Precise Program Analysis, Strategy Iteration and Games

Helmut Seidl

Joint work with: Thomas Gawlitza

Technische Universität München

Warsaw Sept. 11, 2008

Part 1: Program Analysis

Part 2: Games

Part 1: Program Analysis

Overview

- Precise Interval Analysis
- Integer Equations
- Strategy Iteration
- Application to Intervals
- Extensions

Interval Analysis

Given:

program variable **x** and program point *u*;

Wanted:

tight interval containing all values of **x** when reaching *u*.

Interval Analysis

Given:

program variable $\mathbf{x}$ and program point u;

Wanted:

tight interval containing all values of $\mathbf{x}$ when reaching u.

$\Longrightarrow$

Program Proving: Certifying absence of overflow;

Program Optimization: Removing array bound checks.

Example

```
for ($i = 0; i < 42; i{+}{+}$)
        if ($0 \leq i \wedge i < 42$){
                $A_1 = A + i$;
                $M[A_1] = i$;
        }
    //   $A$ initial address of an array
    //   array-bound check on $i$
```

Example

```
for ($i = 0; i < 42; i{+}{+}$)
        if ($0 \leq i \wedge i < 42$){
                $A_1 = A + i$;
                $M[A_1] = i$;
        }
    //   $A$ initial address of an array
    //   array-bound check on $i$
```

The inner check is superfluous :-)

General Approach [Cousot 1977]

- Represent the collecting semantics of the program by a set of constraints:

$$\begin{array}{lll} \mathcal{C}[\text{start}] & \supseteq & \mathbb{R}^k \\ \mathcal{C}[v] & \supseteq & [\![s]\!](\mathcal{C}[u]) \\ & & (u, s, v) \text{ control-flow edge} \end{array}$$

- Choose a suitable complete lattice $\mathbb{D}$ of abstract values which describe sets of concrete program states ...

General Approach (cont.) [Cousot 1977]

- Represent an abstraction of the collecting semantics of the program also by a set of constraints:

$$\begin{array}{lll} \mathcal{D}[\text{start}] & \sqsupseteq & \top \\ \mathcal{D}[v] & \sqsupseteq & [\![s]\!]^\sharp(\mathcal{D}[u]) \\ & & (u, s, v) \text{ controlflow edge} \end{array}$$

General Approach (cont.) [Cousot 1977]

- Represent an abstraction of the collecting semantics of the program also by a set of constraints:

$$\begin{array}{lll} \mathcal{D}[\text{start}] & \sqsupseteq & \top \\ \mathcal{D}[v] & \sqsupseteq & [\![s]\!]^\sharp(\mathcal{D}[u]) \\ & & (u, s, v) \text{ controlflow edge} \end{array}$$

Challenge:

Compute a sufficiently small solution of this system.

Examples of Abstract Lattices

Affine Equations Karr 1976

Examples of Abstract Lattices

Affine Equations	Karr 1976
Intervals	Cousot 1976

Examples of Abstract Lattices

Affine Equations	Karr 1976
Intervals	Cousot 1976
Polyhedra	Cousot, Halbwachs 1978
DBM	Sagiv 1999
Octogons	Miné 2001
TCM	Sankaranarayanan, Sipma, Manna 2005

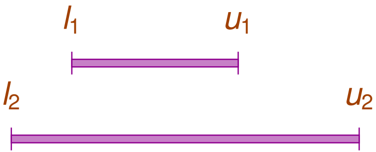
Examples of Abstract Lattices

Affine Equations	Karr 1976
Intervals	Cousot 1976
Polyhedra	Cousot, Halbwachs 1978
DBM	Sagiv 1999
Octogons	Miné 2001
TCM	Sankaranarayanan, Sipma, Manna 2005

The Lattice of Intervals Cousot 1976

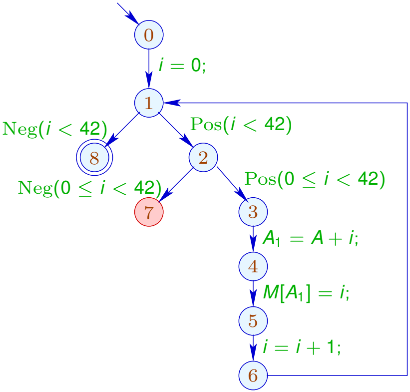
- Ordering:

$$[l_1, u_1] \sqsubseteq [l_2, u_2] \qquad \text{iff} \qquad l_2 \leq l_1 \wedge u_1 \leq u_2$$



- Put up a system of (in-)equations whose least solution approximates the intervals in question !
- Compute some solution !!

