Local Reasoning about Data Update

Cristiano Calcagno, Philippa Gardner and Uri Zarfaty

Imperial College London

References

Context Logic and Tree Update, POPL'05

Context Logic as Modal Logic: Completeness and Parametric Inexpressivity, submitted

Local Reasoning about Data Update, journal paper, submitted

Supported by EPSRC, Microsoft and Royal Academy of Engineering.

Reasoning about Data Update

Examples of data update

heap update, information on hard discs, XML update, term rewriting

Hoare Reasoning for Heap Update

Hoare's original work based on First-order Logic

O'Hearn, Reynolds, Yang's work on local reasoning using Separation Logic (SL), an application of Bunched Logic to heaps (O'Hearn, Pym).

Hoare Reasoning for Data Update

Hardly studied for other forms of data update

No unified account

Reasoning about Trees

Reasoning about Static Trees

Ambient Logic (AL) Cardelli, Gordon

Reasoning about web data Cardelli, Gardner, Ghelli

Similar reasoning to Separation Logic

Local Hoare Reasoning for Tree Update

Not possible using Ambient Logic

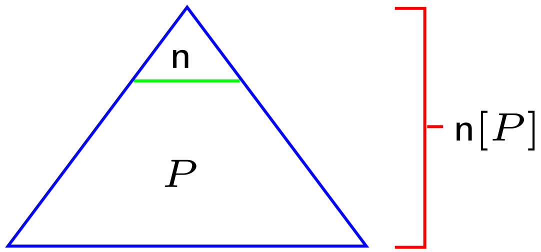
Possible using Context Logic (CL) Calcagno, Gardner, Zarfaty

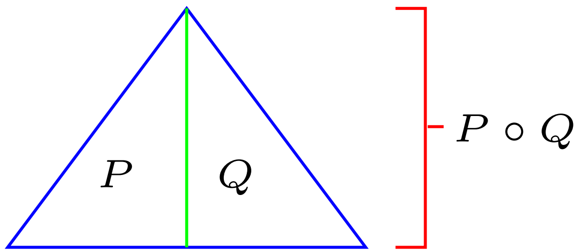
Summary

- Context Logic
- Application to Trees
- Local Reasoning about Tree Update
- Inexpressivity Result for Ambient Logic

Reasoning about Trees

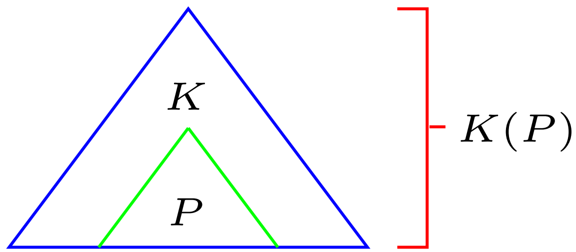
Ambient Logic





P and Q are data formulae

Context Logic for Trees



P is a data formula and K a context formula

CL_0-Formulae

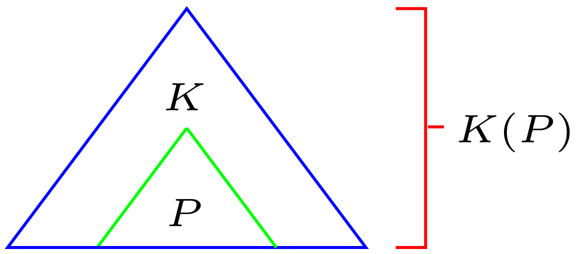
Data Formulae

$P ::=$	0	zero formula
	$K(P)$	application
	$K \triangleleft P$	data application adjoint
	$P \vee P \mid \neg P \mid$ false	boolean additive formulae

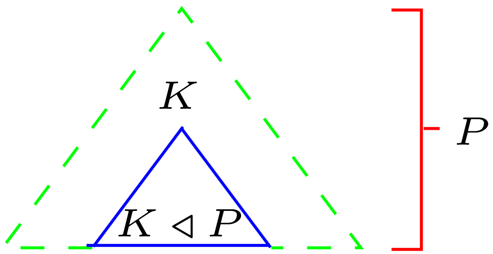
Context Formulae

$K ::=$	I	identity formula
	$P \triangleright P$	context application adjoint
	$K \vee K \mid \neg K \mid$ False	boolean additive formulae

