Hung Son Nguyen
Quang-Thuy Ha · Tianrui Li
Małgorzata Przybyła-Kasperek (Eds.)

LNAI 11103

Rough Sets

International Joint Conference, IJCRS 2018
Quy Nhon, Vietnam, August 20–24, 2018
Proceedings

IJC
RS
2018

Springer

Lecture Notes in Artificial Intelligence 11103

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel
University of Alberta, Edmonton, Canada

Yuzuru Tanaka
Hokkaido University, Sapporo, Japan

Wolfgang Wahlster
DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann
DFKI and Saarland University, Saarbrücken, Germany

More information about this series at http://www.springer.com/series/1244

Hung Son Nguyen · Quang-Thuy Ha
Tianrui Li · Małgorzata Przybyła-Kasperek (Eds.)

Rough Sets

International Joint Conference, IJCRS 2018
Quy Nhon, Vietnam, August 20–24, 2018
Proceedings

Springer

Editors
Hung Son Nguyen
University of Warsaw
Warsaw
Poland

Quang-Thuy Ha
Faculty of Information Technology
Vietnam National University
Hanoi
Vietnam

Tianrui Li
School of Information Science
Southwest Jiaotong University
Chengdu
China

Małgorzata Przybyła-Kasperek
Institute of Computer Science
University of Silesia
Sosnowiec
Poland

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-99367-6 ISBN 978-3-319-99368-3 (eBook)
https://doi.org/10.1007/978-3-319-99368-3

Library of Congress Control Number: 2018951242

LNCS Sublibrary: SL7 – Artificial Intelligence

© Springer Nature Switzerland AG 2018
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The proceedings of the 2018 International Joint Conference on Rough Sets (IJCRS 2018) contain the results of the meeting of the International Rough Set Society held at the International Centre for Interdisciplinary Science and Education (ICISE) and the University of Quy Nhon in Quy Nhon, Vietnam, during August 2018.

Conferences in the IJCRS series are held annually and comprise four main tracks relating the topic rough sets to other topical paradigms: rough sets and data analysis covered by the RSCTC conference series from 1998, rough sets and granular computing covered by the RSFDGrC conference series since 1999, rough sets and knowledge technology covered by the RSKT conference series since 2006, and rough sets and intelligent systems covered by the RSEISP conference series since 2007. Owing to the gradual emergence of hybrid paradigms involving rough sets, it was deemed necessary to organize Joint Rough Set Symposiums, first in Toronto, Canada, in 2007, followed by symposiums in Chengdu, China in 2012, Halifax, Canada, 2013, Granada and Madrid, Spain, 2014, Tianjin, China, 2015, where the acronym IJCRS was proposed, continuing with the IJCRS 2016 conference in Santiago de Chile and IJCRS 2017 in Olsztyn, Poland.

The IJCRS conferences aim at bringing together experts from universities and research centers as well as from industry representing fields of research in which theoretical and applicational aspects of rough set theory already find or may potentially find usage. They also become a place for researchers who want to present their ideas to the rough set community, or for those who would like to learn about rough sets and find out if they can be useful for their problems.

This year's conference, IJCRS 2018, celebrated the 20th anniversary of the first international conference on rough sets called RSCTC, which was organized by Lech Polkowski and Andrzej Skowron during June 22–26, 1998, in Warsaw, Poland. On this occasion, we listened to a retrospective talk delivered by Andrzej Skowron, who summarized the successes of this field and showed directions for further research and development.

IJCRS 2018 attracted 61 submissions (not including invited contributions), which underwent a rigorous reviewing process. Each accepted full-length paper was evaluated by three to five experts on average. The present volume contains 45 full-length regular and workshop submissions, which were accepted by the Program Committee, as well as six invited articles.

The conference program included five keynotes and plenary talks, a fellow talk, eight parallel sessions, a tutorial, the 6th International Workshop on Three-way Decisions, Uncertainty, and Granular Computing, and a panel discussion on rough sets and data science.

The chairs of the Organizing Committee also prepared the best paper award and the best student paper award. From all research papers submitted, the Program Committee nominated five papers as finalists for the award and, based on the final presentations during the conference, selected the winners.

We would like to express our gratitude to all the authors for submitting papers to IJCRS 2018, as well as to the members of the Program Committee for organizing this year's attractive program.

We also gratefully thank our sponsors: Vietnam National University in Ho Chi Minh City, for providing the technical support and human resources for the conference; the University of Quy Nhon, for sponsoring the reception and the conference facilities during the first day and the last day; Ton Duc Thang University, for sponsoring the pre-conference workshops on rough sets and data mining.

The conference would not have been successful without support received from distinguished individuals and organizations. We express our gratitude to the IJCRS 2018 honorary chairs, Andrzej Skowron, Huynh Thanh Dat, and Do Ngoc My, for their great leadership. We appreciate the help of Dinh Thuc Nguyen, Nguyen Tien Trung, Quang Vinh Lam, Quang Thai Thuan, Thanh Tran Thien, Luong Thi Hong Cam, Giang Thuy Minh, Phung Thai Thien Trang, Dao Thi Hong Le, Hung Nguyen-Manh, and all other representatives of Vietnam National University in Ho Chi Minh City and Quy Nhon University, who were involved in the conference organization. We would also like to thank Marcin Szeląg, Sinh Hoa Nguyen, and Dang Phuoc Huy, who supported the conference as tutorial, workshop, and special session chairs. We acknowledge the significant help from Khuong Nguyen-An, Tran Thanh Hai, Ly Tran Thai Hoc, and Marcin Szczuka provided at various stages of the conference publicity, website, and material preparation.

We are grateful to Tu Bao Ho, Hamido Fujita, Hong Yu, Andrzej Skowron, Piero Pagliani, and Mohua Banerjee for delivering excellent keynote and plenary talks and fellow talks. We thank Dominik Ślęzak and Arkadiusz Wojna for the tutorial. We are thankful to Hong Ye, Mohua Banerjee, Mihir Chakraborty, Bay Vo, and Le Thi Thuy Loan for the organization of workshops and special sessions.

Special thanks go to Alfred Hofmann of Springer, for accepting to publish the proceedings of IJCRS 2018 in the LNCS/LNAI series, and to Anna Kramer for her help with the proceedings. We are grateful to Springer for the grant of 1,000 Euro for the best paper award winners. We would also like to acknowledge the use of EasyChair, a great conference management system.

We hope that the reader will find all the papers in the proceedings interesting and stimulating.

August 2018

Hung Son Nguyen
Quang-Thuy Ha
Tianrui Li
Małgorzata Przybyła-Kasperek

Organization

Honorary Chairs

Andrzej Skowron	University of Warsaw, Poland
Thanh Dat Huynh	VNU-HCMC, Vietnam
Ngoc My Do	Quy Nhon University, Vietnam

General Chairs

Davide Ciucci	University of Milano-Bicocca, Italy
Dan Thu Tran	VNU-HCMC, Vietnam

Organizing Committee Chairs

Dinh Thuc Nguyen	VNU-HCMC, Vietnam
Tien Trung Nguyen	Quy Nhon University, Vietnam
Quang Vinh Lam	VNU-HCMC, Vietnam
Quang Thai Thuan	Quy Nhon University, Vietnam
Thanh Thien Tran	Quy Nhon University, Vietnam

Program Committee

Program Committee Chairs

Hung Son Nguyen	University of Warsaw, Poland
Quang-Thuy Ha	College of Technology, VNU-Hanoi, Vietnam
Tianrui Li	Southwest Jiaotong University, Chengdu, China
Małgorzata Przybyła-Kasperek	University of Silesia, Poland

Workshop, Special Sessions, and Tutorial Chairs

Marcin Szeląg	Poznań University of Technology, Poland
Sinh Hoa Nguyen	Polish-Japanese Academy of IT, Poland
Phuoc Huy Dang	Dalat University, Vietnam

Program Committee

Mani A.	Calcutta University, India
Piotr Artiemjew	University of Warmia and Mazury, Poland
Jaume Baixeries	Universitat Politecnica de Catalunya, Spain

Mohua Banerjee	Indian Institute of Technology Kanpur, India
Jan Bazan	University of Rzeszów, Poland
Rafael Bello	Universidad Central de Las Villas, Cuba
Nizar Bouguila	Concordia University, Canada
Jerzy Baszczyski	Poznań University of Technology, Poland
Mihir Chakraborty	Jadavpur University, India
Shampa Chakraverty	Netaji Subhas Institute of Technology, India
Chien-Chung Chan	University of Akron, USA
Mu-Chen Chen	National Chiao Tung University, Taiwan
Costin-Gabriel Chiru	Technical University of Bucharest, Romania
Victor Codocedo	INSA Lyon, France
Chris Cornelis	University of Granada, Spain
Zoltan Erno Csajbok	University of Debrecen, Hungary
Jianhua Dai	Hunan Normal University, China
Rafal Deja	WSB, Poland
Dayong Deng	Zhejiang Normal University, China
Thierry Denoeux	Université de Technologie de Compiegne, France
Fernando Diaz	University of Valladolid, Spain
Pawel Drozda	University of Warmia and Mazury, Poland
Didier Dubois	IRIT/RPDMP, France
Ivo Dntsch	Brock University, Canada
Zied Elouedi	Institut Superieur de Gestion de Tunis, Tunisia
Rafael Falcon	Larus Technologies Corporation, Canada
Victor Flores	Universidad Catolica del Norte, Chile
Wojciech Froelich	University of Silesia, Poland
Brunella Gerla	University of Insubria, Italy
Piotr Gny	Polish-Japanese Academy of IT, Poland
Anna Gomolinska	University of Białystok, Poland
Salvatore Greco	University of Catania, Italy
Rafal Gruszczynski	Nicolaus Copernicus University in Toruń, Poland
Jerzy Grzymala-Busse	University of Kansas, USA
Bineet Gupta	Shri RamSwaroop Memorial University, India
Christopher Henry	University of Winnipeg, Canada
Christopher Hinde	Loughborough University, UK
Qinghua Hu	Tianjin University, China
Van Nam Huynh	JAIST, Japan
Dmitry Ignatov	National Research University HSE, Russia
Masahiro Inuiguchi	Osaka University, Japan
Ryszard Janicki	McMaster University, Canada
Richard Jensen	Aberystwyth University, UK
Xiuyi Jia	Nanjing University of Science and Technology, China
Michal Kepski	University of Rzeszów, Poland
Md. Aquil Khan	Indian Institute of Technology Indore, India
Yoo-Sung Kim	Inha University, South Korea
Marzena Kryszkiewicz	Warsaw University of Technology, Poland
Yasuo Kudo	Muroran Institute of Technology, Japan

Yoshifumi Kusunoki	Osaka University, Japan
Sergei O. Kuznetsov	National Research University HSE, Russia
Xuan Viet Le	Quy Nhon University, Vietnam
Huaxiong Li	Nanjing University, China
Jiye Liang	Shanxi University, China
Churn-Jung Liau	Academia Sinica, Taipei, Taiwan
Tsau Young Lin	San Jose State University, USA
Pawan Lingras	Saint Mary's University, Canada
Caihui Liu	Gannan Normal University, China
Guilong Liu	Beijing Language and Culture University, China
Pradipta Maji	Indian Statistical Institute, India
Benedetto Matarazzo	University of Catania, Italy
Jess Medina	University of Cadiz, Spain
Ernestina Menasalvas	Universidad Politecnica de Madrid, Spain
Claudio Meneses	Universidad Catolica del Norte, Chile
Marcin Michalak	Silesian University of Technology, Poland
Tams Mihlydek	University of Debrecen, Hungary
Fan Min	Southwest Petroleum University, China
Pabitra Mitra	Indian Institute of Technology Kharagpur, India
Sadaaki Miyamoto	University of Tsukuba, Japan
Mikhail Moshkov	KAUST, Saudi Arabia
Michinori Nakata	Josai International University, Japan
Amedeo Napoli	Inria, France
Hoang Son Nguyen	Hue University, Vietnam
Loan T. T. Nguyen	TDTU, Vietnam
Long Giang Nguyen	Institute of Information Technology, VAST, Vietnam
M. C. Nicoletti	FACCAMP and UFSCar, Brazil
Vilem Novak	University of Ostrava, Czech Republic
Agnieszka Nowak-Brzezińska	University of Silesia, Poland
Piero Pagliani	Research Group on Knowledge and Information, Italy
Sankar Pal	Indian Statistical Institute, India
Krzysztof Pancerz	University of Rzeszów, Poland
Vladimir Parkhomenko	SPbPU, Russia
Andrei Paun	University of Bucharest, Romania
Witold Pedrycz	University of Alberta, Canada
Tatiana Penkova	Institute of Computational Modelling SB RAS, Russia
Georg Peters	Munich University of Applied Sciences and Australian Catholic University, Germany
Alberto Pettorossi	Università di Roma Tor Vergata, Italy
Jonas Poelmans	Clarida Technologies, UK
Lech Polkowski	Polish-Japanese Academy of IT, Poland
Henri Prade	IRIT - CNRS, France
Mohamed Quafafou	Aix-Marseille University, France
Elisabeth Rakus-Andersson	Blekinge Institute of Technology, Sweden
Sheela Ramanna	University of Winnipeg, Canada