... in the Example



$$
\begin{array}{lcl}
i_1 & = & [0, 0] \sqcup i_6 \\
i_2 & = & i_1 \sqcap [-\infty, 41] \\
i_3 & = & i_2 \sqcap [0, 41] \\
i_6 & = & i_3 + [1, 1] \\
i_7 & = & i_2 \sqcap [-\infty, -1] \sqcup \\
 & & \qquad\qquad i_2 \sqcap [42, \infty] \\
i_8 & = & i_1 \sqcap [42, \infty]
\end{array}
$$

Discussion

- The least solution provides only a safe approximation to the concrete intervals :-}
- There are infinite ascending chains, e.g.,

$$\emptyset \sqsubset [0,0] \sqsubset [0,1] \sqsubset [-1,1] \sqsubset [-1,2] \sqsubset \ \ldots$$

Discussion

- The least solution provides only a safe approximation to the concrete intervals :-}
- There are infinite ascending chains, e.g.,

$$\emptyset \sqsubset [0,0] \sqsubset [0,1] \sqsubset [-1,1] \sqsubset [-1,2] \sqsubset \ldots$$

$\Longrightarrow$ Naive Kleene iteration may not terminate !

$\Longrightarrow$ Widening for fixpoint acceleration

$\Longrightarrow$ Narrowing as post-processing.

New Developments

polynomially solvable subclass	Su, Wagner, 2004
approximate solution	Costan, Gaubert et al., 2005
revision of pol. case	Reineke, G+S, Wilhelm, 2006 Leroux, 2007
exact least solution	G+S, 2007

New Developments

polynomially solvable subclass	Su, Wagner, 2004
approximate solution	Costan, Gaubert et al., 2005
revision of pol. case	Reineke, G+S, Wilhelm, 2006 Leroux, 2007
exact least solution	G+S, 2007

Simplification: Integers

- Lattice:

$$\mathcal{Z} = \{-\infty < \ldots -2 < -1 < 0 < 1 < 2 < \ldots < \infty\}$$

Simplification: Integers

- Lattice:

$$\mathcal{Z} = \{-\infty < \ldots -2 < -1 < 0 < 1 < 2 < \ldots < \infty\}$$

- Equations $x_j = e_j$ where:

$$e \quad ::= \quad z \mid x_{j'} \mid e_1 + e_2 \mid b \cdot e \mid e_1 \wedge e_2 \mid e_1 \vee e_2$$

for $z \in \mathcal{Z}, b \geq 0$.

Example

$$
\begin{array}{rcl}
i_1 & = & 0 \vee i_6 \\
i_2 & = & i_1 \wedge 41 \\
i_3 & = & i_2 \wedge 41 \\
i_6 & = & i_3 + 1 \\
i_7 & = & i_2 \wedge -1 \vee i_2 \wedge \infty \\
 & & \\
i_8 & = & i_1 \wedge \infty
\end{array}
$$

Example

$$
\begin{array}{rcl}
i_1 & = & 0 \vee i_6 \\
i_2 & = & i_1 \wedge 41 \\
i_3 & = & i_2 \wedge 41 \\
i_6 & = & i_3 + 1 \\
i_7 & = & i_2 \wedge -1 \vee i_2 \wedge \infty \\
\\
i_8 & = & i_1 \wedge \infty
\end{array}
$$

$\Longrightarrow$ almost upper half of an interval analysis :-)

Simplification: Disjunctive Systems

$$\begin{aligned} x_1 &= 2 \cdot x_2 \vee -13 \\ x_2 &= x_1 + 5 \vee -7 \\ x_3 &= 3 \cdot x_3 - 6 \vee 4 \end{aligned}$$

Simplification: Disjunctive Systems

$$\begin{aligned} x_1 &= 2 \cdot x_2 \vee -13 \\ x_2 &= x_1 + 5 \vee -7 \\ x_3 &= 3 \cdot x_3 - 6 \vee 4 \end{aligned}$$

Idea: Bellman-Ford

- Perform n rounds of RR iteration.
- Every variable which then has not yet stabilized receives value ∞ :-)

... in the Example

$$\begin{array}{lll} x_1 & = & 2 \cdot x_2 \vee -13 \\ x_2 & = & x_1 + 5 \vee -7 \\ x_3 & = & 3 \cdot x_3 - 6 \vee 4 \end{array}$$

... in the Example

$$\begin{array}{lcl} x_1 & = & 2 \cdot x_2 \vee -13 \\ x_2 & = & x_1 + 5 \vee -7 \\ x_3 & = & 3 \cdot x_3 - 6 \vee 4 \end{array}$$

	1	2	3	4
x_1	-13	-13	-13	-13
x_2	-7	-7	-7	-7
x_3	4	6	12	∞

The General Case, Idea 1

- Replace any $\wedge$-subexpression $e_1 \wedge e_2$ by one of the e_i $\Longrightarrow$ min strategy
- Solve the resulting disjunctive system
  $\Longrightarrow$ upper bound μ
- Check if μ is a solution of original system.
  Otherwise improve strategy ...

