# Interval CLP

[Benhamou et al]

humbly presented by

Johan Lübcke and Dan Lawesson



## Outline

- Preliminaries
  - ✔ F-intervals
  - ✔ Approximation
- Narrowing
  - ✔ Algorithm
  - ✔ Example
- Convexity
  - ✔ Choice points
- Implementation
  - ✔ Newton
  - ✔ Box consistency
  - ✔ Splitting



## F-intervals

$F = E \cup \{-\infty, \infty\}$, $E \subset \Re$, $E$ finite

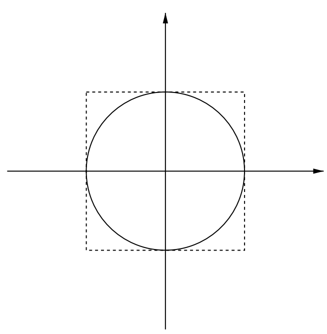
F-interval : $(a, b)$, $\{a, b\} \subseteq F$

$\mathcal{F}$ is the set of all F-intervals



## Approximations

If $\rho$ is a subset of $\Re^n$ then $apx(\rho)$ is the smallest F-interval containing $\rho$.



$$apx(X^2 + Y^2 \leq 1) = ([-1, 1], [-1, 1])$$

## Constraint systems

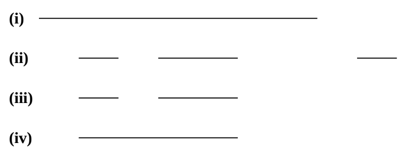
$V$ is a set of variables

- A *constraint* is an expression $\rho(x_1, \ldots, x_n)$, where $\rho \subseteq \Re^n$ and every $x_i \in V \cup E$
- A *system* $\Sigma = (i, S)$, where $i : V \cup E \to \mathcal{F}$ and $S$ is a finite set of constraints
- A *solution* $\sigma : V \cup E \to \Re$ of a system $\Sigma$ satisfies
  - $\forall x \in E, \sigma(x) = x$
  - $\forall x \in V, \sigma(x) \in i(x)$
  - $\forall \rho(x_1, \ldots, x_n) \in S, (\sigma(x_1), \ldots, \sigma(x_n)) \in \rho$



## Narrowing

For every F-block $u$, the *narrowing* function $\overrightarrow{\rho} : \mathcal{F}^n \to \mathcal{F}^n$ of $\rho$ satisfies $\overrightarrow{\rho}(u) = apx(u \cap \rho)$




(i)-(iv) : $u$, $\rho$, $u \cap \rho$, $\overrightarrow{\rho}$

**Contractance** $\overrightarrow{\rho}(u) \subseteq u$

**Correctness** $u \cap \rho = \overrightarrow{\rho}(u) \cap \rho$

**Monotonicity** $u \subseteq v \Rightarrow \overrightarrow{\rho}(u) \subseteq \overrightarrow{\rho}(v)$

**Idempotence** $\overrightarrow{\rho}(\overrightarrow{\rho}(u)) = \overrightarrow{\rho}(u)$



## Narrowing algorithm

Intuition: Calculate narrowing fix point. Whenever an interval (denoted $X$) becomes narrower, re-evaluate all constraints containing $X$.

Terminating!



## Narrowing example

As defined earlier : $\overrightarrow{\rho}(u) = apx(u \cap \rho)$

$c_0 = \{X \in \Re \mid X \geq 0\}$

$c_1 = \{(X, Y) \in \Re^2 \mid Y \geq X\}$

$\Sigma = (i, S) = (\{X \mapsto [0, \infty], Y \mapsto [0, \infty]\}, \{c_0(X), c_1(X, Y)\})$

Introduce $c_2 = \{(X, Y) \in \Re^2 \mid X^2 + Y^2 \leq 1\}$

The resulting stable system is $\Sigma = (\{X \mapsto [0, 1], Y \mapsto [0, 1]\}, \{c_0, c_1, c_2\})$

## Interval convexity

A constraint $\rho$ is *interval convex* if for every F-block $u$ and every $i \in \{1, \ldots, n\}$, $\pi_i(\rho \cap u)$ is an F-interval.

$$add = \{(x, y, z) \in \Re^3 \mid x + y = z\}$$

is interval convex, but

$$mult = \{(x, y, z) \in \Re^3 \mid xy = z\}$$

is not.

Example: Assume $u = ([-2, 3], [-4, 5], [1, 1])$, then $mult \cap u = ([-2, 3], [-3, -\frac{1}{2}] \cup [\frac{1}{3}, 5], [1, 1])$ and $apx(mult \cap u) = u$.



## Choice points

Assume $\rho = \rho_1 \cup \rho_2$, i.e. $c = c_1 \vee c_2$

1. $\overrightarrow{\rho}(u) = apx(u \cap \rho_1) \vee \overrightarrow{\rho}(u) = apx(u \cap \rho_2)$, disjunction implemented by choice point.
2. $\overrightarrow{\rho}(u) = apx(u \cap (\rho_1 \cup \rho_2))$, splitting after propagation (similar to labeling) creates choice point.

Example: $mult = mult^+ \cup mult^-$, where

$$mult^+ = \{(x, y, z) \in \Re^3 \mid x \geq 0, xy = z\}$$

and

$$mult^- = \{(x, y, z) \in \Re^3 \mid x < 0, xy = z\}$$

Both are interval convex.



