AWERProcedia
Information Technology
& Computer Science

Vol 04 (2013) 1022-1028

3rd World Conference on Innovation and Computer Sciences 2013

Boolean SATisfiability problem (SAT) in modelling user roles

Monika Simkova*, University of Hradec Kralove, Rokitanskeho 62, Hradec Kralove 50002, Czech Republic.

Suggested Citation:
Simkova M., University of Hradec Kralove, Rokitanskeho 62, Hradec Kralove 50002, Czech Republic. *AWERProcedia Information Technology & Computer Science*. [Online]. 2013, 04, pp 1022-1028. Available from: www.awer-center.org/pitcs

Received November 09, 2012; revised february 16, 2013; accepted March 15, 2013.
Selection and peer review under responsibility of Prof. Dr. Fahrettin Sadıkoglu, Near East University.

Abstract

This paper describes the results of the research project Connection of system for identity management for algorithms for the analysis of access privileges and modelling of user role within the innovation project which was carried out by company AG COM and Ortex in collaboration with the University of Hradec Kralove, together operating in association Hradec Information Technology Cluster (HIT Cluster). The project focuses on linking the existing identity management system – parts of the system roles on external component that provides sophisticated design and life cycle of roles. The basis of the project is the development of algorithms applicable to the analysis of access rights and modelling user roles from the state of the permission settings to the final systems. This classification process based on roles aims at achieving an optimal minimum number of roles that are consistent with the needs of organizations that all users have all the rights that they need. The specific issues of this article is the use of SAT algoritmus in modeling user roles.

Keywords: User roles, modelling of user roles, access priviliges;

*ADDRESSES FOR CORRESPONDANCE: **Monika Simkova,** University of Hradec Kralove, Rokitanskeho 62, Hradec Kralove 50002, Czech Republic, *E-mail Address:* monika.simkova@uhk.cz

1. Introduction

A user account consists of attributes that contain access information, specific technical settings and personal identification of the user (attributes). Each account is covered by one role. And each account has the same values for the same attribute names. If we try to account cover more than one role, is the addition of roles. If more roles are required, they must be added. Covering the account roles includes a simple example where the aim is complete coverage of the entire set of roles with accounts. [1], [2], [3], [4]

The system includes accounts with one X attribute. The domain of this attribute in all accounts in the system is X={a,b,c} => Role1 : a; Role2 : b; Role3 : c. SAT solves the problem of truthfulness closed existentially quantified Boolean formulas. This algorithm allows to defining, if current formula on logical variables is in conjunctive normal form, if there exists truth assignments of variables, which has the value of 1. [5], [6, [7], [8]

For example the formula $\exists\, x1\, x2\, x3\, x4\, (x1 \vee \neg x2 \vee \neg x3) \wedge (x1 \vee x3 \vee x4)$ is satisfied if $x1, x2, x3, x4 = TRUE$.

2. Transfer of problem to SAT

Description of problem is introduced top-down. The description of complex object is decomposed and consists from descriptions of simpler objects. The sum of coverage of all specified aggregated accounts is decomposed into simpler sub-formulas that describe coverage of simple objects. Addition is described conversion of formula established in the first part to the conjunctive normal form. [7]

Table 1. Basic definitions and marking

Basic definitions	Marking
Aggregated account	$agregAcc$
Set of all specified aggregated accounts	$AGREG\_ACCOUNTS$
i-th aggregate account of set of all specified aggregated accounts	$AGREG\_ACCOUNTS(i)$
Attribute	$attrib$
Attribute name	$NAME(attrib)$
Attribute type $attrib$	$TYPE(attrib)$
Set of all attributes aggregated account $agregAcc$	$ATTRIBS(agregAcc)$
Attribute to cover	$attribToCover$
Attribute name to cover	$NAME(attribToCover)$
Set of all attributes to cover	$ATTRIBS\_TO\_COVER$
Role	$role$
Set of all roles that algorithm will be make	$ROLES$
k-th role that algorithm will be make	$ROLES(k)$
Set of all attributes role	$ATTRIBS(role)$
Attribute value	$attribValue$
Set of all attribute values $attrib$	$VALUES(attrib)$
Boolean variable	$\log Variable$
Value of the logical variable	$BOOL\_VALUE(\log Variable)$

