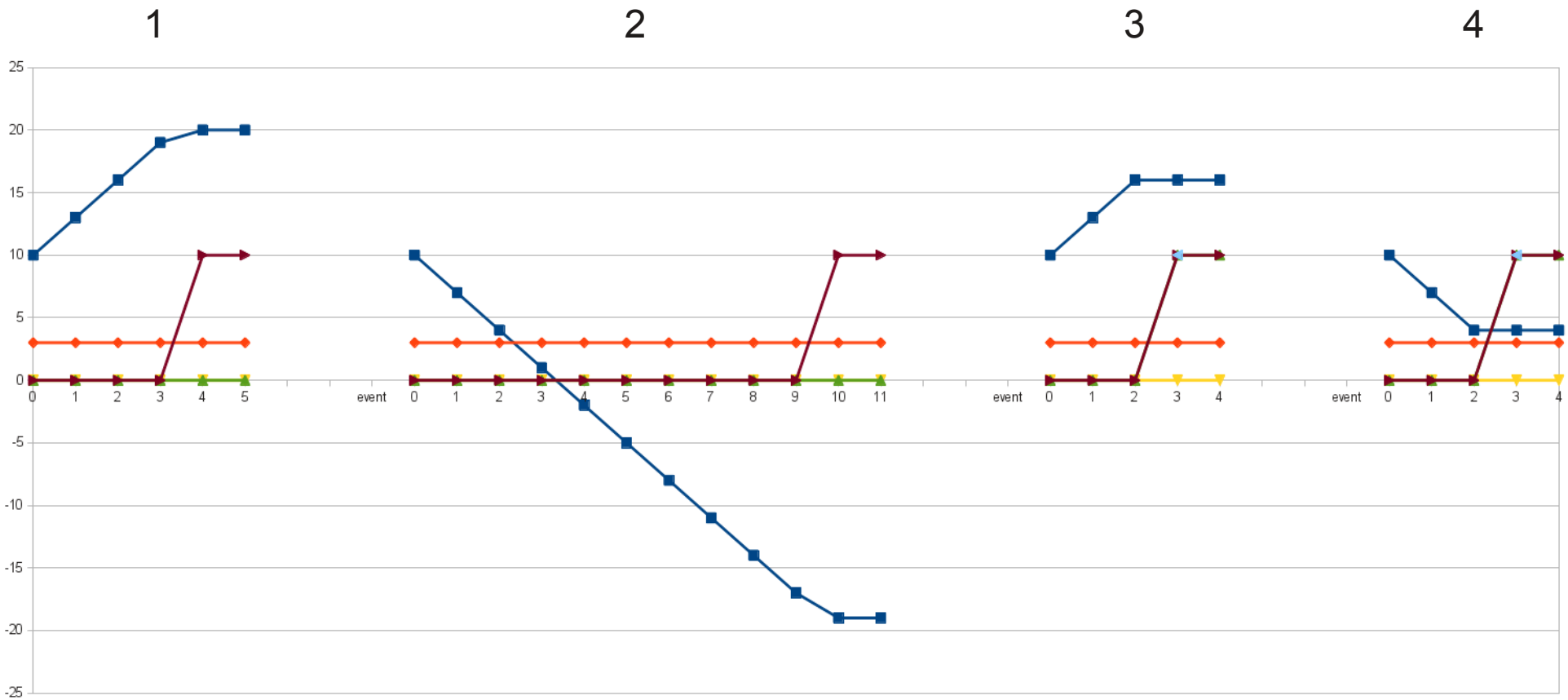
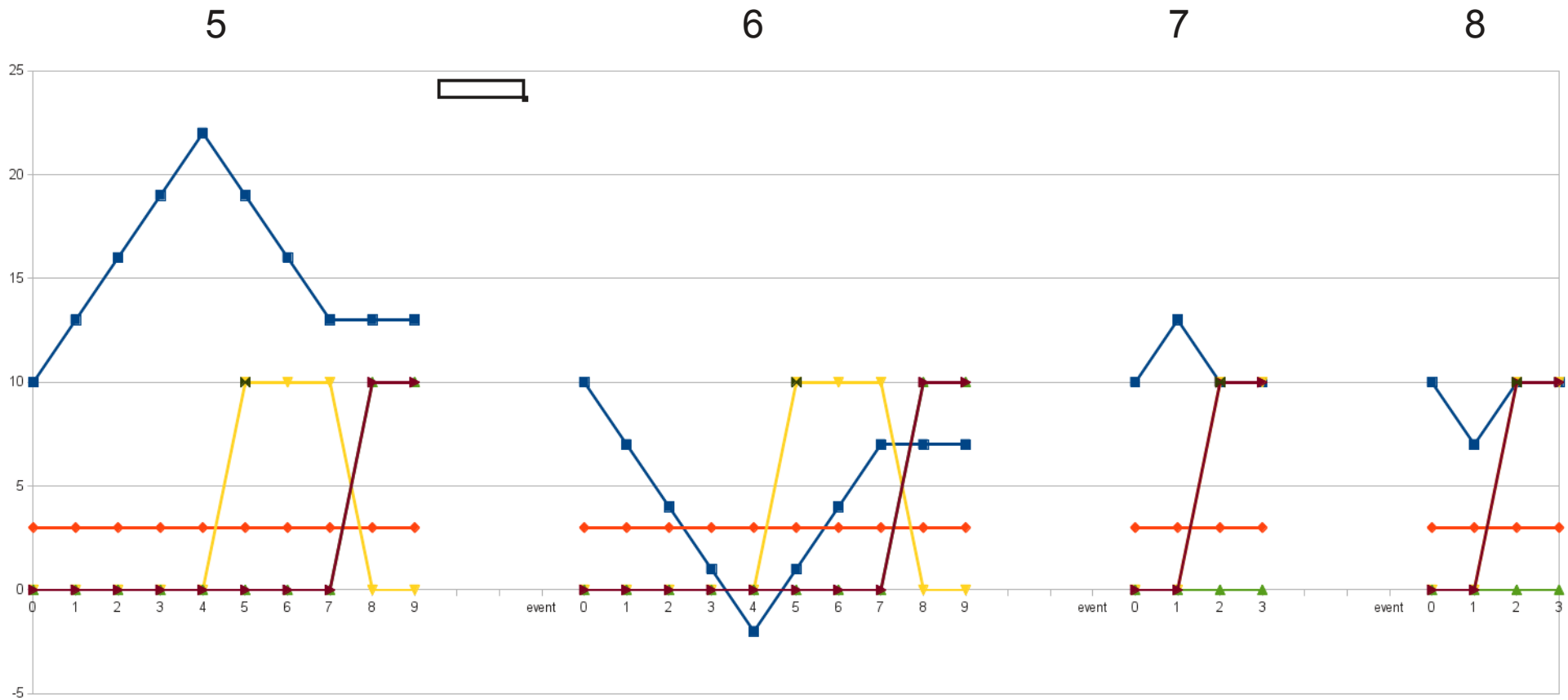
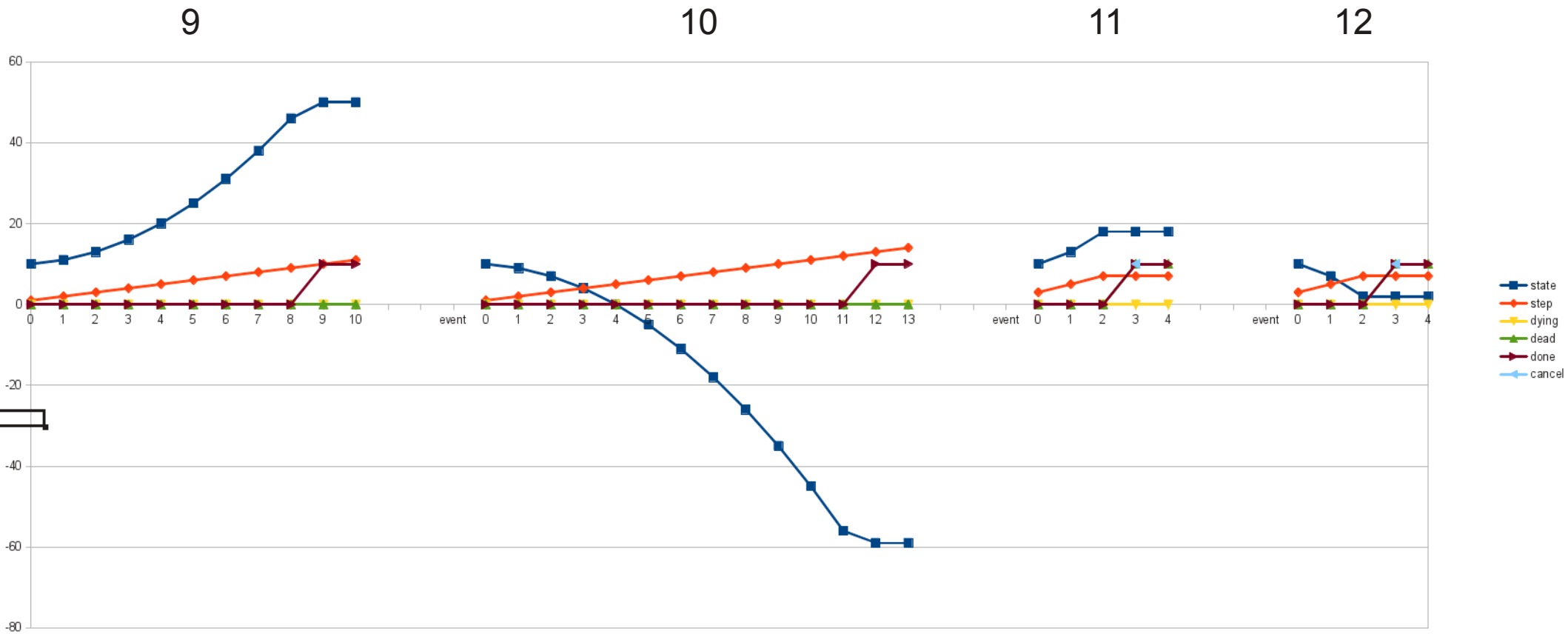
```
// ## LINEAR
// ## increasing
MyComponentLinear componentLinear1 = new MyComponentLinear("linearUp", 10, 20, 3);

execute(componentLinear1, 0, 0);

// ## decreasing
MyComponentLinear componentLinear2 = new MyComponentLinear("linearDown", 10, -19, 3);

execute(componentLinear2, 0, 0);

// ## increasing cancel middle
MyComponentLinear componentLinear3 = new MyComponentLinear("linearUpCancelMiddle", 10, 20, 3);

execute(componentLinear3, 3, 0);

// ## decreasing cancel middle
MyComponentLinear componentLinear4 = new MyComponentLinear("linearDownCancelMiddle", 10, -21, 3);

execute(componentLinear4, 3, 0);

// ## increasing terminate middle
MyComponentLinear componentLinear5 = new MyComponentLinear("linearUpTerminateMiddle", 10, 40, 3);

execute(componentLinear5, 0, 5);

// ## decreasing terminate middle
MyComponentLinear componentLinear6 = new MyComponentLinear("linearDownTerminateMiddle", 10, -25, 3);

execute(componentLinear6, 0, 5);

// ## increasing terminate start
MyComponentLinear componentLinear7 = new MyComponentLinear("linearUpTerminateStart", 10, 25, 3);

execute(componentLinear7, 0, 2);

// ## decreasing terminate start
