Models and semaphores (25 January)

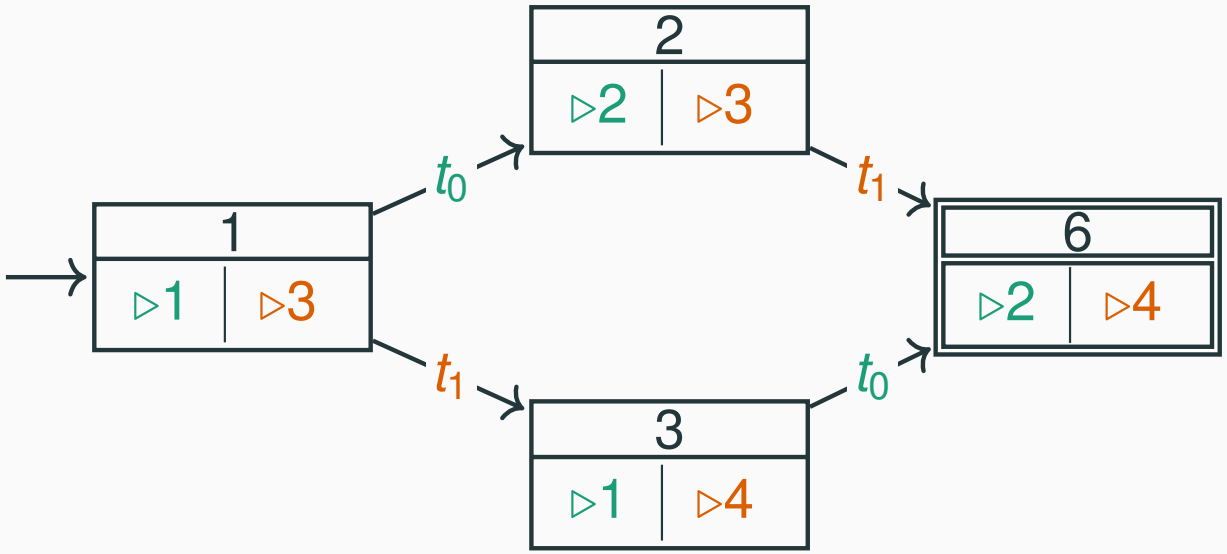
What does Peterson's algorithm achieve?

1. Mutual exclusion using only atomic reads and writes
2. Mutual exclusion and first-come-first-served fairness
3. Mutual exclusion using busy waiting
4. Mutual exclusion using test-and-set operations

What does Peterson's algorithm achieve?

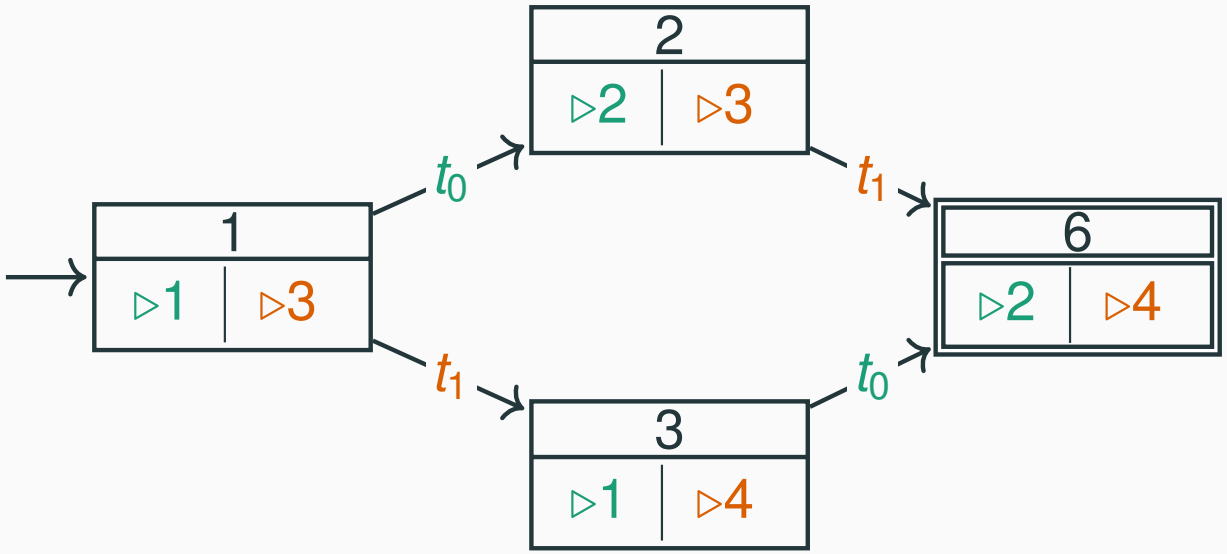
1. Mutual exclusion using only atomic reads and writes
2. Mutual exclusion and first-come-first-served fairness
3. Mutual exclusion using busy waiting
4. Mutual exclusion using test-and-set operations

What properties does the following state/transition diagram show?



1. No deadlocks can occur
2. There are no race conditions
3. No starvation can occur, but deadlocks may occur
4. Neither deadlocks nor race conditions may occur

What properties does the following state/transition diagram show?



1. No deadlocks can occur
2. There are no race conditions
3. No starvation can occur, but deadlocks may occur
4. Neither deadlocks nor race conditions may occur

Which of the following are strategies to avoid deadlocks?

1. Using locks
2. Requiring that all threads acquire locks in the same order
3. Limiting the amount of concurrency
4. Using counting semaphores instead of binary semaphores

Which of the following are strategies to avoid deadlocks?

1. Using locks
2. Requiring that all threads acquire locks in the same order
3. Limiting the amount of concurrency
4. Using counting semaphores instead of binary semaphores

What is the value of semaphore `s` when thread `t` terminates?

```
Semaphore s = new Semaphore(2); // capacity 2
```

thread `t`

```
1  for (int i = 0; i < 10; i++)
2  { s.down();
3    s.up();   }
```

thread `u`

```
for (int i = 0; i < 10; i++)   4
{ s.down();                    5
  s.up();   }                  6
```

1. 1
2. 2
3. Either 1 or 2
4. 0 or 1 or 2

What is the value of semaphore `s` when thread `t` terminates?

```
Semaphore s = new Semaphore(2); // capacity 2
```

thread `t`

```
1  for (int i = 0; i < 10; i++)
2  { s.down();
3    s.up();   }
```

thread `u`

```
for (int i = 0; i < 10; i++)   4
{ s.down();                    5
  s.up();   }                  6
```

1. 1
2. 2
3. Either 1 or 2
4. 0 or 1 or 2

What is the value of semaphore `s` when thread `t` terminates?

```
Semaphore s = new Semaphore(1); // capacity 1
```

thread `t`

```
1  for (int i = 0; i < 10; i++)
2  { s.down();
3    s.up();   }
```

thread `u`

```
for (int i = 0; i < 10; i++)   4
{ s.down();                    5
  s.up();   }                  6
```

1. 1
2. 2
3. Either 1 or 2
4. 0 or 1 or 2

What is the value of semaphore `s` when thread `t` terminates?

```
Semaphore s = new Semaphore(1); // capacity 1
```

thread `t`

```
1  for (int i = 0; i < 10; i++)
2  { s.down();
3    s.up();   }
```

thread `u`

```
for (int i = 0; i < 10; i++)   4
{ s.down();                    5
  s.up();   }                  6
```

1. 1
2. 2
3. Either 1 or 2
4. 0 or 1 or 2