Basic axioms

- $i \in 1..\left|AGREG\_ACCOUNTS\right|$

- $agregAcc \in AGREG\_ACCOUNTS$
- $attrib : \exists\, agregAcc, attrib \in ATTRIBS\ (agregAcc)$
- $attribToCover \in ATTRIBS\_TO\_COVER$
- $role \in ROLES$
- $attribValue : \exists agregAcc, \exists attrib, \exists attribName, NAME(attrib) = attribName, attribValue \in \in VALUES((ATTRIBS(agregAcc))(attribName))$
- $TYPE(attrib) \in \{highest\_Value, priority, union\}$
- $\forall attrib, TYPE(attrib) \in \{highest\_Value, priority\} : |VALUES(attrib)| \leq 1$
- $\forall attribToCover \in ATTRIBS\_TO\_COVER, \forall agregAcc \in AGREG\_ACCOUNTS : \exists attrib \in ATTRIBS(agregAccount) \wedge NAME(attrib) = NAME(attribToCover)$
- $attrib \in ATTRIBS(agregAcc) : (ATTRIBS(agregAcc))(NAME(attrib)) = attrib$
- $attrib \notin ATTRIBS(agregAcc) : (ATTRIBS(agregAcc))(NAME(attrib)) = \varnothing$
- $attrib \in ATTRIBS(role) : (ATTRIBS(role))(NAME(attrib)) = attrib$
- $attrib \notin ATTRIBS(role) : (ATTRIBS(role))(NAME(attrib)) = \varnothing$
- $attribToCover \in ATTRIBS\_TO\_COVER : ATTRIBS\_TO\_COVER(NAME(attribToCover)) = attribToCover$
- $attribToCover \notin ATTRIBS\_TO\_COVER : ATTRIBS\_TO\_COVER(NAME(attribToCover)) = \varnothing$
- $BOOL\_VALUE(\log Variable) \in \{TRUE, FALSE\}$

2.1. Domain attribute

Table 2. Domain attribute - definitions and marking

Basic definitions	Marking
Attribute name to cover	$domainName$
Domain attribute with name $domain\_name$	$DOM(domain\_name)$

Axioms

- $domainName : \exists attribToCover, NAME(attribToCover) = domainName$
- $\forall agregAcc \in AGREG\_ACCOUNTS, \forall attrib \in (ATTRIBS(agregAcc))(domainName) :$ $DOM(domainName) = \bigcup VALUES(attrib)$

Description

Domain of attribute whit specific name is defined as the union of all sets of values of all attributes of the specific name of all aggregated accounts.

2.2. Basic variables

Table 3. Basic variables - definitions and marking

Basic definitions	Marking
Attribute name to cover	$attribToCoverName$
Set of all logical basic variables of role	$BASE\_VARIABLES(role)$
Set of all logical basic variables defined for role and $attribToCoverName$	$BASE\_VARIABLES(role, attribToCoverName)$
Basic variable for specific role, attribute name to cover and specific attribute value to cover specific role	$BASE\_VARIABLE(role, attribToCoverName, attribValue)$

Axioms

- $attribToCoverName : \exists attribToCover, NAME(attribToCover) = attribToCoverName$
- $\forall role : BASE\_VARIABLES(role) = \bigcup_{attribToCoverName} BASE\_VARIABLES(role, attribToCoverName)$
- $\forall role, \forall attribToCoverName : BASE\_VARIABLES(role, attribToCoverName) =$ $= \bigcup_{attribValue \in DOM(attribToCoverName)} BASE\_VARIABLES(role, attribToCoverName, attribValue)$

$\forall role, \forall attribToCoverName, \forall attribValue :$
$BOOL\_VALUE(BASE\_VARIABLE(role, attribToCoverName, attribValue)) \in \{TRUE, FALSE\}$

$BOOL\_VALUE(BASE\_VARIABLE(role, attribToCoverName, attribValue)) = TRUE \Leftrightarrow$
$\Leftrightarrow attribValue \in VALUES((ATTRIBS(role))(attribToCoverName))$

Description

The specific basic variable is always associated with specific role, attribute to cover and some value from domain of this attribute to cover. This basic variable takes value $TRUE$ if the specific role includes given values in specific attribute.

2.3. Rules of basic variables

Table 4. Rules of basic variables - definitions and marking

Basic definitions	Marking
Basic variable	$baseVariable$

Axioms

- $baseVariable \in \bigcup_{\forall role \in ROLES} BASE\_VARIABLES(role)$
- $\forall role, \forall attribToCoverName: \left| \begin{Bmatrix} baseVariable : baseVarianle \in BASE\_VARIABLES(role, \\ attribToCoverName), TYPE(ATTRIBS\_TO\_COVER(\\ attributeToCoverName)) \in \{highest\_value, priority\}, \\ BOOL\_VALUE(baseVariable) = TRUE \end{Bmatrix} \right| \leq 1$

Description

For attribute type with simple value exists maximum one variable, which has value $TRUE$, in the case of variables, which are associated with given role and specific attribute. For attributes of type multi-values this restriction doesn´t apply.

