
void test()
```

```
{
```

```
    int i, n = 5;
```

```
    for( $i = 1; i \leq n; ++i$ )
```

```
        count = count + 1;
```

```
}
```

```
main()
```

```
{ Parbegin
```

```
    P1. test(); ✓
```

```
    P2. test();
```

```
    Parend
```

```
}
```

What is the minimum and maximum value of count?

Count ≠ 10

P1, P2: 10

Preemptively

(challenge)

** H/w

10

Non-Preemptive

Q.6



N = 2

M = 2

fork L3;

fork L4;

S1;

L1: Join N

S3;

L2: Join M

S5: goto next;

L3: S2;

goto L1;

L4: S4;

goto L2;

next: S6;

Draw the Precedence Graph

Q.7

Consider the following pseudocode, where S is a semaphore initialized to 5 in line#2 and **counter** is a shared variable initialized to 0 in line#1. Assume that the increment operation in line #7 is not atomic

```
1. int counter = 0  
2. Semaphore S = init(5);  
3. void parop(void)  
4. {  
5.     wait (S);  
6.     wait (S);  
7.     counter++;  
8.     signal (S);  
9.     signal (S);  
10. }
```

If five threads execute the function **parop** concurrently, which of the following program behavior(s) is/are possible?

- A** There is a deadlock involving all the threads
- B** The value of counter is 5 after all the threads successfully complete the execution of parop
- C** The value of counter is 1 after all the threads successfully complete the execution of parop
- D** The value of counter is 0 after all the threads successfully complete the execution of parop



**THANK
YOU!**

