Relational Constraint Solving in SMT

Baoluo Meng, Andrew Reynolds, Cesare Tinelli, Clark Barrett

THE UNIVERSITY OF IOWA

STANFORD

Introduction

Many computational problems can be modeled **relationally**

- High-level system design
- Reasoning about ontologies
- Architectural configuration of network systems
- Verification of programs with linked data structures
- …

Contributions

- Present **a theory of finite relations** $\mathcal{T}_{\mathcal{R}}$ as an extension to a theory of finite sets $\mathcal{T}_{\mathcal{S}}$ in our earlier work
- Present **a calculus** for the satisfiability of quantifier-free formulas in $\mathcal{T}_{\mathcal{R}}$
- Implement **a modular theory solver** in the SMT solver – **CVC4** based on the DPLL(T) architecture
- Demonstrate useful **applications** of the theory $\mathcal{T}_{\mathcal{R}}$ in **Alloy** and **OWL**

Related Work

Alloy

A **declarative language** for modeling and analyzing structurally-rich systems

Based on **relational logic** with built-in transitive closure and cardinality

SAT-based analysis by the Alloy Analyzer

- **Prove the consistency** of a model
- **Disprove a given property** holds for a model

Relational Reasoning via SMT

El Ghazi et. al introduced an approach that **translates** the Alloy kernel language to the SMT-LIB language, enabling the **solving of Alloy constraints using SMT solvers** (**AlloyPE**)

The resulting SMT formulas are difficult to solve by SMT solvers because of **heavy usage of quantifiers** in the translation

Description Logics (DLs)

Fragments of relational logic for efficient **knowledge representation and reasoning**

Consider on purpose **only unary and binary relations**

Web Ontology Language (OWL): a semantic web ontology language based on **description logics**

- **Efficient solvers**: KONCLUDE, FaCT++, Chainsaw and etc.

A Theory of Finite Sets $\mathcal{T}_\mathcal{S}$

A Theory of Finite Set $\mathcal{T}_\mathcal{S}$

A theory $\mathcal{T}_\mathcal{S}$ of **finite sets with cardinality** was introduced in previous work (IJCAR 2016)

Implemented a **sound**, **complete** and **terminating procedure** for the theory $\mathcal{T}_\mathcal{S}$ in **CVC4**

Set Signature $\Sigma_{\mathcal{S}}$ of $\mathcal{T}_{\mathcal{S}}$

Empty Set: $[\,]$: $\mathrm{Set}(\alpha)$

Set constructor: $[\_]$: $\alpha \to \mathrm{Set}(\alpha)$

Subset: $\sqsubseteq$: $\mathrm{Set}(\alpha) \times \mathrm{Set}(\alpha) \to \mathrm{Bool}$

Membership: $\sqsubset$: $\alpha \times \mathrm{Set}(\alpha) \to \mathrm{Bool}$

Union, intersection, set difference:

$$\sqcap, \sqcup, \setminus \; : \mathrm{Set}(\alpha) \times \mathrm{Set}(\alpha) \to \mathrm{Set}(\alpha)$$

A Tableaux-style Calculus

- The calculus consists of a set of derivation rules in **guarded assignment form**
- The derivation rules modify **a state data structure**, where a state is either the distinguished state **unsat** or **a set $\mathcal{S}$ of constraints**.

A Tableaux-style Calculus

- The premises of a rule refer to the current state $\mathcal{S}$ and the conclusion describes how $\mathcal{S}$ is changed by the rule's application
- Rules with two or more conclusions, separated by the symbol □, are non-deterministic branching rules
- $\mathcal{S}, c$ is an abbreviation for $\mathcal{S}$ □ $\{c\}$, and $\mathcal{T}(\mathcal{S})$ denotes the set of all terms and subterms occurring in $\mathcal{S}$

A Tableaux-style Calculus

We define the following **closure operator** for $\mathcal{S}$ where $\models_{\mathbf{tup}}$ denotes **entailment** in the theory.

