TreeKs: a Functor to Make Abstract Numerical Domains Scalable

Research Internship, advised by Antoine Miné
École normale supérieure, Paris, team Abstraction

Mehdi Bouaziz
Motivation and context

Abstract interpretation is a formal theory of sound approximation of semantics, mainly used in static analyzer, such as:

- Clousot: static verification of Code Contracts
- Astrée: proof of absence of runtime errors on embedded softwares

Abstract numerical domains:

- a set $\mathcal{D}_\mathcal{V}$ of computer-representable abstract values
- effective algorithms to compute sound abstractions of the operations: intersection $\cap^{\mathcal{D}_\mathcal{V}}$, union $\cup^{\mathcal{D}_\mathcal{V}}$, projection $\exists^{\mathcal{D}_\mathcal{V}}$, ...
Numerical abstract domains: examples

Intervals [Cousot Cousot 76]

\[ \bigwedge_i a_i \leq X_i \leq b_i \]

Non-relational
Linear cost

Polyhedra [Cousot Halbwachs 78]

\[ \bigwedge_j \sum_i a_{ij} X_i \leq b_j \]

Relational and very precise
Worst-case exponential cost
Weakly relational numerical abstract domains

Zones [Miné 01]
\[ \wedge_{ij} X_i \pm X_j \leq c_{ij} \]
Cubic cost

Octagons [Miné 01]
\[ \wedge_{ij} \pm X_i \pm X_j \leq c_{ij} \]
Cubic cost

Logahedra [Howe King 09]
\[ \wedge_{ij} 2^{a_i} X_i \pm 2^{b_j} X_j \leq c_{ij} \]
Cubic cost

TVPI [Simon King Howe 02]
\[ \wedge_{ij} a_i X_i + b_j X_j \leq c_{ij} \]
Quasi-cubic cost

Octahedra [Clarisó Cortadella 07]
\[ \wedge \sum_i \pm X_i \leq c \]
Worst-case exponential cost
Closure operation: example

Domain of zones \((\bigwedge_{i,j} X_i - X_j \leq b_{ij})\)

\[\mathcal{V} = \{x, y, z\}\]
Closure operation: example

Domain of zones \((\bigwedge_{i,j} X_i - X_j \leq b_{ij})\)

\(\mathcal{V} = \{x, y, z\}\)

\(-x \leq -1\)
Closure operation: example

Domain of zones $(\bigwedge_{i,j} X_i - X_j \leq b_{ij})$

$V = \{x, y, z\}$

$-x \leq -1$

$x - y \leq 0$
Closure operation: example

Domain of zones \((\bigwedge_{i,j} X_i - X_j \leq b_{i,j})\)

\[\mathcal{V} = \{x, y, z\}\]

\[-x \leq -1\]
\[x - y \leq 0\]
\[y - z \leq -2\]
Closure operation: example

Domain of zones \((\bigwedge_{i,j} X_i - X_j \leq b_{ij})\)
\[\mathcal{V} = \{x, y, z\}\]

\[\begin{align*}
-x &\leq -1 \\
-x &\leq -1 \\
x - y &\leq 0 \\
y - z &\leq -2 \\
y - z &\leq -2
\end{align*}\]
Closure operation: example

Domain of zones \((\bigwedge_{i,j} X_i - X_j \leq b_{ij})\)

\[ V = \{x, y, z\} \]

\[ \begin{align*}
-x & \leq -1 \\
x - y & \leq 0 \\
y - z & \leq -2 \\
\end{align*} \]

\[ \begin{align*}
-y & \leq -1 \\
z & \leq -3 \\
\end{align*} \]
Closure operation: example

Domain of zones \((\bigwedge_{ij} X_i - X_j \leq b_{ij})\)

\[ V = \{x, y, z\} \]

\[ -x \leq -1 \]
\[ x - y \leq 0 \]
\[ y - z \leq -2 \]

\[ -y \leq -1 \]
\[ -z \leq -3 \]
\[ x - z \leq -2 \]
Closure operation: example

Domain of zones \((\bigwedge_{i,j} X_i - X_j \leq b_{ij})\)

\[ V = \{x, y, z\} \]

\[ -x \leq -1 \]
\[ x - y \leq 0 \]
\[ y - z \leq -2 \]

\[ -y \leq -1 \]
\[ -z \leq -3 \]
\[ x - z \leq -2 \]