Zbigniew Ras	University of North Carolina at Charlotte, USA
Grzegorz Rozenberg	Leiden University, The Netherlands
Henryk Rybiski	Warsaw University of Technology, Poland
Wojciech Rzasa	Rzeszów University, Poland
Hiroshi Sakai	Kyushu Institute of Technology, Japan
Guido Santos	Universittsklinikum Erlangen, Germany
Gerald Schaefer	Loughborough University, UK
Zhongzhi Shi	Institute of Computing Technology Chinese Academy of Sciences, China
Marek Sikora	Silesian University of Technology, Poland
Bruno Simões	Vicomtech-IK4, Spain
Roman Słowinski	Poznan University of Technology, Poland
John Stell	University of Leeds, UK
Jaroslaw Stepaniuk	Bialystok University of Technology, Poland
Zbigniew Suraj	University of Rzeszów, Poland
Paul Sushmita	Indian Institute of Technology Jodhpur, India
Piotr Synak	Security On Demand, Poland
Andrzej Szałas	University of Warsaw, Poland
Marcin Szczuka	University of Warsaw, Poland
Ryszard Tadeusiewicz	AGH University of Science and Technology in Kraków, Poland
Bala Krushna Tripathy	VIT University, India
Li-Shiang Tsay	North Carolina A&T State University, USA
Dmitry Vinogradov	Federal Research Center for Computer Science and Control, RAS, Russia
Bay Vo	Ho Chi Minh City University of Technology, Vietnam
Alicja Wakulicz-Deja	University of Silesia, Poland
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Szymon Wilk	Poznan University of Technology, Poland
Arkadiusz Wojna	Security On Demand, Poland
Marcin Wolski	Maria Curie-Skłodowska University, Poland
Wei-Zhi Wu	Zhejiang Ocean University, China
Yan Yang	Southwest Jiaotong University, China
Jingtao Yao	University of Regina, Canada
Yiyu Yao	University of Regina, Canada
Dongyi Ye	Fuzhou University, China
Hong Yu	Chongqing University of Posts and Telecommunications, China
Mahdi Zargayouna	Université Paris Est, France
Nan Zhang	Yantai University, China
Qinghua Zhang	Chongqing University of Posts and Telecommunications, China
Yan Zhang	University of Regina, Canada
Bing Zhou	Sam Houston State University, USA
Wojciech Ziarko	University of Regina, Canada
Beata Zielosko	University of Silesia, Poland

Additional Reviewers

Azam, Nouman
Benítez Caballero, María José
Bui, Huong
Chen, Chun-Hao
Czołombitko, Michał
Jankowski, Dariusz
Le, Tuong
Li, Jinhai
Mai, Son
Nguyen, Dan
Nguyen, Duy Ham
Nguyen, Hoang Son
Nguyen, Van Du
Nguyen, Viet Hung
Pham, Thi-Ngan
Ramírez Poussa, Eloisa
Shah, Ekta
Son, Le Hoang
Su, Ja-Hwung
Vluymans, Sarah

Introducing Histogram Functions into a Granular Approximate Database Engine (Industry Talk)

Dominik Ślęzak[1] and Arkadiusz Wojna[2]

[1] Institute of Informatics, University of Warsaw, Poland
[2] Security On-Demand, USA/Poland

Abstract. We discuss an approximate database engine that we started designing at Infobright, and now we continue its development for Security On-Demand (SOD). At SOD, it is used in everyday data analytics, allowing for fast approximate execution of ad-hoc queries over tens of billions of data rows [1]. In our engine, queries are run against collections of histograms that represent domains of single columns over groupings of consecutively loaded data rows (so-called packrows). Query execution process corresponds to transformation of such granulated summaries of the input data into summaries reflecting query results [2].

We compare our algorithms that generate histogram descriptions of the original data with data quantization methods that are widely used in data mining. We also introduce a new idea of extending SQL with function *hist(a)* that produces quantized representation of column *a* by means of merging *a*'s histograms corresponding to particular packrows into a unified *a*'s histogram over the whole data. We refer to our recent works on summary-based data visualization [3] and machine learning [4] in order to illustrate several scenarios of utilizing *hist* in practice.

Keywords: Big data analytics · Data granulation · Data quantization

References

1. Ślęzak, D., Chądzyńska-Krasowska, A., Holland, J., Synak, P., Glick, R., Perkowski, M.: Scalable cyber-security analytics with a new summary-based approximate query engine. In: Proceedings of BigData, pp. 1840–1849 (2017)
2. Ślęzak, D., Glick, R., Betliński, P., Synak, P.: A new approximate query engine based on intelligent capture and fast transformations of granulated data summaries. J. Intell. Inf. Syst. **50**(2), 385–414 (2018)
3. Chądzyńska-Krasowska, A., Stawicki, S., Ślęzak, D.: A metadata diagnostic framework for a new approximate query engine working with granulated data summaries. In: Polkowski, L., et al. (eds.) IJCRS 2017. LNCS, vol. 10313, pp. 623–643. Springer, Cham
4. Ślęzak, D., Borkowski, J., Chądzyńska-Krasowska, A.: Ranking mutual information dependencies in a summary-based approximate analytics framework. In: Proceedings of HPCS (2018)

Contents

Fuzzy Bisimulations in Fuzzy Description Logics Under the Gödel Semantics

Quang-Thuy Ha[1], Linh Anh Nguyen[2,3](✉), Thi Hong Khanh Nguyen[4], and Thanh-Luong Tran[5]

[1] Faculty of Information Technology, VNU University of Engineering and Technology, 144 Xuan Thuy, Hanoi, Vietnam
thuyhq@vnu.edu.vn

[2] Division of Knowledge and System Engineering for ICT, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
nguyenanhlinh@tdt.edu.vn

[3] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
nguyen@mimuw.edu.pl

[4] Faculty of Information Technology, Electricity Power University, 235 Hoang Quoc Viet, Hanoi, Vietnam
khanhnth@epu.edu.vn

[5] Department of Information Technology, University of Sciences, Hue University, 77 Nguyen Hue, Hue, Vietnam
ttluong@hueuni.edu.vn

Abstract. Description logics (DLs) are a suitable formalism for representing knowledge about domains in which objects are described not only by attributes but also by binary relations between objects. Fuzzy DLs can be used for such domains when data and knowledge about them are vague. One of the possible ways to specify classes of objects in such domains is to use concepts in fuzzy DLs. As DLs are variants of modal logics, indiscernibility in DLs is characterized by bisimilarity. The bisimilarity relation of an interpretation is the largest auto-bisimulation of that interpretation. In (fuzzy) DLs, it can be used for concept learning. In this paper, for the first time, we define fuzzy bisimulation and (crisp) bisimilarity for fuzzy DLs under the Gödel semantics. The considered logics are fuzzy extensions of the DL $\mathcal{ALC}_{reg}$ with additional features among inverse roles, nominals, qualified number restrictions, the universal role and local reflexivity of a role. We give results on invariance of concepts as well as conditional invariance of TBoxes and ABoxes for bisimilarity in fuzzy DLs under the Gödel semantics. We also provide a theorem on the Hennessy-Milner property for fuzzy bisimulations in fuzzy DLs under the Gödel semantics.

© Springer Nature Switzerland AG 2018

H. S. Nguyen et al. (Eds.): IJCRS 2018, LNAI 11103, pp. 559–571, 2018.
https://doi.org/10.1007/978-3-319-99368-3_44

1 Introduction

In traditional machine learning, objects are usually described by attributes, and a class of objects can be specified, among others, by a logical formula using attributes. Decision trees and rule-based classifiers are variants of classifiers based on logical formulas. To construct a classifier, one can restrict to using a sublanguage that allows only essential attributes and certain forms of formulas. If two objects are indiscernible w.r.t. that sublanguage, then they belong to the same decision class. Indiscernibility is an equivalence relation that partitions the domain into equivalence classes, and each decision class is the union of some of those equivalence classes.

There are domains in which objects are described not only by attributes but also by binary relations between objects. Examples include social networks and linked data. For such domains, description logics (DLs) are a suitable formalism for representing knowledge about objects. Basic elements of DLs are concepts, roles and individuals (objects). A concept name is a unary predicate, a role name is binary predicate. A concept is interpreted as a set of objects. It can be built from atomic concepts, atomic roles and individual names (as nominals) by using constructors. As DLs are variants of modal logics, indiscernibility in DLs is characterized by bisimilarity. The bisimilarity relation of an interpretation $\mathcal{I}$ w.r.t. a logic language is the largest auto-bisimulation of $\mathcal{I}$ w.r.t. that language. It has been exploited for concept learning in DLs [6,10,15,17,18].

In practical applications, data and knowledge may be imprecise and vague, and fuzzy logics can be used to deal with them. Fuzzy DLs have attracted researchers for two decades (see [1,3] for overviews and surveys). If objects are described by attributes and binary relations, and data about them are vague, then one of the possible ways to specify classes of objects is to use concepts in fuzzy DLs. Bisimilarity in fuzzy DLs can be used for learning such concepts. Thus, bisimilarity and bisimulation in fuzzy DLs are worth studying.

There are different families of fuzzy operators. The Gödel, Łukasiewicz, Product and Zadeh families are the most popular ones. The first three of them use t-norms for defining implication. The Gödel and Zadeh families define conjunction and disjunction of truth values as infimum and supremum, respectively. Each family of fuzzy operators represents a semantics, which is extended to fuzzy DLs appropriately (see, e.g., [2]).

The objective of this paper is to introduce and study bisimulations in fuzzy DLs under the Gödel semantics. Apart from the works [7,12,14] on bisimulation/bisimilarity in traditional or paraconsistent DLs and the earlier mentioned works on using bisimilarity for concept learning in traditional DLs, other notable related works are [5,8,9]. In [8] Eleftheriou et al. presented (weak) bisimulation and bisimilarity in Heyting-valued modal logics and proved the Hennessy-Milner property for those notions. A Heyting-valued modal logic uses a Heyting algebra as the space of truth values. There is a close relationship between Heyting-valued modal logics and fuzzy modal logics under the Gödel semantics, as every linear Heyting algebra is a Gödel algebra [8] and every Gödel algebra is a Heyting algebra with the Dummett condition [4]. In [5] Ćirić et al. introduced bisimulations for fuzzy automata. Such a bisimulation is a fuzzy relation between the sets of states of the two considered automata. One of the results of [5] states that there is a uniform forward bisimulation between fuzzy automata $\mathcal{A}$ and $\mathcal{B}$ iff there is a special isomorphism between the factor fuzzy automata of them w.r.t. their greatest forward bisimulation fuzzy equivalence relations. It is a kind of the Hennessy-Milner property. In [9] Fan introduced fuzzy bisimulations for some Gödel modal logics, which are fuzzy modal logics using the Gödel semantics. The considered logics include the fuzzy monomodal logic K and its extensions with converse and/or involutive negation. She proved that fuzzy bisimulations in those logics have the Hennessy-Milner property. The work [9] follows the approach of [5] in defining bisimulation as a fuzzy relation and expressing conditions of bisimulation by using relational composition. As discussed in [9], there is a relationship between fuzzy bisimulations in Gödel modal logics and weak bisimulations in Heyting-valued modal logics [8], especially for the case when the underlying Heyting algebra is linear.

In this paper, we define fuzzy bisimulation and (crisp) bisimilarity for fuzzy DLs under the Gödel semantics. The considered logics are fuzzy extensions of the DL $\mathcal{ALC}_{reg}$ with additional features among inverse roles, nominals, qualified number restrictions, the universal role and local reflexivity of a role. The DL $\mathcal{ALC}_{reg}$ is a variant of Propositional Dynamic Logic (PDL) [16]. It extends the basic DL $\mathcal{ALC}$ with role constructors like program constructors of PDL. We give results on invariance of concepts as well as conditional invariance of TBoxes and ABoxes for bisimilarity in fuzzy DLs under the Gödel semantics. Moreover, we provide a theorem on the Hennessy-Milner property for fuzzy bisimulations in fuzzy DLs under the Gödel semantics. Roughly speaking, it states that, if fuzzy interpretations $\mathcal{I}$ and $\mathcal{I}'$ are witnessed and modally saturated, then $Z : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'} \to [0,1]$ is the greatest fuzzy bisimulation between $\mathcal{I}$ and $\mathcal{I}'$ iff $Z(x,x') = \inf\{C^{\mathcal{I}}(x) \Leftrightarrow C^{\mathcal{I}'}(x) \mid C \text{ is a concept}\}$ for all $x \in \Delta^{\mathcal{I}}$ and $x' \in \Delta^{\mathcal{I}'}$, where $\Leftrightarrow$ denotes the Gödel equivalence.