$$\begin{aligned}
\mathcal{S}^* = {} & \{s \approx t \mid s, t \in \mathcal{T}(\mathcal{S}),\ \mathcal{S} \models_{\text{tup}} s \approx t\} \cup \\
& \{s \not\approx t \mid s, t \in \mathcal{T}(\mathcal{S}),\ \mathcal{S} \models_{\text{tup}} s \approx s' \wedge t \approx t' \text{ for some } s' \not\approx t' \in \mathcal{S}\} \cup \\
& \{s \sqsubseteq t \mid s, t \in \mathcal{T}(\mathcal{S}),\ \mathcal{S} \models_{\text{tup}} s \approx s' \wedge t \approx t' \text{ for some } s' \sqsubseteq t' \in \mathcal{S}\}
\end{aligned}$$

A Calculus for $\mathcal{T}_\mathcal{S}$

$$\textsc{Inter Up}\ \frac{x \sqsubseteq s \in \mathcal{S}^* \qquad x \sqsubseteq t \in \mathcal{S}^* \qquad s \sqcap t \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\, x \sqsubseteq s \sqcap t}$$

$$\textsc{Inter Down}\ \frac{x \sqsubseteq s \sqcap t \in \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\, x \sqsubseteq s,\, x \sqsubseteq t}$$

$$\textsc{Union Up}\ \frac{x \sqsubseteq u \in \mathcal{S}^* \qquad u \in \{s, t\} \qquad s \sqcup t \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\, x \sqsubseteq s \sqcup t}$$

$$\textsc{Union Down}\ \frac{x \sqsubseteq s \sqcup t \in \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\, x \sqsubseteq s \ \parallel\ \mathcal{S} := \mathcal{S},\, x \sqsubseteq t}$$

Derivation rules for intersection and union

A Calculus for $\mathcal{T}_{\mathcal{S}}$ cont.

Diff Up

$$\frac{x \sqsubseteq s \in \mathcal{S}^* \qquad s \setminus t \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\ x \sqsubseteq t \ \parallel\ \mathcal{S} := \mathcal{S},\ x \sqsubseteq s \setminus t}$$

Diff Down

$$\frac{x \sqsubseteq s \setminus t \in \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\ x \sqsubseteq s,\ x \not\sqsubseteq t}$$

Single Up

$$\frac{[\,x\,] \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\ x \sqsubseteq [\,x\,]}$$

Single Down

$$\frac{x \sqsubseteq [\,y\,] \in \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\ x \approx y}$$

Empty Unsat

$$\frac{x \sqsubseteq [\,] \in \mathcal{S}^*}{\textsf{unsat}}$$

Set Diseq

$$\frac{s \not\approx t \in \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\ z \sqsubseteq s,\ z \not\sqsubseteq t \quad \parallel \quad \mathcal{S} := \mathcal{S},\ z \not\sqsubseteq s,\ z \sqsubseteq t}$$

Eq Unsat

$$\frac{(t \not\approx t) \in \mathcal{S}^*}{\textsf{unsat}}$$

Derivation rules for set difference, singleton, disequality and contradiction

A Relational Extension $\mathcal{T}_{\mathcal{R}}$ to $\mathcal{T}_{\mathcal{S}}$

Notation

$\mathbf{Tup}_{\boldsymbol{n}}(\boldsymbol{\alpha}_{\mathbf{1}}, \dots, \boldsymbol{\alpha}_{\boldsymbol{n}})$**:** a parametric tuple sort (n > 0)

$\mathbf{Set}(\mathbf{Tup}_{\boldsymbol{n}}(\boldsymbol{\alpha}_{\mathbf{1}}, \dots, \boldsymbol{\alpha}_{\boldsymbol{n}}))$**:** a relational sort and abbreviate it as $\mathbf{Rel}_{\boldsymbol{n}}(\boldsymbol{\alpha}_{\mathbf{1}}, \cdots, \boldsymbol{\alpha}_{\boldsymbol{n}})$

Relational Signature $\Sigma_{\mathcal{R}}$ of $\boldsymbol{\mathcal{T}_{\mathcal{R}}}$

Tuple constructor:

$$\langle \_ ,\ \dots ,\_ \rangle : \alpha_1 \times \cdots \times \alpha_n \to \mathrm{Tup}_n(\alpha_1, \dots, \alpha_n)$$

Product: $\mathrm{Rel}_m(\alpha) \times \mathrm{Rel}_n(\beta) \to \mathrm{Rel}_{m+n}(\alpha, \beta)$

