Verification of Relational Programs and Approximation Algorithms

Peter Höfner

14th Logic and Computation Seminar (November 12, 2015)

CSIRO's Digital Productivity business unit and NICTA have joined forces to create digital powerhouse Data61

Motivation

towards more automation in program verification

- functional correctness
- use algebra to improve proof automation
- use pre-/postconditions (Hoare-style reasoning)

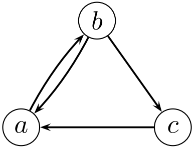
at the moment

- look at ‘simple’ and well-known while programs (pre-/postconditions, invariant proofs)
- use relational algebra
- limited to algorithms where data structure can be modelled by (relational) algebra
- investigate the power of cardinalities over relational algebra

Relation Algebra: The Standard Model

the standard model are relations (sets over $M \times M$)

$$\{(a,b),(b,a),(b,c),(c,a)\}$$



$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

$n \times n$ matrices

operations/constans:

- $\cup, \cap, ^{-}$ are set theoretic definitions
- $R;S = \{(a,c) \mid \exists b : (a,b) \in R \wedge (b,c) \in S\}$
- $R^{\mathsf{T}} = \{(b,a) \mid (a,b) \in R\}$
- $\mathsf{O} = \emptyset$, $\mathsf{L} = M \times M$, $\mathsf{I} = \{(a,a) \mid a \in M\}$

Relation Algebra

a *relation algebra* is a structure $(A, \cup, ;, \overline{}, ^{\mathsf{T}}, \mathsf{I})$ such that

- $(A, \cup, \overline{})$ is Boolean algebra, i.e.,
  $(Q \cup R) \cup S = Q \cup (R \cup S), \quad Q \cup R = R \cup Q,$
  $R = \overline{\overline{R} \cup \overline{S}} \cup \overline{\overline{R} \cup S}$
- provides an operation for *composition*
  $(Q;R);S = Q;(R;S), \quad (Q \cup R);S = Q;S \cup R;S \quad R;\mathsf{I} = R$
- defines an operation of *conversion*
  $R^{\mathsf{T}^{\mathsf{T}}} = R, \quad (R \cup S)^{\mathsf{T}} = R^{\mathsf{T}} \cup S^{\mathsf{T}}, \quad R^{\mathsf{T}};\overline{R;S} \cup \overline{S} = \overline{S}$

Relation Algebra

a *relation algebra* is a structure $(A, \cup, ;, \overline{}, ^{\mathsf{T}}, \mathsf{I})$ such that

- $(A, \cup, \overline{})$ is Boolean algebra, i.e.,
  $(Q \cup R) \cup S = Q \cup (R \cup S), \quad Q \cup R = R \cup Q,$
  $R = \overline{\overline{R} \cup \overline{S}} \cup \overline{\overline{R} \cup S}$
- provides an operation for *composition*
  $(Q;R);S = Q;(R;S), \quad (Q \cup R);S = Q;S \cup R;S \quad R;\mathsf{I} = R$
- defines an operation of *conversion*
  $R^{\mathsf{T}^{\mathsf{T}}} = R, \quad (R \cup S)^{\mathsf{T}} = R^{\mathsf{T}} \cup S^{\mathsf{T}}, \quad R^{\mathsf{T}};\overline{R;S} \cup \overline{S} = \overline{S}$

additional constants/operations:

- intersection: $R \cap S = \overline{\overline{R} \cup \overline{S}}$
- order: $R \subseteq S \Leftrightarrow R \cup S = S$
- $R^* = \bigcup_{i \geq 0} R^i = I \cup R \cup R^2 \cup \ldots$
  (first-order characterisation possible)
- smallest and greatest element: $\mathsf{O} = R \cap \overline{R}$, $\mathsf{L} = R \cup \overline{R}$