Example

$$\begin{aligned} x_1 &= 2 \cdot x_2 \wedge 1 \\ x_2 &= x_1 \vee -1 \end{aligned}$$

Example

$$\begin{array}{lll} x_1 & = & 2 \cdot x_2 \wedge 1 \\ x_2 & = & x_1 \vee -1 \end{array}$$

Approximation:

$$\begin{array}{lll} x_1 & = & 2 \cdot x_2 \wedge 1 \\ x_2 & = & x_1 \vee -1 \end{array}$$

Example

$$\begin{array}{rcl} x_1 & = & 2 \cdot x_2 \wedge 1 \\ x_2 & = & x_1 \vee -1 \end{array}$$

Approximation:

$$\begin{array}{rcl} x_1 & = & 2 \cdot x_2 \wedge 1 \\ x_2 & = & x_1 \vee -1 \end{array}$$

has solution: $\mu = \{x_1 \mapsto 1, x_2 \mapsto 1\}$

which is a solution of the original system
— but not the least one :-(

The General Case, Idea 2

- Approximate general systems by conjunctive systems

  ...

- Approximate the least solution by feasible solutions of conjunctive systems :-)

- Compute these by greatest fixpoint iteration :-))

... in the Example

Initial Assignment:

$$\mu_0 = \{x_1 \mapsto -\infty, x_2 \mapsto -\infty\}$$

... in the Example

Initial Assignment:

$$\mu_0 = \{x_1 \mapsto -\infty, x_2 \mapsto -\infty\}$$

Decorated System of Equations:

$$\begin{array}{lll} x_1 & = & 2 \cdot x_2 \wedge 1 \vee -\infty \\ x_2 & = & x_1 \vee -1 \end{array}$$

... in the Example

Initial Assignment:

$$\mu_0 = \{x_1 \mapsto -\infty, x_2 \mapsto -\infty\}$$

Corresponding Conjunctive System:

$$\begin{aligned} x_1 &= 2 \cdot x_2 \wedge 1 \vee -\infty \\ x_2 &= x_1 \vee -1 \end{aligned}$$

... in the Example

Initial Assignment:

$$\mu_0 = \{x_1 \mapsto -\infty, x_2 \mapsto -\infty\}$$

Corresponding Conjunctive System:

$$\begin{aligned} x_1 &= 2 \cdot x_2 \wedge 1 \vee -\infty \\ x_2 &= x_1 \vee -1 \end{aligned}$$

... with solution:

$$\mu_1 = \{x_1 \mapsto -\infty, x_2 \mapsto -1\}$$

... in the Example

Next Assignment:

$$\mu_1 = \{x_1 \mapsto -\infty, x_2 \mapsto -1\}$$

... in the Example

Next Assignment:

$$\mu_1 = \{x_1 \mapsto -\infty, x_2 \mapsto -1\}$$

Corresponding Conjunctive System:

$$\begin{array}{lll} x_1 & = & 2 \cdot x_2 \wedge 1 \vee -\infty \\ x_2 & = & x_1 \vee -1 \end{array}$$

... in the Example

Next Assignment:

$$\mu_1 = \{x_1 \mapsto -\infty, x_2 \mapsto -1\}$$

Corresponding Conjunctive System:

$$\begin{array}{lll} x_1 & = & 2 \cdot x_2 \wedge 1 \vee -\infty \\ x_2 & = & x_1 \vee -1 \end{array}$$

... with greatest solution:

$$\mu_1 = \{x_1 \mapsto -2, x_2 \mapsto -1\}$$

☐

Why it works ...

- We only make profitable switches of decisions.
- The resulting variable assignments are feasible pre-fixpoints.
- Every occurring conjunctive system has exactly one solution exceeding the given pre-fixpoint — which thus is the greatest.

Complexity

$$\mathcal{O}(n \cdot |\mathcal{E}| \cdot (\Pi(m) + n))$$

where:

n :	number of variables
$\vert\mathcal{E}\vert$:	size of the equation system
m :	number of $\vee$
$\Pi(\,.\,)$:	number of strategy iterations

Remark

- The run-time is independent of the sizes of occurring coefficients :-)
- Strategy iteration is a well-known technique in control-theory as well as algorithmic game theory :-)
- Trivially, $\Pi(m) \leq 2^m$:-}
- No example with more than linear iterations has been found so far !!

Application to Interval Equations

- Sub-expressions which evaluate to $\emptyset$ can be eliminated.

Application to Interval Equations

- Sub-expressions which evaluate to $\emptyset$ can be eliminated.
- If all sub-expressions evaluate to non-empty intervals, the computation of upper and lower interval bounds becomes independent.
- An interval variable x can then be simulated by:

x^+ $==$ upper bound

x^- $==$ negated lower bound

Application to Interval Equations

- Sub-expressions which evaluate to $\emptyset$ can be eliminated.
- If all sub-expressions evaluate to non-empty intervals, the computation of upper and lower interval bounds becomes independent.
- An interval variable x can then be simulated by:

$$\begin{array}{lcr} x^{+} & == & \text{upper bound} \\ x^{-} & == & \text{negated lower bound} \end{array}$$

- Any algorithm for integer systems can also solve interval systems :-))

... in Our Example

$$\begin{array}{lcl}
i_1 & = & [0,0] \sqcup i_6 \\
i_2 & = & i_1 \sqcap [-\infty, 41] \\
i_3 & = & i_2 \sqcap [0, 41] \\
i_6 & = & i_3 + [1,1] \\
i_7 & = & i_2 \sqcap [-\infty, -1] \\
 & & \sqcup \; i_2 \sqcap [42, \infty] \\
i_8 & = & i_1 \sqcap [42, \infty]
\end{array}$$

... in Our Example

$$\begin{array}{lcl} i_1 & = & [0,0] \sqcup i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0, 41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup\ i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

First Assumption: all variables have value $\emptyset$ ...

