Refinement and Validation of Humoral Immunity Based on Event-B

Xuqing Shi, Shengrong Zou\textsuperscript{(E)}, Yudan Shu, and Li Chen

Institute of Information Engineering, Yangzhou University,
Yangzhou, Jiangsu, China
330416490@qq.com

Abstract. By the outbreak of Covid-19, we should focus more eyesight on the human immune system. Humoral immunity plays an important role in the immunologic mechanism. In this process, B cells and other immune cells cooperate each other to produce antibodies and eliminate antigens by series of interactions, activation, proliferation and differentiation. In this paper, we use the formal language Event-B to model the humoral immunity process on the development tool called Rodin. Humoral immunity process is abstract and has complexity in system design. Accidentally, the formal method is used to verify the correctness and consistency of the complex systems, which is an appropriate approach to model this immunity process by stepwise refinements and validation. We also present an instance to demonstrate the differences between the immunity responses after the invasion of influenza viruses and coronavirus respectively in the last refinement and validate it using proof obligations. Experimental results show that events in our model are all validated by the automatic certification tool on Rodin platform.

Keywords: Event-B · Refinements and validation · Coronavirus · Humoral immunity

1 Introduction

The correctness and security of softwares are always the important topic. Recently, the prevalence of Blockchain technology attracts many scholars and scientists from the field of Computer Science. We know its foundation is smart contract. The hackers put efforts into finding bugs in the code of the contract due to its publicity and this will cause serious accidents \cite{1}. For example, in 2017, a bug happened in smart contract caused the loss of 50 million dollars \cite{2}. In addition, in the field of chain design, the formal validation is equally essential to reduce the risk of large losses. There is an instance that when Intel Corporation published Pentium chains, a zero division error occurred. This code mistake costs Intel Corporation nearly 500 million dollars to retrieve Pentium chains. This accident brought huge economic and reputational losses \cite{3}.

Similar to the introduction above, with the development of the global outbreak of Covid-19, human immunology will be paid more attention. Humoral
immunity we concern about is an abstract process occurred among the interactions of diverse immune cells. We use Event-B to model it by stepwise refinements and verify it using proof obligations. Formal method can find amount of bugs and logical mistakes in this procedure. Assume that we validate the correctness of all the states in humoral immunity process, we can realize the software system using traditional programming languages such as java, python and etc. Then, this completed system can be applied for further research in medical community and used as teaching tools in biology classes.

2 Related Work

It’s well known that when viruses or germ enter the body, the immune system in activated state will response and fight against them. As a result, the immune response which has two main lymphocytes B and T helps to clean the infected cells and the viruses. The response which B cells participate in is also called humoral immunity. For malarial infections, the humoral immunity response tends to be more effective as compared to the cellular immunity responses according to [4]. Tian et al. [5] has been inspired by humoral immunity and created a clustering model to recognize unknown antigens. From 2012 to 2017, authors in [6–9] have reported various virus dynamic models with humoral immunity or cellular immunity but due to the complexity and abstractness, these models exist drawbacks such as ignoring the absorption effect. The number of lymphocytes involved in immunity is more than $10^{12}$ which exceeds the total number of nerve cells, and the immune system may be more complex than the nervous system. Motivated by the above studies, we consider adopting formal method to model humoral immunity and validate the correctness.

In the process of humoral immunity, antigens are divided into two groups, TD antigens and TI antigens. TD antigens require collisions and touch with phagocytic cells and helper T cells and to be absorbed and treated by them. Then B cells will receive the information from TD antigens and are activated to proliferate and differentiate plamsa cells. Plasma cells will produce the related antibodies to combine with antigens and generate protein precipitation at last. Unlike TD antigens, TI antigens can directly stimulate B cells to secrete plamsa cells in this process. In addition, some activated B cells can change into effector B cells which are also called memory cells during the proliferation and differentiation. When antigens with the same information as before invade the human body, they will discriminate these antigens swiftly and secrete plamsa cells.

Based on the specification of humoral immunity above, we follow the refinement rules to model step by step, which means that the next model after the refinement is to add details based on the previous one. The model of each step is established by the corresponding requirements. At last, we will prove the consistency of this complex systems of immunity process, which means the logics of the model according to our requirements are reasonable.

