Testing for Termination with Monotonicity Constraints

M. Codish[1], V. Lagoon[2], P.J. Stuckey[2]

mcodish@cs.bgu.ac.il, (lagoon|pjs)@cs.mu.oz.au

[1] Dept. of CS, Ben-Gurion University, Israel

[2] Dept. of CS & SE, University of Melbourne, Australia

In This Talk

The current state of affairs:

- the local approach (one loop at a time): correctness proof given by Dershowitz *et al*, 2001 (also by Codish&Genaim, POPL2001)
- completeness of termination for size-change graphs given by Lee, Jones and Ben-Amram, POPL2001.
- termination analyzers: TermiLog (Lindenstrauss, Sagiv, 1997), TerminWeb (Genaim, Taboch, Codish, 2002)

Our contribution:

- the termination test used in TermiLog and TerminWeb is complete for size-change graphs
- completeness of termination analysis for monotonicity constraints (extension of size-change graphs)

Approximation of Termination

Programs are mapped to sets of loop descriptions

$$\text{PROGRAM} \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu \;,\; p(\bar{y}) \\ \vdots \end{array} \right.$$

Approximation of Termination

Programs are mapped to sets of loop descriptions

$$\text{PROGRAM} \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu,\ p(\bar{y}) \\ \vdots \end{array} \right.$$

Approximation of Termination

Programs are mapped to sets of loop descriptions

$$\text{PROGRAM} \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu,\ p(\bar{y}) \\ \vdots \end{array} \right.$$

Size-Change Graphs

$x_i > y_j$, $x_i \geq y_j$,

$x_i \in \bar{x}$, $y_j \in \bar{y}$

(Lee,Jones,Ben-Amram 2001)

Approximation of Termination

Programs are mapped to sets of loop descriptions

$$\text{PROGRAM} \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu,\ p(\bar{y}) \\ \vdots \end{array} \right.$$

Size-Change Graphs

$x_i > y_j$, $x_i \geq y_j$,

$x_i \in \bar{x}$, $y_j \in \bar{y}$

(Lee,Jones,Ben-Amram 2001)

Monotonicity Constraints

$v > w$, $v \geq w$,

$v, w \in (\bar{x} \cup \bar{y})$

(Lindenstrauss,Sagiv, 1989)

Approximation of Termination

Programs are mapped to sets of loop descriptions

$$\text{PROGRAM} \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu,\ p(\bar{y}) \\ \vdots \end{array} \right.$$

Size-Change Graphs

$x_i > y_j,\ x_i \geq y_j,$

$x_i \in \bar{x},\ y_j \in \bar{y}$

(Lee,Jones,Ben-Amram 2001)

Monotonicity Constraints

$v > w,\ v \geq w,$

$v, w \in (\bar{x} \cup \bar{y})$

(Lindenstrauss,Sagiv, 1989)

Linear Constraints

$\sum c_i v_i > c$

$v_i \in (\bar{x} \cup \bar{y})$

(Podelski,Rybalchenko, 2004)

Approximation of Termination

Programs are mapped to sets of loop descriptions

$$\text{PROGRAM} \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu,\ p(\bar{y}) \\ \vdots \end{array} \right.$$

Size-Change Graphs	Monotonicity Constraints	Linear Constraints
$x_i > y_j,\ x_i \geq y_j,$	$v > w,\ v \geq w,$	$\sum c_i v_i > c$
$x_i \in \bar{x},\ y_j \in \bar{y}$	$v, w \in (\bar{x} \cup \bar{y})$	$v_i \in (\bar{x} \cup \bar{y})$
(Lee,Jones,Ben-Amram 2001)	(Lindenstrauss,Sagiv, 1989)	(Podelski,Rybalchenko, 2004)

- Is termination decidable?
  (It's not about solving the Halting Problem!)
- What's the algorithm? What's the cost?