The motivations of our work are as follows:

- (Fuzzy) bisimulation has potential applications to concept learning in fuzzy DLs, i.e., for machine learning in information systems based on fuzzy DLs. It was not studied for fuzzy DLs under the Gödel semantics.
- The class of fuzzy DLs studied in this paper is large. In comparison with [9], not only are they variants of multimodal (instead of monomodal) logics, but they also allow PDL-like role constructors, qualified number restrictions, nominals, the universal role and the concept constructor that represents local reflexivity of a role.
- To deal with qualified number restrictions, the approach of using relational composition for defining conditions of (fuzzy) bisimulation in [5,9] is not suitable, and we have to use "elementary" conditions for defining bisimulation. Consequently, when restricting to the fuzzy monomodal logic K, our notion of fuzzy bisimulation is different in nature from the one introduced by Fan [9] (see Remark 3), although the *greatest* fuzzy bisimulation relations specified by these two different approaches coincide. This means that our study on fuzzy

bisimulations in fuzzy DLs under the Gödel semantics is not a simple extension of Fan's work [9] on fuzzy bisimulations in Gödel monomodal logics. Due to the mentioned difference, proofs of our results are more complicated.
- This paper serves as a starting point for studying bisimulation and bisimilarity in fuzzy DLs under other t-norm based semantics (e.g., Łukasiewicz and Product).

The remainder of this paper is structured as follows. In Sect. 2, we formally specify the considered fuzzy DLs and their Gödel semantics. In Sect. 3, we define fuzzy bisimulations. In Sect. 4, we present our results on invariance of concepts, TBoxes and ABoxes for bisimilarity in fuzzy DLs under the Gödel semantics. Section 5 contains our results on the Hennessy-Milner property of fuzzy bisimulations. Concluding remarks are given in Sect. 6. Due to the lack of space, all proofs of our results are omitted. They will be made available online or published in an extended version of the paper.

2 Preliminaries

In this section, we recall the Gödel fuzzy operators, fuzzy DLs under the Gödel semantics and define related notions that are needed for this paper.

2.1 The Gödel Fuzzy Operators

The family of Gödel fuzzy operators are defined as follows, where $p, q \in [0, 1]$:

$$\begin{aligned} p \otimes q &= \min\{p, q\} \\ p \oplus q &= \max\{p, q\} \\ \ominus p &= (\text{if } p = 0 \text{ then } 1 \text{ else } 0) \\ (p \Rightarrow q) &= (\text{if } p \leq q \text{ then } 1 \text{ else } q) \\ (p \Leftrightarrow q) &= (p \Rightarrow q) \otimes (q \Rightarrow p). \end{aligned}$$

Note that $(p \Leftrightarrow q) = 1$ if $p = q$, and $(p \Leftrightarrow q) = \min\{p, q\}$ otherwise.

For a set $\mathit{\Gamma}$ of values in $[0, 1]$, we define $\otimes\mathit{\Gamma} = \inf \mathit{\Gamma}$ and $\oplus\mathit{\Gamma} = \sup \mathit{\Gamma}$, where the extrema are taken in the complete lattice $[0, 1]$.

Given $R, S : \mathit{\Delta} \times \mathit{\Delta}' \to [0, 1]$, if $R(x, y) \leq S(x, y)$ for all $\langle x, y\rangle \in \mathit{\Delta} \times \mathit{\Delta}'$, then we write $R \leq S$ and say that S is *greater than or equal to* R. We write $R \oplus S$ to denote the function of type $\mathit{\Delta} \times \mathit{\Delta}' \to [0, 1]$ defined as follows:

$$(R \oplus S)(x, y) = R(x, y) \oplus S(x, y).$$

If $\mathcal{Z}$ is a set of functions of type $\mathit{\Delta} \times \mathit{\Delta}' \to [0, 1]$, then by $\oplus\mathcal{Z}$ we denote the function of the same type defined as follows:

$$(\oplus\mathcal{Z})(x, y) = \oplus\{Z(x, y) \mid Z \in \mathcal{Z}\}.$$

Given $R : \mathit{\Delta} \times \mathit{\Delta}' \to [0, 1]$ and $S : \mathit{\Delta}' \times \mathit{\Delta}'' \to [0, 1]$, the composition $R \circ S$ is a function of type $\mathit{\Delta} \times \mathit{\Delta}'' \to [0, 1]$ defined as follows:

$$(R \circ S)(x, y) = \oplus\{R(x, z) \otimes S(z, y) \mid z \in \mathit{\Delta}'\}.$$

2.2 Fuzzy Description Logics Under the Gödel Semantics

By Φ we denote a set of features among I, O, Q, U and $\mathtt{Self}$, which stand for inverse roles, nominals, qualified number restrictions, the universal role and local reflexivity of a role, respectively. In this subsection, we first define the syntax of roles and concepts in the fuzzy DL $\mathcal{L}_\Phi$, where $\mathcal{L}$ extends the DL $\mathcal{ALC}_{reg}$ with fuzzy (truth) values and $\mathcal{L}_\Phi$ extends $\mathcal{L}$ with the features from Φ. We then define fuzzy interpretations and the Gödel semantics of $\mathcal{L}_\Phi$.

Our logic language uses a set $\mathbf{C}$ of *concept names*, a set $\mathbf{R}$ of role names, and a set $\mathbf{I}$ of individual names. A *basic role* of $\mathcal{L}_\Phi$ is either a role name or the inverse r^- of a role name r (when $I \in \Phi$).

Roles and *concepts* of $\mathcal{L}_\Phi$ are defined as follows:

- if $r \in \mathbf{R}$, then r is a role of $\mathcal{L}_\Phi$,
- if R, S are roles of $\mathcal{L}_\Phi$ and C is a concept of $\mathcal{L}_\Phi$, then $R \circ S$, $R \sqcup S$, R^* and $C?$ are roles of $\mathcal{L}_\Phi$,
- if $I \in \Phi$ and R is a role of $\mathcal{L}_\Phi$, then R^- is a role of $\mathcal{L}_\Phi$,
- if $U \in \Phi$, then U is a role of $\mathcal{L}_\Phi$, called the *universal role* (we assume that $U \notin \mathbf{R}$),
- if $p \in [0, 1]$, then p is a concept of $\mathcal{L}_\Phi$,
- if $A \in \mathbf{C}$, then A is a concept of $\mathcal{L}_\Phi$,
- if C, D are concepts of $\mathcal{L}_\Phi$ and R is a role of $\mathcal{L}_\Phi$, then:
  - $C \sqcap D$, $C \to D$, $\neg C$, $C \sqcup D$, $\forall R.C$, $\exists R.C$ are concepts of $\mathcal{L}_\Phi$,
  - if $O \in \Phi$ and $a \in \mathbf{I}$, then $\{a\}$ is a concept of $\mathcal{L}_\Phi$,
  - if $Q \in \Phi$, R is a basic role of $\mathcal{L}_\Phi$ and $n \in \mathbb{N}$, then $\geq n\, R.C$ and $\leq n\, R.C$ are concepts of $\mathcal{L}_\Phi$,
  - if $\mathtt{Self} \in \Phi$ and $r \in \mathbf{R}$, then $\exists r.\mathtt{Self}$ is a concept of $\mathcal{L}_\Phi$.

The concept 0 stands for $\bot$, and 1 for $\top$.

By $\mathcal{L}_\Phi^0$ we denote the largest sublanguage of $\mathcal{L}_\Phi$ that disallows the role constructors $R \circ S$, $R \sqcup S$, R^*, $C?$ and the concept constructors $\neg C$, $C \sqcup D$, $\forall R.C$, $\leq n\, R.C$.

We use letters A and B to denote *atomic concepts* (which are concept names), C and D to denote arbitrary concepts, r and s to denote *atomic roles* (which are role names), R and S to denote arbitrary roles, a and b to denote individual names.

Given a finite set $\Gamma = \{C_1, \ldots, C_n\}$ of concepts, by $\bigsqcap \Gamma$ we denote $C_1 \sqcap \ldots \sqcap C_n$, and by $\bigsqcup \Gamma$ we denote $C_1 \sqcup \ldots \sqcup C_n$. If $\Gamma = \emptyset$, then $\bigsqcap \Gamma = 1$ and $\bigsqcup \Gamma = 0$.

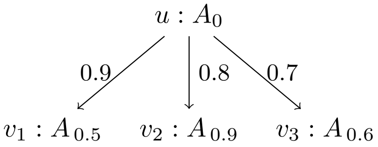
Definition 1. A *(fuzzy) interpretation* is a pair $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$, where $\Delta^\mathcal{I}$ is a non-empty set, called the *domain*, and $\cdot^\mathcal{I}$ is the *interpretation function*, which maps every individual name a to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$, every concept name A to a function $A^\mathcal{I} : \Delta^\mathcal{I} \to [0, 1]$, and every role name r to a function $r^\mathcal{I} : \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to [0, 1]$. The function $\cdot^\mathcal{I}$ is extended to complex roles and concepts as follows (cf. [2]):

$$\begin{aligned}
U^{\mathcal{I}}(x,y) &= 1\\
(r^-)^{\mathcal{I}}(x,y) &= r^{\mathcal{I}}(y,x)\\
(C?)^{\mathcal{I}}(x,y) &= (\text{if } x = y \text{ then } C^{\mathcal{I}}(x) \text{ else } 0)\\
(R \circ S)^{\mathcal{I}}(x,y) &= \oplus\{R^{\mathcal{I}}(x,z) \otimes S^{\mathcal{I}}(z,y) \mid z \in \Delta^{\mathcal{I}}\}\\
(R \sqcup S)^{\mathcal{I}}(x,y) &= R^{\mathcal{I}}(x,y) \oplus S^{\mathcal{I}}(x,y)\\
(R^*)^{\mathcal{I}}(x,y) &= \oplus\{\otimes\{R^{\mathcal{I}}(x_i,x_{i+1}) \mid 0 \le i < n\} \mid\\
&\qquad n \ge 0,\ x_0,\ldots,x_n \in \Delta^{\mathcal{I}},\ x_0 = x,\ x_n = y\}\\
p^{\mathcal{I}}(x) &= p\\
\{a\}^{\mathcal{I}}(x) &= (\text{if } x = a^{\mathcal{I}} \text{ then } 1 \text{ else } 0)\\
(\neg C)^{\mathcal{I}}(x) &= \ominus C^{\mathcal{I}}(x)\\
(C \sqcap D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)\\
(C \sqcup D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)\\
(C \to D)^{\mathcal{I}}(x) &= (C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x))\\
(\exists r.\mathtt{Self})^{\mathcal{I}}(x) &= r^{\mathcal{I}}(x,x)\\
(\exists R.C)^{\mathcal{I}}(x) &= \oplus\{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\}\\
(\forall R.C)^{\mathcal{I}}(x) &= \otimes\{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\}\\
(\ge n\, R.C)^{\mathcal{I}}(x) &= \oplus\{\otimes\{R^{\mathcal{I}}(x,y_i) \otimes C^{\mathcal{I}}(y_i) \mid 1 \le i \le n\} \mid\\
&\qquad y_1,\ldots,y_n \in \Delta^{\mathcal{I}},\ y_i \ne y_j \text{ if } i \ne j\}\\
(\le n\, R.C)^{\mathcal{I}}(x) &= \otimes\{(\otimes\{R^{\mathcal{I}}(x,y_i) \otimes C^{\mathcal{I}}(y_i) \mid 1 \le i \le n+1\} \Rightarrow\\
&\qquad \oplus\{y_j \ne y_k \mid 1 \le j < k \le n+1\}) \mid y_1,\ldots,y_{n+1} \in \Delta^{\mathcal{I}}\}. \blacksquare
\end{aligned}$$

For definitions of the Zadeh, Łukasiewicz and Product semantics for fuzzy DLs, we refer the reader to [2].