## Restricted intervals

**Integers** If $u = [a, b]$ then $\overrightarrow{int}(u) = [\lceil a \rceil, \lfloor b \rfloor]$,

$$neq = \{(x, y) \in \Re^2 \mid x < y\} \cup \{(x, y) \in \Re^2 \mid x > y\}$$

**Booleans** is a special case of integers

Example:
(using $int$ for two variables)

$$\begin{aligned} u &= ([-0.3, 2.4], [1.5, 2.5]) \\ \overrightarrow{int}(u) &= ([0, 2], [2, 2]) \\ \overrightarrow{neq}(\overrightarrow{int}(u)) &= ([0, 2), [2, 2]) \cup \emptyset \\ \overrightarrow{int}(\overrightarrow{neq}(\overrightarrow{int}(u))) &= ([0, 1], [2, 2]) \end{aligned}$$



## Newton

Extending Prolog, *Newton* combines constraints over reals, integers and booleans.

Newton uses a relaxed implementation of narrowing called *box consistency*.

## Interval extension

$f' : \mathcal{F}^n \rightarrow \mathcal{F}$ is an interval extension of $f : \Re^n \rightarrow \Re$ iff

$\forall F_1, \ldots, F_n \in \mathcal{F} : x_1 \in F_1, \ldots, x_n \in F_n \Rightarrow f(x_1, \ldots, x_n) \in f'(F_1, \ldots, F_n)$

Example: $f'([x_1, x_2], [y_1, y_2]) = [\lfloor x_1 + y_1 \rfloor, \lceil x_2 + y_2 \rceil]$ is an interval extension of $f(x, y) = x + y$.

Another example would be $f'(X, Y) = [-\infty, \infty]$...



## Arc consistency revisited

A constraint $c$ is arc-consistent wrt $(D_1, \ldots, D_n)$ and an integer $i$, $1 \leq i \leq n$ iff

$$\begin{aligned} D_i \quad \subseteq \quad & \{r_i \,|\, \exists r_1 \in D_1, \ldots, r_{i-1} \in D_{i-1}, \\ & r_{i+1} \in D_{r+1}, \ldots, r_n \in D_n : c(\overline{r})\} \end{aligned}$$

Arc consistency for systems is defined in the natural way.

Generalized to intervals, a constraint $\rho$ is arc consistent wrt $(F_1, \ldots, F_n)$ and an integer $i$ , $1 \leq i \leq n$ iff

$$\begin{aligned} F_i \quad = \quad & apx(F_i \cap \{r_i \,|\, \exists r_1 \in F_1, \ldots, r_{i-1} \in F_{i-1}, \\ & r_{i+1} \in F_{r+1}, \ldots, r_n \in F_n : \overline{r} \in \rho\}) \end{aligned}$$

This is the fix point criteria for narrowing, i.e.

$$u = \overrightarrow{\rho}(u) = apx(u \cap \rho)$$



## Box consistency

Box consistency is a relaxed version of arc consistency that is less expensive to calculate. An interval constraint $\rho' = C(F_1, \ldots, F_n)$ is box consistent wrt $(F_1, \ldots, F_n)$ and an integer $i$ iff

$$\begin{gathered} C(F_1, \ldots, F_{i-1}, [l, l^+], F_{i+1}, \ldots, F_n) \\ \wedge \\ C(F_1, \ldots, F_{i-1}, [u^-, u], F_{i+1}, \ldots, F_n) \end{gathered}$$

where $l = left(F_i)$ and $u = right(F_i)$.

This is equivalent to

$$\begin{aligned} F_i \quad = \quad & apx(\{r_i \in F_i \,|\, C(F_1, \ldots, F_{i-1}, \\ & apx(\{r_i\}), F_{i+1}, \ldots, F_n)\}) \end{aligned}$$

Box consistency differs from arc consistency when $\rho'$ contains multiple instances of the same variable.



## Box consistency example

$x_1 - x_1 + x_2 = 0$ is not arc consistent wrt $([-1, 1], [0, 2])$ because assigning $x_2 \neq 0$ makes the constraint unsatisfiable. I.e. the intervals should be narrowed to $([-1, 1], [0, 0])$.

$X_1 - X_1 + X_2 = 0$ is box consistent wrt $([-1, 1], [0, 2])$, because

$$[-1, -1] - [-1, -1] + [0, 2] \doteq [0, 0]$$

$$[1, 1] - [1, 1] + [0, 2] \doteq [0, 0]$$

$$[-1, 1] - [-1, 1] + [0, 0] \doteq [0, 0]$$

$$[-1, 1] - [-1, 1] + [2, 2] \doteq [0, 0]$$

## Implementation

The choice of interval extension determines the efficiency of the propagation.

**Natural Interval Extension** Expensive to calculate, the natural extension provides maximum propagation. I.e. the natural interval extension of $x_1(x_2 + x_3)$ is $X_1(X_2 + X_3)$.

**Distributed Interval Extension** Rewriting the constraint as a sum of terms allows for more efficient calculation of box consistency, but weaker pruning. The distributed interval extension of $x_1(x_2 + x_3)$ is $X_1X_2 + X_1X_3$.

**Taylor Interval Extension** is silently ignored in this presentation.



## Splitting

Sometimes propagation just ain't enough...

Splitting corresponds to labeling with domain splitting. To achieve smaller boxes, the box given after propagation can be divided into subranges. The solver is restarted for each sub range, giving smaller new propagation opportunities.



## Summary

Interval constraints allows for a coherent handling of multiple simultaneous variable domain types.

Result: $\exists b \in Benchmarks(\exists S \in OtherSolvers(better_b(Newton, S)))$