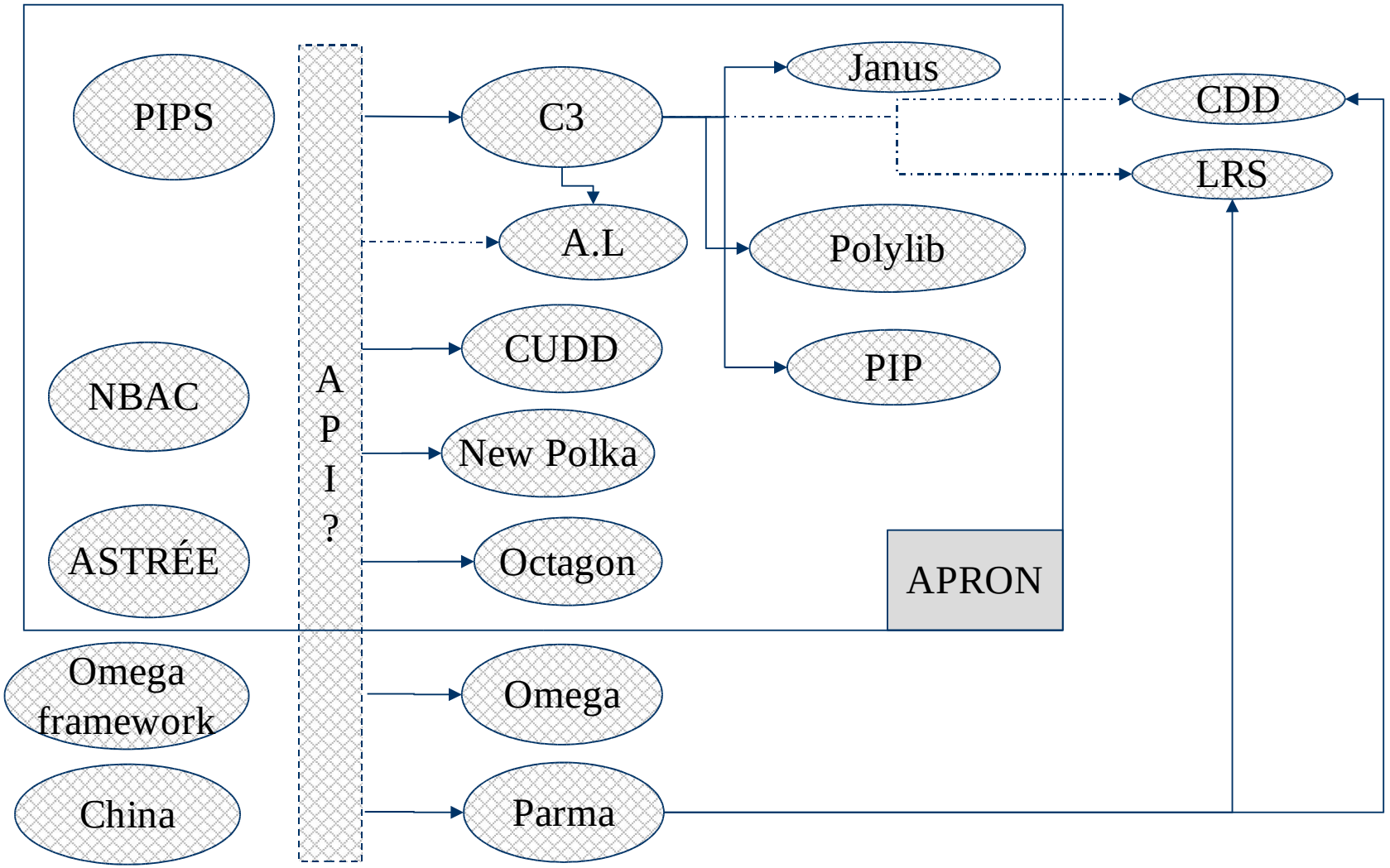
Vers une implantation générique validée expérimentalement des domaines abstraits?