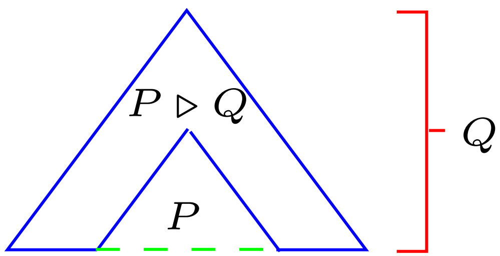
Reasoning about Trees



Application



Data Adjoint



Context Adjoint

CL_0-Models

A CL_0-model $\mathcal{M}$ is a tuple $(\mathcal{D}, \mathcal{C}, \mathsf{ap}, \mathbf{I}, \mathbf{0})$ consisting of

1. data set $\mathcal{D}$ and context set $\mathcal{C}$
2. application $\mathsf{ap} \subseteq (\mathcal{C} \times \mathcal{D}) \times \mathcal{D}$: we write $\mathsf{ap}(\mathsf{c}, \mathsf{d}_1) = \mathsf{d}_2$
3. the left identity $\mathbf{I} \subseteq \mathcal{C}$ to application:
   - $\forall \mathsf{d} \in \mathcal{D}, \exists \mathsf{i} \in \mathbf{I}, \mathsf{d}' \in \mathcal{D}.\ \mathsf{ap}(\mathsf{i}, \mathsf{d}) = \mathsf{d}'$;
   - $\forall \mathsf{d}, \mathsf{d}' \in \mathcal{D}, \forall \mathsf{i} \in \mathbf{I}.\ \mathsf{ap}(\mathsf{i}, \mathsf{d}) = \mathsf{d}'$ implies $\mathsf{d} = \mathsf{d}'$;
4. the projection $\mathsf{p} : \mathcal{C} \to \mathcal{D}$ defined by

$$\mathsf{p}(\mathsf{c}) = \mathsf{d} \Leftrightarrow \exists\, \mathsf{o} \in \mathbf{0}.\ \mathsf{ap}(\mathsf{c}, \mathsf{o}) = \mathsf{d}$$

st. p is a total surjective function and $\forall \mathsf{c}, \mathsf{o}.\ \mathsf{p}(\mathsf{c}) = \mathsf{o} \Rightarrow \mathsf{c} \in \mathbf{I}$.

Example CL_0-Models

- *Mon*$_{\mathcal{D}} = (\mathcal{D}, \mathcal{D}, \cdot, \{e\}, \{e\})$, with (partial) monoid $\cdot$ and unit e.
- *Term*$_\Sigma = (\mathcal{T}_\Sigma, \mathcal{C}_\Sigma, \mathsf{ap}, \{\_\})$, with $\mathcal{T}_\Sigma$ the set of terms, $\mathcal{C}_\Sigma$ the contexts, ap context application and _ the empty context.
- sequences, trees, multisets, heaps
- *Rel*$_{\mathcal{D}} = (\mathcal{D}, \mathcal{P}(\mathcal{D} \times \mathcal{D}), \mathsf{ap}, \{i\})$, with ap relational application and i the identity relation, is a CL-model.
- *Step* $= (\mathbb{N}, \{0, 1\}, +, \{0\})$ is a CL-model.
- $\mathcal{M}_1 + \mathcal{M}_2 = (\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{C}_1 \cup \mathcal{C}_2, \mathsf{ap}_1 \cup \mathsf{ap}_2, I_1 \cup I_2, 0_1 \cup 0_2)$ for CL_0-models $\mathcal{M}_i = (\mathcal{D}_i, \mathcal{C}_i, \mathsf{ap}_i, I_i, 0_i)$, $i = 1, 2$.