MyComponentLinear componentLinear8 = new MyComponentLinear("linearDownTerminateStart", 10, -25, 3);

execute(componentLinear8, 0, 2);

// ## NONLINEAR
// ## increasing
MyComponentNonlinear componentNonlinear1 = new MyComponentNonlinear("nonlinearUp", 10, 50, 1, 1);

execute(componentNonlinear1, 0, 0);

// ## decreasing
MyComponentNonlinear componentNonlinear2 = new MyComponentNonlinear("nonlinearDown", 10, -59, 1, 1);

execute(componentNonlinear2, 0, 0);

// ## increasing cancel middle
MyComponentNonlinear componentNonlinear3 = new MyComponentNonlinear("nonlinearUpCancelMiddle", 10, 50, 3, 2);

execute(componentNonlinear3, 3, 0);

// ## decreasing cancel middle
MyComponentNonlinear componentNonlinear4 = new MyComponentNonlinear("nonlinearDownCancelMiddle", 10, -51, 3, 2);

execute(componentNonlinear4, 3, 0);

// ## increasing terminate middle
MyComponentNonlinear componentNonlinear5 = new MyComponentNonlinear("nonlinearUpTerminateMiddle", 10, 70, 3, 2);

execute(componentNonlinear5, 0, 5);
```

```
private void execute(final A_Component component, final int cancelBeforeUpdate, final int terminateBeforeUpdate)
{
   assert (cancelBeforeUpdate >= 0) : cancelBeforeUpdate;
   assert (terminateBeforeUpdate >= 0) : cancelBeforeUpdate;

   System.out.println("\n,,," + component.toString());

   boolean isDone = false;

   int step = 0;

   System.out.println("event,state,step,dying,dead,done,cancel,terminate");

   System.out.println(step + "," + component.getState_() + "," + component.getStep_() + "," + map(component.isDying_()) + "," + map(component.isDead_())
                      + "," + map(isDone));

   do
   {
      ++step;

      if (step == cancelBeforeUpdate)
      {
         component.cancel_();
      }
      else if (step == terminateBeforeUpdate)
      {
         component.terminate_();
      }

      isDone = component.updateState_();

      System.out.print(step + "," + component.getState_() + "," + component.getStep_() + "," + map(component.isDying_()) + "," + map(component.isDead_())
                       + "," + map(isDone));

      if (step == cancelBeforeUpdate)
      {
         System.out.print(",10,");
      }
      else if (step == terminateBeforeUpdate)
      {
         System.out.print(",,10");
      }

      System.out.println();
   }
   while (!isDone);

   ++step;

   isDone = component.updateState_();

   System.out.println(step + "," + component.getState_() + "," + component.getStep_() + "," + map(component.isDying_()) + "," + map(component.isDead_())
                      + "," + map(isDone));
}
```

