Computer-Aided Verification: Revision Lecture

Sam Barrett

January 9, 2021

1 Modelling Sequential and Parallel Systems

The key ideas and concepts of this week of the submodule are:

- Verification requires precise models of system behaviour over time in order to reason about it.
- We utilise transition systems (LTSs)
- A key concept relating to our use of LTSs is non-determinism, which is the idea that we either have choices at individual nodes or overall alternative paths through the system.
- LTSs and non-determinism are particularly useful when looking at parallel programs where we can compose multiple LTSs into a single LTS synchronously or asynchronously.

Following this section of the submodule you should be able to:

- Construct an LTS representing a sequential or parallel imperative program.
- Construct an LTS representing a reactive, multi-component, system.
- Construct the (synchronous/ asynchronous) parallel composition of LTSs.

Points to remember:

- When drawing an LTS, always make sure the states are clear.
- When deciding what the states *are* make sure they contain all the required information to verify properties or move to the next state.
- When building the product of two LTSs make sure both parts of the component in the product state. I.e. make sure the product LTS has all the information from both the component LTSs.
- When constructing and exploring LTSs be sure to be exhaustive, work as if you are the modelchecker. Be sure to **never** duplicate a state.
- Be sure to layout the states logically to make them easier to read/mark.

2 Temporal Logic

In the second week we focused on specifying properties for these models, to do this we used temporal logic. We assumed a linear time view of execution.

We classified our properties into 3 categories: Invariants, safety properties and liveness properties.

We used LTL equivalence to prove or disprove these properties.

Based on the material from this week you should be able to:

- Identify classes that a property, written as LTL or natural language, belongs to and explain why
- understand the semantics of LTL, i.e. given a LTS and a LTL formula you should be able to say whether it is satisfied.
- Be able to translate a property from natural language to LTL
- Be able to use LTL equivalences to prove or disprove equivalence of two LTL formula.

3 Model Checking

4 Software Model Checking