Done!
Domain of zones: representation

We represent a set of difference constraints between two variables ($X_i - X_j \leq m_{ji}$) by a potential graph or by a DBM (Difference Bound Matrix).

\[
\begin{align*}
0 - x &\leq -1 \\
x - y &\leq 0 \\
y - z &\leq -2
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>$x$</td>
<td>$-1$</td>
<td>0</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>$y$</td>
<td>$+\infty$</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>$z$</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
<td>$-2$</td>
<td>0</td>
</tr>
</tbody>
</table>
We represent a set of difference constraints between two variables \((X_i - X_j \leq m_{ji})\) by a potential graph or by a DBM (Difference Bound Matrix).

\[
\begin{align*}
0 - x &\leq -1 \\
x - y &\leq 0 \\
y - z &\leq -2
\end{align*}
\]

\[
\begin{align*}
0 - y &\leq -1 \\
0 - z &\leq -3 \\
x - z &\leq -2
\end{align*}
\]
Domain of zones: closure and other operators

The closure is a shortest-path closure.

After closure, operators are point-wise.

Join (best approximation of union):

\[(m \sqcup n)_{ij} = \max(m_{ij}, n_{ij})\]

Forget operator (projection):

\[(\exists X_k m)_{ij} = \begin{cases} m_{ij} & \text{if } i \neq k \text{ and } j \neq k \\ 0 & \text{if } i = j = k \\ +\infty & \text{otherwise} \end{cases} \]
How to scale: packing

Principle:
- split variables into packs
- use a DBM per pack

Cost: linear for bounded-size packs
Information loss: no communication between packs!
How to scale: packing

Principle:

- split variables into packs
- use a DBM per pack

Cost: linear for bounded-size packs
Information loss: no communication between packs!
Solution: intervals constraints sharing
Not good enough!
TreeKs: a certain subgraph

Shape:

- a tree of complete graphs (packs)
- sharing borders

packs tree
TreeKs: a certain subgraph

**Shape:**
- a tree of complete graphs (packs)
- sharing **borders**

Abstract value: tuple of DBMs
Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
   Apply closure on this pack in the domain of zones
   Pass the new constraints to his father

for each pack from the root to the leaves
   Apply closure on this pack in the domain of zones
   Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his father

for each pack from the root to the leaves
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his father

for each pack from the root to the leaves
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

**for each** pack from the leaves to the root
- Apply closure on this pack in the domain of zones
- Pass the new constraints to his father

**for each** pack from the root to the leaves
- Apply closure on this pack in the domain of zones
- Pass the new constraints to his children
Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his father

for each pack from the root to the leaves
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his children
Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
    Apply closure on this pack in the domain of zones
    Pass the new constraints to his father

for each pack from the root to the leaves
    Apply closure on this pack in the domain of zones
    Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
   Apply closure on this pack in the domain of zones
   Pass the new constraints to his father

for each pack from the root to the leaves
   Apply closure on this pack in the domain of zones
   Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his father

for each pack from the root to the leaves
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his father

for each pack from the root to the leaves
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
    Apply closure on this pack in the domain of zones
    Pass the new constraints to his father

for each pack from the root to the leaves
    Apply closure on this pack in the domain of zones
    Pass the new constraints to his children
Closure algorithm

Closure algorithm in TreeKs $O(mp^3)$

for each pack from the leaves to the root
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his father

for each pack from the root to the leaves
  Apply closure on this pack in the domain of zones
  Pass the new constraints to his children
Conclusion

- can be applied to many numerical abstract domains (zones, octagons, logahedra, TVPI, octahedra, polyhedra, ...)
- linear cost when pack size is bounded

Future work:
- implementation
- development of packs generation strategies
- application to other domains