Remark 1. Observe that $(\le nR.C)^{\mathcal{I}}(x)$ is either 1 or 0. Namely, $(\le n\, R.C)^{\mathcal{I}}(x) = 1$ if, for every set $\{y_1,\ldots,y_{n+1}\}$ of $n+1$ pairwise distinct elements of $\Delta^{\mathcal{I}}$, there exists $1 \le i \le n+1$ such that $R^{\mathcal{I}}(x,y_i) \otimes C^{\mathcal{I}}(y_i) = 0$. Otherwise, $(\le n\, R.C)^{\mathcal{I}}(x) = 0$. $\blacksquare$

Example 1. Let $\mathbf{R} = \{r\}$, $\mathbf{C} = \{A\}$ and $\mathbf{I} = \emptyset$. Consider the fuzzy interpretation $\mathcal{I}$ illustrated and specified below:



- $\Delta^{\mathcal{I}} = \{u, v_1, v_2, v_3\}$,
- $A^{\mathcal{I}}(u) = 0$, $A^{\mathcal{I}}(v_1) = 0.5$, $A^{\mathcal{I}}(v_2) = 0.9$, $A^{\mathcal{I}}(v_3) = 0.6$,
- $r^{\mathcal{I}}(u, v_1) = 0.9$, $r^{\mathcal{I}}(u, v_2) = 0.8$, $r^{\mathcal{I}}(u, v_3) = 0.7$, and $r^{\mathcal{I}}(x, y) = 0$ for other pairs $\langle x, y\rangle$.

We have that:

- $(\forall r.A)^{\mathcal{I}}(a) = 0.5$, $(\exists r.A)^{\mathcal{I}}(a) = 0.8$, $(\leq 1\, r.A)^{\mathcal{I}}(a) = 0$, $(\geq 2\, r.A)^{\mathcal{I}}(a) = 0.6$,
- for $C = \forall(r \sqcup r^-)^*.A$ and $1 \leq i \leq 3$: $C^{\mathcal{I}}(v_i) = 0$,
- for $C = \exists(r \sqcup r^-)^*.A$: $C^{\mathcal{I}}(v_1) = 0.8$, $C^{\mathcal{I}}(v_2) = 0.9$ and $C^{\mathcal{I}}(v_3) = 0.7$. ■

A fuzzy interpretation $\mathcal{I}$ is *witnessed* (w.r.t. $\mathcal{L}_\Phi$) [11] if any infinite set under the prefix operator $\otimes$ (resp. $\oplus$) in Definition 1 has the smallest (resp. biggest) element. The notion of being "*witnessed w.r.t.* $\mathcal{L}_\Phi^0$" is defined similarly under the assumption that only roles and concepts of $\mathcal{L}_\Phi^0$ are allowed. A fuzzy interpretation $\mathcal{I}$ is *finite* if $\Delta^{\mathcal{I}}$, $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{I}$ are finite, and is *image-finite* w.r.t. Φ if, for every $x \in \Delta^{\mathcal{I}}$ and every basic role R of $\mathcal{L}_\Phi$, $\{y \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(x, y) > 0\}$ is finite. Observe that every finite fuzzy interpretation is witnessed and every image-finite fuzzy interpretation w.r.t. Φ is witnessed w.r.t. $\mathcal{L}_\Phi^0$.

A *fuzzy assertion* in $\mathcal{L}_\Phi$ is an expression of the form $a \doteq b$, $a \not\doteq b$, $C(a) \bowtie p$ or $R(a, b) \bowtie p$, where C is a concept of $\mathcal{L}_\Phi$, R is a role of $\mathcal{L}_\Phi$, $\bowtie \in \{\geq, >, \leq, <\}$ and $p \in [0, 1]$. A *fuzzy ABox* in $\mathcal{L}_\Phi$ is a finite set of fuzzy assertions in $\mathcal{L}_\Phi$.

A *fuzzy GCI* (general concept inclusion) in $\mathcal{L}_\Phi$ is an expression of the form $(C \sqsubseteq D) \rhd p$, where C and D are concepts of $\mathcal{L}_\Phi$, $\rhd \in \{\geq, >\}$ and $p \in (0, 1]$. A *fuzzy TBox* in $\mathcal{L}_\Phi$ is a finite set of fuzzy GCIs in $\mathcal{L}_\Phi$.

Given a fuzzy interpretation $\mathcal{I}$ and a fuzzy assertion or GCI φ, we say that $\mathcal{I}$ *validates* φ, denoted by $\mathcal{I} \models \varphi$, if:

- case $\varphi = (a \doteq b)$: $a^{\mathcal{I}} = b^{\mathcal{I}}$,
- case $\varphi = (a \not\doteq b)$: $a^{\mathcal{I}} \neq b^{\mathcal{I}}$,
- case $\varphi = (C(a) \bowtie p)$: $C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie p$,
- case $\varphi = (R(a, b) \bowtie p)$: $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie p$,
- case $\varphi = (C \sqsubseteq D) \rhd p$: $(C \to D)^{\mathcal{I}}(x) \rhd p$ for all $x \in \Delta^{\mathcal{I}}$.

A fuzzy interpretation $\mathcal{I}$ is a *model* of a fuzzy ABox $\mathcal{A}$, denoted by $\mathcal{I} \models \mathcal{A}$, if $\mathcal{I} \models \varphi$ for all $\varphi \in \mathcal{A}$. Similarly, $\mathcal{I}$ is a model of a fuzzy TBox $\mathcal{T}$, denoted by $\mathcal{I} \models \mathcal{T}$, if $\mathcal{I} \models \varphi$ for all $\varphi \in \mathcal{T}$.

Two concepts C and D are *equivalent*, denoted by $C \equiv D$, if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every fuzzy interpretation $\mathcal{I}$. Two roles R and S are *equivalent*, denoted by $R \equiv S$, if $R^{\mathcal{I}} = S^{\mathcal{I}}$ for every fuzzy interpretation $\mathcal{I}$.

We say that a role R is in *inverse normal form* if inverse constructor is applied in R only to role names. In this paper, we assume that roles are presented in inverse normal form because every role can be translated to an equivalent role in inverse normal form using the following rules:

$$\begin{array}{rr} U^- \equiv U & (R \circ S)^- \equiv S^- \circ R^- \\ (R^-)^- \equiv R & (R \sqcup S)^- \equiv R^- \sqcup S^- \\ (C?)^- \equiv C? & (R^*)^- \equiv (R^-)^*. \end{array}$$

Remark 2. The concept constructors $\neg C$ and $C \sqcup D$ can be excluded from $\mathcal{L}_\Phi$ and $\mathcal{L}^0_\Phi$ because

$$\begin{aligned}\neg C &\equiv (C \to 0)\\ C \sqcup D &\equiv ((C \to D) \to D) \sqcap ((D \to C) \to C).\end{aligned} \qquad \blacksquare$$

3 Fuzzy Bisimulations

Let $\Phi \subseteq \{I, O, Q, U, \mathtt{Self}\}$ be a set of features and $\mathcal{I}$, $\mathcal{I}'$ fuzzy interpretations. A function $Z : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'} \to [0,1]$ is called a *fuzzy $\mathcal{L}_\Phi$-bisimulation* (under the Gödel semantics) between $\mathcal{I}$ and $\mathcal{I}'$ if the following conditions hold for every $x \in \Delta^{\mathcal{I}}$, $x' \in \Delta^{\mathcal{I}'}$, $A \in \mathbf{C}$, $a \in \mathbf{I}$, $r \in \mathbf{R}$ and every basic role R of $\mathcal{L}_\Phi$:

$$Z(x,x') \leq (A^{\mathcal{I}}(x) \Leftrightarrow A^{\mathcal{I}'}(x')) \tag{1}$$

$$\forall y \in \Delta^{\mathcal{I}}\ \exists y' \in \Delta^{\mathcal{I}'}\ Z(x,x') \otimes R^{\mathcal{I}}(x,y) \leq Z(y,y') \otimes R^{\mathcal{I}'}(x',y') \tag{2}$$

$$\forall y' \in \Delta^{\mathcal{I}'}\ \exists y \in \Delta^{\mathcal{I}}\ Z(x,x') \otimes R^{\mathcal{I}'}(x',y') \leq Z(y,y') \otimes R^{\mathcal{I}}(x,y); \tag{3}$$

if $O \in \Phi$, then

$$Z(x,x') \leq (x = a^{\mathcal{I}} \Leftrightarrow x' = a^{\mathcal{I}'}); \tag{4}$$

if $Q \in \Phi$, then, for any $n \geq 1$,

$$\begin{array}{l}\text{if } Z(x,x') > 0 \text{ and } y_1,\ldots,y_n \text{ are pairwise distinct elements of } \Delta^{\mathcal{I}} \text{ such}\\ \text{that } R^{\mathcal{I}}(x,y_j) > 0 \text{ for all } 1 \leq j \leq n, \text{ then there exist pairwise distinct}\\ \text{elements } y'_1,\ldots,y'_n \text{ of } \Delta^{\mathcal{I}'} \text{ such that, for every } 1 \leq i \leq n, \text{ there exists}\\ 1 \leq j \leq n \text{ such that } Z(x,x') \otimes R^{\mathcal{I}}(x,y_j) \leq Z(y_j,y'_i) \otimes R^{\mathcal{I}'}(x',y'_i),\end{array} \tag{5}$$

$$\begin{array}{l}\text{if } Z(x,x') > 0 \text{ and } y'_1,\ldots,y'_n \text{ are pairwise distinct elements of } \Delta^{\mathcal{I}'}\\ \text{such that } R^{\mathcal{I}'}(x',y'_j) > 0 \text{ for all } 1 \leq j \leq n, \text{ then there exist pairwise}\\ \text{distinct elements } y_1,\ldots,y_n \text{ of } \Delta^{\mathcal{I}} \text{ such that, for every } 1 \leq i \leq n, \text{ there}\\ \text{exists } 1 \leq j \leq n \text{ such that } Z(x,x') \otimes R^{\mathcal{I}'}(x',y'_j) \leq Z(y_i,y'_j) \otimes R^{\mathcal{I}}(x,y_i);\end{array} \tag{6}$$

if $U \in \Phi$, then

$$\forall y \in \Delta^{\mathcal{I}}\ \exists y' \in \Delta^{\mathcal{I}'}\ Z(x,x') \leq Z(y,y') \tag{7}$$

$$\forall y' \in \Delta^{\mathcal{I}'}\ \exists y \in \Delta^{\mathcal{I}}\ Z(x,x') \leq Z(y,y'); \tag{8}$$

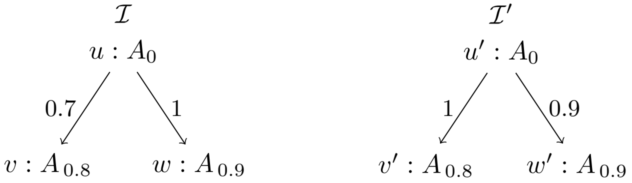
if $\mathtt{Self} \in \Phi$, then

$$Z(x,x') \leq (r^{\mathcal{I}}(x,x) \Leftrightarrow r^{\mathcal{I}'}(x',x')). \tag{9}$$

For example, if $\Phi = \{I, Q\}$, then only Conditions (1)–(3), (5) and (6) are essential. By definition, the function $\lambda\langle x,x'\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'}.0$ is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$.

Remark 3. Observe that Condition (2) (resp. (3)) together with the qualification over x and x' implies $Z^{-1} \circ R^{\mathcal{I}} \leq R^{\mathcal{I}'} \circ Z^{-1}$ (resp. $Z \circ R^{\mathcal{I}'} \leq R^{\mathcal{I}} \circ Z$). However, in general, the converse does not hold. ■

Example 2. Let $\mathbf{R} = \{r\}$, $\mathbf{C} = \{A\}$, $\mathbf{I} = \emptyset$ and $\Phi = \emptyset$. Consider the fuzzy interpretations $\mathcal{I}$ and $\mathcal{I}'$ illustrated below (and specified similarly as in Example 1).



If Z is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$, then:

- $Z(v, w') \leq 0.8$ and $Z(w, v') \leq 0.8$ due to (1),
- $Z(u, u') \leq 0.8$ due to (3) for $x = u$, $x' = u'$ and $y' = v'$,
- $Z(u, v') = Z(u, w') = Z(v, u') = Z(w, u') = 0$ due to (1).

It can be check that the function $Z : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'} \to [0, 1]$ specified by

- $Z(v, v') = Z(w, w') = 1$,
- $Z(v, w') = Z(w, v') = Z(u, u') = 0.8$,
- $Z(u, v') = Z(u, w') = Z(v, u') = Z(w, u') = 0$

is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$, and hence is the greatest fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$. ■

Proposition 1. *Let $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$ be fuzzy interpretations.*

1. *The function $Z : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$ specified by*

$$Z(x, x') = (\textit{if } x = x' \textit{ then } 1 \textit{ else } 0)$$

   is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and itself.
2. *If Z is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$, then Z^{-1} is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}'$ and $\mathcal{I}$.*
3. *If Z_1 is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$, and Z_2 is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}'$ and $\mathcal{I}''$, then $Z_1 \circ Z_2$ is a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}''$.*
4. *If $\mathcal{Z}$ is a finite set of fuzzy $\mathcal{L}_\Phi$-bisimulations between $\mathcal{I}$ and $\mathcal{I}'$, then $\oplus\mathcal{Z}$ is also a fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$.*

The proof of this proposition is straightforward.

