Polyhedral Compilation Foundations

Louis-Noël Pouchet

pouchet@cse.ohio-state.edu

Dept. of Computer Science and Engineering, the Ohio State University

Jan 25, 2010

888.11, Class #1

Objectives for this Class

Objectives for the next few lectures:

- Learning the basic mathematicals concept underlying polyhedral compilation
- Build a survival kit of mathematical results
- Get a good understanding of **why** and **how** things are done

What this class is not about:

- Non topic-related mathematics, advanced polyhedral maths
- Standard program optimization

Requirements: basic (linear) algebra concepts, basic compilation concepts

Polyhedral Program Optimization: a Three-Stage Process

1 Analysis: from code to model

→ Existing prototype tools

- PoCC (Clan-Candl-LetSee-Pluto-Cloog-Polylib-PIPLib-ISL-FM)
- URUK, Omega, Loopo, . . .

→ GCC GRAPHITE (now in mainstream)

→ Reservoir Labs R-Stream, IBM XL/Poly

Polyhedral Program Optimization: a Three-Stage Process

1 Analysis: from code to model

→ Existing prototype tools

- PoCC (Clan-Candl-LetSee-Pluto-Cloog-Polylib-PIPLib-ISL-FM)
- URUK, Omega, Loopo, . . .

→ GCC GRAPHITE (now in mainstream)

→ Reservoir Labs R-Stream, IBM XL/Poly

2 Transformation in the model

→ Build and select a program transformation

Polyhedral Program Optimization: a Three-Stage Process

1 Analysis: from code to model

→ Existing prototype tools

- PoCC (Clan-Candl-LetSee-Pluto-Cloog-Polylib-PIPLib-ISL-FM)
- URUK, Omega, Loopo, ...

→ GCC GRAPHITE (now in mainstream)

→ Reservoir Labs R-Stream, IBM XL/Poly

2 Transformation in the model

→ Build and select a program transformation

3 Code generation: from model to code

→ "Apply" the transformation in the model

→ Regenerate syntactic (AST-based) code

Today

Stage 1: from syntactic code to polyhedral representation

- Modeling iteration domains with polytopes

Underlying mathematical concepts:

- Convexity
- Polyhedra (bounded, rational, integer and parametric)
- Lattices

Next weeks: (1) data dependence, (2) scheduling, (3) optimization I, (4) optimization II, ...

Motivating Example [1/2]

Example

```
for (i = 0; i < 3; ++i)
  for (j = 0; j < 3; ++j)
    A[i][j] = i * j;
```

Program execution:

```
1: A[0][0] = 0 * 0;
2: A[0][1] = 0 * 1;
3: A[0][2] = 0 * 2;
4: A[1][0] = 1 * 0;
5: A[1][1] = 1 * 1;
6: A[1][2] = 1 * 2;
7: A[2][0] = 2 * 0;
8: A[2][1] = 2 * 1;
9: A[2][2] = 2 * 2;
```

Motivating Example [2/2]

A few observations:

- Statement is executed 9 times
- There is a different values for i,j associated to these 9 instances
- There is an order on them (the execution order)

Objective:
find a representation where these 3 characteristics are modeled

Exercise 1: Find a Representation

Find such a representation (not using polyhedra)

Exercise 1: Find a Representation

Find such a representation (not using polyhedra)

- One solution: instance graph (aka extended representation)
  - 1 node per executed instance
  - directed graph: reflect execution ordering
- Another: system of affine recurrence equations (SARE)
- ...

Exercise 2: Listing the Issues

Generalization: exhibit the key problems we can face for the modeling of 1 statement

Exercise 2: Listing the Issues

Generalization: exhibit the key problems we can face for the modeling of 1 statement

- Memory consumption (compact representation)
- Parametric loop bound / unbounded loops
- non-unit loop strides
- conditionals
- ...