Size-Change Graphs by Example

```
int  Ack(int x, int y) {
        if (x==0) return y + 1
        else if (y==0) return Ack(x − 1, 1)
        else return Ack(x − 1, Ack(x, y − 1))
}
```

Size-Change Graphs by Example

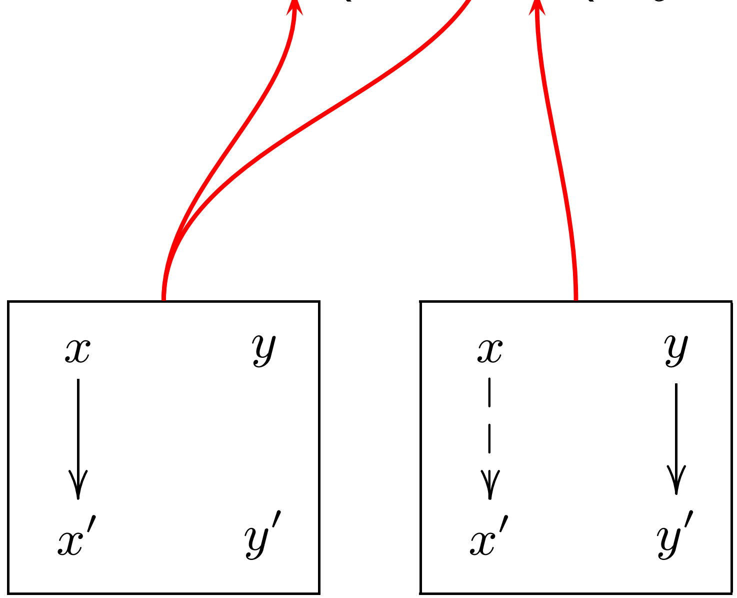
int Ack(int x, int y) {

  if (x==0) **return** $y + 1$

  else if (y==0) **return** Ack($x - 1$, 1)

  else return Ack($x - 1$, Ack(x, $y - 1$))

}



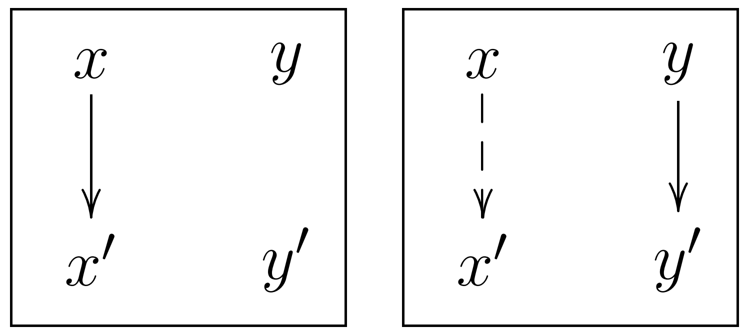
Local vs. Global

Global approach – find a *ranking function* f:

$$\exists f\ \forall loop.\ f \text{ decreases on the } loop$$

$Ack(x, y)$

$f : \min(x, y)$



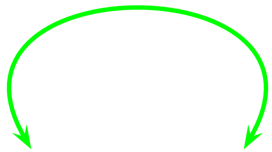
Local approach – one loop at a time

- for a “price” it is sufficient to prove termination for each individual loop
- there is also a “prize”: simpler termination conditions and easier to automate

One Loop at a Time

$\exists f\ \forall loop.\ f$ decreases on the $loop$

One Loop at a Time



$\exists f \; \forall loop.\; f$ decreases on the $loop$

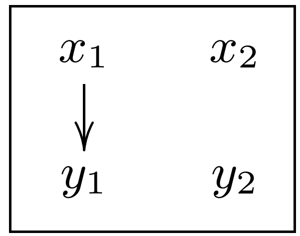
One Loop at a Time

$\forall loop\ \exists f.\ f$ decreases on the $loop$

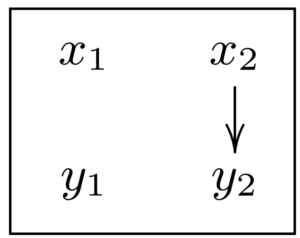
One Loop at a Time

$\forall loop \; \exists f .\; f$ decreases on the $loop$

$p(x_1, x_2) \leftarrow [x_1 > y_1],\; p(y_1, y_2)$:



$p(x_1, x_2) \leftarrow [x_2 > y_2],\; p(y_1, y_2)$:

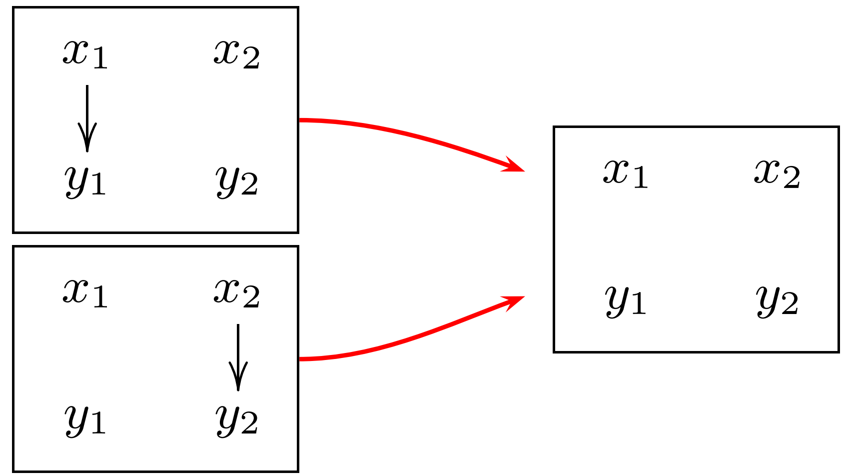


One Loop at a Time

$\forall loop\ \exists f.\ f$ decreases on the $loop$

$p(x_1, x_2) \leftarrow [x_1 > y_1],\ p(y_1, y_2)$:

$p(x_1, x_2) \leftarrow [x_2 > y_2],\ p(y_1, y_2)$:



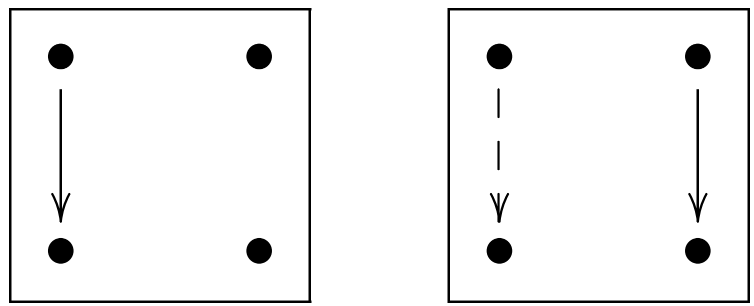
A composition $\mu_1(\bar{x}, \bar{y}) \circ \mu_2(\bar{x}, \bar{y})$ is a size-change graph entailed by $\exists \bar{z}.\mu_1(\bar{x}, \bar{z}) \wedge \mu_2(\bar{z}, \bar{y})$.
Loop descriptions are closed under composition:

$$\mu_1 \in G^* \wedge \mu_2 \in G^* \ \Rightarrow\ \mu_1 \circ \mu_2 \in G^*$$

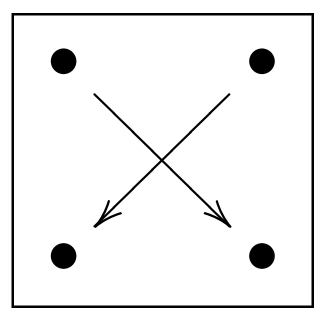
Idempotence

A size-change graph μ is idempotent if $\mu \circ \mu = \mu$.

Example: the two graphs of $Ack(x, y)$ are idempotent:



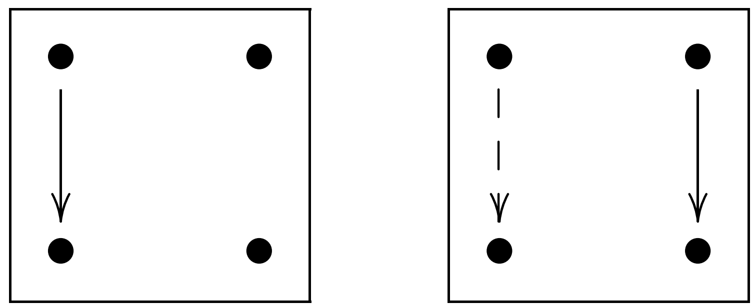
Example: a non-idempotent graph:



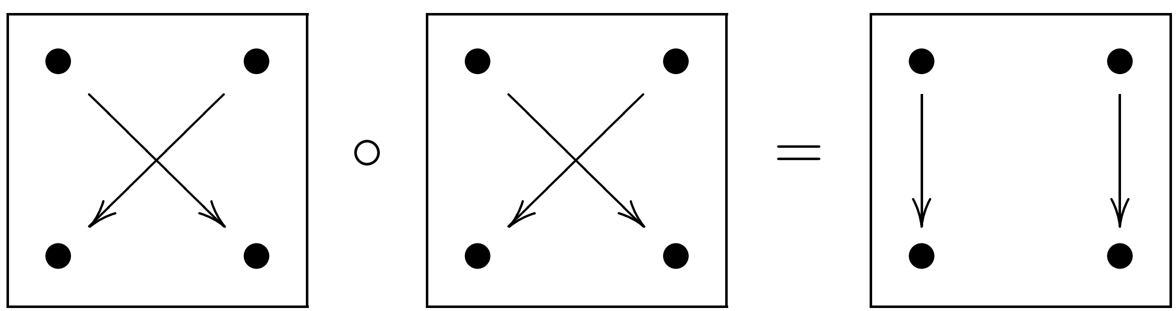
Idempotence

A size-change graph μ is idempotent if $\mu \circ \mu = \mu$.

Example: the two graphs of $Ack(x, y)$ are idempotent:



Example: a non-idempotent graph:



Theorem: each SCG has an idempotent degree

Correctness of Local Approach

Let G be a set of size-change graphs. If every idempotent $\mu \in G^*$ has a ranking function then any program described by G terminates. (Dershowitz *et al*, 2001) (Lee,Jones,Ben-Amram 2001)

Correctness of Local Approach

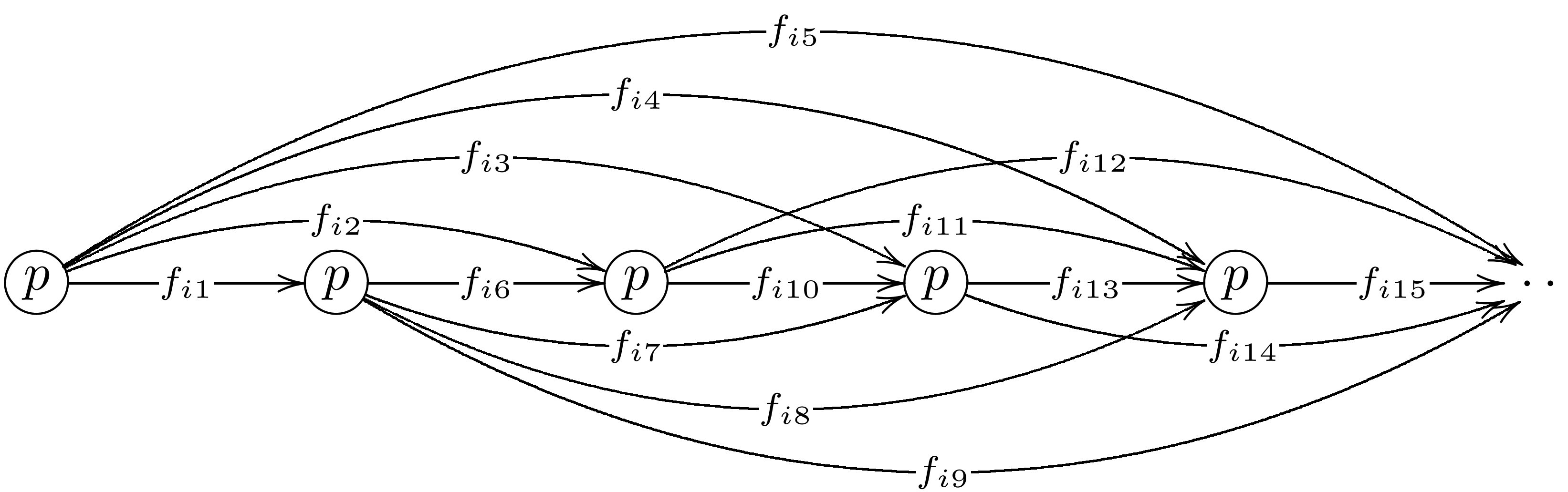
Let G be a set of size-change graphs. If every idempotent $\mu \in G^*$ has a ranking function then any program described by G terminates. (Dershowitz *et al*, 2001) (Lee,Jones,Ben-Amram 2001)

Ramsey's theorem (1930): Let X be some countably infi nite set and colour the pairs in $X \times X$ in C different colours. Then there exists some infi nite $M \subset X$ such that the pairs of M all have the same colour.