CL_0-Satisfaction Relation

For CL_0-model $\mathcal{M} = (\mathcal{D}, \mathcal{C}, \mathsf{ap}, \mathbf{I}, \mathbf{0})$, the CL_0-satisfaction relation $\vDash$ consists of two relations $\mathcal{M}, \mathsf{d} \vDash P$ and $\mathcal{M}, \mathsf{c} \vDash K$ given by:

$\mathcal{M}, \mathsf{d} \vDash 0$ iff $\mathsf{d} \in \mathbf{0}$

$\mathcal{M}, \mathsf{d} \vDash K(P)$ iff $\exists \mathsf{c}, \mathsf{d}'.\ \mathsf{ap}(\mathsf{c}, \mathsf{d}') = \mathsf{d} \wedge \mathcal{M}, \mathsf{c} \vDash K \wedge \mathcal{M}, \mathsf{d}' \vDash P$

$\mathcal{M}, \mathsf{d} \vDash K \lhd P$ iff $\forall \mathsf{c}, \mathsf{d}'.\ \mathcal{M}, \mathsf{c} \vDash K \wedge\ \mathsf{ap}(\mathsf{c}, \mathsf{d}) = \mathsf{d}' \Rightarrow \mathcal{M}, \mathsf{d}' \vDash P$

$\mathcal{M}, \mathsf{c} \vDash I$ iff $\mathsf{c} \in \mathbf{I}$

$\mathcal{M}, \mathsf{c} \vDash P_1 \rhd P_2$ iff $\forall\, \mathsf{d}, \mathsf{d}'.\ \mathcal{M}, \mathsf{d} \vDash P_1 \wedge\ \mathsf{ap}(\mathsf{c}, \mathsf{d}) = \mathsf{d}' \Rightarrow \mathcal{M}, \mathsf{d}' \vDash P_2$

The boolean additive cases are standard.

Derived CL-Data Formulae

Standard derived formulae for the additive connectives.

- $\diamond P \triangleq \mathrm{True}(P)$ somewhere property P holds;

  $\vdash P \Rightarrow \diamond P$ and $\not\vdash \diamond\diamond P \Rightarrow \diamond P$ (holds with context composition)

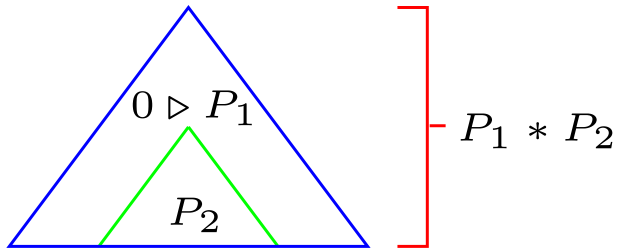
- $K \blacktriangleleft P_2 \triangleq \neg(K \triangleleft \neg P_2)$ there exists a context satisfying property K such that, when the given data element is put in the hole, the resulting data satisfies P_2.

- $P_1 \blacktriangleright P_2 \triangleq \neg(P_1 \triangleright \neg P_2)$ there exists some data element satisfying property P_1 such that, when it is put in the hole of the given context, the resulting data satisfies P_2.

Derived CL_0-Data Formulae

$1 \triangleq \neg 0 \wedge \neg(\neg I)(\neg 0)$ size one

$P_1 * P_2 \triangleq (0 \triangleright P_1)(P_2)$ data can be split into subdata satisfying P_2 and a context satisfying P_1 when a zero is put in the hole.



$P_1 \ast\!\!\leftarrow P_3 \triangleq (0 \triangleright P_1) \triangleleft P_3$ whenever a context applied to a zero satisfies P_1, then the context applied to the given data satisfies P_3.

$P_2 \mathrel{-\!\!*} P_3 \triangleq \neg(\neg(P_2 \triangleright P_3)(0))$ whenever data satisfying P_2 replaces empty subdata of the given data, then result satisfies P_3.

Proof theory and Completeness

Hilbert-style proof theory using $\triangleright$, $\triangleleft$

Modal-logic presentation using $\blacktriangleright$, $\blacktriangleleft$ and specific CL_0-axioms

The CL_0-axioms are well-behaved (very simple Salqvist formulae)

Salqvist's theorem implies completeness

Summary

- Context Logic
- Application to Trees
- Local Reasoning about Tree Update
- Inexpressivity Result for Ambient Logic

Application to Trees

Tree Model

trees $t ::=$	0	empty tree
	$\mathsf{n}[t]$	tree with top node $\mathsf{n} \in N$
	$t \circ t$	horizontal composition

contexts $c ::= \quad \_ \mid \mathsf{n}[c] \mid c \circ t \mid t \circ c, \quad \mathsf{n} \in N$

Equality states that $\circ$ is associative and commutative with unit 0.