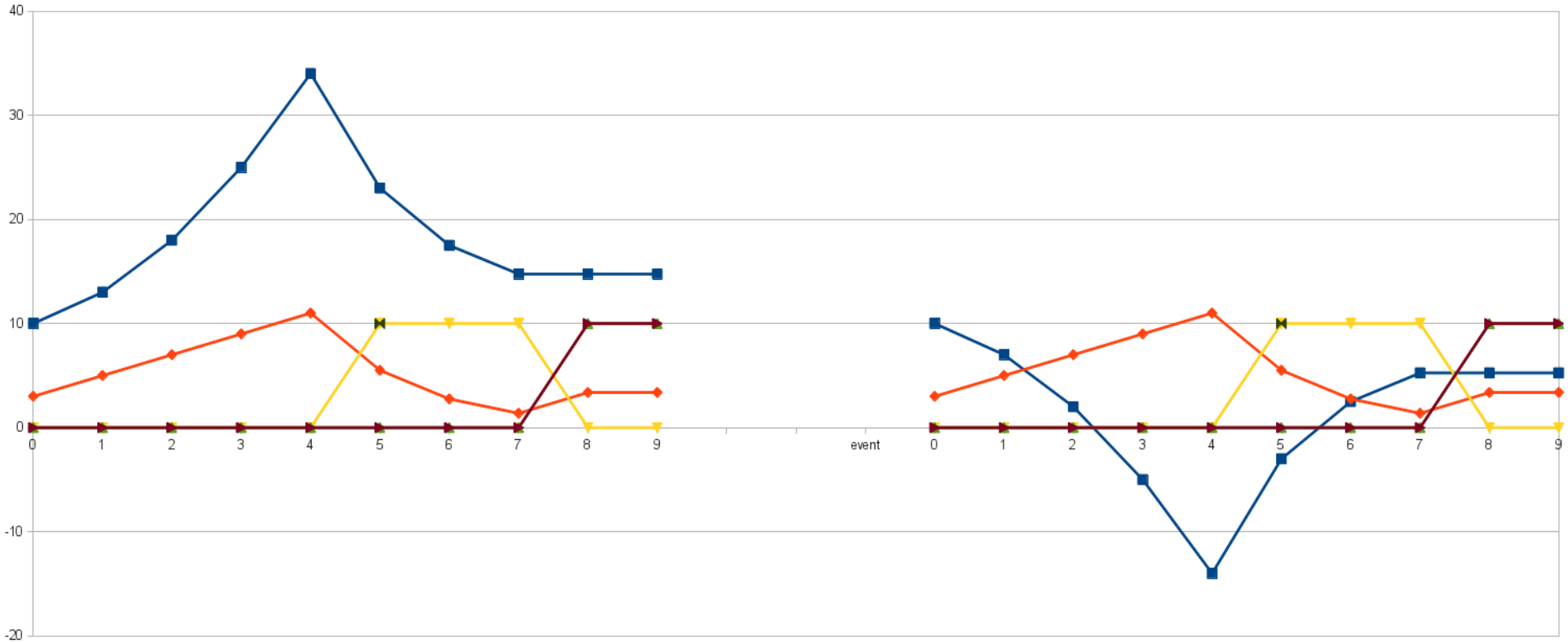

1
2
3
4
25
20
15
10
5
0
-5
-10
-15
-20
-25
0 1 2 3 4 5
event 0 1 2 3 4 5 6 7 8 9 10 11
event 0 1 2 3 4
event 0 1 2 3 4
state
step
dying
dead
done
cancel
terminate


5
6
7
8
25
20
15
10
5
0
-5
0 1 2 3 4 5 6 7 8 9
event
0 1 2 3 4 5 6 7 8 9
event
0 1 2 3
event
0 1 2 3
state
step
dying
dead
done
cancel
terminate


9
10
11
12
60
40
20
0
-20
-40
-60
-80
0 1 2 3 4 5 6 7 8 9 10
event 0 1 2 3 4 5 6 7 8 9 10 11 12 13
event 0 1 2 3 4
event 0 1 2 3 4
state
step
dying
dead
done
cancel

13

14


40
30
20
10
0
-10
-20
0
1
2
3
4
5
6
7
8
9
event
0
1
2
3
4
5
6
7
8
9
state
step
dying
dead
done
cancel
terminate