Primal Infon Logic with Conjunctions as Sets

Carlos Cotrini[1] Yuri Gurevich[2] Ori Lahav[3]
Artem Melentyev[4]

[1] Swiss Federal Institute of Technology, Switzerland

[2] Microsoft Research, USA

[3] Tel Aviv University, Israel

[4] Ural Federal University, Russia

TCS, September 3, 2014

Outline

- Motivation
- (Full) Infon Logic
- Primal Infon Logic (PIL)
- PIL extensions
- PIL with Conjunctions as Sets (SPIL)
- SPIL algorithm

Motivation: DKAL

- Yuri Gurevich and Itay Neeman.
  DKAL: Distributed Knowledge Authorization Language. 2008
- The world of DKAL consists of communicating principals computing their own knowledge in their own states
- They communicate infons, items of information, and reason in terms of infons
- Expressive but efficient logic is wanted

Propositional Intuitionistic (Constructive) Logic

- Richard Statman, *Intuitionistic Propositional Logic is Polynomial-Space Complete*. 1979
- The derivability problem (decide whether a given formula follows from given hypotheses) is PSPACE-complete.

(Full) Propositional Infon Logic

- Yuri Gurevich and Itay Neeman.
  Logic of infons: The propositional case. 2009.
- View infons as statements. Recall that infons are items of information (rather than representations of truth values)
- This logic is a conservative extension of propositional intuitionistic logic (with conjunction and implication only) by means of quotation modalities.
  - Alice said ((Bob said x) $\to$ x)
- The logic is applicable for policy and trust management
- The derivability problem is still PSPACE-complete

Language of Propositional Infon Logic

- Vocabulary:
  Principals (Alice, Bob)
  Propositional variables (x,y)
  Constant $\top$, known to all principals
- Connectives:
  Conjunction $\wedge$
  Implication $\rightarrow$
  Quotation $p : x$ (p said x)
- Example: $(Bob : x) \rightarrow x$
  (Bob is trusted on saying x)

Inference rules of Propositional Infon Logic

The elimination and introduction of $\wedge$ and $\to$ are those of standard intuitionistic logic.
In addition, there is a rule of quotation introduction:

$$\text{(said)}\ \frac{\Gamma \vdash y}{(q\colon\Gamma) \vdash q\colon y}$$

Propositional **Primal** Infon Logic (PIL)

- Yuri Gurevich and Itay Neeman.
  Logic of infons: The propositional case. 2009.
- Fragment of (full) propositional infon logic obtained by weakening the implication-introduction rule

$$(\to i)\ \frac{\Gamma, x \vdash y}{\Gamma \vdash x \to y}$$

  to

$$(\to i_w)\ \frac{\Gamma \vdash y}{\Gamma \vdash x \to y}$$

- While this logic looks weak, it is still very much applicable, and in fact it was suggested by practice
- The multi-derivability problem (decide which of given queries follow from given hypotheses) is **linear time**

PIL with disjunction

- PIL with disjunction is NP-complete:
  Lev Beklemishev and Yuri Gurevich. *Propositional Primal Logic with Disjunction*, 2012.

$$(\vee\mathrm{i})\ \frac{\Gamma \vdash x}{\Gamma \vdash x \vee y}$$

$$(\vee\mathrm{e})\ \frac{\Gamma, x \vdash z \qquad \Gamma, y \vdash z \qquad \Gamma \vdash x \vee y}{\Gamma \vdash z}$$

- PIL with (still useful) primal disjunction (with disjunction introduction ($\vee$I) but no elimination ($\vee$E)) is **linear time**:
  C. Cotrini and Y. Gurevich. *Basic primal infon logic*. 2013.

Transitive PIL (TPIL)

- Carlos Cotrini and Yuri Gurevich.
  Transitive Primal Infon Logic. 2012
- Add transitivity of implication:

$$\text{(trans*)}\ \frac{\vec{q}\colon (A_1 \to A_2) \quad \vec{q}\colon (A_2 \to A_3) \quad \dots \quad \vec{q}\colon (A_{k-1} \to A_k)}{\vec{q}\colon (A_1 \to A_k)}$$

- Multi-derivability problem is $O(n^2)$.