The following are specific requirements and modeling descriptions. Each circulation in humoral immunity process can follow the same stepwise refinements.
3 Methodology

Event-B evolves from classical B language. It supports predicate calculus and theorem proving. It is a formal modeling method based on Action Systems. An Event-B model consists of two basic types of components, namely machine and context [10]. A context describes the static part of a model. In a context, relevant sets, constants, axioms and theorems are defined. On the contrary, a machine is used to describe the dynamic part, which includes relevant variables, invariants and events. An event has its guards and actions [11]. The guards represent the conditions under which the event will be fired. The actions mean what will happen if the event is fired. A context can be ‘seen’ by a machine, which means that the machine will use the static elements defined in the context [12]. The fundamental knowledge will be used in the following parts.

3.1 The Initial Model

As shown in Table 1, we specify the requirements of the initial model. Briefly speaking, it describes the growth of B cells which is from pre-B cells to native B cells. Only native B cells can be interacted with other immune cells [13]. Then we start to model it.

Machine m0 is created which sees c0. M0 includes four events and we will introduce the main three of them. But before that, we need to define variables and the related invariants. As shown in Table 2, there are four variations. It is shown that cell which simulates a set of all the cells in the human body is contained in carrier set Cell defined in c0 and cell can partition into bcell(other cell sets will be joined in the next machines). Bcell_hp and bcell_state are all total surjections which denotes that one of the values of Hp and one State can correspond to multiple B cells and their domains and ranges are certainly equal to the sets revealed in the functions.

The following are events. In Fig. 1.(a), we can see the code of the event new_bcell. Bcellobj is a formal parameter which denotes a B cell. Two guards are constrained conditions. Grd1 is the constrained condition which means that bcellobj is not the element of bcell. If grd1 is satisfied, then take actions. Act1 and act4 merge bcellobj into bcell and cell respectively. Then act2 assigns initialHp to the value of Hp of bcellobj and act3 assign prophaseState to the cell state of it. Then, we will introduce bcell_grow_to_immatureState in Fig. 1.(b). In this event, the formal parameter is bcellobj. Guards constrain that bcellobj must satisfy some conditions. When satisfying these guards, take act1 to change the state into immatureState.

3.2 The First Refinement Model

As shown in Table 1, we add the details of the process which describes how TD antigens and TI antigens stimulate B cells to activation respectively and it is divided into four steps. Then, we establish the model.
### Table 1. Requirements analysis

| Initial | Requirements description | First refinement | Requirements description |
|---------|---------------------------|------------------|--------------------------|
| Req1.1  | B cells are produced from the bone marrow and grow into pre-B cells | Req2.1 | TD antigens are absorbed and treated by mng cells |
| Req1.2  | Pre-B cells grow into immature B cells | Req2.2 | mng cells present DNA information of TD antigens to helper T cells |
| Req1.3  | Immature B cells grow into native B cells | Req2.3 | Helper T cells present DNA information of TD antigens to native B cells and activate them |

| Second refinement | Requirements description | third refinement | Requirements description |
|-------------------|--------------------------|------------------|--------------------------|
| Req3.1            | Most of the activated B cells proliferate and differentiate plasma cells | Req4.1 | Memory cells discriminate antigens |
| Req3.2            | A sub-fraction of the activated B cells proliferate and differentiate memory cells | Req4.2 | Memory cells proliferate and differentiate plasma cells |
| Req3.3            | Plasma cells produce antibodies | Req4.3 | Antibodies and antigens combine into protein precipitation |

| Fourth refinement | Requirements description |
|-------------------|--------------------------|
| Req5.1            | Influenza viruses invade human bodies |
| Req5.2            | Coronavirus invade human bodies |
| Req5.3            | B cells and memory cells discriminate influenza viruses and make them swallowed by neutrophile granulocyte and hyaline leukocyte |
| Req5.4            | T cells kill influenza viruses but kill pulmonary cells and upper respiratory tract cells at the same time |
| Req5.5            | Helper T cells present DNA information of TD antigens to native B cells and activate them |
Table 2. Variables in m0

| Variables | Invariants |
|-----------|------------|
| cell      | cell ⊆ Cell |
| bcell     | partition(cell, bcell) |
| bcell_hp  | bcell_hp ∈ bcell → N1 |
| bcell_state | bcell_state ∈ bcell → State |

Fig. 1. The events defined in m0. (a) Describe the production of a B cell. (b) Describe that a B cell grows into immature state.