Warming Up: Reflexive-Transitive Closure

input R

$C, v := \mathsf{I}, \mathsf{O}$

while $v \neq R;\mathsf{L}$ **do**

  let $p = \mathit{point}(R;\mathsf{L} \cap \overline{v})$;

  $C, v := C \cup C;p;p^{\mathsf{T}};R;C,\ \ v \cup p$;

od

return C

Warming Up: Reflexive-Transitive Closure

input R

$C, v := \mathsf{I}, \mathsf{O}$

while $v \neq R;\mathsf{L}$ **do**

  let $p = point(R;\mathsf{L} \cap \overline{v});$ //function choosing point from $R;\mathsf{L} \cap \overline{v}$

  $C, v := C \cup C;p;p^{\mathsf{T}};R;C,\ \ v \cup p;$

od

return C

Warming Up: Reflexive-Transitive Closure

input R

$C, v := \mathsf{I}, \mathsf{O}$

while $v \neq R;\mathsf{L}$ **do**

  let $p = point(R;\mathsf{L} \cap \overline{v});$ //function choosing point from $R;\mathsf{L} \cap \overline{v}$

  $C, v := C \cup C;p;p^{\mathsf{T}};R;C,\ \ v \cup p;$

od

return C

$\{C = R^*\}$

Warming Up: Reflexive-Transitive Closure

input R

$\{True\}$

$C, v := \mathsf{I}, \mathsf{O}$

while $v \neq R;\mathsf{L}$ **do**

  let $p = point(R;\mathsf{L} \cap \overline{v});$ //function choosing point from $R;\mathsf{L} \cap \overline{v}$

  $C, v := C \cup C;p;p^{\mathsf{T}};R;C,\ \ v \cup p;$

od

return C

$\{C = R^*\}$

Warming Up: Reflexive-Transitive Closure

input R

$\{True\}$

$C, v := \mathsf{I}, \mathsf{O}$

$\{C = (R \cap v)^* \ \wedge \ v = v;\mathsf{L}\}$

while $v \neq R;\mathsf{L}$ **do**

  let $p = point(R;\mathsf{L} \cap \overline{v})$; //function choosing point from $R;\mathsf{L} \cap \overline{v}$

  $C, v := C \cup C;p;p^{\mathsf{T}};R;C, \ v \cup p$;

od

return C

$\{C = R^*\}$

Warming Up: Reflexive-Transitive Closure

- correctness proof: simple exercise ?

Warming Up: Reflexive-Transitive Closure

- correctness proof: simple exercise ?
- $Inv_0(R, C, v) \Leftrightarrow C = (R \cap v)^*$
  $Inv_1(v) \Leftrightarrow v = v;\mathsf{L}$

Warming Up: Reflexive-Transitive Closure

- correctness proof: simple exercise ?
- $Inv_0(R, C, v) \Leftrightarrow C = (R \cap v)^*$
  $Inv_1(v) \Leftrightarrow v = v;\mathsf{L}$
- p is point $\Leftrightarrow p;\mathsf{L} = p \wedge \mathsf{L};p = \mathsf{L} \wedge p;p \subseteq \mathsf{I}$

Warming Up: Reflexive-Transitive Closure

- correctness proof: simple exercise ?
- $Inv_0(R, C, v) \Leftrightarrow C = (R \cap v)^*$
  $Inv_1(v) \Leftrightarrow v = v;\mathsf{L}$
- p is point $\Leftrightarrow p;\mathsf{L} = p \wedge \mathsf{L};p = \mathsf{L} \wedge p;p \subseteq \mathsf{I}$
- proof automation
  (e.g. the automated theorem prover Prover9)

