Poster Presentation : Probabilistic Choice Operators as Global Constraints : Application to Statistical Software Testing

Matthieu Petit and Arnaud Gotlieb

IRISA-INRIA, Campus de Beaulieu, 35042 Rennes cedex, France
{Matthieu.Petit, Arnaud.Gotlieb}@irisa.fr

Probabilistic Concurrent Constraint Programming (PCCP) [3] is an extension of Concurrent Constraint Programming (CCP) [5] where probabilistic choice operators are introduced to represent the randomness or uncertain behaviour of processes. A probabilistic choice between two processes can be though of as flipping a coin : head the first process is triggered, tail it is the second. Based on this theoretical framework, it seems possible to extend the classical CCP over finite domains framework [4] with probabilistic choice operators.

Our aim is to define probabilistic choice operators as global constraints of the CCP over finite domains paradigm [4] and to apply this framework to deal with a specific Software Testing problem [1]. Global constraints are a good way for giving global semantics to complex constraints. Furthermore, such operators appear to the user like single constraints and so can be awaked and treated efficiently by the constraint propagation algorithm. A part of our work is to establish the relationships between probabilistic choice operators, global constraints and the PCCP semantic framework.

Gupta et al. pioneered the inclusion of probabilistic choice operators in CCP to address several applications areas, such as stochastic processes [3].

Example 1 (extracted from [3])
choose X ***from*** $\{0,1\}$ ***with distribution*** $\{\frac{1}{2},\frac{1}{2}\}$ ***in*** [***tell*** $(X = Z)$]
$\|$ ***choose*** Y ***from*** $\{0,1\}$ ***with distribution*** $\{\frac{1}{2},\frac{1}{2}\}$
in [***if*** $Z = 1$ ***then tell*** $(Y = 1)$].

After the example running, Z is constrained to 0 with a probability $\frac{1}{2}$ *(event* $X = 0$*), to 1 with a probability* $\frac{1}{4}$ *(event* $X = 1 \wedge Y = 1$*) and the process fails with a probability* $\frac{1}{4}$ *(event* $X = 1 \wedge Y = 0$*).*

Our current implementation includes new global constraints of SICStus Prolog's library clp(FD) like the `choose_unif` global constraint.

Example 2 (Example 1 running)

```
?-choose_unif(X,0..1,[X#=Z]),choose_unif(Y,0..1,[ask(Z#=1,Y#=1)]).
```

Here is an output sequence of several launches (we get the distribution $p(Z = 0) = \frac{1}{2}$, $p(Z = 1) = \frac{1}{4}$ *and* $p(\texttt{fail}) = \frac{1}{4}$*, as expected).*

X=1, Y=1, Z=1	X=0, Y=0, Z=0	no	X=1, Y=1, Z=1	X=0, Y=1, Z=0	no	X=0, Y=1, Z=0	X=0, Y=0, Z=0	...

In [2], we proposed to transform the problem of the automatic test data generation into a Constraint Logic Programming over finite domains problem. Our work aims at extending this framework to address a new problem : the statistical structural testing application [6]. In this testing method, we use a probabilistic test data selection, i.e. the use of a random test data generator to cover a selected testing criterion (such as the all-paths criterion). It requires constructing a non-uniform generator over the input domain of the program, which aims at giving the highest probability to activate each criterion element, including the most difficult to reach. This allows a good coverage of test criteria and reduces the cost of the test oracle construction.

Example 3 *Test criterion : covering all-paths.*

```
if X = 0 then C1 else C2 ; if Y = 0 then C3 else C4
```

Statistical structural testing aims at constructing a random test data generator where the events $(X = 0 \wedge Y = 0)$, $(X = 0 \wedge Y \neq 0)$, $(X \neq 0 \wedge Y = 0)$ *and* $(X \neq 0 \wedge Y \neq 0)$ *have the same probability* $(\frac{1}{4})$. *Here is a first model of this problem in PCCP :*

$$
\begin{array}{l}
\textbf{\textit{choose}}\ U\ \textbf{\textit{from}}\ \{0,1\}\ \textbf{\textit{with distribution}}\ \{\frac{1}{2},\frac{1}{2}\} \\
\quad \textbf{\textit{in}}\ [\textbf{\textit{if}}\ U=0\ \textbf{\textit{then}}\ [\textbf{\textit{tell}}\ (X=0) \parallel C1] \parallel \textbf{\textit{if}}\ U=1\ \textbf{\textit{then}}\ [\textbf{\textit{tell}}\ (X\neq 0) \parallel C2]] \\
\parallel \textbf{\textit{choose}}\ V\ \textbf{\textit{from}}\ \{0,1\}\ \textbf{\textit{with distribution}}\ \{\frac{1}{2},\frac{1}{2}\} \\
\quad \textbf{\textit{in}}\ [\textbf{\textit{if}}\ V=0\ \textbf{\textit{then}}\ [\textbf{\textit{tell}}\ (Y=0) \parallel C3] \parallel \textbf{\textit{if}}\ V=1\ \textbf{\textit{then}}\ [\textbf{\textit{tell}}\ (Y\neq 0) \parallel C4]]\,.
\end{array}
$$

To conclude, we believe that implementing probabilistic choice operators as global constraints is interesting. In the one hand, this gives the possibility of using a powerful probabilistic choice operator in the CCP over finite domains framework. In the other hand, it seems to be adequate to address the problem of random test data generator for statistical structural testing. The implementation as an extension of the clp(FD) library of SICStus Prolog is in progress.

References

1. Richard A. DeMillo and A. Jefferson Offutt. Constraint-based automatic test data generation. *IEEE Trans. Softw. Eng.*, 17(9):900–910, 1991.
2. A. Gotlieb, B. Botella, and M. Rueher. A clp framework for computing structural test data. In *Computational Logic (CL)*, pages 399–413, LNAI 1891, 2000.
3. V. Gupta, R. Jagadeesan, and P. Panangaden. Stochastic processes as concurrent constraint programs. In *Symposium on POPL*, pages 189–202, 1999.
4. P. Van Hentenryck, V. A. Saraswat, and Y. Deville. Design, implementation, and evaluation of the constraint langage cc(FD). *Journal of Logic Programming*, 1998.
5. V. A. Saraswat, M. Rinard, and P. Panangaden. Semantic foundations of concurrent constraint programming. In *Symposium on POPL*, pages 333–352, 1991.
6. P. Thévenod-Fosse and H. Waeselynck. An Investigation of Statistical Software Testing. *Journal of Software Testing, Verification and Reliability*, 1991.