

Colloque du
Département d'informatique et de recherche opérationnelle

Stefanie Schirmer

*Laboratoire d'ingénierie des ARN, Université de Montréal, et Practical
Computer Science, Bielefeld University*

**Enjoy Dynamic Programming in Bellman's GAP:
Grammars, algebras & products
for dynamic programming at ease**

Résumé :

Dynamic programming (DP) is a classic programming technique that helps to solve complex problems from various fields and topics, such as combinatorics, optimization, and biological sequence analysis. DP algorithms are commonly expressed in the form of recurrences, and reuse of intermediate results happens via accessing a table of solutions at the right position. Developing and implementing such an algorithm can be tedious, error-prone and the resulting programs can be hard to debug.

Algebraic Dynamic Programming (ADP) is a formal framework for specifying dynamic programming algorithms on sequences in an abstract and modular way using tree grammars and evaluation algebras, without the need for subscripts to access intermediate results. By separating the problems of search space description, candidate description, candidate evaluation and tabulation, it helps us in thinking about and formalizing dynamic programming algorithms.

Bellman's GAP is a programming language for DP-Problems derived from the ADP formalism, which allows us to describe the problem in abstract building blocks, and yet get efficient programs. The compiler of the language handles tabulation and access of intermediate results, and we can focus on the algorithm itself, and equip it with different evaluation strategies and scoring schemes.

This is joint work with G. Sauthoff, R. Giegerich.



Le jeudi 28 novembre 2013, 15h30, pavillon André-Aisenstadt, salle 3195
Information : Gilles Brassard, brassard@iro.umontreal.ca