Establishment	
$Inv_0(R, \mathsf{I}, \mathsf{O}) \wedge Inv_1(\mathsf{O})$	$0\,s$
Post-Condition	
$v = R;\mathsf{L} \wedge Inv_0(R, C, v) \wedge Inv_1(v) \Rightarrow C = R^*$	$0\,s$
Maintenance	
$Inv_1(v) \wedge p \text{ is point} \wedge p \subseteq R;\mathsf{L} \cap \overline{v} \;\Rightarrow\; Inv_1(v \cup p)$	$1\,s$
$Inv_0(R, C, v) \wedge p \text{ is point} \wedge p \subseteq R;\mathsf{L} \cap \overline{v}$ $\Rightarrow\; Inv_0(R, C \cup C;p;p^{\mathsf{T}};R;C, v \cup p)$	–

Warming Up: Reflexive-Transitive Closure

- correctness proof: simple exercise ?
- $Inv_0(R, C, v) \Leftrightarrow C = (R \cap v)^*$
  $Inv_1(v) \Leftrightarrow v = v;\mathsf{L}$
- p is point $\Leftrightarrow p;\mathsf{L} = p \wedge \mathsf{L};p = \mathsf{L} \wedge p;p \subseteq \mathsf{I}$
- proof automation
  (e.g. the automated theorem prover Prover9)

Establishment	
$Inv_0(R, \mathsf{I}, \mathsf{O}) \wedge Inv_1(\mathsf{O})$	$0\,s$
Post-Condition	
$v = R;\mathsf{L} \wedge Inv_0(R, C, v) \wedge Inv_1(v) \Rightarrow C = R^*$	$0\,s$
Maintenance	
$Inv_1(v) \wedge p$ is point $\wedge\ p \subseteq R;\mathsf{L} \cap \overline{v} \;\Rightarrow\; Inv_1(v \cup p)$	$1\,s$
$Inv_0(R, C, v) \wedge p$ is point $\wedge\ p \subseteq R;\mathsf{L} \cap \overline{v}$ $\Rightarrow\; Inv_0(R, C \cup C;p;p^{\mathsf{T}};R;C, v \cup p)$	$0\,s$

add 3 theorems about the operation $*$

Conclusion

- shows partial correctness only
  total correctness has to be shown separately
- often automated reasoning helps – but not always
- perfect candidate for interactive theorem proving/proof assistants (preferable with some bits of proof automation)

- more algorithms verified
  - topological sorting
  - node colouring
  - matching algorithms
  - ...
- verification of Relational-While Programs can be done (in)equationally and automatically;
  in particular RA seems to be well suited for graph problems

Verification of Relational-While Program

- can be done equationally and automatically

Verification of Relational-While Program

- can be done equationally and automatically
- BUT: what about cardinalities

Cardinalities on Relation Algebra

- introduced by Prof. Kawahara [Kaw06]
- operation $|.|$ over relation algebra
- Prof. Kawahara looked at basic graph theory, such as the theorem of Hall and König
- we used his approach for verification

Cardinalities on Relation Algebra

(C1) if R is finite, then $|R| \in \mathbb{N}$, and
$|R| = 0$ iff $R = \mathsf{O}$

(C2) $|R| = |R^{\mathsf{T}}|$

(C3) if R and S are finite, then
$|R \cup S| = |R| + |S| - |R \cap S|$.

(C4) if Q is univalent ($Q^{\mathsf{T}};Q \subseteq \mathsf{I}$), then
$|R \cap Q^{\mathsf{T}};S| \leq |Q;R \cap S|$, and
$|Q \cap S;R^{\mathsf{T}}| \leq |Q;R \cap S|$

Cardinalities on Relation Algebra

(C1) if R is finite, then $|R| \in \mathbb{N}$, and
$|R| = 0$ iff $R = \mathsf{O}$

(C2) $|R| = |R^{\mathsf{T}}|$

(C3) if R and S are finite, then
$|R \cup S| = |R| + |S| - |R \cap S|$.

