Towards a Group Theoretical Model for Algorithm Optimization

Sven DE SMET [a,1]
[a] *Student at Ghent University*

Abstract. This paper proposes to use a group theoretical model for the optimization of algorithms. We first investigate some of the fundamental properties that are required in order to allow the optimization of parallelism and communication. Next, we explore how a group theoretical model of computations can satisfy these requirements. As an application example, we demonstrate how this group theoretical model can uncover new optimization possibilities in the polyhedral model.

Keywords. Group theory, Algorithm, Parallelism, Communication, Polyhedral model

1. Introduction

In order to optimally benefit from the incessant increase in parallel computational capacity in modern architectures, the available parallelism in our algorithms must increase accordingly. Since manual parallelisation and optimization of communication in a parallel architecture requires considerable effort from the developer, we must resort to automatic techniques.

In recent decades, much research in the field of automatic optimization of algorithms has focused on structured computations that can be represented in the polyhedral model. This model allows to represent and transform a large class of computationally intensive algorithms based on techniques from linear algebra, the theory of polyhedra and linear programming. Parallelism and communication can be optimized with affine spacetime and data mappings. However, it is well known that some computations contain more parallelism than can be extracted with affine mappings [1] and a similar observation can be made about the optimization of communication. Several techniques have been proposed to resolve this issue [2,3,4,5,6], but each of them suffers from drawbacks such as restricted applicability or complexity issues.

The polyhedral model and the techniques developed for it have several properties that enable efficient optimization of the algorithms they can represent and manipulate:

- A representation of structured sets of operations and data elements and the relation between these sets.
- A representation of hierarchies of partitions of the operations and data elements and the relation between these partitions.

[1] sven.desmet@cubiccarrot.com, http://www.cubiccarrot.com/salmoc/

- The ability to efficiently enumerate the cells (and subcells) of partition hierarchies.
- The ability to efficiently compute an abstraction of the transitive closure of relations on the considered sets.

The reader is strongly encouraged to read [7] and [8] in order to acquire a better understanding of the importance of these properties and the remainder of this paper.

In this paper, we investigate how group theory may allow to represent and manipulate significantly more general algorithms than allowed by the polyhedral model while maintaining these important properties.

2. A Group Theoretical Model

In the polyhedral model, the sets of operations and data elements are restricted to sets that are unions of affine grids intersected with polyhedral bounds or parametrised families of these sets. Each operation or data element is then identified with an integer vector.

For the model based on group theory we aim to represent more general sets. To this end we consider permutation groups. A **permutation representation** (G, A, f) is a group G that acts on a set A through a function $f \in G \to \mathcal{I}A$ (where $\mathcal{I}A$ is the set of bijections on A) such that $f(1)$ is the identity transformation and the function composition operation $\circ$ is compatible with the group operation,

$$\overset{G^2}{\underset{g,h}{\forall}} f(h) \circ f(g) = f(g \cdot h)$$

For a partition $\mathbf{P}$ of a set P, the permutation group $(G, \mathbf{P}, f)$ allows to represent a set of partitions that are coarser than $\mathbf{P}$ using the group structure of G. For any subgroup S of G, two elements of P are contained in the same cell of the partition of P induced by S through the permutation group $(G, \mathbf{P}, f)$ iff their containing cells of $\mathbf{P}$ lie on the same orbit of S. An enumerable set of generators thus allows us to represent a structured partition.

If we use the normal subgroups of a group G to construct partitions in this way, then these partitions can themselves be used as the sets of a permutation group to construct coarser partitions. For a normal subgroup $N \trianglelefteq G$, the quotient group G/N can be used as a group for a permutation representation of the coarser partitions. The lattice of normal subgroups of G thus induces a hierarchy of partitions on P that is a lattice-structured abstraction of the set of all partitions of P. Refining and coarsening the respective partitions induced by two groups H and F can be done by intersecting groups H and F and considering the group generated by the union of generators of H and F respectively.

If the elements of the subgroups of G can be enumerated efficiently, then the cells and subcells in a partition hierarchy can also be enumerated efficiently, since for any normal subgroup N of G, both N and G/N are subgroups of G.

If we let an element of G encode a relation on the elements of P (such as a dependence relation), then for a set $H \subseteq G$ of relations, the subgroup generated by H represents an efficient abstraction of the transitive closure of these relations.

Relations on separate sets can be described using the cardinal product construction of the groups on the respective sets.