Remark 4. It seems that the assertion 4 of Proposition 1 cannot be strengthened for infinite $\mathcal{Z}$. So, the greatest fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$ may not exist. As stated later by Theorem 4, if $\mathcal{I}$ and $\mathcal{I}'$ are witnessed w.r.t. $\mathcal{L}_\Phi^0$ and modally saturated w.r.t. $\mathcal{L}_\Phi^0$ (see Definition 2), then the greatest fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$ exists. ■

Let $\mathcal{I}$ and $\mathcal{I}'$ be fuzzy interpretations. For $x \in \Delta^{\mathcal{I}}$ and $x' \in \Delta^{\mathcal{I}'}$, we write $x \sim_{\Phi} x'$ to denote that there exists a fuzzy $\mathcal{L}_{\Phi}$-bisimulation Z between $\mathcal{I}$ and $\mathcal{I}'$ such that $Z(x, x') = 1$. If $x \sim_{\Phi} x'$, then we say that x and x' are *$\mathcal{L}_{\Phi}$-bisimilar* to each other. Let $\sim_{\Phi,\mathcal{I}}$ be the binary relation on $\Delta^{\mathcal{I}}$ such that, for $x, x' \in \Delta^{\mathcal{I}}$, $x \sim_{\Phi,\mathcal{I}} x'$ iff $x \sim_{\Phi} x'$. By Proposition 1, $\sim_{\Phi,\mathcal{I}}$ is an equivalence relation. We call it the *$\mathcal{L}_{\Phi}$-bisimilarity* relation of $\mathcal{I}$. If $\mathbf{I} \neq \emptyset$ and there exists a fuzzy $\mathcal{L}_{\Phi}$-bisimulation Z between $\mathcal{I}$ and $\mathcal{I}'$ such that $Z(a^{\mathcal{I}}, a^{\mathcal{I}'}) = 1$ for all $a \in \mathbf{I}$, then we say that $\mathcal{I}$ and $\mathcal{I}'$ are *$\mathcal{L}_{\Phi}$-bisimilar* to each other and write $\mathcal{I} \sim_{\Phi} \mathcal{I}'$.

4 Invariance Results

A concept C of $\mathcal{L}_{\Phi}$ is said to be *invariant for $\mathcal{L}_{\Phi}$-bisimilarity between witnessed interpretations* if, for any witnessed interpretations $\mathcal{I}$, $\mathcal{I}'$ and any $x \in \Delta^{\mathcal{I}}$ and $x' \in \Delta^{\mathcal{I}'}$, if $x \sim_{\Phi} x'$, then $C^{\mathcal{I}}(x) = C^{\mathcal{I}'}(x')$.

Theorem 1. *All concepts of $\mathcal{L}_{\Phi}$ are invariant for $\mathcal{L}_{\Phi}$-bisimilarity between witnessed interpretations.*

This theorem is a corollary of the following stronger result.

Lemma 1. *Let $\mathcal{I}$ and $\mathcal{I}'$ be witnessed interpretations and Z a fuzzy $\mathcal{L}_{\Phi}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$. Then, the following properties hold for every concept C of $\mathcal{L}_{\Phi}$, every role R of $\mathcal{L}_{\Phi}$, every $x \in \Delta^{\mathcal{I}}$ and every $x' \in \Delta^{\mathcal{I}'}$:*

$$Z(x, x') \leq (C^{\mathcal{I}}(x) \Leftrightarrow C^{\mathcal{I}'}(x')) \tag{10}$$

$$\forall y \in \Delta^{\mathcal{I}}\ \exists y' \in \Delta^{\mathcal{I}'}\ Z(x, x') \otimes R^{\mathcal{I}}(x, y) \leq Z(y, y') \otimes R^{\mathcal{I}'}(x', y') \tag{11}$$

$$\forall y' \in \Delta^{\mathcal{I}'}\ \exists y \in \Delta^{\mathcal{I}}\ Z(x, x') \otimes R^{\mathcal{I}'}(x', y') \leq Z(y, y') \otimes R^{\mathcal{I}}(x, y). \tag{12}$$

The following lemma differs from Lemma 1 in that $\mathcal{L}_{\Phi}^{0}$ is used instead of $\mathcal{L}_{\Phi}$. Its proof is a shortened version the one of Lemma 1, as (11) and (12) are the same as (2) and (3) when R is a role of $\mathcal{L}_{\Phi}^{0}$, respectively, and we can ignore the cases when C is $\forall R.D$ or $\leq n\, R.D$.

Lemma 2. *Let $\mathcal{I}$ and $\mathcal{I}'$ be witnessed interpretations w.r.t. $\mathcal{L}_{\Phi}^{0}$ and Z a fuzzy $\mathcal{L}_{\Phi}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$. Then, for every concept C of $\mathcal{L}_{\Phi}^{0}$, every $x \in \Delta^{\mathcal{I}}$ and every $x' \in \Delta^{\mathcal{I}'}$, $Z(x, x') \leq (C^{\mathcal{I}}(x) \Leftrightarrow C^{\mathcal{I}'}(x'))$.*

A fuzzy TBox $\mathcal{T}$ is said to be *invariant for $\mathcal{L}_{\Phi}$-bisimilarity between witnessed interpretations* if, for every witnessed interpretations $\mathcal{I}$ and $\mathcal{I}'$ that are $\mathcal{L}_{\Phi}$-bisimilar to each other, $\mathcal{I} \models \mathcal{T}$ iff $\mathcal{I}' \models \mathcal{T}$. The notion of invariance of fuzzy ABoxes for $\mathcal{L}_{\Phi}$-bisimilarity between witnessed interpretations is defined similarly.

Theorem 2. *If $U \in \Phi$ and $\mathbf{I} \neq \emptyset$, then all fuzzy TBoxes in $\mathcal{L}_{\Phi}$ are invariant for $\mathcal{L}_{\Phi}$-bisimilarity between witnessed interpretations.*

Theorem 3. *Let $\mathcal{A}$ be a fuzzy ABox in $\mathcal{L}_{\Phi}$. If $O \in \Phi$ or $\mathcal{A}$ consists of only fuzzy assertions of the form $C(a) \bowtie p$, then $\mathcal{A}$ is invariant for $\mathcal{L}_{\Phi}$-bisimilarity between witnessed interpretations.*

5 The Hennessy-Milner Property

Definition 2. A fuzzy interpretation $\mathcal{I}$ is said to be *modally saturated* w.r.t. $\mathcal{L}_\Phi^0$ (and the Gödel semantics) if the following conditions hold:

- for every $p \in (0, 1]$, every $x \in \Delta^\mathcal{I}$, every basic role R of $\mathcal{L}_\Phi$ and every infinite set Γ of concepts in $\mathcal{L}_\Phi^0$, if for every finite subset Λ of Γ there exists $y \in \Delta^\mathcal{I}$ such that $R^\mathcal{I}(x, y) \otimes C^\mathcal{I}(y) \geq p$ for all $C \in \Lambda$, then there exists $y \in \Delta^\mathcal{I}$ such that $R^\mathcal{I}(x, y) \otimes C^\mathcal{I}(y) \geq p$ for all $C \in \Gamma$;
- if $Q \in \Phi$, then for every $p \in (0, 1]$, every $x \in \Delta^\mathcal{I}$, every basic role R of $\mathcal{L}_\Phi$, every infinite set Γ of concepts in $\mathcal{L}_\Phi^0$ and every $n \in \mathbb{N}$, if for every finite subset Λ of Γ there exist n pairwise distinct $y_1, \ldots, y_n \in \Delta^\mathcal{I}$ such that $R^\mathcal{I}(x, y_i) \otimes C^\mathcal{I}(y_i) \geq p$ for all $1 \leq i \leq n$ and $C \in \Lambda$, then there exist n pairwise distinct $y_1, \ldots, y_n \in \Delta^\mathcal{I}$ such that $R^\mathcal{I}(x, y_i) \otimes C^\mathcal{I}(y_i) \geq p$ for all $1 \leq i \leq n$ and $C \in \Gamma$;
- if $U \in \Phi$, then for every $p \in (0, 1]$ and every infinite set Γ of concepts in $\mathcal{L}_\Phi^0$, if for every finite subset Λ of Γ there exists $y \in \Delta^\mathcal{I}$ such that $C^\mathcal{I}(y) \geq p$ for all $C \in \Lambda$, then there exists $y \in \Delta^\mathcal{I}$ such that $C^\mathcal{I}(y) \geq p$ for all $C \in \Gamma$. ■

Clearly, every finite fuzzy interpretation is modally saturated w.r.t. $\mathcal{L}_\Phi^0$ for any Φ. If $U \notin \Phi$, then every image-finite fuzzy interpretation w.r.t. Φ is modally saturated w.r.t. $\mathcal{L}_\Phi^0$.

Theorem 4. *Let $\mathcal{I}$ and $\mathcal{I}'$ be fuzzy interpretations that are witnessed w.r.t. $\mathcal{L}_\Phi^0$ and modally saturated w.r.t. $\mathcal{L}_\Phi^0$. Let $Z : \Delta^\mathcal{I} \times \Delta^{\mathcal{I}'} \to [0, 1]$ be specified by*

$$Z(x, x') = \otimes\{C^\mathcal{I}(x) \Leftrightarrow C^{\mathcal{I}'}(x) \mid C \textit{ is a concept of } \mathcal{L}_\Phi^0\}.$$

Then, Z is the greatest fuzzy $\mathcal{L}_\Phi$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$.

Given fuzzy interpretations $\mathcal{I}$, $\mathcal{I}'$ and $x \in \Delta^\mathcal{I}$, $x' \in \Delta^{\mathcal{I}'}$, we write $x \equiv_\Phi x'$ to denote that $C^\mathcal{I}(x) = C^{\mathcal{I}'}(x')$ for every concept C of $\mathcal{L}_\Phi$. Similarly, we write $x \equiv_\Phi^0 x'$ to denote that $C^\mathcal{I}(x) = C^{\mathcal{I}'}(x')$ for every concept C of $\mathcal{L}_\Phi^0$.

Corollary 1. *Let $\mathcal{I}$, $\mathcal{I}'$ be fuzzy interpretations and let $x \in \Delta^\mathcal{I}$, $x' \in \Delta^{\mathcal{I}'}$.*

1. *If $\mathcal{I}$ and $\mathcal{I}'$ are witnessed w.r.t. $\mathcal{L}_\Phi^0$ and modally saturated w.r.t. $\mathcal{L}_\Phi^0$, then*
$$x \sim_\Phi x' \quad \textit{iff} \quad x \equiv_\Phi^0 x'.$$
2. *If $\mathcal{I}$ and $\mathcal{I}'$ are image-finite fuzzy interpretations w.r.t. Φ, then*
$$x \sim_\Phi x' \quad \textit{iff} \quad x \equiv_\Phi^0 x'.$$
3. *If $\mathcal{I}$ and $\mathcal{I}'$ are witnessed w.r.t. $\mathcal{L}_\Phi$ and modally saturated w.r.t. $\mathcal{L}_\Phi^0$, then*
$$x \equiv_\Phi x' \quad \textit{iff} \quad x \sim_\Phi x' \quad \textit{iff} \quad x \equiv_\Phi^0 x'.$$

The assertion 1 (resp. 3) directly follows from Theorem 4 and Lemma 2 (resp. 1). The assertion 2 directly follows from the assertion 1. The following corollary directly follows from Theorem 4 and Lemma 1.

Corollary 2. *Let $\mathcal{I}$ and $\mathcal{I}'$ be fuzzy interpretations that are witnessed w.r.t. $\mathcal{L}_\Phi$ and modally saturated w.r.t. $\mathcal{L}_\Phi^0$. Then, $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{L}_\Phi$-bisimilar iff $a^\mathcal{I} \equiv_\Phi^0 a^{\mathcal{I}'}$ for all $a \in \mathbf{I}$.*

6 Concluding Remarks

We have defined fuzzy bisimulations and (crisp) bisimilarity relations for a large class of fuzzy DLs under the Gödel semantics. We have provided results on invariance of concepts as well as conditional invariance of TBoxes and ABoxes for such bisimilarity. We have also provided results on the Hennessy-Milner property for such bisimulations. As far as we know, this is the first time fuzzy bisimulations are defined and studied for fuzzy DLs under the Gödel semantics.

