Une interface commune pour les treillis numériques

Projet APRON
(Bertrand Jeannet)

Plan

1. Introduction
2. Fonctionalités offertes
3. Architecture générale de l'interface
4. Etat d'avancement et perspectives

I. Introduction

Motivations 1/2

Analyse des variables numériques d'un programme: nombreux treillis abstraits implémentés:

- Intervalles (. . .)
- Egalités linéaires (Müller-Olms & Seidl)
- Combinaison des intervalles et égalités (Venet)
- Zones et Octagones
  (Miné, Bagnara & al ?, Métivier $\simeq$)
- Polyèdres convexes (NewPolka, Parma, PolyLib, CRI)
- Octaèdres (Cortadella)
- "Templates" (VMCAI'05)
- Égalités polynomiales (Müller-Olms & Seidl)
- Inégalités polynomiales (SAS'05)
- Ellipsoïdes (Férêt)

Motivations 2/2

Fonctionalités

- Noyau minimal de fonctions offertes
  (meet, join, affectation par expression linéaire, . . .)
- + Fonctions plus spécifiques pour des applications particulières (ex: PolyLib, CRI)

Mais API très diverses !

- Certaines fonctionalités de base peuvent manquer
- Certaines interfaces trop liées aux structures de données internes (ex: NewPolka)

$\Rightarrow$ Complique la diffusion de ces implémentations
$\Rightarrow$ Difficile de comparer

- l'efficacité de 2 implémentations du même treillis
- la précision de deux treillis différents

Objectifs pour une interface commune 1/2

- Identifier les fonctionalités de base que doit fournir (l'implémentation d') un treillis numérique
- Élaborer une interface détaillée (types de données, signatures des fonctions, sémantique associée)
- Implémenter une telle interface pour les bibliothèques maintenues par les membres du projet
- Diffuser auprès de la communauté analyse statique (utilisateurs et implémenteurs)

Objectifs pour une interface commune 2/2

Sous les contraintes suivantes:

- Satisfaire les besoins de tous les membres du projet, tout en restant générique !
- Souci de simplicité et de minimalité, sans trop pénaliser les performances
- Minimiser le travail d'adaptation d'une librairie existante

Initiatives comparables en analyse/vérification

L'interface SIS/VIS pour manipuler les BDDs (University of california Berkeley)

Analyseurs génériques:

- Se préoccupent du calcul de point-fixe
- Souvent un peu spécialisés:
  - en fonction du type d'analyse (ex: analyse type bitvector)
  - incorporent la sémantique du langage source (Marktoberdorf'98 Generic Abstract Interpreter)

II. Fonctionalités offertes

(Au niveau 0 de l'interface)

Fonctionalités offertes (1/5)

Sémantique d'une valeur abstraite:
sous-ensemble $\boldsymbol{X} \subseteq \mathbb{N}^{\boldsymbol{p}} \times \mathbb{R}^{\boldsymbol{q}}$

Les valeurs abstraites sont typées selon leur dimensionalité (niveau 0 de l'interface)

Les dimensions sont donc typées entier ou réel
Elles sont numérotées de $\mathbf{0}$ à $\boldsymbol{p} + \boldsymbol{q} - \mathbf{1}$

Autres types de données manipulés: coefficients, intervalles, expressions (quasi-)linéaires, contraintes.

Fonctionalités offertes (2/5)

Contrôle de la représentation interne:

- Forme canonique (sémantique pas encore très clair)
- Forme minimale (en terme d'occupation mémoire)
- Notion d'approximation laissé au libre choix de l'implémentation (prise en compte des entiers ou non, ...)

Impression:

- d'une valeur abstraite
- de la différence de deux valeurs abstraites

Sérialisation/désérialisation binaire vers une zone mémoire

Fonctionalités offertes (3/5)

Constructeurs: $\bot(\boldsymbol{p}, \boldsymbol{q})$, $\top(\boldsymbol{p}, \boldsymbol{q})$, abstraction d'un hypercube, abstraction d'un polyèdre convexe

Tests:

- du vide, de l'univers
- inclusion, égalité
- inclusion d'une dimension dans un intervalle, satisfaction d'une contrainte linéaire

Extraction de propriétés:

- intervalle de variation d'une dimension, d'une expression linéaire, dans une valeur abstraite
- conversion vers un hypercube, un polyèdre convexe

Fonctionalités offertes (4/5)

Opérations de treillis et variations:

- $\sqcup$, $\sqcap$ de deux valeurs, d'un tableau de valeurs
- intersection avec une plusieurs contraintes linéaires
- ajout de rayons (opérateur de passage du temps généralisé)

