Formal Methods in Software Engineering

Paul Wild

Tutorial session 2

Tuesday 29th October, 2019

Dining Philosophers

Setting

- n philosophers are sitting around a circular dining table and there are n forks, one between each pair of adjacent philosophers.
- Whenever a philosopher is hungry, they first grab the fork to their left, then the fork to their right, and then they eat.
- When they are done they release both forks.

Dining Philosophers

Setting

- n philosophers are sitting around a circular dining table and there are n forks, one between each pair of adjacent philosophers.
- Whenever a philosopher is hungry, they first grab the fork to their left, then the fork to their right, and then they eat.
- When they are done they release both forks.

Modelling in Promela

We will now try to model this in Promela.

- The philosophers will be modelled as processes (`proctype`).
- The forks will be modelled as channels (`chan`).

What are some properties of this system that we may want to check?

Dining Philosophers

Deadlocks

- Model checking showed us that a deadlock can occur when all philosophers simultaneously grab the fork to their left.
- What are some possible changes to our model that ensure that this is no longer possible?

Dining Philosophers

Deadlocks

- Model checking showed us that a deadlock can occur when all philosophers simultaneously grab the fork to their left.
- What are some possible changes to our model that ensure that this is no longer possible?

Approaches

1. Use atomic sequences. How?

Dining Philosophers

Deadlocks

- Model checking showed us that a deadlock can occur when all philosophers simultaneously grab the fork to their left.
- What are some possible changes to our model that ensure that this is no longer possible?

Approaches

1. Use atomic sequences. How?
2. Make a change to the rule that everybody needs to first grab the fork to their left. How?

Dining Philosophers

Deadlocks

- Model checking showed us that a deadlock can occur when all philosophers simultaneously grab the fork to their left.
- What are some possible changes to our model that ensure that this is no longer possible?

Approaches

1. Use atomic sequences. How?
2. Make a change to the rule that everybody needs to first grab the fork to their left. How?
3. Introduce a waiter that needs to be asked for permission first. How?

A lightweight puzzle

Ferryman, Wolf, Goat, and Cabbage

Let us try to formalise and solve this popular puzzle in `Promela`. For those who don't know: A ferryman is supposed to take a wolf, a goat, and a cabbage to the other side of a river. However, his passengers are governed by natural instincts (well, except for the cabbage), so if he leaves the wolf together with the goat, or the goat together with the cabbage, on one side, he will have one passenger less. The question is: can he fulfil his task avoiding this scenario?

- Why could model checking be of any use here?

A lightweight puzzle

Ferryman, Wolf, Goat, and Cabbage

Let us try to formalise and solve this popular puzzle in `Promela`. For those who don't know: A ferryman is supposed to take a wolf, a goat, and a cabbage to the other side of a river. However, his passengers are governed by natural instincts (well, except for the cabbage), so if he leaves the wolf together with the goat, or the goat together with the cabbage, on one side, he will have one passenger less. The question is: can he fulfil his task avoiding this scenario?

- Why could model checking be of any use here?
- ! Output of a counter-example in case a specification is false