As mentioned in the Introduction, we use "elementary" Conditions (2), (3) and (5)–(8) instead of the ones based on relational composition for defining bisimulations. Consequently, our notion of fuzzy bisimulation is different in nature from the one introduced by Fan [9], although the greatest fuzzy bisimulation relations specified by these two different approaches coincide when restricting to the fuzzy modal logics without involutive negation considered in [9]. Furthermore, in comparison with [9], not only is the class of logics considered by us much larger, we also study invariance of TBoxes and ABoxes for bisimilarity, and our theorem on the Hennessy-Milner property is formulated for witnessed and modally saturated interpretations, which are more general than image-finite interpretations.

Like the relationship between [9] and [8], our notion of fuzzy bisimulation is also related to the notion of weak bisimulation introduced by Eleftheriou et al. [8] for Heyting-valued modal logics, especially for the case when the considered logic is K and the underlying Heyting algebra is the complete lattice $\langle [0, 1], \leq \rangle$. In this case, the latter notion can be treated as a cut-based variant of our notion (see [9] for a more detailed discussion). The differences are that the considered classes of logics are essentially different and our approach uses fuzzy relations as in [5,9], while the approach of [8] uses families of crisp relations, where each of the families is specified by a cut-value. Following [9], we use the term "fuzzy bisimulation" instead of"bisimulation" to emphasize its fuzziness.

Our notions and results have potential applications to concept learning in fuzzy DLs. As future work, apart from such applications, it is also worth studying bisimulation and bisimilarity in fuzzy DLs under other t-norm based semantics (e.g., Łukasiewicz and Product). Recently, Nguyen [13] studied bisimilarity in fuzzy DLs under the Zadeh semantics, which does not use t-norms for defining implication. His approach is essentially different, as it uses (crisp) simulation instead of (fuzzy) bisimulation because the latter notion does not seem to be definable for fuzzy DLs under the Zadeh semantics.

Acknowlegements. This paper was partially supported by VNU-UET and VNU.

References

1. Bobillo, F., Cerami, M., Esteva, F., García-Cerdaña, Á., Peñaloza, R., Straccia, U.: Fuzzy description logics. In: Handbook of Mathematical Fuzzy Logic, Volume 58 of Studies in Logic, Mathematical Logic and Foundations, vol. 3, pp. 1105–1181. College Publications (2015)
2. Bobillo, F., Delgado, M., Gómez-Romero, J., Straccia, U.: Fuzzy description logics under Gödel semantics. Int. J. Approximate Reasoning **50**(3), 494–514 (2009)
3. Borgwardt, S., Peñaloza, R.: Fuzzy description logics – a survey. In: Moral, S., Pivert, O., Sánchez, D., Marín, N. (eds.) SUM 2017. LNCS (LNAI), vol. 10564, pp. 31–45. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67582-4_3
4. Cattaneo, G., Ciucci, D., Giuntini, R., Konig, M.: Algebraic structures related to many valued logical systems. Part I: Heyting Wajsberg algebras. Fundamenta Informaticae **63**(4), 331–355 (2004)
5. Ćirić, M., Ignjatović, J., Damljanović, N., Bašic, M.: Bisimulations for fuzzy automata. Fuzzy Sets Syst. **186**(1), 100–139 (2012)
6. Divroodi, A.R., Ha, Q.-T., Nguyen, L.A., Nguyen, H.S.: On the possibility of correct concept learning in description logics. Vietnam J. Comput. Sci. **5**(1), 3–14 (2018)
7. Divroodi, A.R., Nguyen, L.A.: On bisimulations for description logics. Inf. Sci. **295**, 465–493 (2015)
8. Eleftheriou, P.E., Koutras, C.D., Nomikos, C.: Notions of bisimulation for Heyting-valued modal languages. J. Log. Comput. **22**(2), 213–235 (2012)
9. Fan, T.-F.: Fuzzy bisimulation for Gödel modal logic. IEEE Trans. Fuzzy Syst. **23**(6), 2387–2396 (2015)
10. Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L.A., Nguyen, H.S., Szałas, A., Tran, T.-L.: A bisimulation-based method of concept learning for knowledge bases in description logics. In: Proceedings of SoICT 2012, pp. 241–249. ACM (2012)
11. Hájek, P.: Making fuzzy description logic more general. Fuzzy Sets Syst. **154**(1), 1–15 (2005)
12. Lutz, C., Piro, R., Wolter, F.: Description logic TBoxes: model-theoretic characterizations and rewritability. In: Walsh, T. (ed.) Proceedings of IJCAI 2011, pp. 983–988 (2011)
13. Nguyen, L.A.: Bisimilarity in fuzzy description logics under the Zadeh semantics, submitted
14. Nguyen, L.A., Nguyen, T.H.K., Nguyen, N.-T., Ha, Q.-T.: Bisimilarity for paraconsistent description logics. J. Intell. Fuzzy Syst. **32**(2), 1203–1215 (2017)
15. Nguyen, L.A., Szałas, A.: Logic-based roughification. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems (To the Memory of Professor Zdzisław Pawlak), vol. 1, pp. 517–543. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30344-9_19
16. Schild, K.: A correspondence theory for terminological logics: preliminary report. In: Proceedings of IJCAI 1991, pp. 466–471. Morgan Kaufmann (1991)
17. Tran, T.-L., Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L.A., Nguyen, H.S.: Bisimulation-based concept learning in description logics. Fundamenta Informaticae **133**(2–3), 287–303 (2014)
18. Tran, T.-L., Nguyen, L.A., Hoang, T.-L.-G.: Bisimulation-based concept learning for information systems in description logics. Vietnam J. Comput. Sci. **2**(3), 149–167 (2015)

A New Trace Clustering Algorithm Based on Context in Process Mining

Hong-Nhung Bui[1,2](✉), Tri-Thanh Nguyen[1], Thi-Cham Nguyen[1,3],
and Quang-Thuy Ha[1]

[1] Vietnam National University, Hanoi (VNU), VNU-University of Engineering
and Technology (UET), 144, Xuan Thuy, Cau Giay, Hanoi, Vietnam
nhungbth@hvnh.edu.vn, {ntthanh, thuyhq}@vnu.edu.vn,
nthicham@hpmu.edu.vn
[2] Banking Academy of Vietnam, 12, Chua Boc, Dong Da, Hanoi, Vietnam
[3] Hai Phong University of Medicine and Pharmacy, 72A, Nguyen Binh Khiem,
Ngo Quyen, Haiphong, Vietnam

Abstract. In process mining, trace clustering is an important technique that attracts the attention of researchers to solve the large and complex volume of event logs. Traditional trace clustering often uses available data mining algorithms which do not exploit the characteristic of processes. In this study, we propose a new trace clustering algorithm, especially for the process mining, based on the using trace context. The proposed clustering algorithm can automatic detects the number of clusters, and it does not need a convergence iteration like traditional ones like K-means. The algorithm takes two loops over the input to generate the clusters, thus the complexity is greatly reduced. Experimental results show that our method also has good results when compared to traditional methods.

Keywords: Event log · Process mining · Trace context · Clustering algorithm

1 Introduction

Most today's modern information systems have collection of data that describes all the events of the user occur during the execution of the software system so-called event logs. Event logs play an important role in modern software systems, they record information about the system in real-time including a set of events that contain several information, e.g., *case id*, *event id*, *timestamp*, *activity*, etc., Table 1 introduces some examples about an event log. The events in the same *case* are ordered by *timestamp* and have the same "*case id*". These are valuable data for managers to analyze and evaluate the company's business processes.

Process mining includes three tasks process discovery, conformance checking and enhancement is the field that allows the use of the event log data for analysis and improvement of the processes.

© Springer Nature Switzerland AG 2018
H. S. Nguyen et al. (Eds.): IJCRS 2018, LNAI 11103, pp. 644–657, 2018.
https://doi.org/10.1007/978-3-319-99368-3_50

The target of process discovery is to generate a process model that captures all of the behaviors found in the event log [23]. The generated model can be used to analyze what is actually applied in daily activities of the company. It can be used to verify whether the formal process is strictly followed, or to enhance the formal process.

The event log quality is an important factor in process model generation. If the event log is homogeneous and small enough, the process model is easy to analyze as one example in Fig. 1a. However, real-life event logs are extremely huge with diverse characteristics, thus, the discovered process model may be diffuse and very hard to understand as an example in Fig. 1b. To overcome this problem, clustering a complex event log into sub-logs/clusters is one of the most widely used solution. The generated model from an event sub-log will have much lower complexity [5, 7, 9–11, 15–18, 21].

Table 1. A fragment of the event log [23]

Case id	Event id	Properties				
		Timestamp	Activity	Resource	Cost	…
1	4423	30-12-2010:11.02	Register request	Pete	50	
	4424	31-12-2010:10.06	Examine thoroughly	Sue	400	
	4425	06-01-2011:15.12	Check ticket	Mike	100	
	4426	07-01-2011:11.18	Decide	Sara	200	
	4427	07-01-2011:14.24	Reject request	Pete	200	
2	4483	30-12-2010:11.32	Register request	Mike	50	
	4485	30-12-2010:12.12	Check ticket	Mike	100	
	4487	30-12-2010:14.16	Examine casually	Pete	400	
	4488	06-01-2011:11.22	Decide	Sara	200	
	4489	08-01-2011:12.06	Pay compensation	Ellen	200	
3	4521	30-12-2010:14.32	Register request	Pete	50	
	4522	30-12-2010:15.06	Examine casually	Mike	400	
	4524	30-12-2010:16.34	Check ticket	Ellen	100	
	4525	06-01-2011:09.18	Decide	Sara	200	
	4526	08-01-2011:12.18	Reinitie request	Sara	200	
	…					

Traditional approaches use the data mining clustering algorithms such as Agglomerative Hierarchical Clustering, K-Means, K-Modes, etc., to cluster event logs. These algorithms are designed and used in the field of data mining, they do not exploit the specific characteristics of business processes.

In this paper, we propose a new trace clustering algorithm based on a specific characteristic of process, i.e., the context of traces in a process. The contribution of the paper includes: (1) defining a new trace context; (2) introducing a context tree; (3) giving a new event log clustering algorithm. The proposed algorithm can automatically detect the suitable number of clusters, and it does not need a convergence iteration which takes lot of time. The experimental results show that our method has significant contributions to improving the efficiency and the performance time of the process discovery task.

The rest of this article is organized as follows: First, we give an overview of the process discovery. Section 3 introduces the trace context in process mining and the new trace clustering. The experimental evaluation is described in Sect. 4. Section 5 introduces the related work. Conclusions and future work are shown in the last section.

2 The Brief Summary of Process Discovery Task in Process Mining

Event Logs

An event log is the starting point of process mining. Table 1 shows a fragment of the event log related to the handling of compensation requests of an airline. There are three cases corresponds to three compensation requests. The case 1 has five events with *id* from 4423 to 4427 that are ordered by execution time, i.e., property timestamp. For example, event 4423 executes activity "register request" at "30-12-2010:11.02" occurs before event 4424 which executes activity "examine thoroughly" at "31-12-2010:10.06". Each event in event log also is described by *resources* property, i.e., the persons executing the activities or the cost of the activity.

In process mining, the "*case id*" and "*activity*" are minimum properties that can be used to represent a case. For example, *case* 1 is represented by a sequence of five activities Register request, Examine thoroughly, Check ticket, Decide, Reject request. Such a sequence of activities is called a *trace*. For the sake of simplicity for computation, each activity name is assigned by a distinct letter label, e.g., *a* denotes activity register request. Hence, the event log in Table 1 has a more compact representation shown in Table 2, e.g., *case*1 is represented by a trace $\langle a, b, d, e, h \rangle$. This representation is used for computation, such as clustering. For example, in K-means a trace is converted into a vector as the input to the algorithm.

Table 2. The trace in an event log (where *a* = "register request", *b* = "examine thoroughly", *c*= "examine casually", *d* = "check ticket", *e* = "decide", *f* = "reinitiate request", *g* = "pay compensation", *h* = "reject request")

Case id	Trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$
…	…

Process Discovery Task

Process discovery is the first task of process mining. It takes an event log as an input data and produces a model represented in a process modeling language, e.g., Petri net (Fig. 1), which describes the behaviors recorded in the event log by applying a process discovery algorithm, e.g., α-algorithm [23].