We establish m1 which sees c1 and refines m0. There are six events in m1 and we will concretely state three of them. As shown in Table 3, we define related variables and invariants. Cell, ag and aginfo are subsets of their parallel carrier sets in c1 and they have their corresponding elements which have slave relations with them according to the associated invariants. Td_ag_td_aginfo and tiag_tianginfo are both total surjections which denote that each TD antigen and TI antigen have their linked DNA information so domains are td_ag and tiag and ranges are td_aginfo and tiaginfo respectively. Mngcell_td_aginfo, helperTcell_td_aginfo, bcell_td_aginfo and bcell_tianginfo are partial functions because their domains are changeable that cells are constantly produced and dead in the human body and not each phagocyte cell, helper T cell or bcell has opportunity to carry DNA information of antigens.

The following are events. The event new_td_ag has two formal parameters, td_agobj and td_aginfoobj. Grd1 and grd2 respectively constrain that they are not the elements of the linked sets. When grd1 and grd2 are contented, then take actions. Act1 and act2 merge them into the related sets. Act3 indicate that the antigen DNA information which td_agobj carries is assigned to td_aginfoobj. In Event-B method, function is not only relation but also set [14]. The event next event has three formal parameters, td_agobj, mngcellobj and td_aginfoobj. Grd1 and grd2 regulate their slave relations. Grd3 denotes that td_aginfoobj is the value of td_adobj under he map function. Grd4 restricts that mngcellobj is not in the domain of mngcell_td_aginfo because before this event is carried out, mngcellobj doesn’t take any information. When act1 are permitted to take, mngcellobj will carry td_aginfoobj.
Table 3. Variables in m1

| Variables      | Invariants                                           |
|----------------|------------------------------------------------------|
| mngcell        | partition(cell, bcell, mngcell, helperTcell)         |
| helperTcell    | cell ⊆ Cell                                         |
| aginfo         | partition(aginfo, td_aginfo, tiaginfo)              |
| td_aginfo      | aginfo ⊆ AgInfo                                     |
| ti_aginfo      | partition(ag, td_ag, tiag)                           |
| ag             | ag ⊆ Ag                                             |
| td_ag          | td_ag td_aginfo ∈ td_ag → td_aginfo                 |
| ti_ag          | tiag_tianginfo ∈ tiag → tianginfo                   |
| td_ag td_aginfo| mngcell td_aginfo ∈ mngcell → td_aginfo             |
| ti_ag ti_aginfo| helperTcell td_aginfo ∈ helperTcell → td_aginfo      |
| mngcell td_aginfo| bcell td_aginfo ∈ bcell → td_aginfo               |
| helperTcell td_aginfo| bcell_tianginfo ∈ bcell → tianginfo         |
| bcell td_aginfo|                                            |
| bcell_ti_aginfo|                                            |

3.3 The Second Refinement Model

As shown in Table 1, the details of the process which describes how the activated B cells change into plasma cells and produce antibodies are added into the previous model. In addition, the appearance of memory cells will work soon. The second refinement requirements are divided into three steps. After stating the requirements, we will complete modeling.

We describe the dynamic machine m2. Similar as before, in Table 4, we define the variables plasmacell, memorycell, ab, td_ab and ti_ab, which imply the sets of these kinds of cells in the actual human body and the relevant invariants regulate the slave relations among them. Plasmacell td_aginfo, plasmacell ti_aginfo and etc. are all partial surjections because the entire cells in them are directly or indirectly produced by the B cells which have DNA information of antigens and they will also carry it. Thus, the domains are equal to their correlated cell sets. But doubtfully, B cells will not take all kinds of antigen information so the range is included by td_aginfo or ti_aginfo.