(C4) if Q is univalent ($Q^{\mathsf{T}};Q \subseteq \mathsf{I}$), then
$|R \cap Q^{\mathsf{T}};S| \leq |Q;R \cap S|$, and
$|Q \cap S;R^{\mathsf{T}}| \leq |Q;R \cap S|$

(C5) $|\mathsf{I} \quad| =$

Cardinalities on Relation Algebra

(C1) if R is finite, then $|R| \in \mathbb{N}$, and
$|R| = 0$ iff $R = \mathsf{O}$

(C2) $|R| = |R^{\mathsf{T}}|$

(C3) if R and S are finite, then
$|R \cup S| = |R| + |S| - |R \cap S|$.

(C4) if Q is univalent ($Q^{\mathsf{T}};Q \subseteq \mathsf{I}$), then
$|R \cap Q^{\mathsf{T}};S| \leq |Q;R \cap S|$, and
$|Q \cap S;R^{\mathsf{T}}| \leq |Q;R \cap S|$

(C5) $|\mathsf{I}_{11}| = 1$

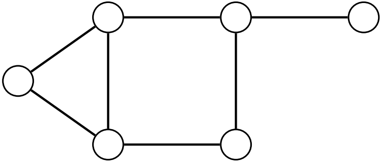
we have to calculate in a heterogenous setting ($m \times n$ matrices)

Relations, Points, Vectors and Cardinalities

useful properties (sanity check)

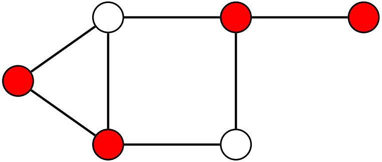
- $|.|$ is monotone, i.e. $R \subseteq S \Rightarrow |R| \subseteq |S|$
- if p is point, then $|p| = 1$
- if v is vector, then $|v| == |\bigcup_{p \in \mathcal{P}_v} p| = \sum_{p \in \mathcal{P}_v} |p|$
- if R is univalent and S is a mapping, then $|R;S| = |R|$
- if R is symmetric, P is injective and Q is univalent, then $|R \cap P;Q^\mathsf{T}| = |R;P \cap Q|$

Minimum Vertex Covers



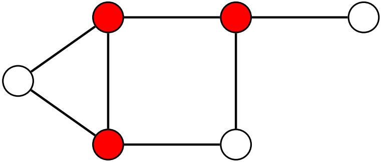
- problem is NP-complete
- approximation algorithm of Garvil and Yannakakis
- cardinalities are used to give quality of approximation

Minimum Vertex Covers



- problem is NP-complete
- approximation algorithm of Garvil and Yannakakis
- cardinalities are used to give quality of approximation

Minimum Vertex Covers



- problem is NP-complete
- approximation algorithm of Garvil and Yannakakis
- cardinalities are used to give quality of approximation

Minimum Vertex Covers

input R

$c, S \quad := \mathsf{O}_{X\mathbf{1}}, R \quad ;$

while $S \neq \mathsf{O}$ **do**

  let $e \; = edge(S)$;

  $c, S \qquad := \; c \cup e;\mathsf{L}, \; S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \qquad ;$

od

return c

Minimum Vertex Covers

input R

$\{R \subseteq \bar{\mathsf{I}},\ R = R^{\mathsf{T}}\}$

$c, S \quad := \mathsf{O}_{X\mathbf{1}}, R \quad ;$

while $S \neq \mathsf{O}$ **do**

  let $e \ = edge(S)$;

  $c, S \quad := \ c \cup e;\mathsf{L},\ \ S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \qquad ;$

od

return c

Minimum Vertex Covers

input R

$\{R \subseteq \bar{\mathsf{I}},\ R = R^{\mathsf{T}}\}$

$c, S \quad := \mathsf{O}_{X\mathbf{1}}, R \quad ;$

while $S \neq \mathsf{O}$ **do**

  let $e \ = edge(S)$;

