

# Recursion, Parallelism and Sequentiality From Prolog to Feynman Path Integral

Mircea Namolaru, Thierry Goubier

mircea.namolaru@gmail.com, thierry.goubier@cea.fr

December 16, 2019

## ▶ **Background**

- ▶ Computation restructuring work started at CEA. Support for machine learning algorithms, huge data sets, online processing.
- ▶ Hard mathematical and modelling problem. Used powerful "weapons" to attack it (report on arxiv). Theory of groups, Measure (Coloring), Bayesian networks.
- ▶ Post CEA (main author): Detailed mathematical description. Simplification/generalization (after reading more of H. Weyl works). Relations with other mathematical theories.

## ▶ **Structure**

- ▶ Results. Contributions.
- ▶ The problem, using an example (composition).
- ▶ Basic mathematical ideas underlying the approach.
- ▶ The solution exemplified.
- ▶ Conclusions.

- ▶ **Main results:**
- ▶ **(1) Structuring of array/relational computations, into a kind of tree-like recursive tree.**
- ▶ **(2) The structuring defines a partition (measure) of the computation domain. Its enumeration, may be view as summation (integration) over this measure.**
  - ▶ **Parallelization of array/relational computations.** Enumeration complexity reduced from exponential in number of variables to width of a tree.
  - ▶ **Structuring method, simple in a few dimensions, mechanical in many.** Theoretically and computationally scalable.
  - ▶ Rubik's cube and coloring used to exemplify (up to a point) abstract ideas.
  - ▶ Computation: Loop structures.

# Contributions (1)

- ▶ **Mathematical modelling. Use of group representation/invariant theory powerful algorithms for performing the computation structuring.**
  - ▶ Mathematically: Weyl: “The theory of groups and quantum mechanics” (1931).
  - ▶ Existing works and results could be view from the perspective of group theory representation/invariants.
  - ▶ For instance, affine partitioning framework (Lim) and Pugh framework/polyhedra model (shortly discussed further, no prior assumption assumed).
  - ▶ **Have we solved this problem, yet ?**. Sets a very high mathematical standard for works presenting new algorithms in related areas.
  - ▶ Algebrization of analysis: relations with Bayesian Networks, Convolutional Neural Networks (discussed elsewhere).

## Contributions (2)

- ▶ **Measure associated to structuring, that may be view as inferring a Dynamic Bayesian Networks. Made possible the use of Bayesian Networks that revolutionized expert system technology**
  - ▶ BN (Perl) uses independence to reduce the enumeration complexity from exponential in number of variable, to width of a tree.
  - ▶ BN solved the monotonic reasoning crisis for artificial intelligence.
  - ▶ "Rubik's cube and Bayesian Networks".
  - ▶ Many works use for experimental validation, heuristics (cost model), relating between computation, hardware and performance.
  - ▶ **Problem: Does the heuristics works for new computations ? hardware ?**
  - ▶ Using Measure (coloring), Bayesian Networks could construct sound ones.

## The problem

- ▶  $f(M, N) = \sum_{P, Q} g(M, N) h(P, Q)$ .  
 for (M=0; M<MS; M++)  
   for (N=0; N<NS; N++)  
     for (P=0; P<PS; P++)  
       for (Q=0; Q<QS; Q++)  
         f(M, N) += g(M, N) h(P, Q)
  
- ▶ Lexicographic order traversal of the cartesian product of variables  $M, N, P, Q$  where  $M = \{0, 1, \dots, MS\}$ , etc.
- ▶ Traversal complexity exponential in the number of variables. Reduce it, while preserving the dependencies.
- ▶ Dependencies: A partial order between variables:  
 $Q < N, P < M$  and  $Q, N < P, M$ .
- ▶ Simpler problem:  $MS = NS = PS = QS = 2$ . AI/ML people probably will know what to do !

# Group representation and invariants

- ▶ **Vector space  $V$ , linearity**, linear combination,  $\sum \alpha_i v_i$  where  $v_i \in V$ .  $\alpha_i$  are coefficients (numbers).
- ▶ **Groups of linear transformation  $T$  over  $V$ .**
- ▶ **Group representation**, to each  $a \in V$  corresponds a linear transformation  $T_a$ ,  $a \rightarrow T(a)$ . For any  $a, b \in V$ ,  $a + b \rightarrow T_a T_b$ .
- ▶ **Problem: Covering of the vector space, via compositions of representations.**
- ▶ Transformation  $T$ , invariant subspace  $S$ , i.e.  $T(S) = S$ . Invariants reduce the space of solutions. Ex: Rubik's cube.
- ▶ **Solution: Irreducible representation/invariant spaces.**

# Linear transformations

- ▶ For a linear transformation, the "type" of the coefficients are important.
- ▶ **Klein, Earlegen program:** geometries are defined by invariance under groups of linear transformations.
  - ▶ Affine: integer coefficients. Preserves the coliniarity, coplanarity etc. (Pugh framework/polyhedra model, affine transformations).
  - ▶ Projective: rational coefficients. Preserve incidence relations. (Affine partitioning framework, invariant spaces of projective transformations)
  - ▶ Orthogonal: real coefficients. Preserve distance.
  - ▶ Invariants, relations between the are known.
- ▶ **Problem: Tiling.** Measure.
- ▶ Translations and Rotations. Complex numbers. Solutions of polynomial equations.
- ▶ Wavefront and Convolutional Parallelism.

- ▶ **Solution: Linear transformations whose coefficients are a group with "good" properties** induce the "tiling". We consider groups whose elements are powers of two.
- ▶ Mapping of the group elements to variable, compatible with dependencies.
  - ▶ The mapping transforms the original space into another one, where only vectors with size powers of two are used.
  - ▶ The group induce an irreducible representation/invariant spaces in this space. May be view as a coloring of this space (where a color denote a group element).
  - ▶ **The coloring of a  $k$ -cube main diagonal induce a coloring of a  $k$ -cube points.** Coloring of  $p^k$  points, ( $p$  cube size) with  $k$  colors compatible with dependencies, where  $k = \sum c_i m_i$ .
  - ▶ Symmetries and anti-symmetries. Ex: Rubik's cube.

# The example

- ▶ Mapping. Two colors, each with multiplicity 2.  
 $M, P \rightarrow 2^2 = 4, N, Q \rightarrow 2^1 = 2$ , compatible with dependencies.
- ▶ Triangulated matrix.

|   |   |   |   |   |
|---|---|---|---|---|
| Q | 2 | 2 | 4 | 4 |
| N |   | 2 | 4 | 4 |
| P |   |   | 4 | 4 |
| M |   |   |   | 4 |

- ▶ The loop below is the “conjugate” of the triangulated form. (Reverse order of variables  $Q, N, P, M$  an alternative).

```
for (M=0;M<8,M+=4)
  for (P=M,P<M+8;P+=4)
    for (N=M+P;N<M+P+4;N+=2)
      for (Q=N;Q<N+4;Q+=2)
        f(M,N-P-M)+=g(M,N-P-M)h(P-M,Q-N)
```



# Conclusions

- ▶ Structuring (parallelization) of complex computations is mathematically a formidable task.
- ▶ Taking advantage of the increasingly complex hardware architecture and fitting the structure to it is another one.
- ▶ We need powerful tools for solving them. This paper advocates:
  - ▶ **The use of group theory/representations for computation structuring.**
  - ▶ **The importance of associated measures (coloring) to structuring, enabling the use of Bayesian Networks technology.**