Automated Consistency Checking of Expressive Ontologies — Beware of the Wrong Interpretation of Success!

Christoph Benzmüller and Marco Ziener

Freie Universität Berlin

ARCOE-LogIC 2013, Corunna/Spain, 15 Sep 2013

Expressive Ontologies

- SUMO, Cyc, ...
- McCarthy-type treatment of context: `(knows A F)`, ...

Translations into FO logic exist

- first-orderized Cyc [RamachandranEtAl, WS-AAAI, 2005]
- TPTP-SUMO [Pease&Sutcliffe, ESARLT, 2007]
- Adimen-SUMO [AlvezLucioRigau, JSemWebInfSys, 2012]
- these have been analysed with proof tools
- revision process terminated(?), no more errors(?)

Question

- Can we now trust these translated ontologies?
- How about the faithfulness of the translation results?

Freie Universität Berlin

Expressive Ontologies

- SUMO, Cyc, ...
- McCarthy-type treatment of context: `(knows A F)`, ...

Translations into FO logic exist

- first-orderized Cyc [RamachandranEtAl, WS-AAAI, 2005]
- TPTP-SUMO [Pease&Sutcliffe, ESARLT, 2007]
- Adimen-SUMO [AlvezLucioRigau, JSemWebInfSys, 2012]
- these have been analysed with proof tools
- revision process terminated(?), no more errors(?)

Question

- Can we now trust these translated ontologies?
- How about the faithfulness of the translation results?

Expressive Ontologies

- SUMO, Cyc, ...
- McCarthy-type treatment of context: `(knows A F)`, ...

Translations into FO logic exist

- first-orderized Cyc [RamachandranEtAl, WS-AAAI, 2005]
- TPTP-SUMO [Pease&Sutcliffe, ESARLT, 2007]
- Adimen-SUMO [AlvezLucioRigau, JSemWebInfSys, 2012]
- these have been analysed with proof tools
- revision process terminated(?), no more errors(?)

Question

- Can we now trust these translated ontologies?
- How about the faithfulness of the translation results?

"The procedure we have used for finding inconsistencies in the ontology can be sketched as follows:

1. *An automated procedure translates a large part of SUMO (and discharges the remaining axioms).*
2. *Then, the whole resulting formula is given (as input) to a theorem prover for automatically finding an inconsistency (that is, without providing a goal).*
3. *When a refutation is found, the theorem prover provides a description of the proof, from which we select the collection of axioms involved in that refutation.*
4. *With the help of theorem provers (for example, for finding minimal inconsistent subcollections of axioms), we identify the source of the inconsistency and repair it.*
5. *Once we repair the problem, the process is repeated from the beginning."*

Figure : Automated error and inconsistency checking as has been applied for Adimen-SUMO; the text is copied from [AlvezLucioRigau, 2012]

Unfaithfulness may be hard to detect (this way)

- **Faithfulness** of a translation α from source logic S to target logic T:

$$\Gamma \models^{S} \Phi \quad \text{iff} \quad \alpha(\Gamma) \models^{T} \alpha(\Phi)$$

Ideally, faithfulness should be formally proved

- ... pen and paper
- ... for implementation

However, this is not always possible

- complex translations
- missing formal semantics
- verification of implementation usually difficult

Unfaithfulness may be hard to detect (this way)

- **Faithfulness** of a translation α from source logic S to target logic T:

$$\Gamma \models^{S} \Phi \quad \text{iff} \quad \alpha(\Gamma) \models^{T} \alpha(\Phi)$$

Ideally, faithfulness should be formally proved

- … pen and paper
- … for implementation

However, this is not always possible

- complex translations
- missing formal semantics
- verification of implementation usually difficult

Unfaithfulness may be hard to detect (this way)

- **Faithfulness** of a translation α from source logic S to target logic T:
$$\Gamma \models^S \Phi \quad \text{iff} \quad \alpha(\Gamma) \models^T \alpha(\Phi)$$

Ideally, faithfulness should be formally proved

- ... pen and paper
- ... for implementation

However, this is not always possible

- complex translations
- missing formal semantics
- verification of implementation usually difficult

Promising work (mainly in Bremen?)