  $c, S \quad := \ c \cup e;\mathsf{L},\ \ S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \qquad ;$

od

return c

$\{R \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}},$
$\ \forall d : X \leftrightarrow \mathbf{1} \bullet R \subseteq d;\mathsf{L} \cup (d;\mathsf{L})^{\mathsf{T}} \Rightarrow |c| \leq 2 \cdot |d|\}$

Minimum Vertex Covers

input R

$\{R \subseteq \bar{\mathsf{I}},\ R = R^{\mathsf{T}}\}$

$c, S, M := \mathsf{O}_{X\mathbf{1}}, R, \mathsf{O};$

while $S \neq \mathsf{O}$ **do**

  let $e\ = edge(S);$

  $c, S, M\ :=\ c \cup e;\mathsf{L},\ S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e},\ M \cup e;$

od

return c

$\{R \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}},$
$\ \forall d : X \leftrightarrow \mathbf{1} \bullet R \subseteq d;\mathsf{L} \cup (d;\mathsf{L})^{\mathsf{T}} \Rightarrow |c| \leq 2 \cdot |d|\}$

Minimum Vertex Covers

input R

$\{R \subseteq \bar{\mathsf{I}},\ R = R^{\mathsf{T}}\}$

$c, S, M := \mathsf{O}_{X\mathbf{1}}, R, \mathsf{O};$

$\{M \subseteq R,\ M = M^{\mathsf{T}},\ M;M \subseteq \mathsf{I},\ M;\mathsf{L} \cap S = \mathsf{O}$ $\quad M$ aux. variable
$R \cap \overline{S} \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}},\ S \subseteq R,\ S = S^{\mathsf{T}},\ |c| \leq |M|\}$

while $S \neq \mathsf{O}$ **do**

let $e = edge(S);$

$c, S, M := c \cup e;\mathsf{L},\ S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e},\ M \cup e;$

od

return c

$\{R \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}},$
$\forall d : X \leftrightarrow \mathbf{1} \bullet R \subseteq d;\mathsf{L} \cup (d;\mathsf{L})^{\mathsf{T}} \Rightarrow |c| \leq 2 \cdot |d|\}$

Correctness Proof

- most invariant proofs are equational and "easy";
  they are verified using the proof assistant Coq
  (there are just more and invariants)
- short proofs
- proof automation would be useful
  (e.g. try Isabelle's tool sledgehammer)

Correctness Proof: Cardinalities

Invariant

$$
\begin{aligned}
|c \cup e;\mathsf{L}| &\leq |c| + |e;\mathsf{L}| - |c \cap e;\mathsf{L}| && \text{// by (C3)} \\
&= |c| + |e;\mathsf{L}| && \text{// isotonicity} \\
&\leq |M| + |e;\mathsf{L}| && \text{// invariant} \\
&= |M| + |e| && \text{// } e \text{ vector} \\
&= |M| + |e| - |M \cap e| && \text{// as } M \cap e = \mathsf{O}\text{, by (C1)} \\
&= |M \cup e| && \text{// by (C3)}
\end{aligned}
$$

Correctness Proof: Cardinalities

Postcondition

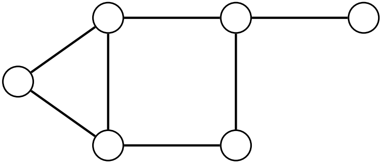
$$
\begin{aligned}
|M| &= |M;(d \cup \overline{d})| && \text{// } M \text{ univalent, } d \cup \overline{d} \text{ mapping, aux. Lemma} \\
&= |M;d \cup M;\overline{d}| \\
&\leq |M;d| + |M;\overline{d}| && \text{// by (C3), isotony} \\
&\leq |M;d| + |R;\overline{d}| && \text{// invariant, isotonicity} \\
&\leq |M;d| + |d| && \text{// as } R;\overline{d} \subseteq d\text{, isotonicity} \\
&= |\mathsf{L} \cap M^{\mathsf{T}\mathsf{T}};d| + |d| \\
&\leq |M^{\mathsf{T}};\mathsf{L} \cap d| + |d| && \text{// } M^{\mathsf{T}} \text{ univalent, by (C4)} \\
&\leq |d| + |d| && \text{// isotonicity} \\
&= 2 \cdot |d|
\end{aligned}
$$