3. The transfer equation in conjunctive normal form (CNF)

SAT solver allows use only formula in CNF doesn't allow specific any formula. Formula has to be converted before transfer to the prepared form. The process is based on the use of logical implications. For each formula, which is an instance of one of the types of formulas, it´s made a new variable, which will be able to imply the right side of this formula. This formula, newly made, will be converted to CNF. These clauses that arise transfer to CNF will be directly put into the solver [9].

3.1. Example

$$F\_ACC\_COV = ATTRIBS\_COV(1) \wedge ATTRIBS\_COV(2) \wedge .. \wedge ATTRIBS\_COV(m)$$

Formula 1. Example

1. Create a new logical variable.
2. Creates the implications:

$$F\_ACC\_VAR \Rightarrow ATTRIBS\_COV(1) \wedge ATTRIBS\_COV(2) \wedge ... \wedge ATTRIBS\_COV(m)$$

3. New clauses which has been created after convert to CNF. These clauses should be directly put into solver.

$$(\neg F\_ACC\_VAR \vee ATTRIBS\_COV(1)) \wedge (\neg F\_ACC\_VAR \vee ATTRIBS\_COV(2)) \wedge ...$$
$$... \wedge (\neg F\_ACC\_VAR \vee ATTRIBS\_COV(m))$$

4. If it´s the value of variable F_ACC_COV equal $TRUE$, then is an association account covered.

3.2. Hierarchical approach

Hierarchical approach creates a whole major formula, instead of primary sub-formulas are used in this way the newly created variable. Each object gets its logical variable whose value will represent a kind of state of this object. Usually, it will be the state cover of the object. [6]

4. Conclusion

Using the equivalence instead of implication in the processing of different sub-formulas is natural and more accurate. In used of implication is possible show the case where the value of some variable on the left side of the implication is $FALSE$ and the right side is $TRUE$. This means that although all attribute of some account are covered, the account is not covered itself, it´s inconsistent with the original definition and description of the problem. However, in practice shows a disadvantage using the equity method. One reason is that the number of clauses using the equity method considerable increases – an average roughly doubled thus increased memory consumption, which especially in the calculation of larger instances plays a big role. The second reason, which is partly related to first, is a considerable extension of the calculation, which is especially noticeable on larger instances, occurs. Although the use implications are inconsistent with the definition and description of the problem, this approach doesn´t bring problems. Solver will try to evaluate all the variables such that all clauses (disjunction) are met. Because the variables of aggregated accounts are in the unit disjunction, has only one member, the solver has to assign a value $TRUE$ to these variables. This value but already implies the fulfilment of the right sides of the implication, in other words has conditions on the values of the objects in the lower parts of the hierarchy, not vice versa, so the whole equivalence doesn´t apply here. [7], [8], [9]

Acknowledgements

This work was supported by the project No. CZ.1.07/2.2.00/28.0327 Innovation and support of doctoral study program (INDOP), financed from EU and Czech Republic funds. This article was created based on project of user roles called Connection of system for identity management for algorithms for the analysis of access privileges and modeling of user role within the innovation project which was carried out by company AG COM C. in collaboration with the University of Hradec Kralove, together operating in association HIT cluster.

References

Al-Kahtani, M.A., A model for attribute-based user-role assignment, IEEE Computer Society, Computer Security Applications Conference, 2002, pp.353-362.

D. Richard Kuhn, Edward J. Coyne, Timothy R. Weil, Adding Attributes to Role-based Access Control, IEEE Computer Society, National Institute of Standards and Technology, 2010, pp. 79-81.

D. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli, The role of site features, user attributes, and information verification behaviors on the perceived credibility of web-based information, New Media & Society, Vol.9, No. 2, 2007, pp. 319-342.

Mosse, G., Modeling Roles a practical series of analysis patterns, Journal of Object Technology, Vol. 1, No.4, 2002, pp. 27-37

[Šimková, M., Poulová, P.: The system of user roles and modeling of access privileges, International Conference on COMPUTERS (ICCOMP12) 2012, Kos Island, Greece

Šimková, M., Štěpánek, J.: The optimal setting of access rights by generating output set of access roles, International Conference on DATA NETWORKS, COMMUNICATIONS, COMPUTERS

Šimková, M., Tomášková, H.: Application of max-min algebra for modeling of system of user roles, MATHEMATICAL and COMPUTATIONAL METHODS in SCIENCE and ENGINEERING

Štěpánek, J., Šimková, M.: Modeling roles – description of used algorithms in the system for the analysis of petting of access permissions, International Conference on DATA NETWORKS, COMMUNICATIONS, COMPUTERS

Tomášková, H., Šimková, M.: Use of modelling roles in internet/mobile communications, WORLD CONFERENCE on INFORMATION TECHNOLOGY