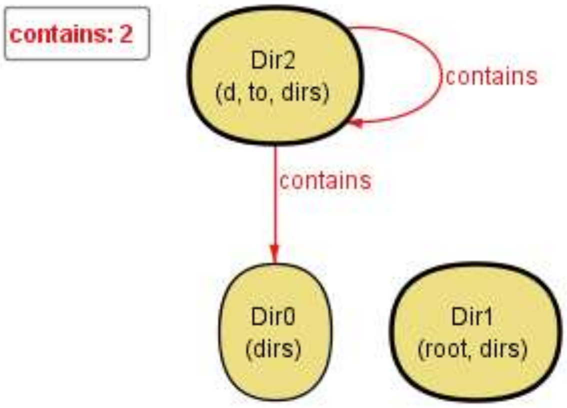
1. The mars climate orbiter was a good example of this. This project failed because there was a conceptual mistake in the units that would be used.
2. Alloy finds a counterexample where dir2 is unreachable from the root because it was moved into itself. You can tell this since Dir2 contains itself and its label contains both ‘d’ and ‘to’.



3. This argument was about how unit testing focuses on the small issues that cause bugs. If there is an issue with the design of the system, unit tests only show you small places where this occurs. The forest is the design and the tree is the thing that we are unit testing. This relates to Ptolemy’s system because unit tests helps you fine-tune sections of the code, but if the fundamental idea of the system is flawed, you will never have a good solution.
4. To verify integrated circuits, developers have been using a method called model checking. The key difference between this and the author’s technique is that integrated circuits are exhaustively tested. We cannot test every state that a program will go through since there are too many. In software verification, we have to abstract the design so that we can test it without going through each possibility.
5. They found flaws with a scheduling system and a creeping angle of gantry.
6. You want to identify the components of your system. You want to come up with some abstractions of the problem so that you can build a model. Once you have your abstractions you want to come up with the rules for the system. Once that is set up, you can run some tests on the model and make some assertions to verify that your design does what you think. If there is an assertion that fails, you need to tweak the design or the model and repeat this process until all assertions are met.