- OntoIOp (`http://ontoiop.org`)
  Mossakowski, Lange, Kutz
- LATIN (`http://trac.omdoc.org/LATIN/`)
  Kohlhase, Mossakowski, Rabe, Codescu, Horozal

Formal assurance of meta-logical properties of logic translations

Contribution of our paper

A Non-trivial translation of entire SUMO ontology to TPTP THF0 syntax (classical HO logic)

B Framework for error detection and inconsistency checking based in HO proof tools (e.g. LEO-II, Satallax, Nitpick)

C Simple errors detected in SUMO within experiments

D Translation in A is unfaithful; however, this could not be detected in experiments

E What is the reason for this: missing A-Box information

- we focus on small SUMO subontology
- translation leads to satisfiable THF0 version
- some queries are answered fine
- when adding A-Box information, unfaithfulness can be revealed

F Suggestion: integrate A-Box information and (annotated) user queries in experiments with proof tools

A: TPTP THF0 Translation of SUMO

Two mappings presented in [BenzmüllerPease, JWebSem, 2013]

- Extensional translation THF0-SUMO-I (implemented):

```
(=> (knows ?AGENT ?FORMULA) (believes ?AGENT ?FORMULA))    (1)
(=> (knows ?AGENT ?FORMULA) (truth ?FORMULA True))         (2)
```

are translated into the following THF0 representation

```
% Type declarations
thf(truth,type,(truth:  ($o>$o>$o))).
thf(believes,type,(believes:  ($i>$o>$o))).
thf(knows,type,(knows:  ($i>$o>$o))).
% Axioms
thf(ax1126,axiom,((!  [FORMULA: $o,AGENT: $i]:                  (3)
 ((knows @ AGENT @ FORMULA) => (believes @ AGENT @ FORMULA))))).
thf(ax3303,axiom,((!  [FORMULA: $o,AGENT: $i]:                  (4)
 ((knows @ AGENT @ FORMULA) => (truth @ FORMULA @ $true))))).
```

- translation into quantified modal logic (not yet implemented)

A: TPTP THF0 translation of SUMO

- THF0-SUMO-I translation of

  SUMO + MILO + domain ontologies

- not translated: `documentation`, `names`, `termFormat`, `abbreviation`, `externalImage`
- THF0-SUMO-I has approx. 45000 axioms

`http://www.christoph-benzmueller.de/papers/SUMOMILODOMAINS.thf`

B: Framework for error and inconsistency detection

- Idea was to perform error and inconsistency detection as in Adimen-SUMO; but here on THF0-SUMO-I with ATPs
  - LEO-II (`http://leoprover.org`)
  - Satallax (`http://www.ps.uni-saarland.de/~cebrown/satallax`)
  - Nitpick (`http://www4.in.tum.de/~blanchet/nitpick.html`)
- ontology-clustering needed; provers otherwise get stuck
- clustering idea: use type information to identify subsets of axioms which are semantically independent to a large degree
- technical set-up of framework: see our paper
- 3.047.128 problems were passed to the provers
- some simple errors detected (and corrected)
- unfaithfulness of the THF0-SUMO-I remained undetected

C: Simple errors detected

```
(=>
  (and
    (instance ?AGENT Agent)
    (potentialCustomer ?CUST ?AGENT)
    (modalAttribute
      (and
        (instance ?R Reserving)
        (destination ?R ?AGENT)) Necessity)
    (conditionalProbability
      (exists (?RES1)
        (and
          (instance ?RES1 Reservation)
          (reservingEntity ?CUST ?RES1)
          (fulfillingEntity ?AGENT ?RES1)))
      (customer ?CUST ?AGENT) ?NUM1)
    (conditionalProbability
      (not
        (exists (?RES2)
          (and
            (instance ?RES1 Reservation)
            (reservingEntity ?CUST ?RES2)
            (fulfillingEntity ?AGENT ?RES2))))
      (customer ?CUST ?AGENT) ?NUM2))
  (lessThan ?NUM2 ?NUM1))
```

D: Unfaithfulness is hard to detect (E1 and E1*)

SUMO toy ontology E1

```
(=> (knows ?AGENT ?FORMULA) (believes ?AGENT ?FORMULA))     (5)
(=> (knows ?AGENT ?FORMULA) (truth ?FORMULA True))          (6)
```

THF0-SUMO-I toy ontology E1*

```
% Type declarations
thf(truth,type,(truth: ($o>$o>$o))).
thf(believes,type,(believes: ($i>$o>$o))).
thf(knows,type,(knows: ($i>$o>$o))).
% Axioms
thf(ax1126,axiom,((! [FORMULA: $o,AGENT: $i]:                  (7)
 ((knows @ AGENT @ FORMULA) => (believes @ AGENT @ FORMULA))))).
thf(ax3303,axiom,((! [FORMULA: $o,AGENT: $i]:                  (8)
 ((knows @ AGENT @ FORMULA) => (truth @ FORMULA @ $true))))).
```