Join: $\bowtie$ $\mathrm{Rel}_{p+1}(\alpha, \gamma) \times \mathrm{Rel}_{q+1}(\gamma, \beta) \to \mathrm{Rel}_{p+q}(\alpha, \beta)$
with $p + q > 0$

Transpose: $\_^{-1}$: $\mathrm{Rel}_m(\alpha_1, \cdots, \alpha_m) \to \mathrm{Rel}_m(\alpha_m, \cdots, \alpha_1)$

Transitive Closure: $\_^{+}$: $Rel_2(\alpha, \alpha) \to Rel_2(\alpha, \alpha)$

TRANSPOSE Derivation Rule ($\_^{-1}$)

TRANSP UP

$$\frac{\langle x_1, \ldots, x_n \rangle \sqsubseteq R \in \mathcal{S}^* \quad R^{-1} \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\ \langle x_n, \ldots, x_1 \rangle \sqsubseteq R^{-1}}$$

TRANSP DOWN

$$\frac{\langle x_1, \ldots, x_n \rangle \sqsubseteq R^{-1} \in \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\ \langle x_n, \ldots, x_1 \rangle \sqsubseteq R}$$

JOIN Derivation Rule ($\bowtie$)

JOIN UP

$$\frac{\langle x_1, \ldots, x_m, z\rangle \sqsubseteq R_1, \langle z, y_1, \ldots, y_n\rangle \sqsubseteq R_2 \in \mathcal{S}^* \quad m+n>0 \quad R_1 \bowtie R_2 \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\ \langle x_1, \ldots, x_m, y_1, \ldots, y_n\rangle \sqsubseteq R_1 \bowtie R_2}$$

JOIN DOWN

$$\frac{\langle x_1, \ldots, x_m, y_1, \ldots, y_n\rangle \sqsubseteq R_1 \bowtie R_2 \in \mathcal{S}^* \quad \mathrm{ar}(R_1) = m+1}{\mathcal{S} := \mathcal{S},\ \langle x_1, \ldots, x_m, z\rangle \sqsubseteq R_1,\ \langle z, y_1, \ldots, y_n\rangle \sqsubseteq R_2}$$

z is a fresh variable

PRODUCT Derivation Rule (□)

PROD UP

$$\frac{\langle x_1, \ldots, x_m \rangle \sqsubseteq R_1 \in \mathcal{S}^* \quad \langle y_1, \ldots, y_n \rangle \sqsubseteq R_2 \in \mathcal{S}^* \quad R_1 * R_2 \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\ \langle x_1, \ldots, x_m, y_1, \ldots, y_n \rangle \sqsubseteq R_1 * R_2}$$

PROD DOWN

$$\frac{\langle x_1, \ldots, x_m, y_1, \ldots, y_n \rangle \sqsubseteq R_1 * R_2 \in \mathcal{S}^* \quad \mathrm{ar}(R_1) = m}{\mathcal{S} := \mathcal{S},\ \langle x_1, \ldots, x_m \rangle \sqsubseteq R_1,\ \langle y_1, \ldots, y_n \rangle \sqsubseteq R_2}$$

TRANSITIVE CLOSURE Derivation Rule ($\_^{+}$)

TCLOS UP I

$$\frac{\langle x_1, x_2\rangle \sqsubseteq R \in \mathcal{S}^* \quad R^+ \in \mathcal{T}(\mathcal{S})}{\mathcal{S} := \mathcal{S},\ \langle x_1, x_2\rangle \sqsubseteq R^+}$$

TCLOS UP II

$$\frac{\langle x_1, x_2\rangle \sqsubseteq R^+,\ \langle x_2, x_3\rangle \sqsubseteq R^+ \in\ \mathcal{S}^*}{\mathcal{S} := \mathcal{S},\ \langle x_1, x_3\rangle \sqsubseteq R^+}$$