3. Application Example: The Polyhedral Model

In [7] a first application example of the group theoretical model has been provided. The sets that were considered are $\mathcal{Z}$-polyhedra. The set of bijections on the $\mathcal{Z}$-polyhedra that were considered are the integral translations. The group G was identified with the set of bijections. Since the integral translations are commutative, every subgroup of G is a normal subgroup. The commutativity allowed to significantly simplify the necessary operations.

It is natural to ask whether a more general application of the group theoretical model to the polyhedral model is possible by considering more general transformations as bijections. Since a bijective transformation must be invertible, the group of unimodular transformations is precisely the set of bijections we can consider.

The group of unimodular transformations is a subgroup of the general linear group. The structure of the general linear group has been well-studied. The Bruhat decompositions of the general linear group allow us to analyze the elements of the general linear group as a combination of permutation and triangular matrices. For the unimodular transformations, the determinant of these matrices must be equal to 1 or -1. We can therefore reduce the problem to the study of unitary skewing transformations and permutation transformations.

The group structure of permutation transformations is identical to the symmetric group of a finite set with a number of elements equal to the number of dimensions. The normal structure of the finite symmetric groups is well-studied and it is possible to scan the elements of the symmetric group and its subgroups.

We thus see that the group of unimodular transformations provides an interesting candidate to further extend the optimization methods that have been developed for the polyhedral model. For instance, the example provided by Lim and Lam as an example of a computation in the polyhedral model that contains no communication-free parallelism ([9,10], example 2), can be parallelised without requiring communication by considering the symmetries of the computation which can be efficiently represented using a permutation representation based on unimodular transformations. The group theoretical approach provides an alternative to the index set splitting approach that does not suffer from the increase in complexity that results from splitting the index sets into multiple subsets (when multiple symmetries that lead to multiple splits are involved).

4. Related work

Besch and Pohl studied a first, more restricted application of group theory to parallelisation that did not consider the theory of permutation representations or the lattice structure of normal subgroups and the partitions it induces [11].

5. Conclusion

This paper is a slightly modified version of a draft paper that was submitted to ParCo 2011 and is very preliminary. Since I do not have the resources to complete this paper by increasing its clarity, extending the experimental evaluation and adding a section on related work, I'm making it available so that it may be useful to others.

6. Acknowledgements

Useful information or feedback was gratefully received from anonymous reviewers.

Initial research that provided the starting point for this paper was supported in part by a PhD grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen)[2] and a BOF/GOA project[3] and was also morally supported by the Flexware (IWT/060068) project.

References

[1] Beletska, A., San Pietro, P.: Extracting coarse-grained parallelism with the affine transformation framework and its limitations. Electronic Modelling **5** (2006) 1–14

[2] Beletska, A., Bielecki, W., San Pietro, P.: Extracting coarse-grained parallelism in program loops with the slicing framework. In: Proceedings of the Sixth International Symposium on Parallel and Distributed Computing, Washington, DC, USA, IEEE Computer Society (2007) 29

[3] Yu, Y., D‘Hollander, E.H.: Non-uniform dependences partitioned by recurrence chains. In: Proceedings of the 2004 International Conference on Parallel Processsing (ICPP‘04), IEEE (8 2004) 100–107

[4] Yu, Y., D‘Hollander, E.H.: Partitioning loops with variable dependence distances. In Lilja, D., ed.: Proceedings of the 2000 29th International Conference on Parallel Processing. Volume I., Toronto, Canada, The IEEE Computer Society (8 2000) 209–218

[5] Griebl, M., Feautrier, P.A., Lengauer, C.: Index set splitting. Int. J. Parallel Programming **28**(6) (2000) 607–631

[6] Griebl, M., Feautrier, P.A., Lengauer, C.: On index set splitting. (October 1999) 274–282

[7] De Smet, S.: Parallelisation with a grid abstraction. (2011)

[8] De Smet, S.: A lattice model for the optimization of communication in parallel algorithms. (2011)

[9] Lim, A.W., Lam, M.S.: Maximizing parallelism and minimizing synchronization with affine transforms. In: Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, ACM Press (1997) 201–214

[10] Lim, A.W., Lam, M.S.: Maximizing parallelism and minimizing synchronization with affine partitions. Parallel Computing **24**(3–4) (May 1998) 445–475

[11] Besch, M., Pohl, H.W.: A group theoretic approach to data and code partitioning. (1998) 575–583

[2] from 01/01/2008 to 31/08/2009

[3] from 01/07/2006 to 31/12/2007