We choose the node labels to be unique.

CL_0 for Trees

Specific CL_0-formulae for trees

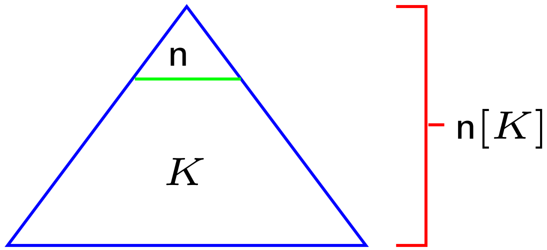
data formulae $P ::= \ldots \mid \mathsf{n}[P] \mid P \circ P, \quad \mathsf{n} \in N$

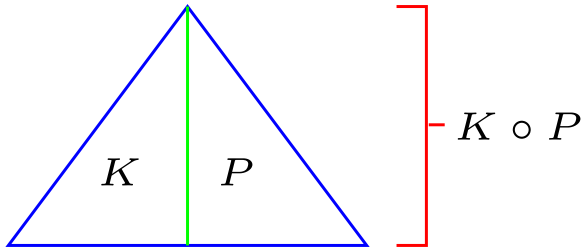
context formulae $K ::= \ldots \mid \mathsf{n}[K] \mid K \circ P \mid P \circ K, \quad \mathsf{n} \in N$

Satisfaction Relation

$\mathit{Tree}_N, \mathsf{c} \models \mathsf{n}[K]$ iff $\exists \mathsf{c}'.\ \mathsf{c} = \mathsf{n}[\mathsf{c}'] \wedge \mathit{Tree}_N, \mathsf{c}' \models K$

$\mathit{Tree}_N, \mathsf{c} \models K \circ P$ iff $\exists \mathsf{c}', \mathsf{d}.\ \mathsf{c} = \mathsf{c}' \circ \mathsf{d} \wedge \mathit{Tree}_N, \mathsf{c}' \models K \wedge \mathit{Tree}_N, \mathsf{d} \models P$





Adjoints $\hat{\mathsf{n}}[P] \triangleq n[I] \triangleleft P$ and $P_1 \multimap P_2 \triangleq (P_1 \circ I) \triangleleft P_2$.

Derived Tree Formulae

- $\mathsf{n}[0]$, the tree $\mathsf{n}[0]$; $\mathsf{n}[\mathrm{true}]$, a tree with root node labelled n
- $\diamond\mathsf{n}[\mathrm{true}]$, a tree containing a node n
- $\mathsf{n}[\mathrm{true}] \circ \mathsf{n}[\mathrm{true}]$, $\mathsf{n}[\mathrm{true}] * \mathsf{n}[\mathrm{true}]$, unsatisfied as n unique
- $\mathsf{m}[\mathrm{true}] \circ \mathsf{n}[\mathrm{true}]$, two trees with top nodes m and n
- $\mathsf{m}[\mathrm{true}] * \mathsf{n}[\mathrm{true}]$, either two trees with top nodes m and n, or one tree with top node m and a subtree with top node n
- $(0 \triangleright P)(\mathsf{n}[\mathrm{true}])$ and $P * \mathsf{n}[\mathrm{true}]$, a tree containing n that satisfies P if the subtree at n is replaced by 0.
- $(\mathsf{m}[\mathrm{true}] \triangleright P)(\mathsf{n}[\mathrm{true}])$, a tree containing n that satisfies P whenever the subtree at n is replaced by a tree with top node m

Summary

- Context Logic
- Application to Trees
- Local Reasoning about Tree Update
- Inexpressivity Result for Ambient Logic

Local Reasoning about Data Update

Update commands tend to operate in a local way, by accessing a small part of the data called the footprint O'Hearn, Reynolds, Yang

Local Hoare reasoning reflects this locality intuition:

small axioms specify the behaviour of commands on their footprints;

the frame rule automatically infers that the rest of the data (the context) remains unchanged.

CL-reasoning is ideally suited to this style of reasoning.

Here we focus on tree update. For our tree commands to be local, the node values must be unique.

