CONFORMIQ

A Conformiq White Paper

ARTIFICIAL INTELLIGENCE
in Automated Test Design

Artificial Intelligence or AI is intelligence exhibited by a machine. The term AI is applied when a machine mimics a cognitive function such as learning and problem solving.

AI is making all sorts of headlines lately and the recent innovation around AI has made it a hot topic especially in the media. The media focus has primarily been around Machine Learning (ML for short) and quite often the terms AI and ML are used interchangeably. However, AI research is actually much more than "just" Machine Learning and in fact the central problems in AI research include things such as:

- *reasoning,*
- *planning,*
- *knowledge,*
- *learning and*
- *natural language processing*

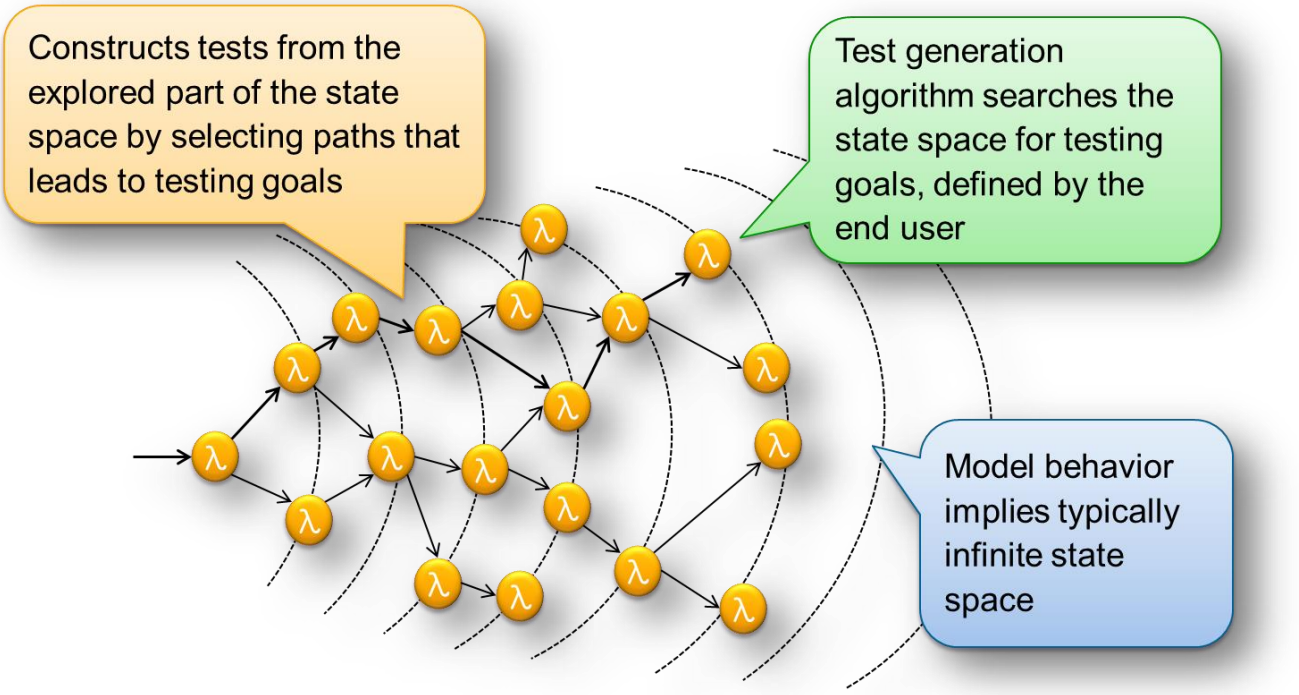
In this article I will, on a very abstract and high level, walk through the core of the Conformiq test generation technology and describe how the results of AI research have been applied with great success in the Conformiq automated test design software. But let's first start by looking what the technology is all about.

CONFORMIQ TECHNOLOGY

Instead of using test cases or test models, Conformiq technology derives tests automatically from *executable system models*, i.e., artifacts that represent and model the *desired behavior* of the application under test. Conformiq uses *semantics-driven methods* for generating test suites, which means that test generation is guided by deep state space analysis of the behavior implied by the model, instead of being based on syntactic analysis or simple heuristics. The algorithmic approach creates full *test oracles* (tests that embed a mechanism for determining whether test has passed or failed, with full test data, time, and expected results) that will automatically design and create test cases with data combinations used as stimuli to the application under test combined with the exact expected response from the system.

PLANNING AND SEARCHING

The core of Conformiq test generation technology is a semantics driven, symbolic execution based, test generation algorithm. The algorithm traverses a part of the (usually infinite) state space of the system model. The explored part in itself is also infinite, but yet is only a part of the whole state space. The test generation algorithm searches this part of the state space for testing goals and the number of testing goals in a model depends on the testing heuristics selected by the user. The algorithm then constructs tests from the explored part of the state space by selecting paths that lead to testing goals, then converting those paths to tests. Every input on an execution path (to the system model) becomes a test stimulus (to be sent to the real system), and every output of an execution path becomes an expected output (to be verified during test execution).