PIL with variables and universal quantifiers: Reduction to Datalog

- Andreas Blass and Yuri Gurevich.
  Hilbertian Deductive Systems, Infon Logic, and Datalog. 2010
- Nikolaj Bjorner, Guido de Caso and Yuri Gurevich.
  From Primal Infon Logic with Individual Variables to Datalog. 2011

Limitations of PIL

- The principle of equivalent formula substitution fails (replacing a subformula with an equivalent one should not affect the derivability of the formula)
- for example $x \wedge y$ is equivalent to $y \wedge x$, but
  $(x \wedge y) \rightarrow z$ does not entail $(y \wedge x) \rightarrow z$
  Similary $w \rightarrow ((x \wedge y) \wedge z)$ does not entail $w \rightarrow (x \wedge (y \wedge z))$
- Imposing the full principle leads to an NP-hardness:
  Lev Beklemishev and Igor Prokhorov.
  On computationally efficient subsystems of propositional logic.
  To appear.

Motivation for PIL with Conjunctions as Sets

- Increase the expressibility (and therefore applicability) of PIL as much as possible
- Have to be very careful about the use of conjunction because the order of conjuncts matters in PIL

$$x \wedge y \vdash y \wedge x$$

$$(x \wedge y) \to z \nvdash (y \wedge x) \to z$$

PIL with Conjunctions as Sets (SPIL)

Definition

$x \sim y$ if x and y are the same formulas modulo the properties of $\wedge$: commutativity, associativity, idempotence, contraction of the identity element $\top$, distributivity of quotations over $\wedge$:

$$
\begin{array}{ll}
(x_1 \wedge x_2) \sim (x_2 \wedge x_1) & (x_1 \wedge \top) \sim x_1 \\
((x_1 \wedge x_2) \wedge x_3) \sim (x_1 \wedge (x_2 \wedge x_3)) & q{:}(x_1 \wedge x_2) \sim (q{:}x_1) \wedge (q{:}x_2) \\
(x_1 \wedge x_1) \sim x_1 & q{:}\top \sim \top
\end{array}
$$

- Principle of equivalent formula substitution holds for equivalence relation $\sim$
- Reasoning exploits this equivalence

SPIL calculus

- Abstract formulas — equivalence classes w.r.t. $\sim$
- Hilbertian calculus for **SPIL**:

$$(\tilde{\top})\ \frac{}{[\top]} \qquad (\tilde{\wedge}\mathrm{i})\ \frac{X_1 \quad X_2 \quad \dots \quad X_n}{\bigwedge S} \text{ where } S = \{X_1, \dots, X_n\} \text{ and } n \geq 2$$

$$(\tilde{\wedge}\mathrm{e})\ \frac{\bigwedge S}{X} \text{ where } X \in S \qquad (\tilde{\vee}\mathrm{i})\ \frac{\vec{q}\text{:}\, X}{\vec{q}\text{:}\, (X \vee Y)} \qquad \frac{\vec{q}\text{:}\, Y}{\vec{q}\text{:}\, (X \vee Y)}$$

$$(\tilde{\to}\mathrm{i})\ \frac{\vec{q}\text{:}\, Y}{\vec{q}\text{:}\, (X \to Y)} \qquad (\tilde{\to}\mathrm{e})\ \frac{\vec{q}\text{:}\, X \qquad \vec{q}\text{:}\, (X \to Y)}{\vec{q}\text{:}\, Y}$$

$\vec{q}$: — quotation prefixes, X, Y — abstract formulas, S — conjunction sets, $[x]$ — equivalence class of x.

SPIL extends PIL

Definition

The consequence relation $\vdash$ between concrete formulas in **SPIL** is given by:
$\Gamma \vdash x$ if $\{[y] \mid y \in \Gamma\} \vdash [x]$.