... in Our Example

$$\begin{array}{rcl}
i_1 & = & [0,0] \sqcup i_6 \\
i_2 & = & i_1 \sqcap [-\infty, 41] \\
i_3 & = & i_2 \sqcap [0, 41] \\
i_6 & = & i_3 + [1,1] \\
i_7 & = & i_2 \sqcap [-\infty, -1] \\
 & & \sqcup\ i_2 \sqcap [42, \infty] \\
i_8 & = & i_1 \sqcap [42, \infty]
\end{array}$$

System for interval bounds:

$$\boxed{\begin{array}{rcl}
i_1^- & = & 0 \\
i_1^+ & = & 0
\end{array}}$$

... in Our Example

$$\begin{array}{lcl} i_1 & = & [0,0] \sqcup i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0, 41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup\ i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

System for interval bounds:

$$\boxed{\begin{array}{lcl} i_1^- & = & 0 \\ i_1^+ & = & 0 \end{array}}$$

results in

$$\mu_1 = \{ i_1 \mapsto [0,0], i_2 \mapsto \emptyset, i_3 \mapsto \emptyset, i_6 \mapsto \emptyset, i_7 \mapsto \emptyset, i_8 \mapsto \emptyset \}$$

... in Our Example

$$
\begin{array}{lcl}
i_1 & = & [0,0] \sqcup i_6 \\
i_2 & = & i_1 \sqcap [-\infty, 41] \\
i_3 & = & i_2 \sqcap [0, 41] \\
i_6 & = & i_3 + [1,1] \\
i_7 & = & i_2 \sqcap [-\infty, -1] \\
 & & \sqcup\ i_2 \sqcap [42, \infty] \\
i_8 & = & i_1 \sqcap [42, \infty]
\end{array}
$$

System for interval bounds:

$$
\boxed{
\begin{array}{lcl}
i_1^- & = & 0 \\
i_2^- & = & i_1^- \wedge \infty \\
i_1^+ & = & 0 \\
i_2^+ & = & i_1^+ \wedge 41
\end{array}
}
$$

since

$$\mu_1 = \{ i_1 \mapsto [0,0], i_2 \mapsto \emptyset, i_3 \mapsto \emptyset, i_6 \mapsto \emptyset, i_7 \mapsto \emptyset, i_8 \mapsto \emptyset \}$$

... in Our Example

$$\begin{array}{lcl} i_1 & = & [0,0] \sqcup i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0, 41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup\ i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

System for interval bounds:

$$\boxed{\begin{array}{lcl} i_1^- & = & 0 \\ i_2^- & = & i_1^- \wedge \infty \\ i_1^+ & = & 0 \\ i_2^+ & = & i_1^+ \wedge 41 \end{array}}$$

results in

$$\mu_2 = \{ i_1 \mapsto [0,0], i_2 \mapsto [0,0], i_3 \mapsto \emptyset, i_6 \mapsto \emptyset, i_7 \mapsto \emptyset, i_8 \mapsto \emptyset \}$$

... in Our Example

$$\begin{array}{lcl} i_1 & = & [0,0] \sqcup i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0, 41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup\, i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

System for interval bounds:

$$\boxed{\begin{array}{lcl} i_1^- & = & 0 \wedge i_6^- \\ i_2^- & = & i_1^- \vee 41 \\ i_3^- & = & i_2^- \vee 41 \\ i_6^- & = & i_3^- - 1 \\ \ldots \end{array}}$$

since

$$\mu_2 = \{i_1 \mapsto [0,0], i_2 \mapsto [0,0], i_3 \mapsto \emptyset, i_6 \mapsto \emptyset, i_7 \mapsto \emptyset, i_8 \mapsto \emptyset\}$$

... in Our Example

$$\begin{array}{lcl} i_1 & = & [0,0] \sqcup \; i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0, 41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup \; i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

System for interval bounds:

$$\boxed{\begin{array}{lcl} \ldots & & \\ i_1^+ & = & 0 \vee i_6^+ \\ i_2^+ & = & i_1^+ \wedge 41 \\ i_3^+ & = & i_2^+ \wedge 41 \\ i_6^+ & = & i_3^+ + 1 \end{array}}$$

results in

$$\begin{array}{ll} \mu_3 = \{ & i_1 \mapsto [0, 42], i_2 \mapsto [0, 41], i_3 \mapsto [0, 41], \\ & i_6 \mapsto [0, 42], i_7 \mapsto \emptyset, i_8 \mapsto \emptyset \; \} \end{array}$$