Duong NGUYEN, CRI/ENSMP

Duong NGUYEN
CRI, ENSMP

Interface commune (API)

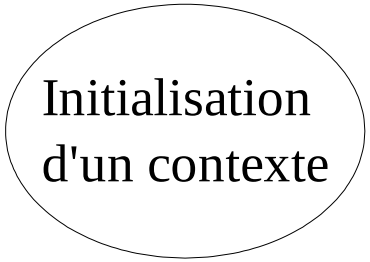


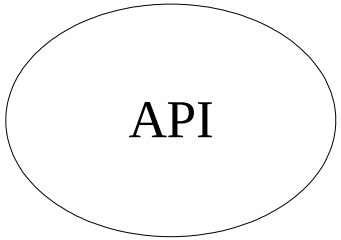
Duong NGUYEN
CRI, ENSMP

Utilisation d'une API commune

Au dela des opérateurs de base d'un treillis

- Gestion d'exceptions (paramètres: e.g. timeout)
- Gestion de la mémoire (version recycle, référence, mémoire utilisée)
- Gestion de la précision numérique (valeur abstraite, 32-bit, 64-bit, gmp)
- Gestion d'approximations (exact, sous, sur)
- Gestion d'optimisation (choix d'algorithmes, paramètres heuristiques, par nombre d'arguments)
- Sélection d'algorithme par l'utilisateur (e.g. widening)
- Gestion du typage(valeur abstraite, entier, rationnel)





Duong NGUYEN
CRI, ENSMP

1. Niveau des opérateurs

« Manque » d'operateurs:

- Octagons: opérateurs de permutation (de dimension)
- De différentes implantations d'élargissements
- Conversions de format (opérateurs de conversion)

Redondance(?)

- Intersection des ensembles, des contraintes
- add_constraint, add_constraints

Efficacité d'opérateurs (niveau algorithmique):

- Plusieurs arguments (e.g. convex hull)
- and_minimized (ou _lazy) version

Algorithmes:

- Entier, rationnel (pas forcement tout disponible)

Duong NGUYEN
CRI, ENSMP

2. Niveau de l’abstraction

Accessibilités de plusieurs types:

- Polyèdres: accès aux vecteurs
- Octagons: pas d'accès direct aux contraintes
- New Polka: accès aux matrices de saturation

Au-delà d'une interface « treillis »?

- Fonctions de transfert:

  Ensemble, dimension et expression affine (e.g. objet « tab » dans New Polka, Octagon)

- Types des objets:

  Objet vs liste des objets (contrainte, générateur, ensemble et expression)

  Expression abstraite vs expression affine

Duong NGUYEN
CRI, ENSMP

3. Gestion de l'environnement

Identificateur -> Location -> Valeur = Union { Int, real, value,Location}

Expression -> Location -> Valeur = Union {Int, real, value,Location}

Expression d'adresses

Typage

Duong NGUYEN
CRI, ENSMP

4. Gestion de mapping et packing

Mapping:

- Base (Ensemble de variables/dimensions)
- Fonctions de permutation sur les dimensions

Packing

Expand dimension; Fold dimensions

Factorisation?

5. Gestion d'exceptions

Overflow (stop/changement de précision/approximation)