Summarizing the Problems

Step 1:

- Find a compact representation (critical)
- 1 point in the set $\leftrightarrow$ 1 executed instance (to allow optimization operations, such as counting points)
- Can retrieve when the instance is executed (total order on the set)
- Easy manipulation: scanning code must be re-generated

Step 2:

- Deal with parametric and infinite domains
- Non-unit strides

Step 3:

- Generalized affine conditionals (union of polyhedra)
- Data-dependent conditionals

Overview of the Solution

- Iteration domain: set of totally ordered n-dimensional vectors
  - **Iteration vector** $\vec{x}_S = (i,j)$
  - Iteration domain: the set of values of $\vec{x}_S$
- Convenient approach: **polytopes model sets** of totally ordered n-dimensional vectors
- **One condition: the set must be convex**

Convexity [1/2]

Convexity is the central concept of polyhedral optimization

Definition (Convex set)

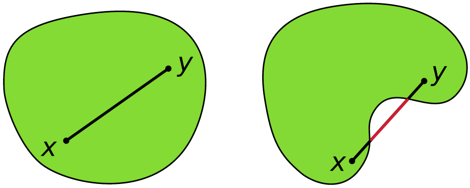
Given S a subset of $\mathbb{R}^n$. S is convex iff, $\forall \mu, \lambda \in S$ and given $c \in [0,1]$:

$$(1-c).\mu + c.\lambda \in S$$

With words: drawing a line segment between any two points of S, each point on this segment is also in S.

Warning: when $\mathbb{K} = \mathbb{Z}$, we use another definition

Convexity [2/2]



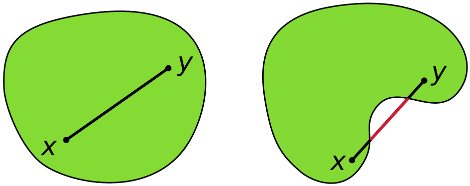
Definition (Convex combination)

Given S a convex set. For any family of vectors $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r \in S$, and any nonnegative numbers $\lambda_1, \lambda_2, \ldots, \lambda_r$ such that $\sum_{i=1}^{r} \lambda_i = 1$, then:

$$\vec{v} = \sum_{i=1}^{r} u_i \lambda_i \in S$$

$\vec{v}$ is a convex combination of $\{\vec{u}_i\}$.

Convexity [2/2]



Definition (Convex combination)

Given S a convex set. For any family of vectors $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r \in S$, and any nonnegative numbers $\lambda_1, \lambda_2, \ldots, \lambda_r$ such that $\sum_{i=1}^{r} \lambda_i = 1$, then:

$$\vec{v} = \sum_{i=1}^{r} u_i \lambda_i \in S$$

$\vec{v}$ is a convex combination of $\{\vec{u}_i\}$.

Exercise: Prove a statement surrounded by loops with unit-stride, no conditional and simple loop bounds has a convex iteration domain.

The Affine Qualifier

Definition (Affine function)

A function $f : \mathbb{K}^m \rightarrow \mathbb{K}^n$ is affine if there exists a vector $\vec{b} \in \mathbb{K}^n$ and a matrix $A \in \mathbb{K}^{m \times n}$ such that:

$$\forall \vec{x} \in \mathbb{K}^m, \ f(\vec{x}) = A\vec{x} + \vec{b}$$

Definition (Affine half-space)

An affine half-space of $\mathbb{K}^m$ (affine constraint) is defined as the set of points:

$$\{\vec{x} \in \mathbb{K}^m \mid \vec{a}.\vec{x} \leq \vec{b}\}$$

Polyhedron (Implicit Representation)

Definition (Polyhedron)

A set $\mathcal{S} \in \mathbb{K}^m$ is a polyhedron if there exists a system of a finite number of inequalities $A\vec{x} \leq \vec{b}$ such that:

$$\mathcal{P} = \{\vec{x} \in \mathbb{K}^m \mid A\vec{x} \leq \vec{b}\}$$

Equivalently, it is the intersection of finitely many half-spaces.