Adaptation to Hitting Sets

- same algorithm, different relation algebra (calculating on incidence relation $I : X \leftrightarrow E$)
- this models hypergraphs (edges are set of nodes)
- cardinality of all maximal hyperedges: $\max\{|I;p| \mid p : E \leftrightarrow \mathbf{1} \text{ point}\}$
- algorithm generalises to hyper graph with approximation

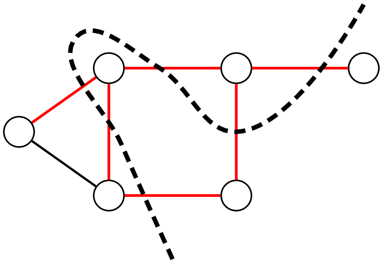
$$\forall d : X \leftrightarrow \mathbf{1} \bullet \mathsf{L} = I^{\mathsf{T}};d \Rightarrow |c| \leq k \cdot |d|$$

Maximum Cuts (Max-Cut)



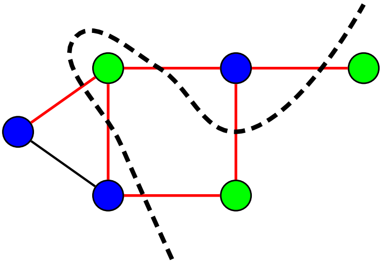
- problem is NP-complete
- approximation algorithm
- cardinalities are used to give quality of approximation
- cardinality of cut: $|R \cap (s;\overline{s}^{\mathsf{T}} \cup \overline{s};s^{\mathsf{T}})|$

Maximum Cuts (Max-Cut)



- problem is NP-complete
- approximation algorithm
- cardinalities are used to give quality of approximation
- cardinality of cut: $|R \cap (s;\overline{s}^{\mathsf{T}} \cup \overline{s};s^{\mathsf{T}})|$

Maximum Cuts (Max-Cut)



- problem is NP-complete
- approximation algorithm
- cardinalities are used to give quality of approximation
- cardinality of cut: $|R \cap (s;\overline{s}^{\mathsf{T}} \cup \overline{s};s^{\mathsf{T}})|$

Minimum Vertex Covers

input R

$v, s, t := \mathsf{L}_{X\mathbf{1}}, \mathsf{O}, \mathsf{O};$

while $v \neq \mathsf{O}$ **do**

    let $p \ = point(v);$

    if $|R;p \cap s| < |R;p \cap t|$

        then $v, s := v \cap \overline{p}, s \cup p$

        else $v, t := v \cap \overline{p}, t \cup p$

    fi

od

return s

Minimum Vertex Covers

input R

$\{R \subseteq \bar{\mathsf{I}},\ R = R^{\mathsf{T}}\}$

$v, s, t := \mathsf{L}_{X\mathbf{1}}, \mathsf{O}, \mathsf{O};$

$\{s \cap t = \mathsf{O},\ s \cup t = \overline{v},\ |R \cap (s;s^{\mathsf{T}} \cup t;t^{\mathsf{T}})| \leq |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})|\}$

while $v \neq \mathsf{O}$ **do**

  let $p = point(v);$

  if $|R;p \cap s| < |R;p \cap t|$

    then $v, s := v \cap \overline{p}, s \cup p$

    else $v, t := v \cap \overline{p}, t \cup p$

  fi

od

return s

$\{\forall c : X \leftrightarrow \mathbf{1} \bullet |R \cap (c;\overline{c}^{\mathsf{T}} \cup \overline{c};c^{\mathsf{T}})| \leq 2 \cdot |R \cap (s;\overline{s}^{\mathsf{T}} \cup \overline{s};s^{\mathsf{T}})|\}$