... in Our Example

$$\begin{array}{lll} i_1 & = & [0,0] \sqcup \; i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0,41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup \; i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

since

$$\mu_3 = \{ \quad \begin{array}{l} i_1 \mapsto [0,42], i_2 \mapsto [0,41], i_3 \mapsto [0,41], \\ i_6 \mapsto [0,42], i_7 \mapsto \emptyset, i_8 \mapsto \emptyset \; \} \end{array}$$

... in Our Example

$$\begin{array}{lcl} i_1 & = & [0,0] \sqcup \; i_6 \\ i_2 & = & i_1 \sqcap [-\infty, 41] \\ i_3 & = & i_2 \sqcap [0, 41] \\ i_6 & = & i_3 + [1,1] \\ i_7 & = & i_2 \sqcap [-\infty, -1] \\ & & \sqcup \; i_2 \sqcap [42, \infty] \\ i_8 & = & i_1 \sqcap [42, \infty] \end{array}$$

gives us the final result:

$$\begin{array}{ll} \mu_4 = \{ & i_1 \mapsto [0,42], i_2 \mapsto [0,41], i_3 \mapsto [0,41], \\ & i_6 \mapsto [0,42], i_7 \mapsto \emptyset, i_8 \mapsto [42,42] \; \} \end{array}$$

Extensions

Additionally, we need constructions for:

- negations (easy)
- multiplication (complicated)

Perspectives

- The run-time of the fixpoint algorithm is independent of the sizes of numbers.
- It computes the least solution of the interval constraints precisely, not just an upper bound.

Perspectives

- The run-time of the fixpoint algorithm is independent of the sizes of numbers.
- It computes the least solution of the interval constraints precisely, not just an upper bound.
- Techniques can be extended also to deal with equations over rationals and thus solve simple forms of polyhedral analysis!

Perspectives

- The run-time of the fixpoint algorithm is independent of the sizes of numbers.
- It computes the least solution of the interval constraints precisely, not just an upper bound.
- Techniques can be extended also to deal with equations over rationals and thus solve simple forms of polyhedral analysis!

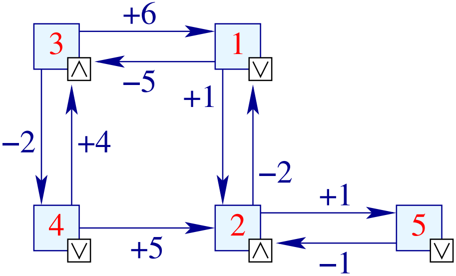
Are interval equations in P?

Part 2: Games

Overview

- Mean Payoff Games
- Total Payoff Games
- Finite Hierarchical Integer Equations
- Simple Integer Equations
- A Lower Bound for Precise Interval Analysis

2-Person-Zero-Sum Infinite Games



Mean Payoff

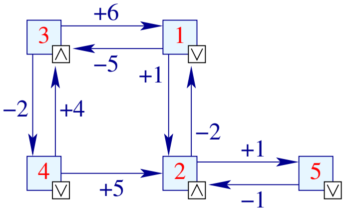
Play Value for the infinite play $\pi = v_1 v_2 \cdots v_n \cdots$:

$$val(\pi) \;=\; \liminf_{n\to\infty} \frac{\rho\,(v_1, v_2) + \ldots + \rho\,(v_n, v_{n+1})}{n}$$

Mean Payoff

Play Value for the infinite play $\pi = v_1 v_2 \cdots v_n \cdots$:

$$val(\pi) = \liminf_{n\to\infty} \frac{\rho(v_1, v_2) + \ldots + \rho(v_n, v_{n+1})}{n}$$



$$val(v_3(v_1 v_2)^\omega) = -\frac{1}{2}$$

Mean Payoff Games

Game Value:

$\mu(v) :=$ maximal play value which
$\vee$-player can enforce
for plays starting at v

Mean Payoff Games

Game Value:

$\mu(v) :=$ maximal play value which
$\vee$-player can enforce
for plays starting at v

Property:

Both players have a positional optimal strategy :-)

Total Payoff

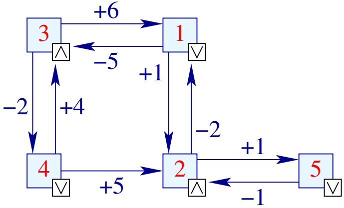
Play Value for the infinite play $\pi = v_1 v_2 \cdots v_n \cdots$:

$$val(\pi) \;=\; \liminf_{n\to\infty} \rho\,(v_1, v_2) + \ldots + \rho\,(v_n, v_{n+1})$$

Total Payoff

Play Value for the infinite play $\pi = v_1 v_2 \cdots v_n \cdots$:

$$val(\pi) \;=\; \liminf_{n\to\infty} \rho\,(v_1, v_2) + \ldots + \rho\,(v_n, v_{n+1})$$



$$val(v_3(v_1 v_2)^\omega) \;=\; -\infty$$
$$val(v_5(v_2 v_1 v_3 v_1)^\omega) \;=\; -8$$

Total Payoff Games

Game Value:

$\tau(v) :=$ maximal total play value which
$\vee$-player can enforce
for plays starting at v

Total Payoff Games

Game Value:

$\tau(v) :=$ maximal total play value which
$\vee$-player can enforce
for plays starting at v

Property: Zielonka, Gimbert 2004

Both players have a positional optimal strategy :-)

Results

Mean Payoff Games $\Longleftrightarrow$

Total Payoff Games $\Longleftrightarrow$

Finite Hierarchical Systems $\Longleftrightarrow$

Simple Integer Equations

Mean Payoff Through Total Payoff

Lemma

For every position v,

$$\begin{array}{llllll} \mu(v) & < & 0 & \text{iff} & \tau(v) & = -\infty \\ \mu(v) & = & 0 & \text{iff} & \tau(v) & \text{finite} \\ \mu(v) & > & 0 & \text{iff} & \tau(v) & = \infty \end{array}$$

Mean Payoff Through Total Payoff

Lemma

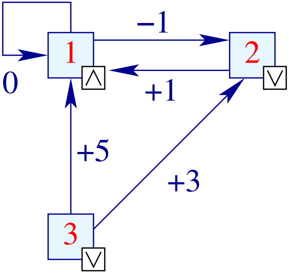
For every position v,

$$\begin{array}{llllll} \mu(v) & < & 0 & \text{iff} & \tau(v) & = -\infty \\ \mu(v) & = & 0 & \text{iff} & \tau(v) & \text{finite} \\ \mu(v) & > & 0 & \text{iff} & \tau(v) & = \infty \end{array}$$

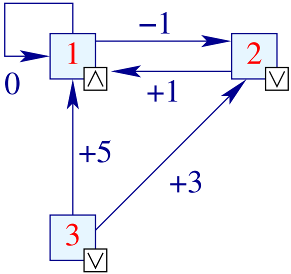
Theorem

Mean-payoff games are polynomially reducible to total-payoff games.

Total Payoff through Equations

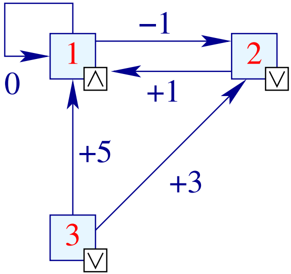


Total Payoff through Equations



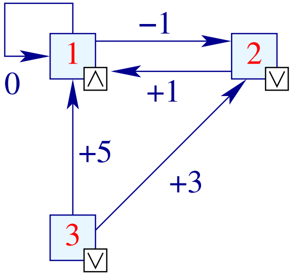
$$
\begin{aligned}
x_1 &= x_1 \wedge x_2 - 1 \\
x_2 &= x_1 + 1 \\
x_3 &= x_1 + 5 \vee x_2 + 3
\end{aligned}
$$

Total Payoff through Equations



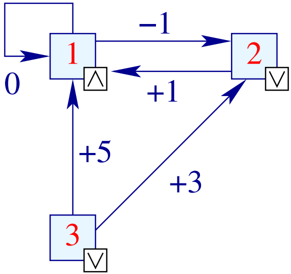
$$
\begin{aligned}
x_1 &= y_1 \wedge x_1 \wedge x_2 - 1 \\
x_2 &= y_2 \wedge x_1 + 1 \\
x_3 &= y_3 \wedge (x_1 + 5 \vee x_2 + 3)
\end{aligned}
$$

Total Payoff through Equations



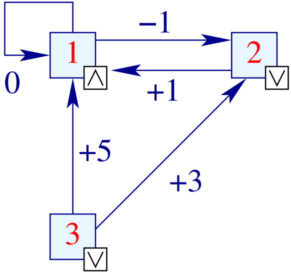
$$
\begin{array}{lcl}
y_1 & = & 0 \vee (x_1 \wedge x_2 - 1) \\
y_2 & = & 0 \vee \ \ x_1 + 1 \\
y_3 & = & 0 \vee \ \ x_1 + 5 \vee x_2 + 3 \\
x_1 & = & y_1 \ \wedge \ \ x_1 \wedge x_2 - 1 \\
x_2 & = & y_2 \ \wedge \ \ x_1 + 1 \\
x_3 & = & y_3 \ \wedge \ (x_1 + 5 \vee x_2 + 3)
\end{array}
$$

Total Payoff through Equations



$$
\begin{aligned}
y_1 &= 0 \vee (x_1 \wedge x_2 - 1) \\
y_2 &= 0 \vee \;\; x_1 + 1 \\
y_3 &= 0 \vee \;\; x_1 + 5 \vee x_2 + 3 \\
x_1 &= y_1 \wedge \;\; x_1 \wedge x_2 - 1 \;\; \vee -3 \\
x_2 &= y_2 \wedge \;\; x_1 + 1 \;\; \vee -3 \\
x_3 &= y_3 \wedge (x_1 + 5 \vee x_2 + 3) \;\vee -3
\end{aligned}
$$

Total Payoff through Equations



$$
\begin{array}{lcl}
y_1 & = & 0 \vee (x_1 \wedge x_2 - 1) \;\wedge\; 20 \\
y_2 & = & 0 \vee \;\; x_1 + 1 \;\; \wedge\; 20 \\
y_3 & = & 0 \vee (x_1 + 5 \vee x_2 + 3) \wedge 20 \\
x_1 & = & y_1 \;\wedge\;\; x_1 \wedge x_2 - 1 \;\; \vee -3 \\
x_2 & = & y_2 \;\wedge\;\; x_1 + 1 \;\; \vee -3 \\
x_3 & = & y_3 \;\wedge\; (x_1 + 5 \vee x_2 + 3) \;\vee -3
\end{array}
$$

Total Payoff through Equations

- All solutions are finite.
- The outer y_i are least-fixpoint variables whereas the inner x_i are greatest-fixpoint variables!