E1* is satisfiable (milliseconds for LEO-II and Satallax)

D: Unfaithfulness is hard to detect (E2)

SUMO toy ontology E2 = E1 +

```
~(truth ((father bruce ben) & (father bruce bill)) True)   (aboxAx1)
(knows peter (father bruce ben))                             (aboxAx2)
(believes peter (father bruce bill))                         (aboxAx3)
```

D: Unfaithfulness is hard to detect (E2*)

```
% Axioms
thf(ax1126,axiom,(
    ! [FORMULA: $o,AGENT: $i] :
      ( ( knows @ AGENT @ FORMULA ) => ( believes @ AGENT @ FORMULA ) ) )).

thf(ax3303,axiom,(
    ! [FORMULA: $o,AGENT: $i] :
      ( ( knows @ AGENT @ FORMULA ) => ( truth @ FORMULA @ $true ) ) )).

thf(aboxAx0,axiom,
    ( ( ben != bill ) & ( bruce != ben ) & ( bruce != bill )
    & ( peter != ben ) & ( peter != bill ) & ( peter != bruce ) )).

thf(aboxAx1,axiom,(
    ~ ( truth
      @ ( ( father @ bruce @ ben ) & ( father @ bruce @ bill ) )
      @ $true ) )).

thf(aboxAx2,axiom,
    ( knows @ peter @ ( father @ bruce @ ben ) )).

thf(aboxAx3,axiom,
    ( believes @ peter @ ( father @ bruce @ bill ) )).
```

Fig. 2. The satisfiable toy ontology **E2***. Adding `~(believes peter ~(father bruce bill))` results in an unsatisfiable set of THF0 axioms. This is clearly counter-intuitive.

D: Unfaithfulness is hard to detect (E2*)

E2* is satisfiable (milliseconds for LEO-II and Satallax)
The following extension of E2 resp. E2* is unsatisfiable

```
~(believes peter ~(father bruce bill))
```
(aboxAx4)

LEO-II, Satallax and Nitpick detect this

The following query holds for E2 resp. E2*

```
believes peter ~(father bruce bill))
```
(query)

LEO-II and Satallax show this

This is counterintuitive! But the problem can only be detected if the A-Box information is present

D: Unfaithfulness is hard to detect (E2*)

E2* is satisfiable (milliseconds for LEO-II and Satallax)
The following extension of E2 resp. E2* is unsatisfiable

```
~(believes peter ~(father bruce bill))
```
(aboxAx4)

LEO-II, Satallax and Nitpick detect this

The following query holds for E2 resp. E2*

```
believes peter ~(father bruce bill))
```
(query)

LEO-II and Satallax show this

This is counterintuitive! But the problem can only be detected if the A-Box information is present

D: Unfaithfulness is hard to detect (E2*)

E2* is satisfiable (milliseconds for LEO-II and Satallax)
The following extension of E2 resp. E2* is unsatisfiable

```
~(believes peter ~(father bruce bill))
```
(aboxAx4)

LEO-II, Satallax and Nitpick detect this

The following query holds for E2 resp. E2*

```
believes peter ~(father bruce bill))
```
(query)

LEO-II and Satallax show this

This is counterintuitive! But the problem can only be detected if the A-Box information is present

D: Insufficient for detecting unfaithfulness

"The procedure we have used for finding inconsistencies in the ontology can be sketched as follows:

1. *An automated procedure translates a large part of SUMO (and discharges the remaining axioms).*
2. *Then, the whole resulting formula is given (as input) to a theorem prover for automatically finding an inconsistency (that is, without providing a goal).*
3. *When a refutation is found, the theorem prover provides a description of the proof, from which we select the collection of axioms involved in that refutation.*
4. *With the help of theorem provers (for example, for finding minimal inconsistent subcollections of axioms), we identify the source of the inconsistency and repair it.*
5. *Once we repair the problem, the process is repeated from the beginning."*

Figure : Automated error and inconsistency checking as has been applied for Adimen-SUMO; the text is copied from [AlvezLucioRigau, 2012]

Conclusion

- Ideally, faithfulness should be formally ensured
- Error and inconsistency checking, if performed only on the translated ontology, may not reveal unfaithfulness
- The particular problem here is an extensional treatment of intensional concepts (modalities) in the translation
- Additional A-Box information, in combination with (annotated) user queries, may help to detect unfaithfulness

- Future work:
  Systematic creation of respective A-Box information?