ELTE
Thematic Excellence Program
Industry and Digitalisation
Application Domain Specific Highly Reliable IT Solutions

Semantic consistency behind ontology learning and schema mapping for heterogeneous data integration

Molnár Bálint, Ma Chuangtao
Department of Information Systems, Faculty of Informatics,
Eötvös Loránd University (ELTE)
{molnarba, machuangtao}@inf.elte.hu

Introduction

- It is a common phenomenon that some inconsistencies and semantic conflicts will inevitably occur during (semi-)automatic ontology learning [1]. Therefore, the issues of inconsistencies and redundancies are becoming the bottleneck in the scenario of (semi-)automatic ontology learning from relational database (RDB).
- How to efficiently identify the inconsistencies between learned ontologies and their original databases is one of the critical tasks in the ontology learning from RDB.

Method and Formalization

- To tackle above problem, a semantic consistency method based on model checking is presented in this wok. The main workflow of the presented method is depicted in Figure 1.

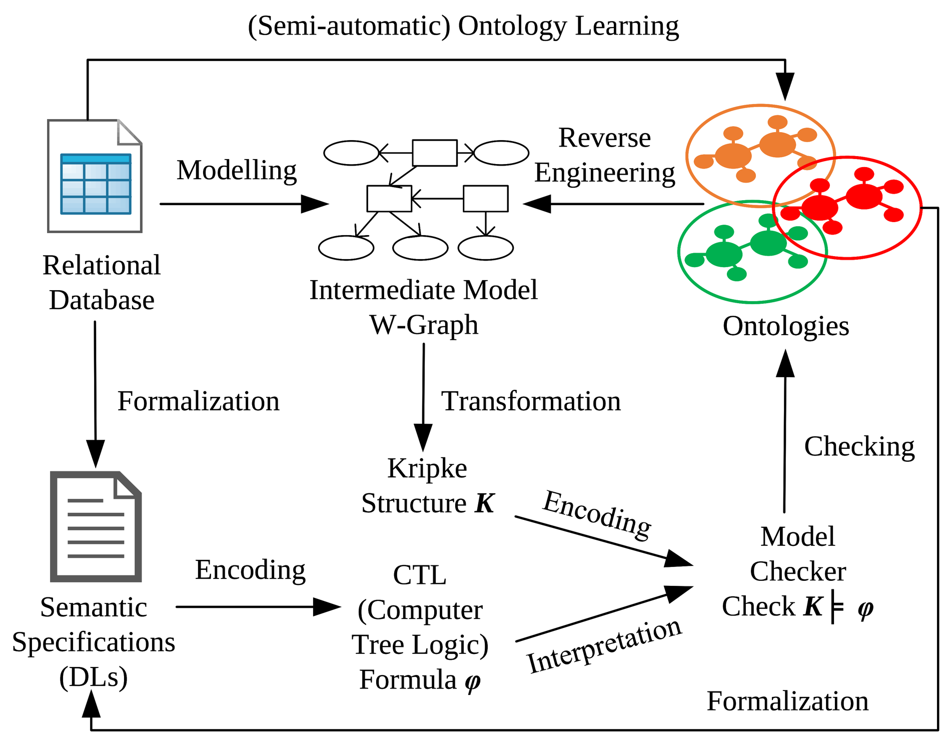


Fig. 1. Semantic consistency checking for learning ontology from relational database.

- The `Mini University` ontology [2], and its corresponding database, are selected as an example to describe the specific steps of presented method.
- We formalized the RDB with W-graph and Kripke structure, thereby, the W-Instance was transformed into a Kripke structure (Figure 2).