Later, we will state the two main events. The first event specifies the process that B cells with DNA information of TD antigens proliferate and differentiate. B cells will mostly change into plasma cells, and a little portion of them will translate into memory cells. To simulate this process, we set seven formal parameters in the event as before. Guards constrain that td_aginfoobj is carried by bcellobj, and plasma cells and memorycellobj are not produced before taking actions. Hence, they are not the elements of the relevant cell sets. When satisfying guards, bcellobj disappears to change into plasma cells and
Refinement and Validation of Humoral Immunity Based on Event-B

Table 4. Variables in m2

| VARIABLES                  | INVARIANTS                                      |
|----------------------------|-------------------------------------------------|
| Plasmacell [partition(cell, bcell, mngecel, helperTcell, plasmacell, memorycell)] |                                                 |
| Memorycell [cell ⊆ Cell]  |                                                 |
| ab [partition(ab, td_ab, tiab)] |                                                 |
| td_ab [ab ⊆ Ab]            |                                                 |
| ti_ab [partition(ag, td_ag, tiag)] |                                |
| plasmacell_td_aginfo [plasmacell_td_aginfo ∈ plasmacell ↔ td_aginfo] |                      |
| plasmacell_ti_aginfo [plasmacell_ti_aginfo ∈ plasmacell ↔ ti_aginfo] |                      |
| memorycell_td_aginfo [memorycell_td_aginfo ∈ memorycell ↔ td_aginfo] |                      |
| memorycell_ti_aginfo [td_ab_td_aginfo ∈ td_ab ↔ td_aginfo] |                      |
| ti_ab_ti_aginfo [memorycell_ti_aginfo ∈ memorycell ↔ ti_aginfo] |                      |

In addition, the new produced cells carry the antigen information. The second event new td_ab imitates the process of producing antibodies whose goal is to eliminate TD antigens. Formal parameters are td_abobj, plasmacellobj and td_aginfoobj. Guards regulate them. When guards are contented, td_abobj is truly produced so it will join the set and it will carry the DNA information of the related antigens.

3.4 The Third Refinement Model

As shown in Table 1, this refinement consists of two procedures. One is that memory cells discriminate the antigens which the ones of the same type invaded before and they secret plasma cells to produce antibodies. The other is that antibodies combine with antigens and induce immunological effects. Subsequently, we start to build up the model.

In this refinement, we didn’t append novel elements in the static context so we just need to create dynamic machine m3 which sees c2. In Table 5, we define the new entrants. Memorycell_td_ag and memorycell_ti_ag are both total functions which imply the interactions between the memory cells and antigens. Then, we introduce the events.

Table 5. Variables in m3

| VARIABLES      | INVARIANTS                                      |
|----------------|-------------------------------------------------|
| memorycell_td_ag | memorycell_td_ag ∈ memorycell → td_ag           |
| memorycell_ti_ag | memorycell_ti_ag ∈ memorycell → ti_ag           |

The first event specifies how memory cells indentify TD antigens. When guards are all satisfied, act1 will be taken that ‘memorycellobj→td_agobj’ will be
integrated into memorycell_td_ag. After that, we describe the immune responses during the process of combining antibodies with TD antigens. In this event, grd1 to grd4 make sure that they both belong to the related sets and possess antigen information. Grd5 to grd8 regulate that td_abobj and td_agobj carry the same type of DNA information of antigens. After fulfilling the guards, we take ac1 and act2 to extract td_abobj and td_agobj from the related sets because immune responses happen between them and they are combined and changed into precipitate.

3.5 The Fourth Refinement Model

As shown in Table 1, in the last refinement, for further research on Covid-19, we present an instance to specify the differences of the immune responses between normal influenza viruses and coronavirus when they invade human bodies [15–17]. This refinement includes five steps in the table. Then, we introduce our designed model.