Tree Update

Node variables $n, m, \ldots$

Tree variables $x, y, \ldots$

Stores map variables to values, denoted by s

Specific CL_0-formulae

data formulae	$P ::=$	$\ldots \mid x$	x tree variable
		$\ldots \mid \exists n.\, P \mid \exists x.\, P$	quantification
context formulae	$K ::=$	$\ldots \mid n[K] \mid K \circ P$	n node variable
		$\ldots \mid \exists n.\, K \mid \exists x.\, K$	quantification

Satisfaction relation *Tree*$_N$, s, t $\models P$ and *Tree*$_N$, s, c $\models K$

Commands for Tree Update

$\mathbb{C} ::=$	$n := n' \mid x := x'$	variable assignment
	$\mathbb{C}_{\mathsf{up}}(n)$	update at location n
	$\mathbb{C}\ ;\ \mathbb{C}$	sequencing

$\mathbb{C}_{\mathsf{up}}(n) ::=$	$[n]_{\mathsf{T}} := 0$	$[n]_{\mathsf{SF}} := 0$	dispose
	$[n]_{\mathsf{T}} \mathrel{*}= x$	$[n]_{\mathsf{SF}} \mathrel{*}= x$	append
	$x := [n]_{\mathsf{T}}$	$x := [n]_{\mathsf{SF}}$	lookup
	$n' := \text{new } [n]_{\mathsf{T}}$	$n' := \text{new } [n]_{\mathsf{SF}}$	new

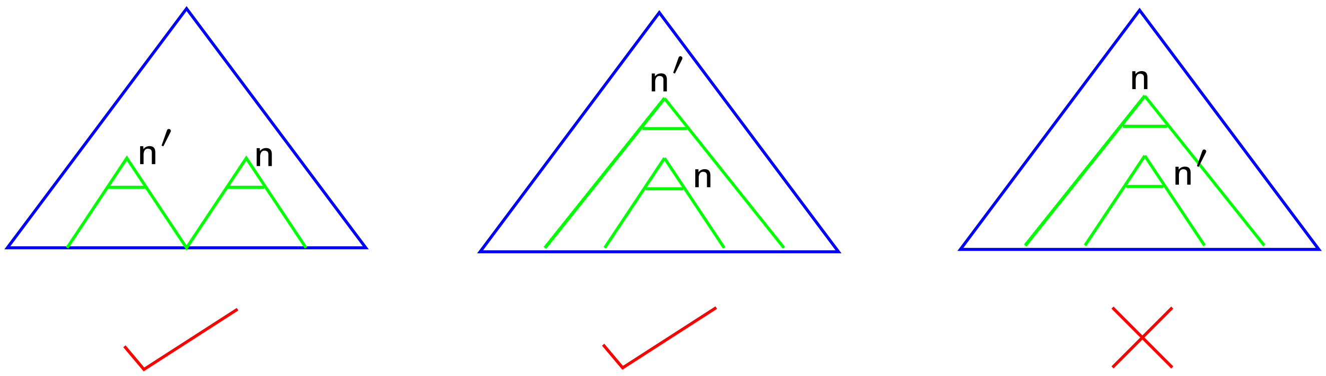
free$(\mathbb{C})$ = set of variables in $\mathbb{C}$; mod$(\mathbb{C})$ given by the red variables.

All the commands are local.

Move Example

$$
\begin{aligned}
\textit{move}(n, n') \triangleq\ & x := [n]_{\mathsf{T}}\ ; \\
& [n]_{\mathsf{T}} := 0\ ; \\
& [n']_{\mathsf{SF}} \mathrel{*}= x
\end{aligned}
$$

Three cases



Local Hoare Reasoning

Hoare triples $\{P\}\ \mathbb{C}\ \{Q\}$ partial, fault-avoiding interpretation

Frame Rule

$$\frac{\{P\}\ \mathbb{C}\ \{Q\}}{\{K(P)\}\ \mathbb{C}\ \{K(Q)\}}\ \mathsf{mod}(\mathbb{C}) \cap \mathsf{free}(K) = \emptyset$$

Plus consequence, auxiliary variable elimination and sequencing.

Soundness The rules are sound if the commands are local.