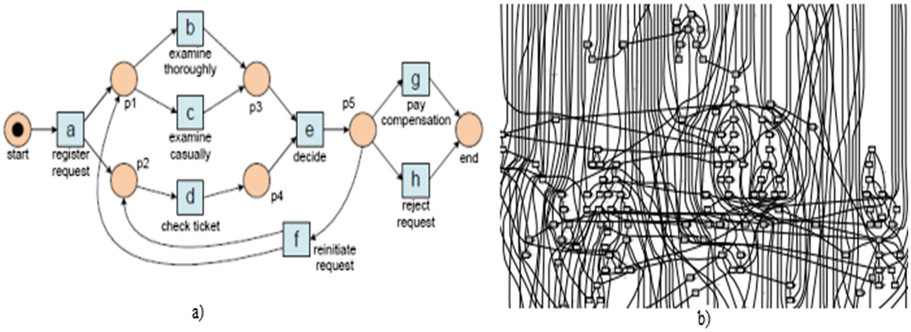


Fig. 1. The process model discovered from the event log by the α-algorithm

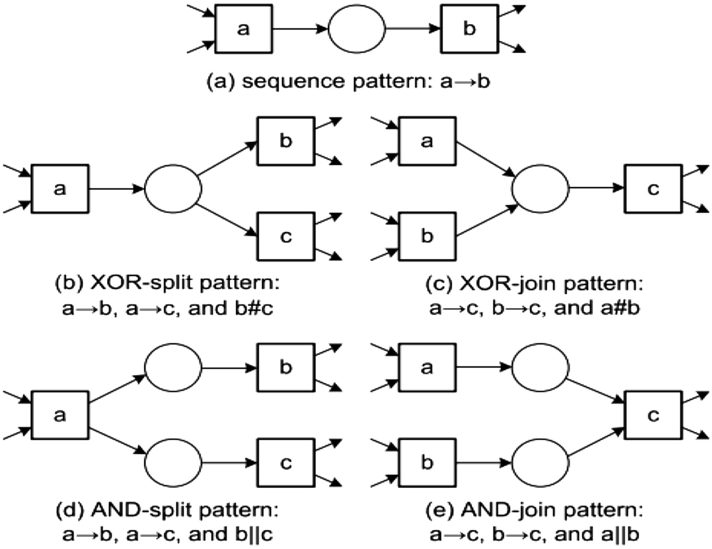


Fig. 2. Typical process patterns in Petri net [23]

α-Algorithm

The α-algorithm was one of the first process discovery algorithms. It generates the process model by reconstructing causality from a set of sequences of events in the event logs.

Given an event log of a business process L, α-algorithm scans L to find the relationships between activities based on the execution order. There are four ordering relations, e.g., *direct succession*, *causality*, *parallel*, *choice*. Let a, b are two activities in L.

1. Direct succession $a > b$: if some case a is followed by b.
2. Causality $a \rightarrow b$: if activity a is followed by b but b is never followed by a.
3. Parallel $a||b$: if activity a is followed by b and b is followed by a.
4. Choice $a\#b$: if activity a is never followed by b and b is never followed by a.

To reflect those dependencies, the Petri net has corresponding notations to connect activities as illustrated in Fig. 2.

As mentioned above, to mine easy-to-understand process models from the complex event log, the trace clustering is the effective approach. The key idea of trace clustering algorithms is to create clusters that the traces within a cluster are more similar to each other than the traces in the different clusters. Next section we introduce our proposed trace clustering algorithm.

3 A Context Approach to Trace Clustering

3.1 Context in Process Mining

In the middle of the 1990s, the context was mentioned by many researchers [2, 3, 14]. It had the important contribution to improving the performance of practical systems. Each different research fields usually have different ideas and definitions of context. It is defined as the object's location, environment, identity and execution time or object's emotional state as well as hobbies and habits of objects, etc. [12].

In process mining, the context was defined as the environment surrounding a business process, e.g., the weather conditions or holiday seasons [13]. In another study, the context was defined as the time, location, and frequency of events as well as related communication, tools, devices, or operators [22]. In [19], the context of activity a was the set of two surrounding activities xy, i.e., xay, by using 3-g in an event log.

3.2 Trace Context

In this paper we introduce a new context definition based on the fact that each business process has a number of different procedures. For example, the credit process has procedures for personal loan, corporate loan, home loan, consumer loan, etc. Each procedure may start with a set of *common activities* which are the clue to separate traces into different clusters. In this paper we define common activities as the trace contexts.

Definition 1. Let $L = \{t_0, t_{1,} \ldots\}$ be an event log, where t_i is a trace. Let p be the longest common prefix p of a trace subset, i.e., $SP = \{t \in L | t = p|d\}$, such that $|SP| > 1$, where d is a sequence of activities, notation '|' in $p|d$ denotes sequence concatenation operation, then p is called as a *trace context*.

3.3 Context Tree

Since the common prefix of traces can be represented by a prefix tree, to efficiently identify the context, we introduce a *Context-tree* based on the idea of *frequent pattern tree* (FP-tree) [8].

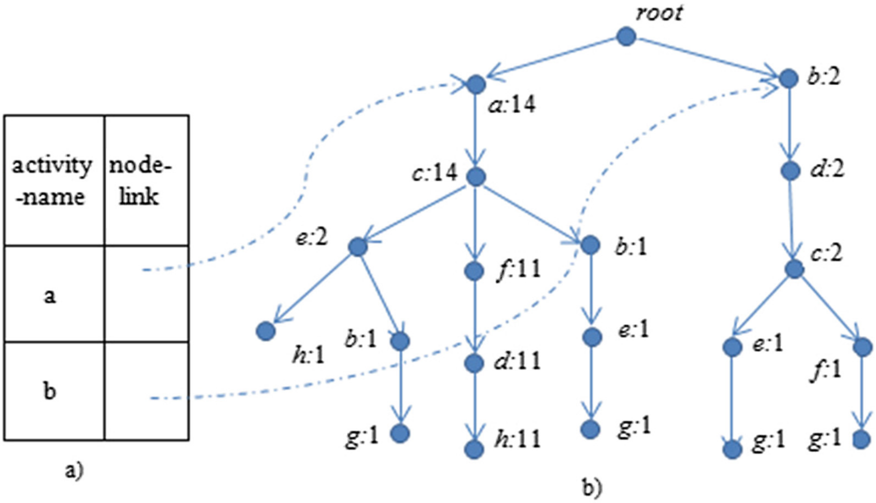


Fig. 3. (a) Header table; (b) The context-tree

Definition 2. A context tree is a tree that has:

1. One root labeled as "*root*" to form a complete tree.
2. A header table helps to access the tree faster during tree construction and traversal. Each entry in the Context-tree header table consists of two fields, (1) *activity-name*, and (2) head of *node-link* which points to the first node below the root carrying this activity.
3. Each node in the context tree consists of three fields except for the root node:

   activity-name: registers which activity is represented by the node;
   count: the number of traces that travel to this node;
   node-link: the pointers to its children, or null if there is none.

4. A trace in the event log is placed on a certain branch of the tree with the top- down fashion. Traces with the same prefix share a chunk of branch from the root node.

The idea is to map traces with the same prefix into the same chunk of tree branch as depicted in Fig. 3. The context tree construction procedure is described as follows:

Algorithm 1. ContextTreeConstruction.

```
Input:  An event log $L$
Output: A corresponding context tree $T$
1. Create a node of a Context-tree $T$ and label it as "root",
    i.e., the root node and $T$ = $root$.
2. Foreach trace $t$ in L do
      Let $t=ac|q$, where $ac$ is the first activity, and q is the
      rest of the activity sequence
      call insert_activity($ac|q$, $T$);
   EndFor
3. Create HeaderTable and update the node-link based on the direct children of the root node.
```

And the $insert\_activity(.)$ is defined as:

Algorithm 2. insert_activity($ac|q$, T)

```
Input: A trace in term of $ac|q$ where $ac$ is the first activity, and $q$ is the rest activities
  $T$ is a tree node
Output: $T$ is updated with new activities
1.  If $T$ has a child $N$ such that $N.activity\text{-}name=ac$ then
      Increase $N'$s count by 1
    Else
      Create a new node $N$, with its count = 1,
      Create a new node-link linked from $T$ to N.
    EndIf
2.  If $q$ is nonempty then
      call insert_activity($q$, $N$) recursively.
    EndIf
```

Let L = [<*aceh*>, <*acfdh*>^{10}, <*acebg*>, <*acbeg*>, <*bdceg*>, <*bdcfg*>] be an event log, which includes 15 traces, the trace $\langle acfdh \rangle$ appears 10 times. The corresponding context-tree is illustrated in Fig. 3.

Mapping the context tree with the Definition 1 it is clear that, for each trace on the tree, the longest common prefix is the sequence of activities that have *count* > 1. From the context-tree in Fig. 3, the set of trace contexts of L is $\{ace, acfdh, ac, bdc\}$.

If a trace is distinct from the others, then it has no context. The following procedure is responsible for identifying the context of a given trace.

Algorithm 3. ContextDetection(*ac*|*q*, *T, context*)

```
Input: A trace in term of $ac|q$ where $ac$ is the first activity, and $q$ is the rest activities
       $T$ is a context tree node
Output: The context of the trace
1.  If $T$ is root then
      context={};
      Get the node $N$ pointed by node-link from the
      HeaderTable of $T$ in the entry corresponding to $ac$;
    Else
      Find the child node $N$ of $T$ that has the label $ac$;
    EndIf
2.  If the node $N$ has count > 1 then
      Context = context|$ac$; //Concatenate a sequence
      If $q$ is nonempty then
         call $ContextDetection(q, N, context)$;
      EndIf
    EndIf
```

3.4 Context Trace Clustering Algorithm

A new trace clustering algorithm called ContextTracClus which aims at creating clusters of traces based on contexts is proposed. The algorithm consists of two distinct phases: (1) Determining trace contexts and Building clusters; (2) Adjusting clusters.

The first phase, *Determining trace contexts and Building clusters*, includes two steps.

Step 1 builds a compact data structure called the Context-tree that stores quantitative information about activities of each trace in a event log. *Step 2* traverses the Context-tree for each trace to find its trace context, and assigns the trace to the cluster corresponding to this context. Based on the Context-tree construction process, for any trace *t* in event log, there exists a path *p* in the Context-tree starting from the root. The trace context of this trace is the sequence of nodes of *p* that have *count* $\geq$ 2. In case a trace has no context, a new cluster is created for storing this trace for later adjustment in Phase 2.

The second phase, *Adjusting clusters*, handles the case where small clusters are generated. If a cluster size, i.e., the number of traces in the cluster, is smaller than a given minimum cluster size threshold *mcs* (e.g., each cluster size should be at least 10% of the number of traces in the event log), this cluster will be added to its closest cluster. The distance between to clusters is defined as the distance between two corresponding trace contexts. In the case that a trace has no context, it will be added to the cluster whose trace context includes the maximum number of duplicate activities with this trace. The pseudo-code of the proposed algorithm, denoted ContextTracClus, is shown in Algorithm 4.

Algorithm 4. ContextTracClus.

```
Input:  An event log $L$
        A minimum cluster size threshold $mcs$
Output: The complete set of clusters $C$
Phase 1: Determining trace contexts and Building clusters
1. $C$ = {};
2. $T$ = ContextTreeConstruction($L$);//$T$ is the context tree
3. Foreach trace $t$ in event log $L$ do
      ContextDetection($t$,$T$,$context$);
      If context is empty then
        Create a new cluster $c$;//$c$ has no label
        Add $t$ to $c$; //This cluster has only one trace
        $C = C \cup c$;
      Else
        If $C$ has no cluster labeled $context$ then
          Create a new cluster $c$ labeled $context$;
          Add $t$ to $c$;
          $C = C \cup c$;
        Else
          Add $t$ to the cluster labeled $context$;
        EndIf
      EndIf
    EndFor
Phase 2: Adjusting clusters
4.  Foreach cluster $c$ in $C$ do
      If size($c$) < $mcs$ then
        Merge $c$ to its closest cluster in $C$;
      EndIf
    EndFor
```

Our algorithm can automatically detect a suitable number of clusters. Unlike traditional clustering algorithms which need convergence loops, our algorithm takes only one loop to identify the clusters, and one loop to merge small clusters.

In K-means algorithm, it randomly selects some data points as the initial center of clusters, and the quality of clustering greatly depends on this selection, especially on event log, where a same trace can occur several times as depicted in Fig. 3, where the trace *acfdh* repeats 10 times. The repeated traces with a big number of times should be a cluster candidate. One more advantage of the algorithm is the ability to put repeated traces into a cluster candidate and removes the uncertainty of random.

The proposed algorithm needs one loop for context tree construction, one loop for clustering. Thus, its complexity is much less than that of traditional clustering algorithms such as K-means, K-modes. Furthermore, the proposed algorithm does not need to transform trace in an intermediate representation (e.g., binary, k-gram, maximal pair, maximal repeat, super maximal repeat and near super maximal repeat, etc.), convert this representation into vector, since it works directly with the traces, then the pre-processing time is greatly reduced.

3.5 An Application Framework for ContextTracClus Algorithm

In process discovery application, we propose a framework as described in Fig. 4, which consists of 5 steps.