$\Longrightarrow$ finite hierarchical system

Total Payoff through Equations

- All solutions are finite.
- The outer y_i are least-fixpoint variables whereas the inner x_i are greatest-fixpoint variables!

$\Longrightarrow$ finite hierarchical system

Property: G+S, 2007

Finite hierarchical systems have optimal positional strategies.

Total Payoff through Equations

Theorem

Let ν denote the canonical solution of the hierarchical system. Then

$$\tau(v_i) = \begin{cases} -\infty & \text{if} \quad \nu(x_i) < A \\ \infty & \text{if} \quad \nu(x_i) > B \\ \nu(x_i) & \text{otherwise} \end{cases}$$

for all positions v_i of the total payoff game where:

$$\begin{array}{lcl} A & = & \text{sum of occurring negative values} \\ B & = & \text{sum of occurring positive values} \end{array}$$

Hierarchical through Simple Equations

Idea:

- Nested finite fixpoints over $\mathcal{Z}$ can be read of from a unique fixpoint over an instrumented lattice

G+S, 2007

Hierarchical through Simple Equations

Idea:

- Nested finite fixpoints over $\mathcal{Z}$ can be read of from a unique fixpoint over an instrumented lattice

  G+S, 2007

- The instrumented lattice can be coded into a scaled finite range of integers ...

Hierarchical through Simple Equations

Idea:

- Nested finite fixpoints over $\mathcal{Z}$ can be read of from a unique fixpoint over an instrumented lattice

  G+S, 2007

- The instrumented lattice can be coded into a scaled finite range of integers ...

- Scaling factor: $2 \cdot (n + 1)^d$ where:

  n number of variables
  d number of different ranks of fixpoints

Hierarchical through Simple Equations

... in the Example:

$$y_1 \quad = \quad 0 \vee x_1 \wedge x_2 - 1 \wedge 20$$

scaled with $2 \cdot 7^2 = 98$ becomes:

$$y_1 \quad = \quad 0 \vee x_1 \wedge x_2 - 98 \wedge 1960$$

Hierarchical through Simple Equations

... in the Example:

$$y_1 \quad = \quad 0 \vee x_1 \wedge x_2 - 1 \wedge 20$$

scaled with $2 \cdot 7^2 = 98$ becomes:

$$y_1 \quad = \quad (0 \vee x_1 \wedge x_2 - 98 \wedge 1960) - 7^1$$

Hierarchical through Simple Equations

... in the Example:

$$y_1 = 0 \vee x_1 \wedge x_2 - 1 \wedge 20$$

scaled with $2 \cdot 7^2 = 98$ becomes:

$$y_1 = (0 \vee x_1 \wedge x_2 - 98 \wedge 1960) - 7^1$$

$$x_1 = (y_1 \wedge x_1 \wedge x_2 - 1) \vee -3$$

scaled with 98 becomes:

$$x_1 = (y_1 \wedge x_1 \wedge x_2 - 98) \vee -294$$

Hierarchical through Simple Equations

... in the Example:

$$y_1 \quad = \quad 0 \vee x_1 \wedge x_2 - 1 \wedge 20$$

scaled with $2 \cdot 7^2 = 98$ becomes:

$$y_1 \quad = \quad (0 \vee x_1 \wedge x_2 - 98 \wedge 1960) - 7^1$$

$$x_1 \quad = \quad (y_1 \wedge x_1 \wedge x_2 - 1) \vee -3$$

scaled with 98 becomes:

$$x_1 \quad = \quad ((y_1 \wedge x_1 \wedge x_2 - 98) \vee -294) + 7^0$$

Hierarchical through Simple Equations

All together:

$$\begin{array}{lcl} y_1 & = & -7 \vee x_1 - 7 \wedge x_2 - 105 \wedge 1953 \\ y_2 & = & -7 \vee \ \ x_1 + 91 \ \ \wedge 1953 \\ y_3 & = & -7 \vee (x_1 + 483 \vee x_2 + 287) \wedge 1953 \\ \\ x_1 & = & (y_1 + 1 \wedge x_1 + 1 \wedge x_2 - 97) \vee -293 \\ x_2 & = & (y_2 + 1 \wedge x_1 + 99) \vee -293 \\ x_3 & = & (y_3 + 1 \wedge (x_1 + 491 \vee x_2 + 295)) \vee -293 \end{array}$$

Hierarchical through Simple Equations

- All solutions of the resulting system are still finite :-)
- Every simple cycle in the resulting system has cost different from 0

$\Longrightarrow$ simple system of equations

Hierarchical through Simple Equations

- All solutions of the resulting system are still finite :-)
- Every simple cycle in the resulting system has cost different from 0

$\Longrightarrow$ simple system of equations

- Simple systems have unique fixpoints !