Definition (Polytope)

A polytope is a bounded polyhedron.

Integer Polyhedron

Definition ($\mathbb{Z}$-polyhedron)

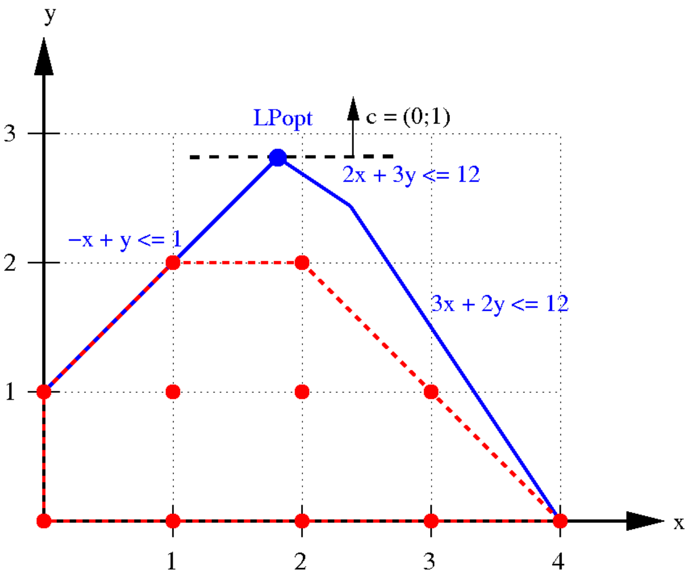
It is a polyhedron where all its extreme points are integer valued

Definition (Integer hull)

The integer hull of a rational polyhedron $\mathcal{P}$ is the largest set of integer points such that each of these points is in $\mathcal{P}$.

For the moment, we will "say" an integer polyhedron is a polyhedron of integer points (language abuse)

Rational and Integer Polytopes



Returning to the Example

Modeling the iteration domain:

- Polytope dimension: set by the number of surrounding loops
- Constraints: set by the loop bounds

$$\mathcal{D}_R : \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} . \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \\ 0 \\ 2 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & -1 & 2 \end{bmatrix} . \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} \geq \vec{0}$$

$$0 \leq i \leq 2, \quad 0 \leq j \leq 2$$

Another View of Polyhedra

The dual representation models a polyhedron as a combination of lines L and rays R (forming the polyhedral cone) and vertices V (forming the polytope)

Definition (Dual representation)

$$\mathcal{P} : \{\vec{x} \in \mathbb{Q}^n \mid \vec{x} = L\vec{\lambda} + R\vec{\mu} + V\vec{\nu},\ \vec{\mu} \geq 0,\ \vec{\nu} \geq 0,\ \sum_i \nu_i = 1\}$$

Definition (Face)

A face $\mathcal{F}$ of $\mathcal{P}$ is the intersection of $\mathcal{P}$ with a supporting hyperplane of $\mathcal{P}$. We have:

$$dim(\mathcal{F}) \leq dim(\mathcal{P})$$

Definition (Facet)

A facet $\mathcal{F}$ of $\mathcal{P}$ is a face of $\mathcal{P}$ such that:

$$dim(\mathcal{F}) = dim(\mathcal{P}) - 1$$

Getting Some Intuition...

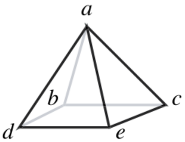
Exercise:

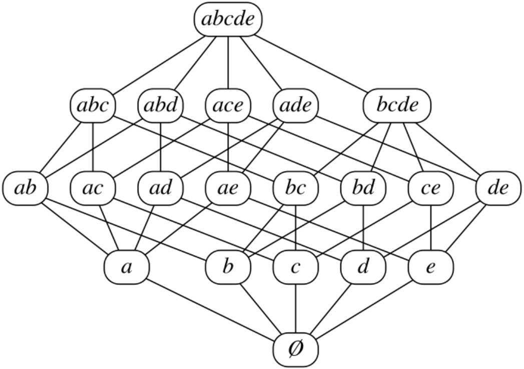
- Give the facets of $\mathcal{D}_S$
- Give some faces of $\mathcal{D}_S$