| m4          | Invariants                               |
|-------------|------------------------------------------|
| Variables   |                                          |
| virus       | cell $\subseteq$ Cell                    |
| in_virus    | partition(cell, bcell, mngcell, helperTcell, plasmacell, memorycell, tcell, upp_cell, pul_cell) |
| co_virus    | virus $\subseteq$ Virus                  |
| virusinfo   | partition(virus, in_virus, co_virus)     |
| in_virusinfo| virusinfo $\subseteq$ VirusInfo          |
| co_virusinfo| partition(virusinfo, in_virusinfo, co_virusinfo, harmless_cellinfo) |
| harmless_cellinfo | in_virus_in_virusinfo $\in$ in_virus $\rightarrow$ in_virusinfo |
| in_virus_in_virusinfo | co_virus_co_virusinfo $\in$ co_virus $\rightarrow$ co_virusinfo |
| co_virus_co_virusinfo | bcell_in_virusinfo $\in$ bcell $\rightarrow$ in_virusinfo |
| bcell_in_virusinfo | memorycell_in_virusinfo $\in$ memorycell $\rightarrow$ in_virusinfo |
| memorycell_in_virusinfo | bcell_harmless_cellinfo $\in$ bcell $\rightarrow$ harmless_cellinfo |
| bcell_harmless_cellinfo | memorycell_harmless_cellinfo $\in$ memorycell $\rightarrow$ harmless_cellinfo |
| memorycell_harmless_cellinfo | upp_cell_state $\in$ upp_cell $\rightarrow$ State |
| upp_cell_state | pul_cell_state $\in$ pul_cell $\rightarrow$ State |
| pul_cell_state |                                          |
Fig. 2. The experimental results for proof. (a) Show the results for verifying events in m0. (b) Show the results for verifying events in m1. (c) Show the results for verifying events in m2. (d) Show the results for verifying events in m3. (e) Show the results for verifying events in m4.

We build up the dynamic machine m4. As usual, we specify the variables and their invariants in Table 6. Virus, virusinfo and their internal elements are defined as same as ones in c4. In_virus_in_virusinfo and co_virus_co_virusinfo are both total surjections because each virus has DNA or RNA information and each kind of virus information can correspond to multiple viruses. Bcell_in_virusinfo, memorycell_in_virusinfo, bcell_harmless_cellinfo and memorycell_harmless_cellinfo are all partial functions because not each bcell or memory cell got in touch with viruses or harmless cells and carry their information. For that, domains and ranges of these functions are included by the relevant sets. Upp_cell_state and pul_cell_state are both partial surjections because each upper respiratory tract cell or pulmonary cell has a cell state (alive or dead), so their domains are certain.

After elaborating the variables and invariants, we can characterize the events. The event new_in_virus has two parameters and guards constrain that they are not the elements of the related sets before taking actions. When guards are satisfied, the two parameters will be combined into the sets. Finally, we introduce the event which describes how coronavirus avoid the examination from B cells or memory cells, and then cause fatal damage to the upper respiratory tracts cell, pulmonary cells and other useful cells which are playing important roles. This event has four parameters, which are bcellobj, co_virusobj, upp_cellobj and pul_cellobj. Grd1 and grd2 represent that bcellobj is in activeState so it has...
abilities to receive virus information and indirectly produce antibodies to fight against viruses. Grd5 to grd8 constrain that upp_cellobj and pul_cellobj are regularly working in the human body. The most important guard is grd4. This guard imitates that when bcell collides with co_virusobj, it receives its information but this information is wrong which is mocked by co_virusobj. Consequently, it can be reckon as DNA information of the harmless cells and bcellobj will not handle it. When these guards are passed, co_virusobj will do harm to upp_cellobj and pul_cellobj. Therefore, act1 and act2 assign deadState to them to show the consequences.

4 Experiments

Rodin is a complex tool platform which supports the application of the Event-B formal method. It provides core functionality for syntactic analysis and proof-based verification of Event-B models. The Rodin Platform is an Eclipse-based IDE for Event-B that provides effective support for refinement and mathematical proof [18]. We utilize the automatic proof plugin on Rodin to validate our established model and the events which cannot be automatically verified are proved manually. In Fig. 2, we display the results. In Fig. 2.(a), there are a total of 16 proof obligations in the initial model and 10 obligations of them are automatically verified. For the left 6 ones, we add some constrained conditions and hypotheses to make the obligations pass through the validation. In Fig. 2.(b), there are a total of 41 proof obligations in the first refinement model and they are all validated automatically. In Fig. 2.(c), there are a total of 34 proof obligations in the second refinement model and they are all validated automatically. In Fig. 2.(b), there are a total of 28 proof obligations in the third refinement model and they are all validated automatically. In Fig. 2.(b), there are a total of 41 proof obligations in the last refinement model and they are all verified by the automatic plugin.