Correctness of Local Approach

Let G be a set of size-change graphs. If every idempotent $\mu \in G^*$ has a ranking function then any program described by G terminates. (Dershowitz *et al*, 2001) (Lee,Jones,Ben-Amram 2001)

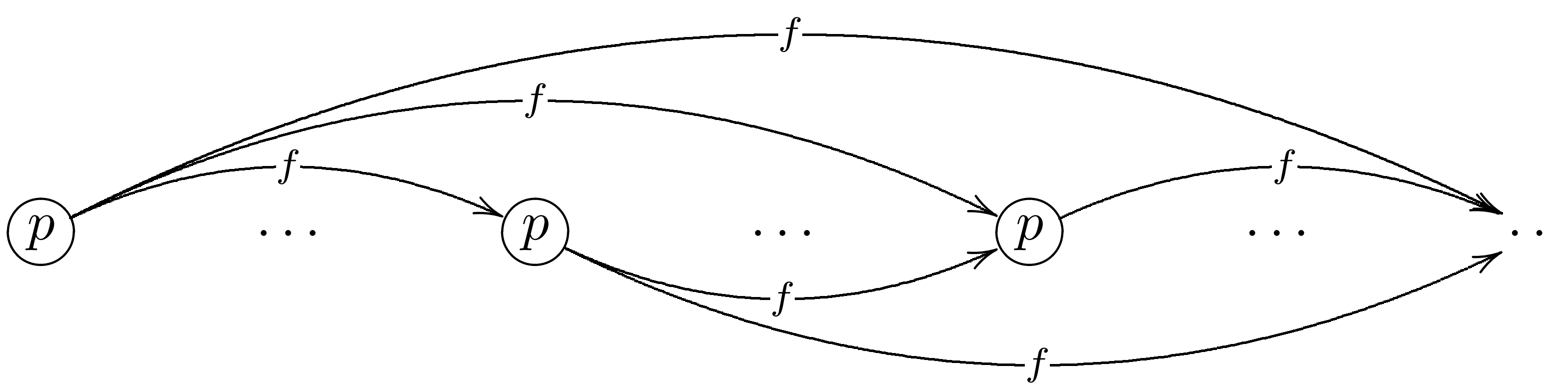
Ramsey's theorem (1930): Let X be some countably infi nite set and colour the pairs in $X \times X$ in C different colours. Then there exists some infi nite $M \subset X$ such that the pairs of M all have the same colour.



Correctness of Local Approach

Let G be a set of size-change graphs. If every idempotent $\mu \in G^*$ has a ranking function then any program described by G terminates. (Dershowitz *et al*, 2001) (Lee,Jones,Ben-Amram 2001)

Ramsey's theorem (1930): Let X be some countably infi nite set and colour the pairs in $X \times X$ in C different colours. Then there exists some infi nite $M \subset X$ such that the pairs of M all have the same colour.



Completeness

Completness means that we can eiher find a ranking function or prove that none exists.

If there is ANY ranking function. . .

Completeness

Completness means that we can eiher find a ranking function or prove that none exists.

If there is ANY ranking function. . .

$$f(\langle x_1, \ldots, x_n \rangle) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})}$$

Completeness

Completness means that we can eiher find a ranking function or prove that none exists.

If there is ANY ranking function. . .

$$f(\langle x_1, \ldots, x_n \rangle) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})}$$

$$f(\langle x_0, \ldots, x_{n-1} \rangle) = \sum_{j=0}^{n-1} \left| \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \right|$$

Completeness

Completness means that we can eiher find a ranking function or prove that none exists.

If there is ANY ranking function…

$$f(\langle x_1, \ldots, x_n \rangle) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})}$$

$$f(\langle x_0, \ldots, x_{n-1} \rangle) = \sum_{j=0}^{n-1} \left| \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \right|$$

…then we can find it.

Completeness

Completness means that we can eiher find a ranking function or prove that none exists.

If there is ANY ranking function…

$$f(\langle x_1, \ldots, x_n \rangle) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})}$$

$$f(\langle x_0, \ldots, x_{n-1} \rangle) = \sum_{j=0}^{n-1} \left| \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \right|$$

…~~then we can find~~ it.
…then there is a ranking function of the certain kind.