Example

```
for (i = 0; i < 3; ++i)
  for (j = 0; j < 3; ++j)
    A[i][j] = i * j;
```

The Face Lattice





Some Equivalence Properties

Theorem (Fundamental Theorem on Polyhedral Decomposition)

If $\mathcal{P}$ is a polyhedron, then it can be decomposed as a polytope $\mathcal{V}$ plus a polyhedral cone $\mathcal{L}$.

Theorem (Equivalence of Representations)

Every polyhedron has both an implicit and dual representation

- Chernikova's algorithm can compute the dual representation from the implicit one
- The Dual representation is heavily used in polyhedral compilation
- Some works operate on the constraint-based representation (Pluto)

Some Useful Algorithms

- **Compute the facets of a polytope**
- Compute the volume of a polytope (number of points)
- Scan a polytope (**code generation**)
- Find the lexicographic minimum

Increasing the Expressiveness

Problems:

- Unbounded domains: use polyhedra!
- Parametric loop bounds: use parametric polyhedra!
- Non-unit loop bounds: normalize the loop!

- Conditionals:
  - Those which preserve convexity: ok! (add affine constraints)
  - Problem remains for the others...

Parametric Polyhedra

Definition (Paramteric polyhedron)

Given $\vec{n}$ the vector of symbolic parameters, $\mathcal{P}$ is a parametric polyhedron if it is defined by:

$$\mathcal{P} = \{\vec{x} \in \mathbb{K}^m \mid A\vec{x} \leq B\vec{n} + \vec{b}\}$$

- Requires to adapt theory and tools to parameters
- Can become nasty: case distinctions (QUAST)
- Reflects nicely the program **context**

Some Useful Algorithms

All extended to parametric polyhedra:

- Compute the facets of a polytope: **PolyLib** [Wilde et al]
- Compute the volume of a polytope (number of points): **Barvinok** [Claus/Verdoolaege]
- Scan a polytope (code generation): **CLooG** [Quillere/Bastoul]
- Find the lexicographic minimum: **PIP** [Feautrier]

Practicing Your Knowledge

Find the iteration domain for the following programs:

Example

```
for (i = 0; i < N; ++i)
  for (j = i; j < N; ++j)
    A[3i + j] = K;
```

Example

```
for (i = 0; i < N; ++i)
  for (j = 0; j < i; ++j)
    A[j] = 0;
```

Practicing Again!

Example

```
for (i = 0; i < N; ++i)
  for (j = 0; j < i; ++j)
    if (i > M)
      A[j] = 0;
```

Example

```
for (i = 0; i < N; i += 2)
  for (j = 0; j < N; ++j)
    A[i] = 0;
```

Example

```
for (i = 0; i < N; i += 2)
  for (j = 0; j < N; ++j)
    if (i % 3 == 1 && j % 2 == 0)
      A[i] = 0;
```

Generalized Conditionals

Case distinction:

- **Conjunctions** (a && b)
- **Disjunctions** (a || b)
- **Non-affine** (i * j < 2)
- **Data-dependent** (a[i] == 0)

Relation with Operations on Polyhedra

Considering conjunctions:

Definition (Intersection)

The intersection of two convex sets $\mathcal{P}_1$ and $\mathcal{P}_2$ is a convex set $\mathcal{P}$:

$$\mathcal{P} = \{\vec{x} \in \mathbb{K}^m \mid \vec{x} \in \mathcal{P}_1 \wedge \vec{x} \in \mathcal{P}_2\}$$

Considering disjunctions:

Definition (Union)

The union of two convex sets $\mathcal{P}_1$ and $\mathcal{P}_2$ is a set $\mathcal{P}$:

$$\mathcal{P} = \{\vec{x} \in \mathbb{K}^m \mid \vec{x} \in \mathcal{P}_1 \vee \vec{x} \in \mathcal{P}_2\}$$

