Process algebraic system verification

Muck van Weerdenburg

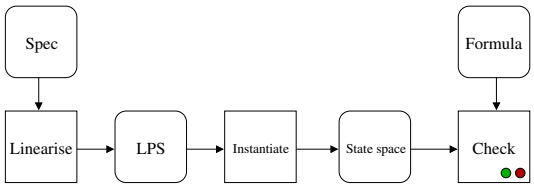
M.J.van.Weerdenburg@tue.nl

Process Algebra

Verification of systems with process algebra is a formal and effective method. Toolsets, like mCRL2[1], have proven to be extremely effective at showing whether protocols and distributed systems behave as desired. On a regular basis design flaws in real life systems are exposed.

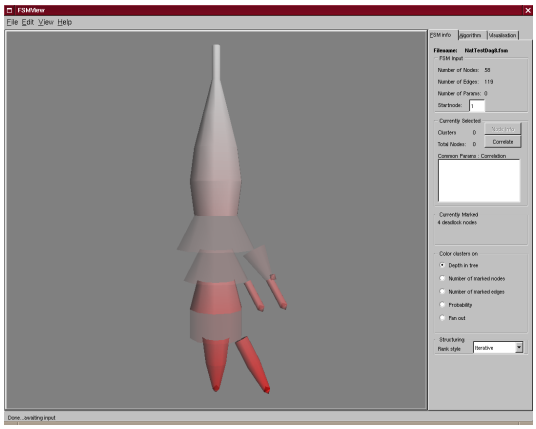
Verification Process

Given a specification of the system under study, we transform it to an intermediate format, a linear process specification (or LPS), which allows for effective analysis. Typically, one then generates the state space of the system, after which formulas, describing the desired properties of the system, can be verified.



Verification process

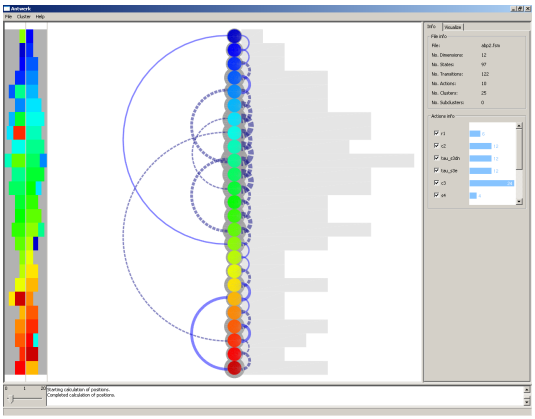
We can also easily detect irregular behaviour with use of state space visualisation tools. For example, in the visualised system below there are three branches which seem to indicate anomalous behaviour.



Visualised state space of a Petri net[2]

Process Manipulation

Special tools can manipulate LPSs such that, for example, unreachable states are removed beforehand and performance of state space generation increases significantly. State spaces of over 10^9 states can be verified by using distributed tools. Larger systems can be handled by using confluence or abstraction.



Clustering of state space of a simple protocol[3]

Future Work

We are continuously working on new and better techniques to be able to handle even larger systems. One aim is to avoid generating state spaces altogether and use symbolic reasoning on the higher level behaviour descriptions (especially LPSs). For example, we can combine an LPS and a requirement formula into a parameterised boolean equation system (PBES) and try to solve it.

1. Information about the mCRL2 language and toolset can be found at http://www.mcrl2.org/.
2. Image generated with FSMView, which is made by Frank van Ham.
3. Image generated with a tool made by Hannes Pretorius.