Correctness Proof

- non-cardinality proofs are again standard;
  they are verified using the proof assistant Coq
- approximation bound is $\frac{1}{2}$:
  $\forall c : X \leftrightarrow \mathbf{1} \bullet |R \cap (c;\overline{c}^{\mathsf{T}} \cup \overline{c};c^{\mathsf{T}})| \leq 2 \cdot |R \cap (s;\overline{s}^{\mathsf{T}} \cup \overline{s};s^{\mathsf{T}})|$

Correctness Proof: Cardinalities

Invariant

$$
\begin{array}{ll}
|R \cap ((s \cup p);(s \cup p)^\top \cup t;t^\top)| & \\
= |(R \cap (s;s^\top \cup t;t^\top \cup s;p^\top \cup p;s^\top \cup p;p^\top)| & \\
= |(R \cap (s;s^\top \cup t;t^\top)) \cup (R \cap s;p^\top) \cup (R \cap p;s^\top)| & \text{// 1st aux. result} \\
\leq |R \cap (s;s^\top \cup t;t^\top)| + |R \cap s;p^\top| + |R \cap p;s^\top| & \text{// by (C3)} \\
\leq |R \cap (s;t^\top \cup t;s^\top)| + |R \cap s;p^\top| + |R \cap p;s^\top| & \text{// invariant} \\
= |R \cap (s;t^\top \cup t;s^\top)| + |R \cap p;s^\top| + |R \cap p;s^\top| & \text{// by (C2), } R = R^\top \\
= |R \cap (s;t^\top \cup t;s^\top)| + |R;p \cap s| + |R;p \cap s| & \text{// 1st aux. lemma} \\
< |R \cap (s;t^\top \cup t;s^\top)| + |R;p \cap t| + |R;p \cap t| & \text{// as } |R;p \cap s| < |R;p \cap t| \\
= |R \cap (s;t^\top \cup t;s^\top)| + |R \cap p;t^\top| + |R \cap p;t^\top| & \text{// aux. lemma} \\
= |R \cap (s;t^\top \cup t;s^\top)| + |R \cap p;t^\top| + |R \cap t;p^\top| & \text{// by (C2), } R = R^\top \\
= |R \cap (s;t^\top \cup t;s^\top)| + |(R \cap p;t^\top) \cup (R \cap t;p^\top)| & \text{// 2nd auxiliary result} \\
= |(R \cap (s;t^\top \cup t;s^\top)) \cup (R \cap p;t^\top) \cup (R \cap t;p^\top)| & \text{// 3rd auxiliary result} \\
= |R \cap ((s \cup p);t^\top \cup t;(s \cup p)^\top)| &
\end{array}
$$