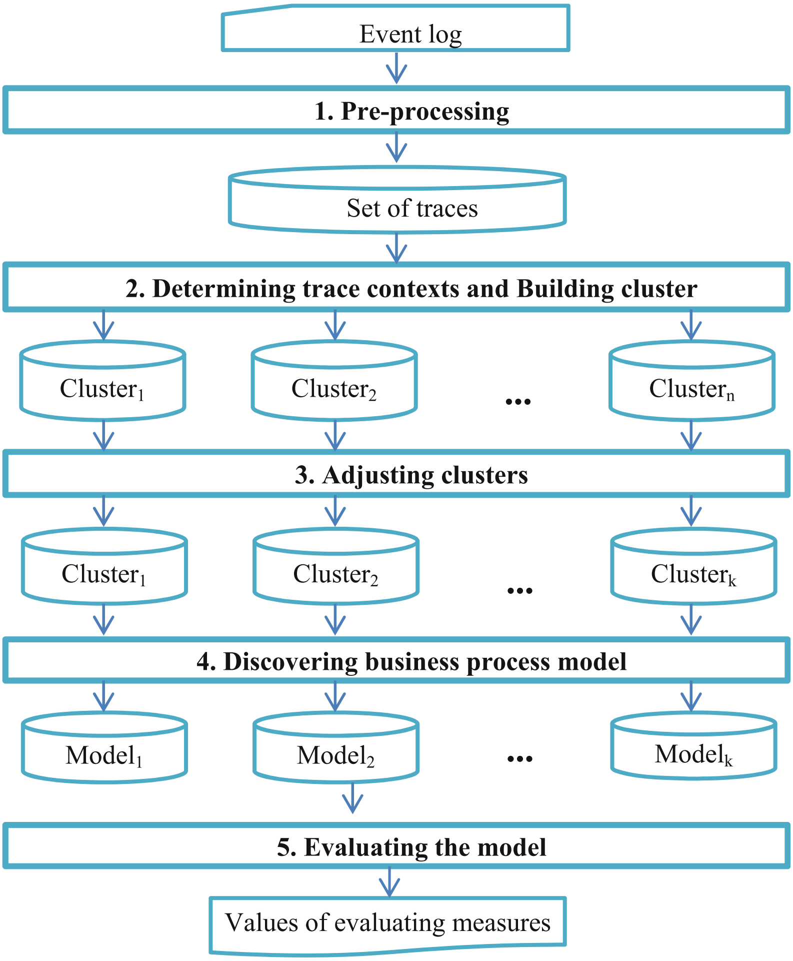


Fig. 4. An application framework of the ContextTracClus algorithm

The Pre-processing step transforms the input event log into a list of traces, i.e., merger all the events with the same *caseid* in the event log into a sequence of activities (sorted by recorded time) to form a trace [20, 23].

Step 2 and 3 use ContextTracClus algorithm to determine the contexts that appear in the event log, and generate n clusters. After adjustment, the number of clusters is k, where $k \leq n$. Each cluster is used to create a sub-log for process discovery.

In step 4, the α-algorithm is used to generate the sub-process models corresponding to each event sub-log.

The Evaluating model step evaluates the quality of each generated process models by two *Fitness* and *Precision*. The fitness measure determines whether all traces in the log can be replayed by the model from beginning to end. The precision measure determines whether the model has behavior very different from the behavior seen in the event log. Additional explanation about the fitness: consider an event log L of 600 traces, and M is the correspondingly generated model. If only 548 traces in L can be replayed correctly in M, then the fitness of M is $\frac{548}{600} = 0.913$. The range of those measures is between 0 and 1, its best value is 1 meaning that the generated process models have the highest quality. Since k models are generated corresponding to k clusters, the final measures, i.e., fitness and precision, are calculated as formula (1).

$$w_{avg} = \sum_{1}^{k} \frac{n_i}{n} w_i \tag{1}$$

where w_{avg} is the aggregated value of the fitness or precision measure, k is the number of clusters, n is the number of traces in the event log, n_i and w_i are the number of traces and the value of the measure in the i^{th} cluster, correspondingly [18].

4 Experimental Result Evaluation

To evaluate the effectiveness of the proposed trace clustering algorithm, we compare our proposed algorithm with K-means clustering algorithm, on three different event logs, i.e., Lfull[1], prAm6[2] and prHm6 (see Footnote 2). Lfull includes 1391 cases with 7539 events; prAm6 consists of 1200 cases with 49792 events; and prHm6 contains 1155 cases with 1720 events.

In the experiment with K-means clustering algorithm, the k-grams trace representation $(k = 1, 2, 3)$ for binary vectors was used. To generate the process model and evaluate the processes, ProM 6.6[3], a process mining tool, was used. The experimental results are shown in Table 3.

[1] www.processmining.org/event_logs_and_models_used_in_book/Chapter7.zip

[2] http://data.3tu.nl/repository/uuid:44c32783-15d0-4dbd-af8a-78b97be3de49

[3] http://www.processmining.org/prom/start

Table 3. Results of K-means and ContextTracClus trace clustering algorithm

Algorithm	Event log					
	Lfull		prAm6		prHm6	
	Fitness	Precision	Fitness	Precision	Fitness	Precision
Scenario 1: Using K-means algorithm						
1-g	**0.991**	0.754	0.968	0.809	0.902	0.66
2-g	0.951	0.958	0.968	0.809	0.902	0.66
3-g	0.955	0.962	0.968	0.809	0.902	0.66
Scenario 2: Using ContextTracClus algorithm						
	0.982	**1**	**0.975**	**0.904**	**0.922**	**0.673**

The experimental results show that ContextTracClus always has a higher precision, i.e., it ensures that the generated process model has the least behaviors not seen in the event log. This is because the traces in a cluster have the same context, i.e., they have the same set of actions so the generated model will have at least superfluous behaviors.

In the scenario 1, we found out the most suitable number of clusters for the data set is 3 after trying with different numbers of clusters, such as 2, 3, 4, 5. The scenario 2 automatically detected the number of clusters based on the input size threshold.

5 Related Work

Greco et al. [4] proposed a clustering solution on traces in event log using bag-of-activities trace representation for *K*-means algorithm.

Song et al. [11] presented a trace clustering approach based on log profiles which captured the information typically available in event logs e.g., activity profile, originator profile. In their approach, the *K*-means, Quality Threshold, Agglomerative Hierarchical Clustering, and SelfOrganizing Maps clustering algorithms were used.

Jagadeesh Chandra Bose et al. [20] proposed a trace representation method based on using some control-flow context information e.g., Maximal Pair, Maximal Repeat, Super Maximal Repeat and Near Super Maximal Repeat. They used some of the clustering algorithms such as Agglomerative Hierarchical Clustering, *K*-means.

Weerdt et al. [6] proposed the ActiTraC algorithm, a three-phase algorithm for clustering an event log into a collection of sub-logs to increase the quality of the process discovery task. The ActiTraC algorithm includes three phases: Selection, Look ahead, and Residual trace resolution. The important idea of this algorithm is the sampling strategy, i.e., a trace is added to the current cluster if and only if it does not decrease the process model accuracy too much.

Ha et al. [18] provided a trace representation solution based on the distance graph model for *K*-Modes, *K*-means clustering algorithms. In this representation, it can describe the ordering and the relationship between the activities in a trace. Distance graphs order k of a trace describe the activity pairs which has distance at most k activities in the trace.

Baldauf et al. [12] presented a survey on an architecture of context-aware systems, which includes the design principles, the common context models. They introduced the existent context-aware systems and discussed their advantages and disadvantages. Their paper mentioned a number of different definitions of "context" such as location, identities of nearby people, objects and changes to those objects (Schilit and Theimer 1994); The user's location, environment, identity and time (Ryan et al. 1997); The user's emotional state, focus of attention, location and orientation, date and time, as well as objects and people in the user's environment (Dey 1998); The aspects of the current situation (Hull et al. 1997). The elements of the user's environment which the computer knows about (Brown 1996).

Becker et al. [22] introduced the support of context information in analyzing and improving processes in logistics. They defined the context as time, location, and frequency of events, tools, devices, or operators. In the experiments, they used the frequency of a process and its overall cycle time as the context data. In addition, they used K-Medoids clustering algorithm for the identification of process groups and for the evaluation of context information.

Bolt et al. [1] presented an unsupervised technique to detect relevant process variants in event logs by applying existing data mining techniques. This technique splits a set of instances based on dependent and independent attributes.

Leyer [13] presented a new approach to identify the effect of context factors on business process performance in the aspect of processing time. They proposed a two-stage approach to identify the relevant data and to determine the context impact by applying the statistical methods.

6 Conclusions and Future Work

This paper proposed a definition of context in business process and a new trace clustering algorithm base on contexts. A context tree was introduced to make the complexity of the algorithm is reduced with two loops over the input for finding clusters, and one small loop over the clusters for adjustment. The ability to work directly with the traces without transforming to an immediate representation is an additional advantage of the algorithm. Another ability to automatically detect the optimal number of clusters makes algorithm to remove the disadvantage of traditional clustering algorithms and produce determined results. As future work, we plan to study the impact of the context in other tasks of the process mining.

References

1. Bolt, A., van der Aalst, W.M.P., de Leoni, M.: Finding process variants in event logs. In: Panetto, H., et al. (ed.) On the Move to Meaningful Internet Systems. OTM 2017 Conferences. OTM 2017. LNCS, vol. 10573, pp. 45–52. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_4
2. Dey, A.K.: Context-aware computing: the CyberDeskProject. In: Proceedings of the AAAI, Spring Symposium on Intelligent Environments, pp. 51–54 (1998)

3. Schilit, B.N., Adams, N., Want, R.: Context-aware computing applications. In: WMCSA, pp. 85–90 (1994)
4. Greco, G., Guzzo, A., Pontieri, L., Saccà, D.: Discovering expressive process models by clustering log traces. IEEE Trans. Knowl. Data Eng. **18**, 1010–1027 (2006)
5. Fischer, I., Poland, J.: New methods for spectral clustering. In: Proceedings of ISDIA (2004)
6. Weerdt, J.D., vanden Broucke, S.K.L.M., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Trans. Knowl. Data Eng. **25**(12), 2708–2720 (2013)
7. Poland, J., Zeugmann, T.: Clustering the Google distance with eigenvectors and semidefinite programming. Knowl. Media Technol. **21**, 61–69 (2006)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: SIGMOD Conference, pp. 1–12 (2000)
9. Weerdt, J.D.: Business process discovery_new techniques and applications. Runner up Ph.D. thesis (2014)
10. Evermann, J., Thaler, T., Fettke, P.: Clustering traces using sequence alignment. In: Reichert, M., Reijers, Hajo A. (eds.) BPM 2015. LNBIP, vol. 256, pp. 179–190. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_15
11. Song, M., Günther, Christian W., van der Aalst, Wil M.P.: Trace clustering in process mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008. LNBIP, vol. 17, pp. 109–120. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00328-8_11
12. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context aware systems. IJAHUC **2**(4), 263–277 (2007)
13. Leyer, M.: Towards a context-aware analysis of business process performance. In: PACIS, vol. 108 (2011)
14. Ryan, N., Pascoe, J., Morse, D.: Enhanced reality fieldwork: the context-aware archaeological assistant. In: Proceeding of the 25th Anniversary Computer Applications in Archaeology (1997)
15. Vitányi, P.M.B.: Information distance: new developments. CoRR abs_1201.1221 (2012)
16. De Koninck, P., De Weerdt, J., vanden Broucke, S.K.L.M.: Explaining clusterings of process instances. Data Min. Knowl. Discov. **31**(3), 774–808 (2017)
17. Koninck, P.D., Weerdt, J.D.: Determining the number of trace clusters_a stability-based approach. In: ATAED@Petri Nets_ACSD, pp. 1–15 (2016)
18. Ha, Q.-T., Bui, H.-N., Nguyen, T.-T.: A trace clustering solution based on using the distance graph model. In: Nguyen, N.-T., Manolopoulos, Y., Iliadis, L., Trawiński, B. (eds.) ICCCI 2016. LNCS (LNAI), vol. 9875, pp. 313–322. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45243-2_29
19. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Context aware trace clustering: towards improving process mining results. In: SDM 2009, pp. 401–412 (2009)
20. Jagadeesh Chandra Bose, R.P.: Process mining in the large preprocessing, discovery, and diagnostics. Ph.D. thesis, Eindhoven University of Technology (2012)
21. Thaler, T., Ternis, S.F., Fettke, P., Loos, P.: A comparative analysis of process instance cluster techniques. Wirtschaftsinformatik **2015**, 423–437 (2015)
22. Becker, T., Intoyoad, W.: Context aware process mining in logistics. Procedia CIRP **63**, 557–562 (2017)
23. Van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19345-3