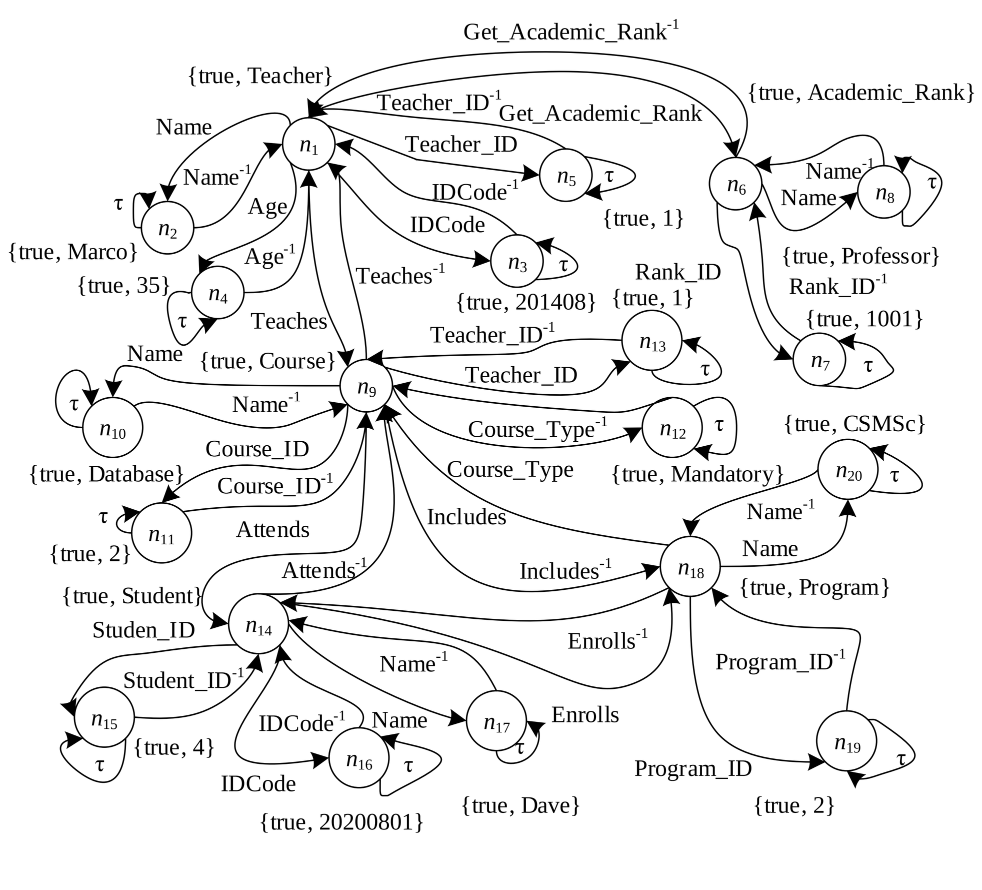


Fig. 2. Kripke structure of `Mini University` data model.

Algorithm and Verification

- Considering that nuXmv [3] is an extended version of NuSMV, which provides a strong verification based on advanced SAT-based algorithms, thereby the nuXmv is employed to verify the presented method. The specific process of consistency checking based on graph intermediate representation and model checking is shown in Algorithm 1.

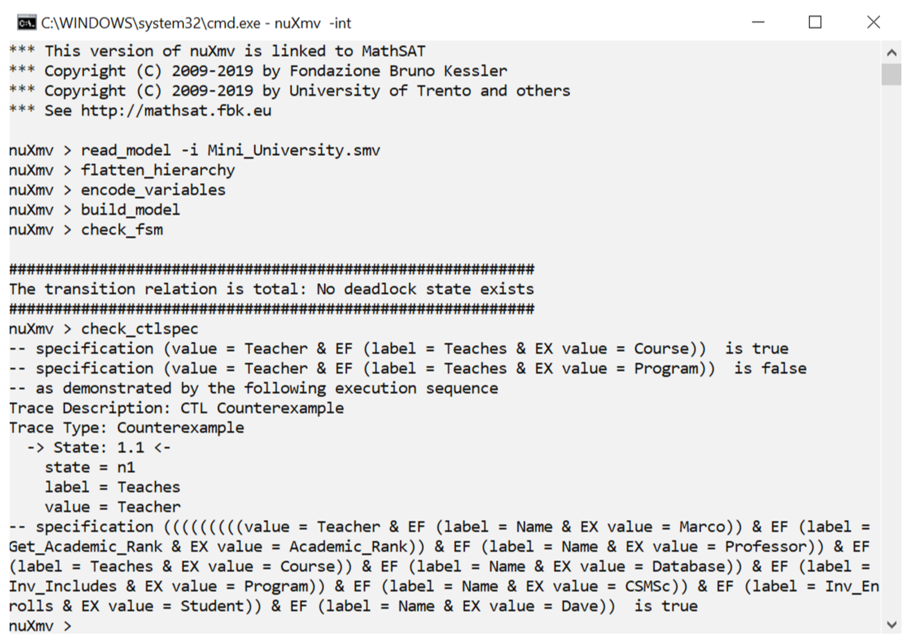
Algorithm 1 Semantic Consistency Checking based on nuXmv Model Checker

Input:
$\mathcal{K}_{\mathcal{G}}$: Kripke structure of original RDB represented based on W-Graph.
$\mathcal{O}$: Ontologies generated from RDB schema and instance $\mathcal{R}$.
Output:
$\mathcal{R}_{\mathcal{C}}$: The result of consistency checking: **true** or **false**.