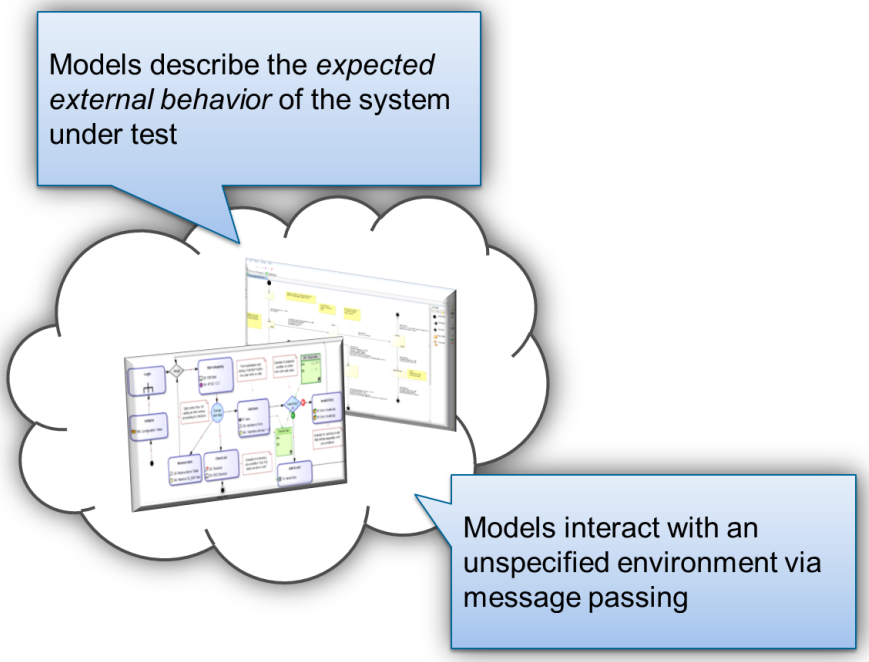


Algorithms that are based on state space enumeration are all computationally difficult and typically suffer from a problem known as state space explosion. This means that the state or search space gets astronomically large compared to the size of the program itself. Even worse, the actual state space of a Conformiq system model is typically infinite, even though in practice it does not matter whether it is mathematically truly infinite or only extremely large.

The planning and search algorithm of the test generation core that selects which paths to expand and to what extent has been carefully tuned during the last decade with details that are a trade secret to create an appropriate size state space for the model being tested. The algorithm has also been carefully crafted for not only multi-threaded, but additionally for fully distributed, parallel test generation operation that generates the same set of test cases deterministically, regardless of the number of processor cores, their speed, and load. This parallel test generation algorithm parallelizes efficiently, providing savings of up to 90% of test generation time by scaling from one to sixteen processor cores, and even more when deployed on a computation cluster with tens or even hundreds of cores. This is important as it improves the productivity of model driven test engineers by cutting down wait time before seeing their newly generated tests.

REASONING AND PROBLEM SOLVING

Conformiq models are said to be "open" meaning they communicate with an unspecified environment, i.e. the user doesn't need to describe the environment of the system or the tester. This means that the model cannot directly be explored or simulated (the environment is unspecified so the inputs from the environment are unspecified as well), so in order to explore the model the algorithm needs to generate inputs to the model. One could generate the inputs "randomly" like some random testing approaches, but this is not a scalable approach which is very easy to demonstrate. This is why the Conformiq test generation algorithm uses reasoning and problem solving to efficiently handle the unspecified inputs by deploying an algorithmic approach. The ability to reason about the unspecified inputs and derive concrete values for the inputs is a cornerstone of the full Conformiq technology. This is actually why the approach is called symbolic execution – the input messages are internal within the tool and are represented as symbolic values that are only fixed later by applying an algorithmic approach.



FINDING IMPORTANT TESTS

There are numerous tools that boast their capability of creating huge number of tests as if that would be the end goal. Creating large numbers of tests is not a problem and, on the contrary, spitting out a huge number of tests is actually quite easy. What is difficult is to find those important tests; tests that actually find issues in the application; tests that are cost-effective to execute and manage; tests that are understandable and compact. What really matters is finding good quality tests.

In order to generate good quality tests, Conformiq technology employs an internal mechanism to make predictions on ways the system might fail. This technology approach combines a large set of heuristics that aim to find parts of the application logic most likely be defective such as:

- Boundary value analysis
- Combinatorial testing such as pair wise and n-wise
- Mutation testing
- MC/DC

On top of these heuristics, the risk based testing feature enables user to identify model areas to be prioritized in test selection.

SEARCHING AND OPTIMIZATION

Because there are often many different ways to put together a set of test cases that cover every testing target reached, the symbolic state space explorer typically finds huge (impractical) number of test cases. In order to keep things practical, the tool triggers yet another algorithm that selects an optimized subset of all the test cases that cover all the testing goals with minimum cost. This step eliminates redundancy in the generated test collection which makes sure that each and every test case has a reason to be in the test suite. The reason is documented from the generated requirements traceability and model coverage reports available to the user.

NATURAL LANGUAGE PROCESSING

A topic that hasn't been discussed so far in this article is where Conformiq is applying numerous AI techniques including Natural Language Processing (NLP) for automatic model creation. This is a capability to "reverse engineer" a formal model from existing testing assets such as manually written test cases. With this capability, duplicate tests and subtests are automatically eliminated from the import to generate an optimized model. It is a highly valuable and important piece of technology allowing legacy to be maintained when deploying Conformiq technology but also allowing the user to jump start their modeling efforts, including for additions to Brownfield projects, further improving their productivity and making the whole process less error prone.

SUMMARY

Conformiq technology has greatly benefitted from the ongoing AI research and over the years has increasingly incorporated more algorithmic advances into its core technology. At the same time, Conformiq has significantly advanced the state-of-the-art by taking this research further. These advances make Conformiq a unique solution in the testing market where no other tool or technology really provides anything comparable. However, Conformiq is still not fully there yet and is constantly looking for ways to improve its capabilities further so that the users of Conformiq products can be even more productive.

Author Kimmo Nupponen is the Chief Scientist at Conformiq.

www.conformiq.com