Theorem

If Γ entails x in **PIL**, *then it does so in* **SPIL** *as well.*

So SPIL is conservative extension of PIL

Our Kripke models

Definition

Kripke model is any structure M whose vocabulary comprises of
(i) binary relations S_q where q ranges over the principal constants
(ii) unary relations V_X where X ranges over non-conjunctive formulas.
The elements of (the universe of) M are called *worlds*.

Kripke semantics for SPIL

Definition

Given a Kripke model M, we define when a world $w \vDash X$:

- $X = [\top]$: $w \vDash X$ for every w.
- $X = [v]$ (where v is a propositional variable): $w \vDash X$ if $w \in V_{[v]}$.
- $X = Y \to Z$: $w \vDash X$ if $w \vDash Z$ or ($w \not\vDash Y$ and $w \in V_X$).
- $X = Y \vee Z$: $w \vDash X$ if $w \vDash Z$ or $w \vDash Y$ or $w \in V_X$.
- $X = q{:}Y$ (for non-conjunctive formula Y): $w \vDash X$ if $w' \vDash Y$ for all w' with wS_qw'.
- $X = \bigwedge S$: $w \vDash X$ if $w \vDash Y$ for all $Y \in S$.

SPIL Soundness and Completeness

Theorem (Soundness and Completeness)

Let Γ *be a set of concrete formulas and* x *a concrete formula.* $\Gamma \vdash x$ *if and only if, for every Kripke model and world* w, $w \vDash [x]$ *whenever* $w \vDash \{[y] \mid y \in \Gamma\}$.

Local formulas

- Local formulas:
  - X is local to X
  - If $\vec{q}{:}(Y * Z)$ is local to X (for $* \in \{\to, \vee\}$)
    then $\vec{q}{:}Y$ and $\vec{q}{:}Z$ are local to X
  - If $\bigwedge S$ is local to X
    then every $Y \in S$ is local to X
- Only $O(n)$ local formulas

Theorem

Any shortest derivation in **SPIL** *contains only local formulas.*

Algorithm overview

- Computation model: standard RAM machine with register size $O(\log(n))$, basic register operations are constant time. Function Random generates $\lceil \log(n) \rceil$ random bits in constant time.
- **Linear** on average for all inputs: for every input the expected running time is linear
  (no probability distribution on inputs, only on coin tosses)
- Worst case $O(n^2)$

Algorithm stages

- Parse concrete formulas to parse tree
- Build local prefixes dictionary
- Compress parse tree to directed acyclic multigraph of abstract formulas
- Derive local formulas

Compressing to DAG

- Initialization: Assign random $Hash(u)$ to every node u.
- Iteratively. From leafs to root.
- List C of nodes to process. Children of nodes in C are already processed / compressed in DAG.
- For conjunction set nodes:
  $SL(\wedge S) = Hash(u_1) \oplus \cdots \oplus Hash(u_k)$ where $u_i \in S$, $\oplus$ is XOR
  Use hash table with hash $= SL$ to separate different conjunction sets.
  $Hash$ is random $\Rightarrow$ SL is random $\Rightarrow$ uniform hashing assumption holds $\Rightarrow$ $O(n)$ average.

Implementation

- Open Source code of Infon Logic algorithms available at
  `http://dkal.codeplex.com/`
- Try it online:
  `http://rise4fun.com/dkal/`

Future works

- Deterministic algorithm for **SPIL**. In preparation
- Transitive **SPIL**. In preparation
- More extensions of Primal Infon Logic

Conclusion

- Conservative extension of Primal Infon Logic
- Principle of equivalent formula substitution holds for equivalence induced by conjunction properties:
  commutativity, associativity, idempotence, contraction of the identity element $\top$, distributivity of quotations over $\wedge$
- Randomized algorithm for multi-derivability problem with $O(n)$ complexity on average for all inputs. $O(n^2)$ worst case.

Questions?