Correctness Proof: Cardinalities

Invariant

$$
\begin{aligned}
& |R \cap ((s \cup p);(s \cup p)^{\mathsf{T}} \cup t;t^{\mathsf{T}})| \\
&= |(R \cap (s;s^{\mathsf{T}} \cup t;t^{\mathsf{T}} \cup s;p^{\mathsf{T}} \cup p;s^{\mathsf{T}} \cup p;p^{\mathsf{T}})| \\
&= |(R \cap (s;s^{\mathsf{T}} \cup t;t^{\mathsf{T}})) \cup (R \cap s;p^{\mathsf{T}}) \cup (R \cap p;s^{\mathsf{T}})| && \text{// 1st aux. result} \\
&\leq |R \cap (s;s^{\mathsf{T}} \cup t;t^{\mathsf{T}})| + |R \cap s;p^{\mathsf{T}}| + |R \cap p;s^{\mathsf{T}}| && \text{// by (C3)} \\
&\leq |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |R \cap s;p^{\mathsf{T}}| + |R \cap p;s^{\mathsf{T}}| && \text{// invariant} \\
&= |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |R \cap p;s^{\mathsf{T}}| + |R \cap p;s^{\mathsf{T}}| && \text{// by (C2), } R = R^{\mathsf{T}} \\
&= |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |R;p \cap s| + |R;p \cap s| && \text{// 1st aux. lemma} \\
&< |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |R;p \cap t| + |R;p \cap t| && \text{// as } |R;p \cap s| < |R;p \cap t| \\
&= |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |R \cap p;t^{\mathsf{T}}| + |R \cap p;t^{\mathsf{T}}| && \text{// aux. lemma} \\
&= |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |R \cap p;t^{\mathsf{T}}| + |R \cap t;p^{\mathsf{T}}| && \text{// by (C2), } R = R^{\mathsf{T}} \\
&= |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |(R \cap p;t^{\mathsf{T}}) \cup (R \cap t;p^{\mathsf{T}})| && \text{// 2nd auxiliary result} \\
&= |(R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})) \cup (R \cap p;t^{\mathsf{T}}) \cup (R \cap t;p^{\mathsf{T}})| && \text{// 3rd auxiliary result} \\
&= |R \cap ((s \cup p);t^{\mathsf{T}} \cup t;(s \cup p)^{\mathsf{T}})|
\end{aligned}
$$

- not nice, but still (in)equational reasoning
  hence proof assistants can easily be used
- verification of postcondition is similar (but shorter)

Conclusion and Remarks

- verification of graph algorithms using cardinalities
- made use of point axiom $\mathsf{L}_{X\mathbf{1}} = \bigcup_{p \in \mathcal{P}_{\mathsf{L}_{X\mathbf{1}}}} p$
- made use of RelView (Berghammer et. al) to check proof invariants

Proof support and Proof Automation

- automated theorem provers (ATPs)
  - Prover9 (best for algebraic reasoning): no types
  - other have types, but difficult to encode heterogeneous RA
  - no (proper) support of intermediate lemmas
- Isabelle/HOL
  - excellent library for homogeneous RA [Str14]
  - no (proper) library for heterogeneous RA (Guttmann)
  - good connection to ATPs (via the Sledgehammer tool) allows proof automation
- Coq
  - good support for types
  - excellent library for (homogenous and heterogenous) RA [Pou]
  - lots of tactics available (decision procedures, normalisation ...)
  - cardinalities have been implemented (Stucke)
  - however, no tool such as sledgehammer

DATA 61

Thank you

Peter Höfner
Software and Computational Systems
senior researcher

t +61 2 8306 0561
e peter.hoefner@nicta.com.au
w www.csiro.au/data61

CSIRO

Literature

[Str14] Armstrong, A., Foster, S., Struth, G., Weber, T.: Relation algebra. Archive of Formal Proofs, 2014.

[BHS14] Berghammer, R., Höfner, P., Stucke, I.: Automated verification of relational while-programs. In: Höfner et. al. Relational and Algebraic Methods in Computer Science. LNCS 8248, 309-326. Springer (2014)

[BHS16] Berghammer, R., Höfner, P., Stucke, I.: Cardinality of Relations and Relational Approximation Algorithms to appear in JLAMP, Elsevier (2016)

[BS10] Berghammer, R., Struth, G.: On automated program construction and verification. In: Bolduc, C., Desharnais, J., Ktari, B. Mathematics of Program Construction. LNCS 6120, 22-41. Springer (2010)

[HS08] Höfner, P., Struth, G.: On automating the calculus of relations. In: Armando, A., Baumgartner, P., Dowek, G. Automated Reasoning. LNAI 5195, 50-66. Springer (2008)

[Kaw06] Kawahara, Y.: On the cardinality of relations. In: Schmidt, R.A. (ed.): Relations and Kleene Algebra in Computer Science. LNCS 4136, 251-265. Springer (2006)

[Pou] Pous, D.: Relation algebra and KAT in Coq. `http://perso.ens-lyon.fr/damien.pous/ra/`