The union of two convex sets may not be a convex set

Generalized Conditionals

Case distinction (with a,b two affine expressions):

- Conjunctions (a && b) $\longrightarrow$ OK! Convexity preserved
- Disjunctions (a || b) $\longrightarrow$ Use a list of iteration domains
- Non-affine (i * j < 2) $\longrightarrow$ Use affine hull (loss of precision)
- Data-dependent (a[i] == 0) $\longrightarrow$ Use predicates + affine hull [Benabderrahmane]

Polyhedra in Use [1/2]

Exercise: Compute the footprint of A

Example

```
for (i = 0; i < N; ++i)
  for (j = 0; j < N; ++j)
    A[i][j] = i * j;
```

Example

```
for (i = 0; i < N; ++i)
  for (j = i; j < N; ++j)
    A[2i + 3][4j] = i * j;
```

Polyhedra in Use [2/2]

Exercise: Compute the set of cells of A accessed

Example

```
for (i = 0; i < N; ++i)
  for (j = 0; j < N; ++j)
    A[i][j] = i * j;
```

Example

```
for (i = 0; i < N; ++i)
  for (j = i; j < N; ++j)
    A[2i + 3][4j] = i * j;
```

Lattices

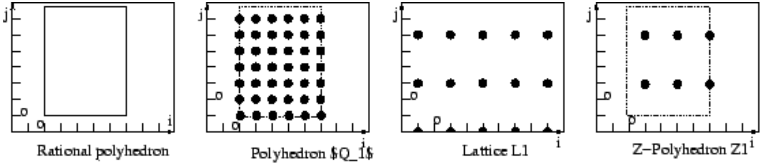
Definition (Lattice)

A subset L in $\mathbb{Q}^n$ is a lattice if is generated by integral combination of finitely many vectors: $a_1, a_2, \ldots, a_n$ ($a_i \in \mathbb{Q}^n$). If the a_i vectors have integral coordinates, L is an integer lattice.

Definition ($\mathcal{Z}$-polyhedron)

A $\mathcal{Z}$-polyhedron is the intersection of a polyhedron and an affine integral full dimensional lattice.

Pictured Example



Example of a $\mathcal{Z}$-polyhedron:

- $Q_1 = \{i,j \mid 0 \le i \le 5,\ 0 \le 3j \le 20\}$
- $L_1 = \{2i+1, 3j+5 \mid i,j \in \mathbb{Z}\}$
- $Z_1 = Q_1 \cap L_1$

Complex Example

Computing the set of cells of A accessed

Example

```
for (i = 0; i < N; ++i)
  for (j = i; j < N; ++j)
    A[2i + 3][4j] = i * j;
```

- $\mathcal{D}_S$: $\{i,j \mid 0 \leq i < N,\ i \leq j < N\}$
- Function: $f_A : \{2i+3, 4j \mid i,j \in \mathbb{Z}\}$
- Image($\mathcal{D}_S, f_A$) is the set of cells of A accessed (a $\mathcal{Z}$-polyhedron):
  - Polyhedron: $Q : \{i,j \mid 3 \leq i < 2N+2,\ 0 \leq j < 4N\}$
  - Lattice: $L : \{2i+3, 4j \mid i,j \in \mathbb{Z}\}$

Quick Facts on $\mathcal{Z}$-polyhedra

- **Iteration domains are in fact $\mathcal{Z}$-polyhedra with unit lattice**
- Intersection of $\mathcal{Z}$-polyhedra is not convex in general
- Union is complex to compute
- Parametric lattices are challenging!
- **Can count points, can optimize, can scan**

- **Implementation available for most operations in PolyLib**

Some Interesting Problems

- Write generalized loop normalization algorithms
  - Stride normalization
  - while loop / do loop conversion
  - Conditional normalization
- ...