Sample Small Axioms

$$\{n[\mathrm{true}]\} \quad [n]_{\mathsf{T}} := 0 \quad \{0\}$$

$$\{n[y]\} \quad n' := \mathsf{new}\ [n]_{\mathsf{T}} \quad \{n[y] \circ n'[0]\}$$

Weakest Preconditions

$$\{(0 \triangleright P)(n[\mathrm{true}])\} \quad [n]_{\mathsf{T}} := 0 \quad \{P\}$$

$$\{\exists y.\forall n'.\,((n[y] \circ n'[0]) \triangleright P)(n[y])\} \quad n' := \mathsf{new}\ [n]_{\mathsf{T}} \quad \{P\}$$

where $y \notin \mathsf{free}(P)$

Derivations

$$\dfrac{\dfrac{\{n[\mathrm{true}]\}\ [n]_{\mathsf{T}} := 0\ \{0\}}{\{(0 \rhd P)(n[\mathrm{true}])\}\ [n]_{\mathsf{T}} := 0\ \{(0 \rhd P)(0)\}}\ \textsc{Frame}}{\{(0 \rhd P)(n[\mathrm{true}])\}\ [n]_{\mathsf{T}} := 0\ \{P\}}\ \textsc{Cons}$$

$$\dfrac{\dfrac{\{n[y]\}\ n' := \mathsf{new}\ [n]_{\mathsf{T}}\ \{n[y] \circ n'[0]\}}{\{K(n[y])\}\ n' := \mathsf{new}\ [n]_{\mathsf{T}}\ \{K(n[y] \circ n'[0])\}}\ \textsc{Frame}}{\{\exists y. K(n[y])\}\ n' := \mathsf{new}\ [n]_{\mathsf{T}}\ \{P\}}\ \textsc{Cons/Vars}$$

$K = (\forall n'.\,(n[y] \circ n'[0]) \rhd P)$ and $y \notin \mathsf{free}(P)$

Uses structural modus ponens $(P \rhd P')(P) \vdash P' \wedge \mathrm{True}(P)$.

Reasoning about Move

Safety property for move(n, n')

$$\{(0 \triangleright \mathrm{True}(n'[\mathrm{true}]))(n[\mathrm{true}])\}$$

$$x := [n]_{\mathsf{T}}$$

$$\{(0 \triangleright \mathrm{True}(n'[\mathrm{true}]))(n[\mathrm{true}])\}$$

$$[n]_{\mathsf{T}} := 0$$

$$\{\mathrm{True}(n'[\mathrm{true}])\}$$

$$[n']_{\mathsf{SF}} *= x$$

$$\{\mathrm{true}\}$$

Reasoning about Move

Specification of move(n, n')

$$\{(0 \triangleright \mathrm{True}(n'[x]))(n[y])\}$$

move(n, n')

$$\{\mathrm{True}(n'[x \circ n[y]\,])\}$$

Other Examples of Update

CL_0-reasoning about heap update is exactly analogous to SL-reasoning about heap update.

CL-reasoning about term rewriting possible, not possible using SL.

CL-reasoning about tree update, heap update and term rewriting is strikingly similar.

Challenge unified Hoare reasoning about data update

Summary

- Context Logic
- Application to Trees
- Local Reasoning about Tree Update
- Inexpressivity Result for Ambient Logic

Expressivity for AL and CL

Assume no quantification

Result AL is as expressive as CL for trees minus $\triangleright$

$$\text{Requires} \quad \hat{\Diamond} P \quad \mathsf{n}^{\perp}[P] \quad \mathsf{n}^{\hat{\perp}}[P]$$

Result AL is as expressive as AL minus structural adjoints

Lozes, then Dawar, Gardner, Ghelli

Result SL is as expressive as SL minus structural adjoints Lozes

Conjecture CL is as expressive as CL minus $\triangleright$

Requires context composition, probably requires multi-holed contexts

CL-reasoning is still essential for reasoning about tree update.