Affectation/Substitution:

- d'une dimension par une expression linéaire
- en parallèle de plusieurs dimensions par des expressions linéaires

Élargissement avec seuil (ensemble de contraintes linéaires)

Fonctionalités offertes (5/5)

Projection/Oubli d'une ou plusieurs dimensions
à dimensionalité constante

Ajout/Retrait/Permutation de dimensions

Expansion et pliage de dimensions
(utiles pour l'abstraction de tableaux, par ex)

Clôture topologique (relaxation des contraintes strictes)

III. Principes généraux

Notion de niveau d'interface

On a pour objectifs: performance des implantations, confort de l'utilisateur, et non-duplication de code $\neq$ bibliothèques

- Niveau 0: on se préoccupe des performances, et de la précision
- Niveau 1: on se préoccupe du confort, et de ce qui est factorisable

Notion de niveau d'interface

On a pour objectifs: performance des implantations, confort de l'utilisateur, et non-duplication de code $\neq$ bibliothèques

- Niveau 0: on se préoccupe des performances, et de la précision
- Niveau 1: on se préoccupe du confort, et de ce qui est factorisable

Niveau 0:

- En prise directe avec la librairie sous-jacente
- Contient toutes les opérations spécifiques au domaine abstrait (ne pouvant être partagées)
- Interface minimale, sauf si avantage algorithmique fort à y inclure une combinaison

Notion de niveau d'interface

On a pour objectifs: performance des implantations, confort de l'utilisateur, et non-duplication de code $\neq$ bibliothèques

- Niveau 0: on se préoccupe des performances, et de la précision
- Niveau 1: on se préoccupe du confort, et de ce qui est factorisable

Niveaux supérieurs: fonctions factorisables
Deux exemples envisagés:

- abstraction d'expressions non linéaires en expressions linéaires d'intervalle
- Appel automatique aux opérations de redimensionnement et de permutation pour calculer $\boldsymbol{P}(\boldsymbol{x},\boldsymbol{y}) \sqcap \boldsymbol{Q}(\boldsymbol{z},\boldsymbol{y})$

Notion de niveau d'interface

Combinaison de treillis/bibliothèques: deux interfaces/bibliothèques de niveau 0 pourront être combinées pour offrir une nouvelle bibliothèque de niveau 0.

- implémentations différentes du même domaine
- produit cartésien ou réduit de domaines
- décomposition de polyèdres en produit cartésien de polyèdres (thèse de D. Merchat)

Langage de programmation

La version de référence est la version C de l'interface

- C s'interface aisément avec la plupart des langages
- La plupart des librairies existantes sont en C (ou C++)

Une version OCaml sera certainement élaborée (ENS, IRISA)
L'interfaçage OCaml/C sera sans doute même générique pour toutes les librairies

Architecture générale (1/3)

Compatibilité avec les threads

Un contexte d'appel qui sera explicitement passé à chaque fonctions afin d'assurer:

- La transmission de données spécifiques à chaque librairie (options non standard, espace de travail, ...)
- La transmission des options (leviers de réglages des algos, de la précision)
- La gestion des exceptions (`not_implemented`, `invalid_argument`, `overflow`, `timeout`, `out_of_space`)

Architecture générale (2/3)

Gestion mémoire:

- Pas de ramasse-miettes dans l'interface C
- Utilisation des mécanismes du runtime pour les interfaces OCaml, Prolog, ...

Mécanismes d'interruption en cas de temps trop long ou de consomnation mémoire trop élevée (`timeout`, `out_of_space`, `overflow`)

Signatures fonctionnelles et impératives (effet de bord) supportées

Architecture générale (3/3)

Représentation des nombres:

- En interne, propre à chaque bibliothèque
- Dans l'interface, choix entre
  - rationels multi-précision GMP (arithmétique exacte)
  - `double` (arithmétique flottante)

IV. Avancement et perspectives

Avancement

Interface niveau 0:

- Document de travail synthétisant discussions et décisions (4 réunions)
- Semble satisfaire les besoins des participants
- Adaptation en cours de NewPolka et d'un treillis d'intervalle $\Rightarrow$ retour d'expérience

Pas résolu:

- Égalités ou inégalités polynomiales: besoin d'ajouter les expressions polynômiales au niveau 0
- Congruences ?
- Utilisation transparente de domaines de type Constraint Template ou Octogones creux

Perspectives

D'utilisation:

Au sein d'APRON, les plus intéressés: Verimag, IRISA, CRI, ENS

De diffusion:

Comment convaincre les implémenteurs ?