TCLOS DOWN

$$\frac{\langle x_1, x_2\rangle \sqsubseteq R^+ \in \mathcal{S}^*}{\begin{array}{l}\mathcal{S} := \mathcal{S},\ \langle x_1, x_2\rangle \sqsubseteq R \quad \| \quad \mathcal{S} := \mathcal{S},\ \langle x_1, z\rangle \sqsubseteq R,\ \langle z, x_2\rangle \sqsubseteq R \\ \| \quad \mathcal{S} := \mathcal{S},\ \langle x_1, z_1\rangle \sqsubseteq R,\ \langle z_1, z_2\rangle \sqsubseteq R^+,\ \langle z_2, x_2\rangle \sqsubseteq R,\ z_1 \not\approx z_2\end{array}}$$

z, z_1, z_2 are fresh variables

Example

$$\mathcal{S} = \{\langle a,b\rangle \notin \mathsf{R}^{-1}, \mathsf{R} \approx \mathsf{Q}, \boxed{\langle a\rangle \in \mathsf{P}, \langle b\rangle \in \mathsf{P}}, \mathsf{P} * \mathsf{P} \approx \mathsf{Q} \sqcap \mathsf{T}\}$$

PROD UP

$$\mathcal{S}_1 := \mathcal{S} \cup \{\boxed{\langle a,b\rangle \in \mathsf{P} * \mathsf{P}, \langle b,a\rangle \in \mathsf{P} * \mathsf{P}, \langle a,a\rangle \in \mathsf{P} * \mathsf{P}}, \dots\}$$

$\mathsf{P} * \mathsf{P} \approx \mathsf{Q} \sqcap \mathsf{T}$ **INTER DOWN**

$$\mathcal{S}_2 := \mathcal{S}_1 \cup \{\langle a,b\rangle \in \mathsf{Q}, \boxed{\langle b,a\rangle \in \mathsf{Q}}, \langle a,a\rangle \in \mathsf{Q}, \dots\}$$

$\langle b,a\rangle \in \mathsf{Q}, \mathsf{R} \approx \mathsf{Q}$ **TRANSP UP**

$$\mathcal{S}_3 := \mathcal{S}_2 \cup \{\boxed{\langle a,b\rangle \in \mathsf{R}^{-1}}, \dots\} \xrightarrow[\langle a,b\rangle \notin \mathsf{R}^{-1}]{\textbf{EQ UNSAT}} \textbf{UNSAT}$$

Example

$$\mathcal{S} = \{\boxed{\langle a, b\rangle \in R^{+},}\ \langle a, b\rangle \in \mathrm{R}, \langle a, b\rangle \in \mathrm{R} \bowtie \mathrm{R}\}$$

TCLOS DOWN

$$\mathcal{S}_1 := \mathcal{S} \cup \{\langle a, b\rangle \in \mathrm{R}\}$$

EQ UNSAT $\langle a, b\rangle \notin \mathrm{R}$

UNSAT

Example

$$\mathcal{S} = \{\boxed{\langle a, b\rangle \;\square\; R^{+},}\, \langle a, b\rangle \;\square\; \mathrm{R}, \langle a, b\rangle \;\square\; \mathrm{R} \bowtie \mathrm{R}\}$$

$\big\downarrow$ **TCLOS DOWN**

$$\mathcal{S}_1 := \mathcal{S} \cup \{\langle a, z\rangle \in \mathrm{R}, \langle z, b\rangle \;\square\; \mathrm{R}\}$$

$\big\downarrow$ **JOIN UP**

$$\mathcal{S}_2 := \mathcal{S}_1 \;\square\; \{\langle a, b\rangle \;\square\; \mathrm{R} \bowtie \mathrm{R}\}$$

$\langle a, b\rangle \notin \mathrm{R} \bowtie \mathrm{R}$ $\big\downarrow$ **EQ UNSAT**

UNSAT

Example

$$\mathcal{S} = \{\boxed{\langle a, b\rangle \; \square \; R^{+},} \langle a, b\rangle \; \square \; \mathrm{R}, \langle a, b\rangle \; \square \; \mathrm{R} \bowtie \mathrm{R}\}$$

TCLOS DOWN

$$\mathcal{S}_1 \; \square{=} \; \mathcal{S} \; \square \; \{\langle a, z_1\rangle \; \square \; \mathrm{R}, \langle z_1, z_2\rangle \; \square \; \mathrm{R}, \left\langle z_{2,} \; b\right\rangle \; \square \; \mathrm{R}, z_1 \not\approx z_2\}$$