Hierarchical through Simple Equations

- All solutions of the resulting system are still finite :-)
- Every simple cycle in the resulting system has cost different from 0

$\Longrightarrow$ simple system of equations

- Simple systems have unique fixpoints !

Theorem

Finite hierarchical systems can be reduced to simple systems of equations.

Simple Equations through Mean Payoff

System:

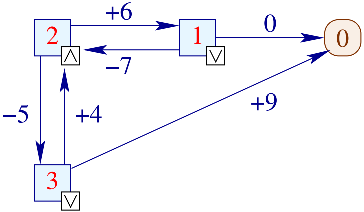
$$\begin{aligned} x_1 &= 0 \vee x_2 - 7 \\ x_2 &= x_1 + 6 \wedge x_2 - 5 \\ x_3 &= 9 \vee x_2 + 4 \end{aligned}$$

Simple Equations through Mean Payoff

System:

$$\begin{aligned} x_1 &= 0 \vee x_2 - 7 \\ x_2 &= x_1 + 6 \wedge x_2 - 5 \\ x_3 &= 9 \vee x_2 + 4 \end{aligned}$$

Graphical Representation:

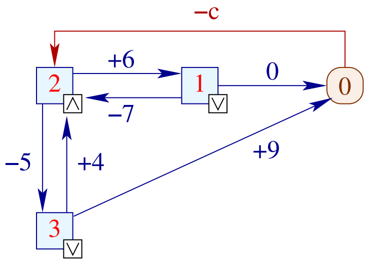


Simple Equations through Mean Payoff

System:

$$\begin{aligned} x_1 &= 0 \vee x_2 - 7 \\ x_2 &= x_1 + 6 \wedge x_2 - 5 \\ x_3 &= 9 \vee x_2 + 4 \end{aligned}$$

Mean-Payoff Game:



Simple Equations through Mean Payoff

Lemma

Let τ denote the game values of the mean-payoff game with back-edge to x_i with weight $-c$.
Then

$$\nu(x_i) \geq c \quad \text{iff} \quad \tau(x_i) \geq 0$$

Simple Equations through Mean Payoff

Lemma

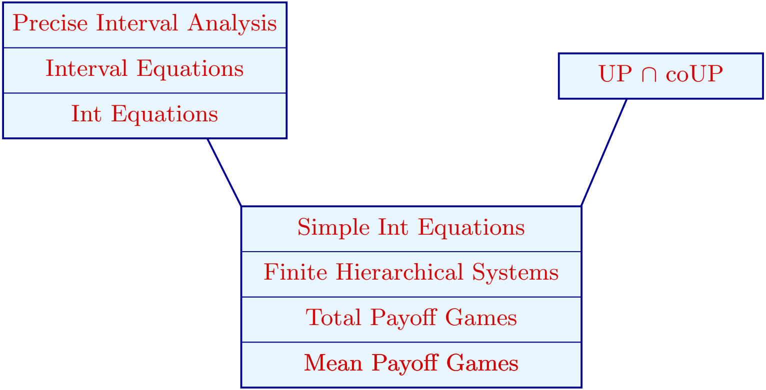
Let τ denote the game values of the mean-payoff game with back-edge to x_i with weight $-c$.
Then

$$\nu(x_i) \geq c \quad \text{iff} \quad \tau(x_i) \geq 0$$

Theorem

Simple equations are polynomially reducible to mean-payoff games.

Summary



Precise Interval Analysis

Interval Equations

Int Equations

UP ∩ coUP

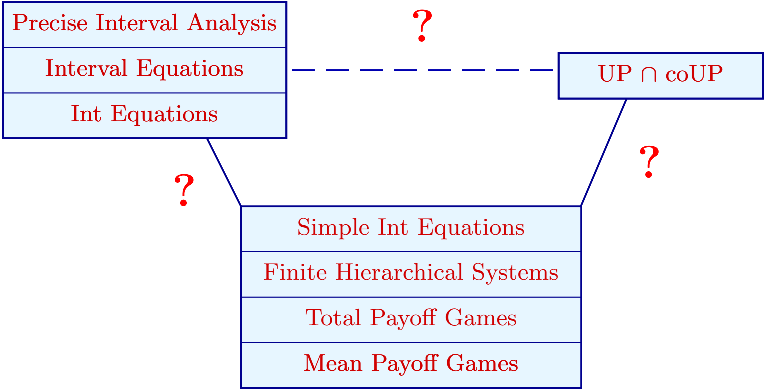
Simple Int Equations

Finite Hierarchical Systems

Total Payoff Games

Mean Payoff Games

Open Problems



Precise Interval Analysis

Interval Equations

Int Equations

UP ∩ coUP

Simple Int Equations

Finite Hierarchical Systems

Total Payoff Games

Mean Payoff Games