5 Conclusion

In this paper, we extract requirements from the abstract process of humoral immunity and show an instance which reflects on the differences between the methods of the attacks which aim at harming the human bodies from influenza viruses and coronavirus. This instance is our try to hope to help push the research on Covid-19. Then, based on the requirements, we define models by stepwise refinements and prove its correctness and consistency. However, we know this is just a portion of the enormous and complex immune system. Thus, we will do further research and develop the superb system. In the future, we also want to use real data to validate our Event-B model to increase its rationality.
References

1. Singla, V., Malav, I.K., Kaur, J., Kalra, S.: Develop leave application using blockchain smart contract. In: 2019 11th International Conference on Communication Systems Networks (COMSNETS), pp. 547–549 (2019)
2. Dirgantoro, K.P., Lee, J.M., Kim, D.: Generative adversarial networks based on edge computing with blockchain architecture for security system. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Fukuoka, Japan, pp. 039–042 (2020)
3. Sulskus, G., Poppleton, M., Rezazadeh, A.: An interval-based approach to modelling time in Event-B. In: Fundamentals of Software Engineering, vol. 9392, pp. 292–307 (2015)
4. Deans, J., Cohen, S.: Immunology of malaria. Ann. Rev. Microbiol. 37, 25–50 (1983)
5. Tian, Y., Ren, P.: A clustering model inspired by humoral immunity. In: 2009 International Workshop on Intelligent Systems and Applications, Wuhan, pp. 1–4 (2009). https://doi.org/10.1109/IWISA.2009.5072611
6. Pradeep, B.G.S.A., Ali, H.: Global stability properties for a delayed virus dynamics model with humoral immunity response and absorption effect. In: 2017 International Conference on Electrical Engineering (ICEE), Lahore, pp. 1–6 (2017)
7. Wang, S.F., Zou, D.Y.: Global stability of in-host viral models with humoral immunity and intracellular delays. Appl. Math. Model. 36, 1313–1322 (2012)
8. Wang, T., Hu, Z., Liao, F.: Stability and Hopf bifurcation for a virus infection model with delayed humoral immunity response. J. Math. Anal. Appl. 411, 63–74 (2014)
9. Elaiw, A.M.: Global stability analysis of humoral immunity virus dynamics model including latently infected cells. J. Biol. Dyn. 9(1), 215–228 (2015)
10. Siyuan, H., Hong, Z.: Towards transformation from UML to event-B. In: 2015 IEEE International Conference on Software Quality, Reliability and Security - Companion, Vancouver, BC, pp. 188–189 (2015)
11. El Mimouni, S., Bouhdadi, M.: Formal modeling of the simple text oriented messaging protocol using Event-B method. In: IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), Marrakech 2015, pp. 1–4 (2015)
12. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press, Cambridge (2010)
13. Ningthoujam, S., Chingkeisingba, T., Chakraborty, S.K.: Finding an effective distance between T-cell and B-cell using S/W ARQ in an immune system communication. China Commun. 17(1), 174–185 (2020)
14. Predut, S., Ipaté, F., Gheorghe, M., Campean, F.: Formal modelling of cruise control system using Event-B and Rodin platform. In: 2018 IEEE 26th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmarCity/DSS), Exeter, United Kingdom, pp. 1541–1546 (2018)
15. Wu, L., Horng, J., Huang, H., Chen, W.: Identifying discriminative amino acids within the hemagglutinin of human influenza A H5N1 virus using a decision tree. IEEE Trans. Inf. Technol. Biomed. 12(6), 689–695 (2008). https://doi.org/10.1109/TITB.2008.986871
16. Dong, D., et al.: The role of imaging in the detection and management of COVID-19: a review. IEEE Rev. Biomed. Eng. (2020). https://doi.org/10.1109/RBME.2020.2990959
17. Pavesi, A., Tan, A.T., Chen, M.B., Adriani, G., Bertoletti, A., Kamm, R.D.: Using microfluidics to investigate tumor cell extravasation and T-cell immunotherapies. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, pp. 1853–1856 (2015). https://doi.org/10.1109/EMBC.2015.7318742

18. Hoang, T.S., Fürst, A., Abrial, J.: Event-B patterns and their tool support. In: 2009 Seventh IEEE International Conference on Software Engineering and Formal Methods, Hanoi, pp. 210–219 (2009). https://doi.org/10.1109/SEFM.2009.17