NO RULES APPLY

SAT

Calculus $\mathcal{C}_{\mathcal{R}}$ Correctness

Refutation and Model Soundness

Proposition 1 (Refutation Soundness). *If there is a closed derivation tree with root node $\mathcal{S}$, then $\mathcal{S}$ is T_R-unsatisfiable.*

Proposition 2 (Model Soundness). *Let $\mathcal{S}$ be the leaf of a saturated branch in a derivation tree. There is a model $\mathcal{I}$ of T_R that satisfies $\mathcal{S}$ and is such that (i) for all $S \in \mathrm{Vars}(\mathcal{S})$ of set sort, $S^{\mathcal{I}} = \left\{x^{\mathcal{I}} \,\middle|\, x \sqsubseteq S \in \mathcal{S}^*\right\}$, and (ii) for all other $x, y \in \mathrm{Vars}(\mathcal{S})$, $x^{\mathcal{I}} = y^{\mathcal{I}}$ if and only if $x \approx y \in \mathcal{S}^*$.*

Detailed proof can be found in the paper!

Termination of a Fragment of $\mathcal{T}_{\mathcal{R}}$

$$
\begin{array}{ll}
\text{(element)} & e := x \\
\text{(unary relation)} & u := x \mid [\,] \mid u_1 \sqcup u_2 \mid u_1 \sqcap u_2 \mid [\,\langle e \rangle\,] \mid b \bowtie u \\
\text{(binary relation)} & b := x \mid [\,] \mid b_1 \sqcup b_2 \mid b_1 \sqcap b_2 \mid [\,\langle e_1, e_2 \rangle\,] \mid b^{-1} \\
\text{(constraint)} & \varphi := e_1 \approx e_2 \mid \langle e \rangle \sqsubseteq u \mid \langle e_1, e_2 \rangle \sqsubseteq b \mid \neg \varphi_1
\end{array}
$$

Proposition 3 (Termination): *If S is a finite set of constraints generated by the grammar in above figure, then all derivation trees with root node S are finite.*

Detailed proof can be found in the paper!

Applications of $\mathcal{T}_{\mathcal{R}}$

A Mapping from Alloy to CVC4

Full support for Alloy kernel language in SMT **natively**

Finite model finding of CVC4 can reason in the presence of quantified formulas

Can **prove and disprove properties** with respect to Alloy models

ALLOY KERNEL LANGUAGE	CVC4
Signature sig S	S : $Rel_1(Atom)$
Field $f : S_1 \rightarrow \cdots \rightarrow S_n$ of a sig S	$f : Rel_{n+1}(Atom, \ldots, Atom)$ $f \sqsubseteq S \,\square\, S_1 \,\square\, \cdots \,\square\, S_n$
sig $S_1, \ldots, S_n$ extends S	$S_1 \sqsubseteq S, \ldots, S_n \sqsubseteq S$ $S_i \sqcap S_j = [\ ]$ for $1 \leq i < j \leq n$ $S_1 \sqcup \cdots \sqcup S_n = S$ **if S is abstract**
sig $S_1, \ldots, S_n$ in S,	$S_1 \sqsubseteq S, \ldots, S_n \sqsubseteq S$

ALLOY KERNEL LANGUAGE	CVC4
Set Operators: +, &, −, =, in	$\sqcup, \sqcap$ −, $\approx, \sqsubseteq$
Relational Operators: ~, □, →, ^	$\_^{-1}$, $\bowtie$, □, $\_^{+}$
Logical operators: and, or, not	AND, OR, NOT
Quantifiers: all, some	FORALL, EXISTS

A File System Example

Alloy Model	**CVC4 Encoding**

abstract sig FileSystemObj{}

sig File **extends** FileSystemObj{}

sig Dir **extends** FileSystemObj{
  contents: **Set** FileSystemObj
}

all f: File | **some** d: Dir |
  f **in** d.contents

Atom : **TYPE**;
FileSystemObj : **Rel$_1$(Atom);**
File : **Rel$_1$(Atom);**
Dir : **Rel$_1$(Atom);**
contents : **Rel$_2$(Atom, Atom);**
contents ⊑ Dir □ FileSystemObj;
Dir ⊓ File ≈ [ ];
Dir ⊔ File ≈ FileSystemObj;

□ f : **Atom** | <f> □ File =>
 □ d : **Atom** | <d> □ Dir □
  <f> □ [<d>] ⋈ contents