Timeout: paramètres

Contrôle de la consommation mémoire

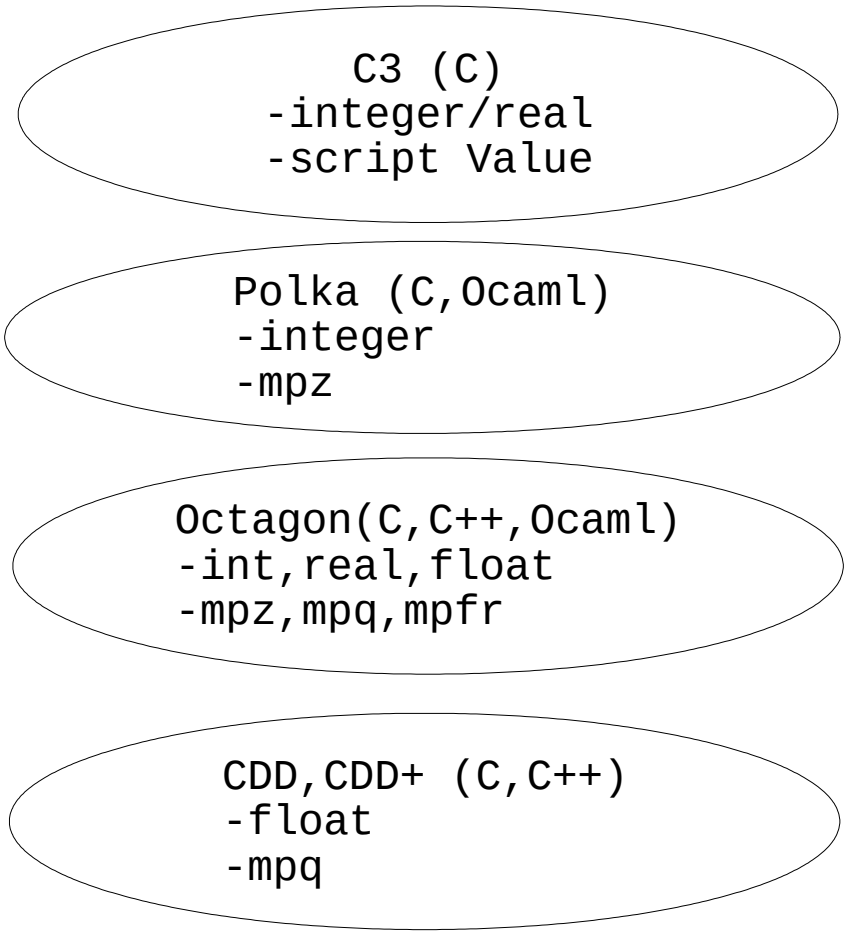
Duong NGUYEN
CRI, ENSMP

6. Langages utilisés

C (C3, Octagon,Polka)

C++(Octagon,Omega)

Ocaml(Octagon,Polka)



7. Différences de précision numérique

1. 32-bit, 64-bit, gmp
2. Sous-libraries utilisés (problème de link)
   - mpz
   - mpq
   - mpfr/mpf

Duong NGUYEN
CRI, ENSMP

8. Algorithmes en développement

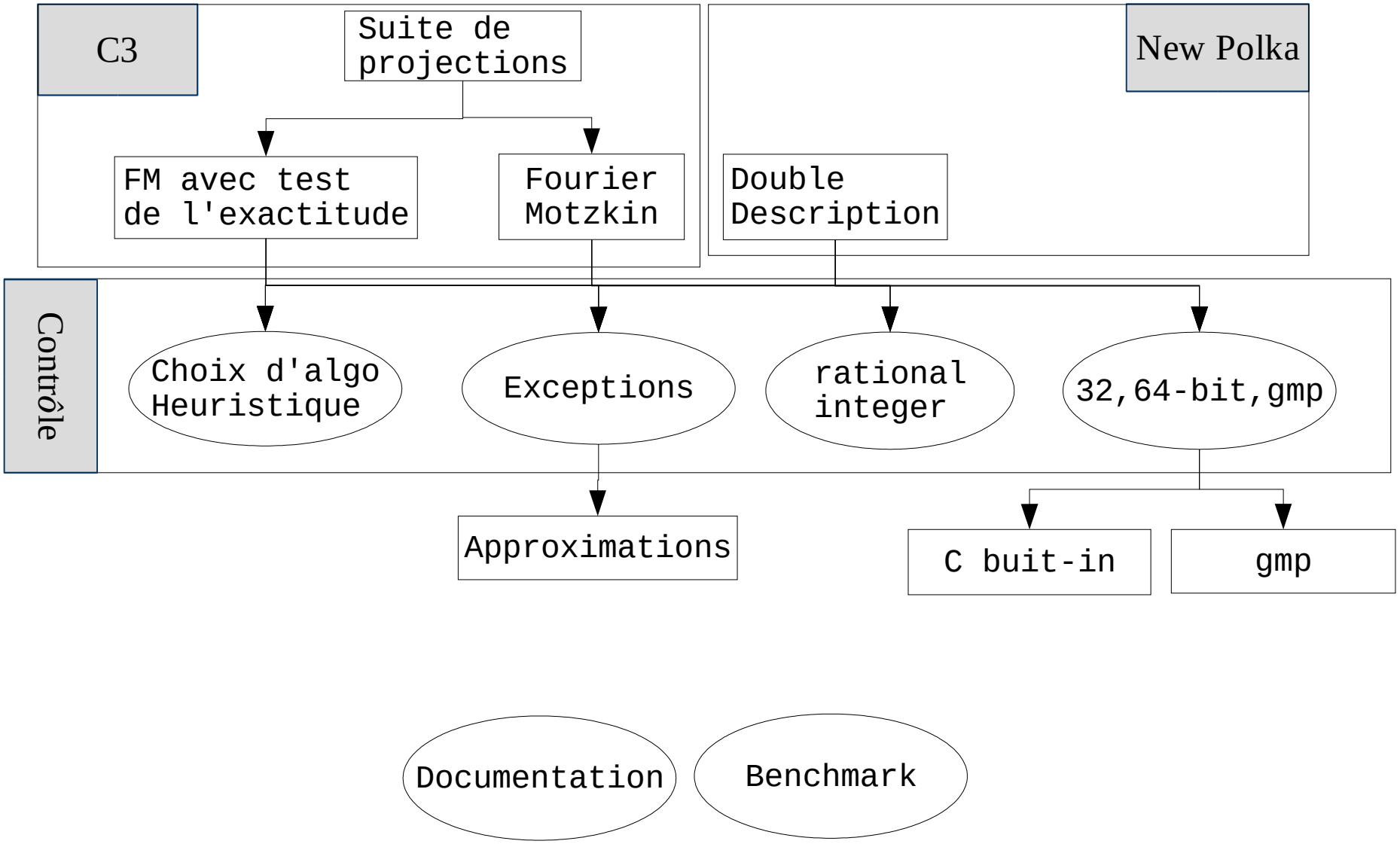
Widening (élargissement)