Completeness Results

$$\text{PROGRAM} \quad \Rightarrow \left\{ \begin{array}{c} \vdots \\ p(\bar{x}) \leftarrow \mu,\ p(\bar{y}) \\ \vdots \end{array} \right.$$

Size-Change Graphs	Monotonicity Constraints	Linear Constraints
$x_i > y_j$, $x_i \geq y_j$,	$v > w$, $v \geq w$,	$\sum c_i v_i > c$
$x_i \in \bar{x}$, $y_j \in \bar{y}$	$v, w \in (\bar{x} \cup \bar{y})$	$v_i \in (\bar{x} \cup \bar{y})$
Completeness: $\surd$	Completeness: $\surd\surd$	Completeness: $\surd$ (wrt. linear f)

Completeness for Size-Change Graphs

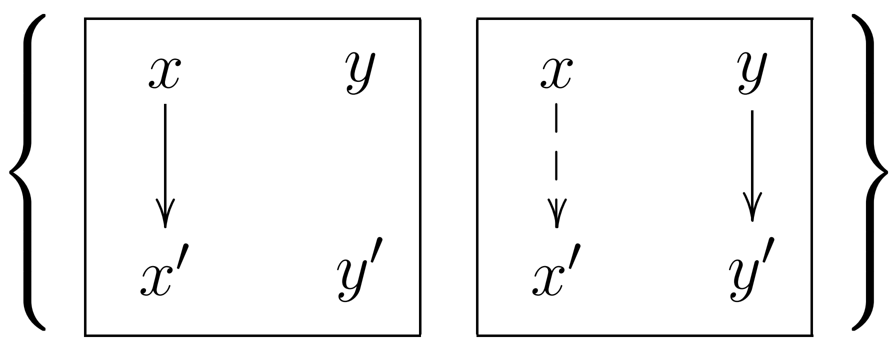
For an idempotent SCG if there is any ranking function f then there is one of the form $f(\bar{x}) = x_i$. (Lee *et al*, 2001)

The algorithm:

1. Compute the closure G^*
2. Compute the subset of idempotent graphs $I \subseteq G^*$
3. For each $\mu(\bar{x}, \bar{y}) \in I$ check that $\exists i. \mu(\bar{x}, \bar{y}) \rightarrow (x_i > y_i)$

Example: $Ack(x, y)$ is terminating

$I = G^* =$



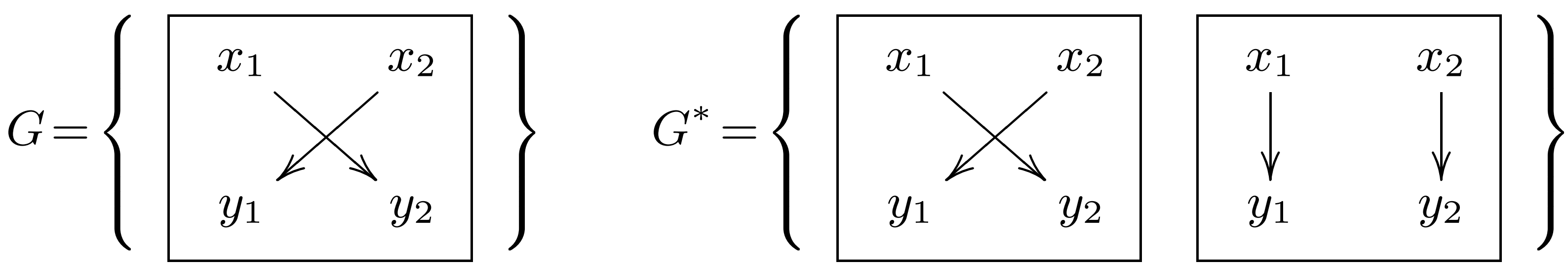
Completeness for Size-Change Graphs

For any SCG if there is a ranking function f then there is one of the form $f(\bar{x}) = \sum_{i \in I} x_i$. (this work)

The algorithm:

1. Compute the closure G^*
2. For each $\mu(\bar{x}, \bar{y}) \in G^*$ check that $\mu(\bar{x}, \bar{y}) \models \bigvee_i (x_i > y_i)$