Evaluation on Alloy Benchmarks

Evaluate CVC4 with two configurations

- **CVC4**: enables full native support for relational operators
- **CVC4+AX:** encodes all relational operators as uninterpreted functions with axioms

Compare with **Alloy Analyzer** and **AlloyPE** on two sets of benchmarks: AlloyPE and one selected from an academic class

Evaluation on Alloy Benchmarks

Compared to the **Alloy Analyzer**

- **CVC4** is **overall slower for SAT** benchmarks
- **CVC4** **solves UNSAT** benchmarks, whereas the Alloy Analyzer can only answer bounded UNSAT

Compared to **AlloyPE**

- **CVC4** **solves SAT** benchmarks, whereas AlloyPE solves none
- **CVC4** **solves most** of AlloyPE's benchmarks

Compared to **CVC4+AX**

- **CVC4** **solves SAT** benchmarks, whereas CVC4+AX solves none
- **CVC4** **solves significantly more UNSAT** benchmarks

Experimental Evaluation on *AlloyPE* Benchmarks

	Alloy Analyzer		CVC4		CVC4+AX		AlloyPE	
Problem	**Res.**	**Time**	**Res.**	**Time**	**Res.**	**Time**	**Res.**	**Time**
mem-wr	b-uns	195.98/35	uns	0.43	uns	0.48	uns	0.44
mem-wi	b-uns	260.66/29	uns	0.45	uns	0.50	uns	0.42
ab-ai	b-uns	185.06/28	uns	0.46	uns	0.79	uns	0.49
ab-dua	b-uns	193.33/27	uns	0.49	uns	0.48	uns	0.70
abt-dua	b-uns	137.87/14	uns	0.60	uns	0.81	uns	0.70
abt-ly-u	b-uns	261.23/9	uns	0.81	uns	28.26	uns	1.4
abt-ly-p	b-uns	277.86/8	uns	0.81	uns	1.77	uns	175.19
gp-nsf	b-uns	152.55/69	uns	0.41	uns	0.59	uns	0.43
gp-nsg	b-uns	166.75/66	uns	0.42	-	to	uns	0.44
com-1	b-uns	297.18/13	uns	2.95	-	to	uns	0.59
com-2	b-uns	295.73/13	-	to	-	to	uns	0.55
com-3	b-uns	295.33/14	uns	4.29	-	to	uns	0.64
com-4a	b-uns	301.57/13	uns	9.39	-	to	uns	0.99
com-4b	b-uns	299.77/13	uns	0.90	-	to	uns	0.61
fs-sd	b-uns	157.90/70	uns	0.42	-	to	uns	0.89
fs-nda	b-uns	271.38/44	uns	0.55	-	to	uns	0.83
gc-s1	b-uns	270.07/14	uns	4.92	uns	8.14	uns	14.27
gc-s2	b-uns	288.44/8	-	to	-	to	uns	10.66
gc-c	b-uns	287.73/8	-	to	-	to	uns	42.31
hr-l	b-uns	275.80/7	-	to	-	to	-	to

