Distributed Systems - Assignment 3

Markus Roth

October 17, 2016

1 Semantics

1.1 async seq void setID(int i); sync seq int getID();

```
Hello World from object : 1
Hello World from object : 2
Hello World from object : 3
Hello World from object : 0
Sum of objects objects id = 6
```

Since the setId method is asynchronous, all the three setID calls are started shortly after each other and happen concurrently. Since the set and get methods are sequential, the get methods waits until the last set method is finished, therefore giving the correct sum of 6.

1.2 async seq void setID(int i); sync conc int getID();

```
Sum of objects objects id = -4
Hello World from object : 1
Hello World from object : 3
Hello World from object : 2
Hello World from object : 0
```

Now the get method is concurrent rather than synchronous, it will get the values immediately (there is no defined order between conc and seq calls). This does not give the set methods time to actually set the values in time, therefore we get the initial values of -1 added four times.

1.3 async conc void setID(int i); sync seq int getID();

```
Sum of objects objects id = -4
Hello World from object : 0
Hello World from object : 3
Hello World from object : 1
Hello World from object : 2
```

There is no defined order between conc and seq methods. As the set is async and conc, they are all executed at the same time and they return immediately. The seq getId method runs immediately, and adds the initial -1 value four times.

1.4 async conc void setID(int i); sync conc int getID();

```
Sum of objects objects id = -4
Hello World from object : 1
Hello World from object : 0
Hello World from object : 2
Hello World from object : 3
```

Since both methods are conc, they don't wait for each other. Since the setters are async, the getters immediately start running and don't wait for the setters to be finished. Therefore, we get the initial values added four times.

1.5 sync seq void setID(int i); sync seq int getID();

```
Hello World from object : 0
Hello World from object : 1
Hello World from object : 2
Hello World from object : 3
Sum of objects objects id = 6
```

Both methods are sync and seq, so the caller always wits for the method to be over before calling the next one. This gives us the correct sum at the end.

1.6 sync seq void setID(int i); sync conc int getID();

```
Hello World from object : 0
Hello World from object : 1
Hello World from object : 2
Hello World from object : 3
Sum of objects objects id = 6
```

Since the set calls are sync, the program only calls the getters after all the setters have returned. Therefore, we get the correct sum of 6 at the end.

1.7 sync conc void setID(int i); sync seq int getID();

```
Hello World from object : 0
Hello World from object : 1
Hello World from object : 2
Hello World from object : 3
Sum of objects objects id = 6
```

Since the set calls are sync, the program only calls the getters after all the setters have returned. Therefore, we get the correct sum of 6 at the end.

1.8 sync conc void setID(int i); sync conc int getID();

```
Hello World from object : 0
GHello World from object : 1
Hello World from object : 2
Hello World from object : 3
Sum of objects objects id = 6
```

Since the set calls are sync, the program waits to execute the next call until they are done. Therefore, we get the setters executed in order and the correct sum at the end.