Example:



$$\mu(\bar{x}, \bar{y}) \models \bigvee_i (x_i > y_i)$$

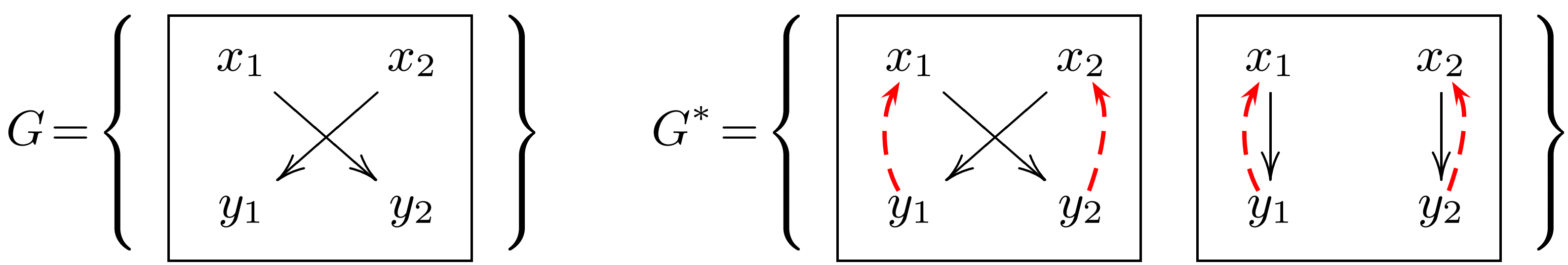
Completeness for Size-Change Graphs

For any SCG if there is a ranking function f then there is one of the form $f(\bar{x}) = \sum_{i \in I} x_i$. (this work)

The algorithm:

1. Compute the closure G^*
2. For each $\mu(\bar{x}, \bar{y}) \in G^*$ check that $\mu(\bar{x}, \bar{y}) \models \bigvee_i (x_i > y_i)$

Example:



$$\mu(\bar{x}, \bar{y}) \models \bigvee_i (x_i > y_i) \;\equiv\; \neg\Big(\mu(\bar{x}, \bar{y}) \wedge \bigwedge_i (x_i \leq y_i)\Big)$$

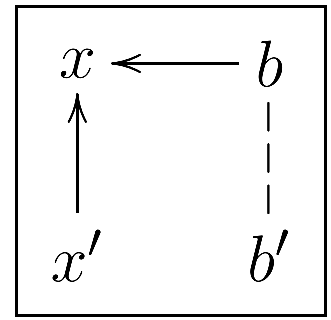
Test for Ranking Function

$\bigvee_i \mu(\bar{x}, \bar{y}) \models (x_i > y_i)$	idempotent graphs	(Lee *et al*)
$\mu(\bar{x}, \bar{y}) \models \bigvee_i (x_i > y_i)$	all graphs	(this work)

Monotonicity Constraints

We want to analyze programs like: **while** `x<b` **do** `x=x+1`

$$\mu(\langle x, b\rangle, \left\langle x', b'\right\rangle) = (x < x', x < b, b = b')$$



Neither $f(x, b) = x$ nor $f(x, b) = b$ is a ranking function.
The ranking function is $f(x, b) = b - x$.

Completeness for MC's

Theorem: for an idempotent monotonicity constraint $\mu(\bar{x}, \bar{y})$ if there is a ranking function then there is a ranking function of the form $f(\bar{x}) = x_i$ or of the form $f(\bar{x}) = x_i - x_j$

Theorem: for a monotonicity constraint $\mu(\bar{x}, \bar{y})$ if there is a ranking function then there is a linear ranking function.

Unfortunately, finding the ranking function is not as easy as for SCGs. We can

- check for idempotence and examine $x_i - x_j$ for all pairs
- apply the method of Podelski and Rybalchenko (based on linear programming)

Using Completeness of Podelski *et al*

- If there is a linear ranking function for a linear constraint then the method of Podelski & Rybalchenko can find it.
- Monotonicity constraints are special case of linear constraints.
- We show that for termination of monotonicity constraints there must exist a linear ranking function.
- So, we can apply the method of Podelski & Rybalchenko, and it’s complete.

Conclusion

We've shown that:

- The TermiLog/TerminWeb test for ranking function is complete for size-change graphs.
- The TermiLog/TerminWeb test for ranking function is incomplete for monotonicity constraints.
- (Size-change graphs and monotonicity constraints are not the same.)
- Termination analysis is complete for monotonicity constraints.
- There is no loss of precision when not checking for idempotence.