Towards Constraint-Based Local Search for Automatic Test Data Generation

Arnaud Gotlieb*, Nadjib Lazaar*, Yahia Lebbah**

* IRISA-INRIA, Campus Beaulieu, 35042 Rennes Cedex, France
Arnaud.Gotlieb@irisa.fr, Nadjib.Lazaar@irisa.fr

** Université d'Oran Es-Senia, B.P. 1524 EL-M'Naouar, 31000 Oran, Algeria
ylebbah@gmail.com

Constraint-Based Testing was introduced fifteen years ago, in the context of mutation testing [1], to generate test cases by using constraint solving techniques. In our previous work [2, 3], we built a constraint-based model for generating automatically test data that cover all the decisions of a C program. Constraints are solved by using classical deductive methods such as domain reduction over finite domains and complete labelling. However, we found that some constraint systems cannot be solved in a reasonable amount of time with these techniques. Hence, we started looking at *Constraint-Based Local Search (CBLS)*, a framework introduced by Michel and Van Hentenryck [4], to speed up the test data generation process in Constraint-Based Testing. CBLS is a generic paradigm that proposes to implement multiple Local Search techniques such as Hill-climbing and simulated annealing by using the high-level concepts of invariants and differentiable objects. Invariants are simple relations expressed over incremental variables that must be maintained under assignments of new values while differentiable objects are relations that can be queried to evaluate the effect of local moves. Their significance comes from the many algorithms that evaluate the effect of various moves before selecting the neighbor to visit. Unlike invariants, a differentiable object can be queried through an interface shown below to evaluate the effect of local moves such as assignments and swaps.

```
Abstract class Constraint {
  bool    isTrue();                          // returns the Truth value of the Constraint
  name[]  getVariables();                    // returns the variables involved in the constraint
  int     violations();                      // returns its violation degree (constraint-dependant)
  int     violations(name x) ;               // returns the viol. degree associated to a single variable
  int     getAssignDelta(name x, int v) ;    // returns the variation in the viol. degree when x evaluates to v
  int     getSwapDelta(name x, name y) ;     // returns the variation when variables x and y are swapped
}
```

Invariants and differentiable objects form the modelling component of CBLS. The search component aims at simplifying the implementation of heuristics and meta-heuristics. It includes several interesting control structures such as selectors, abstractions to manipulate valuations and advanced simulation techniques. By separating the constraint model and search component, Constraint-Based Local Search is a unique architecture that facilitate the implementation of local search.

Perspectives of this work include the implementation of CBLS within our testing tool and an experimental validation on large C programs.

References

[1] R. DeMillo and J. Offut. Constraint-based automatic test data generation. *IEEE Transactions on Software Engineering*, 17(9):900–910, September 1991.

[2] A. Gotlieb, B. Botella, and M. Rueher. Automatic test data generation using constraint solving techniques. In *Proc. of ISSTA'98*, pages 53–62, Clearwater Beach, FL, USA, March 1998.

[3] A. Gotlieb, B. Botella, and M. Watel. Inka: Ten years after the first ideas. In *19th Int. Conf. on Soft. and Systems Eng. and their Applications (ICSSEA'06)*, Paris, France, Dec. 2006.

[4] L. Michel and P. Van Hentenryck. *Constraint-Based Local Search*. MIT Press, 2005.