Experimental Evaluation on Academic Benchmarks

	Alloy Analyzer		CVC4		CVC4+AX		AlloyPE	
Problem	**Res.**	**Time/Scope**	**Res.**	**Time**	**Res.**	**Time**	**Res.**	**Time**
academia_0	sat	0.60/3	sat	1.55	-	to	unk	84.76
academia_1	sat	0.53/2	sat	1.93	-	to	-	to
academia_2	sat	0.45/2	sat	0.49	-	to	unk	0.15
social_1	sat	0.52/3	sat	1.20	-	to	n/a	-
social_5	sat	1.56/2	sat	0.49	-	to	n/a	-
social_6	sat	0.49/2	sat	0.52	-	to	n/a	-
cf_0	sat	0.47/3	sat	0.51	-	to	n/a	-
cf_1	sat	0.49/3	sat	0.78	-	to	n/a	-
javatypes	sat	0.50/3	sat	0.42	-	to	uns	2.35
set	sat	0.45/2	sat	0.46	-	to	unk	0.92
loc_int	sat	0.57/1	sat	2.82	-	to	n/a	-
genealogy	sat	0.64/6	sat	89.20	-	to	n/a	-
number_1	sat	0.81/2	sat	8.65	-	to	n/a	-
railway	sat	0.67/4	sat	156.45	-	to	n/a	-
academia_3	b-uns	162.17/63	uns	0.49	uns	1.05	uns	0.28
academia_4	b-uns	246.92/162	uns	0.43	uns	0.54	uns	0.13
family_1	b-uns	146.62/68	uns	0.41	uns	0.44	uns	0.15
family_2	b-uns	279.77/30	uns	1.02	uns	48.78	uns	0.23
social_2	b-uns	256.98/56	uns	0.66	-	to	n/a	-
social_3	b-uns	191.45/57	uns	0.49	uns	35.91	n/a	-
social_4	b-uns	171.26/64	uns	0.46	uns	18.13	n/a	-
birthday	b-uns	156.08/53	uns	0.45	uns	0.61	uns	0.13
library	b-uns	259.54/119	uns	0.42	uns	0.40	uns	1.11
lights	b-uns	228.89/122	uns	32.69	-	to	n/a	-
INSLabel	b-uns	198.53/8	uns	1.46	-	to	n/a	-
farmers_1	sat	1.04/8	-	to	-	to	n/a	-
views	sat	9.91/9	-	to	-	to	n/a	-

A Mapping from OWL DL to SMT

OWL DL based on the **expressive**, yet **decidable**, description logic $\mathcal{SHOIN}(\boldsymbol{D})$

Built a **translation** from $\mathcal{SHOIN}(\boldsymbol{D})$ constructs to their **SMT counterparts** in $\mathcal{T}_{\mathcal{R}}$

Perform **consistency checks** on OWL models using CVC4

OWL DL	**CVC4**
Individual name a	a : Atom
Nominal {a}	{<a>}
Top concept T	Univ, {□ a : Atom | <a> □ Univ}
Bottom concept □	[ ]
Atomic concept C	C : Rel_1(Atom)
Role R	R : Rel_2(Atom, Atom)
Union C ⊔ D	C ⊔ D
Intersection C ⊓ D	C ⊓ D
Inverse role R^{-}	R^{-1}
Complement ¬C	Univ \ C

OWL DL	CVC4
Concept, role assertion C(a), R(a; b)	a □ C, <a, b> □ R
Individual (dis)equality $a \approx b$, $a \not\approx b$	$a \approx b$, $a \not\approx b$
Concept, role inclusion $C \sqsubseteq D$, $R \sqsubseteq S$	$C \sqsubseteq D$, $R \sqsubseteq S$
Concept, role equiv. $C \equiv D$, $R \equiv S$	$C \approx D$, $R \approx S$
Complex role inclusion $R_1 \circ R_2 \sqsubseteq S$	$R_1 \bowtie R_2 \sqsubseteq S$
Role disjointness Disjoint(R, S)	$R \sqcap S \approx [\,]$

OWL DL	CVC4
Existential restriction □R.C	R ⋈ C
Universal restriction ∀R.C	[x | x □ Univ □ [x] ⋈ R ⊑ C]
At-least restriction $\geq_n$R.C	[x | x □ Univ □ (□ a_1, … , a_n: Atom [<a_1>, … , <a_n>] ⊑ (([x] ⋈ R) ⊓ C) □ Dist(a_1, … , a_n))]
At-most restriction $\leq_n$R.C	[x | x □ Univ □ (□ a_1, … , a_n: Atom (([x] ⋈ R) ⊓ C) ⊑ [<a_1>, … , <a_n>] □ [<a_1>, … , <a_n>] ⊑ C)]
Local reflexivity □R.Self	[<x, y> | <x, y> □ R x ≈ y]

Evaluation on OWL Benchmarks

Experiment on **3936 OWL models** from 4th OWL Reasoner competition with comparison to the state-of-the-art **DL reasoner KONCLUDE**

KONCLUDE gave **answers for all benchmarks** with an average solving time **0.02 sec**

CVC4 found **3,639 consistent**, found **7 inconsistent**, and **timed out (30s) on the remaining 290** with an average solving time **1.7 secs**

Conclusion

- Presented **a calculus** for an extension to the theory of finite sets that includes support for **relations and relational operators**
- Implemented the calculus as **a modular extension** to the set subsolver in our SMT solver CVC4
- Evaluated the solver on **Alloy and OWL benchmarks** showing **promising results**

Future Work

- Investigate more **expressive fragments** for which our calculus terminates

- Devise an approach for a theory that includes both **relational** constraints and **cardinality** constraints

- Extend our logic with the **set complement operator** and a constant for the **universal set**

Thanks for Listening!