procedure CONSISTENCY CHECKING
  Encoding Kripke structure $\mathcal{K}_{\mathcal{G}}$ with *SMV* program.
  Translating semantics of ontologies to CTL formula $\varphi(\mathcal{O}, \mathcal{R})$.
  Verifying if the $\varphi(\mathcal{O}, \mathcal{R})$ satisfies the $\mathcal{K}_{\mathcal{G}}$.
  if $\mathcal{K}_{\mathcal{G}} \models \varphi(\mathcal{O}, \mathcal{R})$ **then**
    $\mathcal{R}_{\mathcal{C}}$=true.
  else if $\mathcal{K}_{\mathcal{G}} \not\models \varphi(\mathcal{O}, \mathcal{R})$ **then**
    $\mathcal{R}_{\mathcal{C}}$=false.
  end if
  return $\mathcal{R}_{\mathcal{C}}$.
end procedure

Results and Conclusion

- Before running the nuXmv model checker, it is necessary to check whether the given specification satisfies the Kripke structure. We checked if there exist deadlock states by using `check_fsm` command.
- We checked not only the specifications that are consistent with the original database but also the specifications that are inconsistent with the original database. Accordingly, the result is shown in Figure 3.


```
C:\WINDOWS\system32\cmd.exe - nuXmv -int
*** This version of nuXmv is linked to MathSAT
*** Copyright (C) 2009-2019 by Fondazione Bruno Kessler
*** Copyright (C) 2009-2019 by University of Trento and others
*** See http://mathsat.fbk.eu

nuXmv > read_model -i Mini_University.smv
nuXmv > flatten_hierarchy
nuXmv > encode_variables
nuXmv > build_model
nuXmv > check_fsm

##########################################################
The transition relation is total: No deadlock state exists
##########################################################
nuXmv > check_ctlspec
-- specification (value = Teacher & EF (label = Teaches & EX value = Course))  is true
-- specification (value = Teacher & EF (label = Teaches & EX value = Program))  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -> State: 1.1 <-
    state = n1
    label = Teaches
    value = Teacher
-- specification ((((((((value = Teacher & EF (label = Name & EX value = Marco)) & EF (label =
Get_Academic_Rank & EX value = Academic_Rank)) & EF (label = Name & EX value = Professor)) & EF
(label = Teaches & EX value = Course)) & EF (label = Name & EX value = Database)) & EF (label =
Inv_Includes & EX value = Program)) & EF (label = Name & EX value = CSMSc)) & EF (label = Inv_En
rolls & EX value = Student)) & EF (label = Name & EX value = Dave))  is true
nuXmv >
```

Fig. 3. Result of consistency checking based on nuXmv model checker.

- We can observe that there is no deadlock state in the current Kripke model. The results given by nuXmv indicate whether the given specifications satisfy the specific Kripke model.
- The results shown that this method could correctly check and return the results of whether the given semantic specification of the learned ontology satisfies the original RDB.

Publications

- On the basis of the above work, there are two related publications:

1. C. Ma, B. Molnár, and A. Benczúr, “A semi-automatic semantic consistency checking method for learning ontology from relational database,” *Information*, vol. 12, no. 5, 2021.
2. C. Ma and B. Molnár, “Semantic consistency behind ontology learning and schema mapping for heterogeneous data integration,” in *Collection of Abstracts: 13th Joint Conference on Mathematics and Informatics (MaCS2020)*, 2020, pp. 115–115

References

[1] L. Zhu, G. Hua, S. Zafar, and Y. Pan, “Fundamental ideas and mathematical basis of ontology learning algorithm,” *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 4, pp. 4503–4516, 2018.

[2] K. Čerāns and G. Būmans, “RDB2OWL: A RDB-to-RDF/OWL mapping specification language,” in *Proceedings of the 2011 Conference on Databases and Information Systems*. IOS Press, 2011, p. 139–152.

[3] R. Cavada, A. Cimatti, M. Dorigatti, A. Griggio, A. Mariotti, A. Micheli, S. Mover, M. Roveri, and S. Tonetta, “The nuXmv symbolic model checker,” in *Computer Aided Verification*, A. Biere and R. Bloem, Eds. Cham: Springer International Publishing, 2014, pp. 334–342.

Application Domain Specific Highly Reliable IT Solutions project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

National Research, Development and Innovation Office
Hungary

PROGRAM FINANCED FROM THE NRDI FUND