Parametric inexpressivity

Result

AL is not as expressive as CL_0 for trees using parametric expressivity

Intuition

CL-formula $(0 \triangleright \mathsf{m}_1[\mathsf{m}_2[0]])(\mathsf{n}[\mathrm{true}])$

AL-formula

$\mathsf{m}_1[\mathsf{m}_2[\mathsf{n}[\mathrm{true}]]] \vee (\mathsf{m}_1[\mathsf{m}_2[0] \circ \mathsf{n}[\mathrm{true}]]) \vee (\mathsf{m}_1[\mathsf{m}_2[0]] \circ \mathsf{n}[\mathrm{true}])$.

CL-formula $(0 \triangleright \diamond\, \mathsf{m}_2[\mathrm{true}])(\mathsf{n}[\mathrm{true}])$

AL-formula $\diamond\, \mathsf{m}_2[\mathrm{true}] \wedge \diamond\, \mathsf{n}[\neg \diamond\ \mathsf{m}_2[\mathrm{true}]]$

The CL-reasoning is more uniform than the AL-reasoning.

Parametric inexpressivity

Result

AL is not as expressive as CL_0 for trees using parametric expressivity

Proof For simplicity, we assume that the node labels are not unique.

Consider formula $(0 \triangleright p)(\mathsf{n}(\mathrm{true}))$, where p is a propositional variable. This formula describes a function from sets of trees to sets of trees. We prove that it is not expressible in AL.

Let p denote the set of trees whose node labels are equal.

Consider $\mathsf{m}[\mathsf{m}[0] \circ \mathsf{n}[0]]$ and $\mathsf{m}[\mathsf{m}'[0] \circ \mathsf{n}[0]]$ for arb. $\mathsf{m} \neq \mathsf{m}'$.

These trees cannot be distinguished by an AL-formula using p.

We work with finite set $N' \subseteq N$, with $\mathsf{n} \in N'$ and $\mathsf{m}, \mathsf{m}' \notin N'$.

AL-Bisimulation

Result from Modal Logic If $\sim$ is an AL-bisimulation, then $\mathsf{t} \sim \mathsf{t}'$ implies t, t' are logically indistinguishable in AL.

Definition Define the symmetric relation $\sim$ by $\mathsf{t} \sim \mathsf{t}'$ iff

- t has equal nodes iff t' has equal nodes
- $\mathsf{t} = \mathsf{n}[\mathsf{t}_1]$ implies there exists $\mathsf{n}', \mathsf{t}'_1$ st. $\mathsf{t}' = \mathsf{n}'[\mathsf{t}'_1]$and $\mathsf{t}_1 \sim \mathsf{t}'_1$ and if $\mathsf{n} \in N'$ then $\mathsf{n} = \mathsf{n}'$, and vice versa
- $\mathsf{t} = \mathsf{t}_1 \circ \mathsf{t}_2$ implies there exists $\mathsf{t}'_1, \mathsf{t}'_2$ st. $\mathsf{t}' = \mathsf{t}'_1 \circ \mathsf{t}'_2$ and $\mathsf{t}_1 \sim \mathsf{t}'_1$ and $\mathsf{t}_2 \sim \mathsf{t}'_2$, and vice versa.

$\mathsf{m}[\mathsf{m}[0] \circ \mathsf{n}[0]] \sim \mathsf{m}[\mathsf{m}'[0] \circ \mathsf{n}[0]]$ for $\mathsf{m}, \mathsf{m}' \notin N'$.

Result $\sim$ is a AL-bisimulation

Other Parametric Inexpressivity Results

Heaps

SL is parametrically as expressive as CL_0 for heaps.

SL is as expressive as first-order logic plus atomic heap formulae Lozes. It is not parametrically as expressive. Direct proof

Bisimulation proof technique still interesting $p * q$ not parametrically expressible in first-order logic using $p = list(3)$ and $q = list(4)$.

Sequences CL_0 for sequences is as expressive as BL for sequences

It describes the $\star$-free regular languages

It is not parametrically as expressive as BL for sequences

Conclusions

Context Logic is a fundamental logic for reasoning about data

Reasoning about data update requires reasoning about contexts

Parametric inexpressivity results are intriguing

Future

Combination of tree update with queries Gardner, Zarfaty, MFPS'06

Small-axiom approach prob. requires multi-holed contexts and wiring

Integration of high-level and low-level reasoning

Unified Hoare reasoning about data update

Other applications of Context Logic