- Relational solver implemented in CVC4
  - Open source
  - Available at: http://cvc4.cs.stanford.edu/web/
  - Working on *.smt2 standard format for relations

References

1. F. Baader. The description logic handbook: Theory, implementation and applications. Cambridge university press, 2003.

2. F. Baader, I. Horrocks, and U. Sattler. Description logics. In V. L. Frank van Harmelen and B. Porter, editors, Handbook of Knowledge Representation, volume 3 of Foundations of Artificial Intelligence, pages 135 – 179. Elsevier, 2008.

3. K. Bansal, A. Reynolds, C. W. Barrett, and C. Tinelli. A new decision procedure for finite sets and cardinality constraints in SMT. In Proceedings of IJCAR'16, volume 9706 of LNCS, pages 82–98. Springer, 2016.

4. C. Barrett, C. L. Conway, M. Deters, L. Hadarean, D. Jovanovi´c, T. King, A. Reynolds, and C. Tinelli. CVC4. In Proceedings of CAV'11, volume 6806 of LNCS, pages 171–177. Springer, 2011.

5. C. Barrett, P. Fontaine, and C. Tinelli. The SMT-LIB standard—Version 2.6. In A. Gupta and D. Kroening, editors, SMT 2010, 2010.

References

6. C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. Satisfiability modulo theories. In A. Biere, M. J. H. Heule, H. van Maaren, and T. Walsh, editors, Handbook of Satisfiability, volume 185, chapter 26, pages 825–885. IOS Press, February 2009.

7. B. Dutertre and L. D. Moura. The YICES SMT solver. Technical report, SRI International, 2006.

8. A. A. E. Ghazi and M. Taghdiri. Analyzing alloy constraints using an SMT solver: a case study. In 5th International Workshop on Automated Formal Methods (AFM), 2010.

9. A. A. E. Ghazi and M. Taghdiri. Relational reasoning via SMT solving. In Proceedings of FM'11, volume 6664 of LNCS, pages 133–148. Springer, 2011.

10. A. A. E. Ghazi, M. Taghdiri, and M. Herda. First-order transitive closure axiomatization via iterative invariant injections. In Proceedings of NFM'15, volume 9058 of LNCS. Springer, 2015.

References

11. I. Horrocks and U. Sattler. Decidability of shiq with complex role inclusion axioms. Artificial Intelligence, 160(1-2):79–104, 2004.

12. D. Jackson. Alloy: a lightweight object modelling notation. ACM Trans. Softw. Eng. Methodol., 11(2):256–290, 2002.

13. D. Jackson. Software Abstractions - Logic, Language, and Analysis. MIT Press, 2006.

14. R. Nieuwenhuis, A. Oliveras, and C. Tinelli. Solving SAT and SAT Modulo Theories: from an abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T). Journal of the ACM, 53(6):937–977, Nov. 2006.

15. A. Reynolds, C. Tinelli, A. Goel, and S. Krstic. Finite model finding in SMT. In Proceedings of CAV'13, volume 8044 of LNCS, pages 640–655. Springer, 2013.

References

16. A. Steigmiller, T. Liebig, and B. Glimm. Konclude: System description. Web Semantics: Science, Services and Agents on the World Wide Web, 27(1), 2014.

17. E. Torlak and D. Jackson. Kodkod: a relational model finder. In Proceedings of TACAS'07, volume 4424 of LNCS, pages 632–647. Springer, 2007.

18. D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: system description. In Proceedings of IJCAR'06, volume 4130 of LNCS. Springer, 2006.

19. D. Tsarkov and I. Palmisano. Chainsaw: a metareasoner for large ontologies. In I. Horrocks, M. Yatskevich, and E. Jim´enez-Ruiz, editors, ORE, 2012.

20. W3C. OWL 2 web ontology language, https://www.w3.org/2007/OWL/wiki/Syntax.