- Combien de versions? (e.g. Polka: 2 versions,PPL:4 versions)

Accéleration (Grenoble)

Projection d'une variable, d'une liste de variables

Duong NGUYEN
CRI, ENSMP



Duong NGUYEN
CRI, ENSMP

API proposée (1)

Treillis de HQSet:

- HQSet
- HQSysCon (liste d'une ou plusieurs contraintes)
- HQSysGen (liste d'une ou plusieurs générateurs)
- HQBase/HQVariable (sémantique de la dimension)
- HQExpression (abstraction d'expression affine?)

Différences:

- Octagons (pas de générateurs) -> conversion d'objets?
- Liste de polyèdres
- Rélations (Omega)

Représentation: (http://www.cri.ensmp.fr/people/duong/HQ_API/doc/index.html)

- Javadoc -> C (règles)
- l'objet courrant (premier argument)
- polymorphisme

API proposée (2)

Avantages d'utilisation de HQBase:

- Nom des variables -> l'origine des variables du programme analysé (debugging)
- Gestion partagée de mapping/ packing/ factorisation/ partition
- Permutation de colonne par changement de base

Inconvénients d'utilisation de HQBase:

- Implantation de la gestion des variables (nommage)
- Espace mémoire
- Space-dimension compatibilité test plus lent
- Complexité

Duong NGUYEN
CRI, ENSMP

API proposée (3)

Fonctions de transfert (expression affine)

Intersection des ensembles avec des contraintes

pas de fonctions comme add_constraint, add_constraints

Conversion de format (paramètres)

Expand dimension/Fold dimensions

Debugging/analyse:

- HQSet.getWeight() methode?

Opérateur time_elapse()?

Duong NGUYEN
CRI, ENSMP

API proposée (4)

Initialisation d'un contexte

- Gestion de timeout, d'overflow
- Gestion de la précision numérique (valeur abstraite, 32-bit, 64-bit, gmp)
- Gestion d'optimisation (choix d'algorithmes, paramètres heuristiques)
- Gestion du typage(valeur abstraite, entier, rationnel)

API:

- Gestion de la mémoire (version recycle, référence, mémoire utilisée)
- Gestion d'approximations (exact, sous, sur)
- Gestion d'optimisation (minimized/lazy version)
- Sélection de type (entier/rationel)
- Sélection d'une version d'algorithme (e.g. widening)

Duong NGUYEN
CRI, ENSMP

Conclusion

Octagons: fonction de permutation

Fonctions de transfert (expression affine)

HQBase (format dense, creuse)

Accessibilités de plusieurs types:

- Vecteurs
- Contraintes
- Matrices

Packing, Expand/Fold dimensions

Produit de domaines

Factorisation?

API proposée (4)