DOI [https://doi.org/10.15589/znp2020.1\(479\).10](https://doi.org/10.15589/znp2020.1(479).10)
УДК 519.766.2

A SURVEY OF PARAMETRIC MODEL VERIFICATION METHODS

ОГЛЯД МЕТОДІВ ВЕРИФІКАЦІЇ ПАРАМЕТРИЗОВАНИХ МОДЕЛЕЙ

Yaroslav Yu. Dorogyu
cisco.rna@gmail.com
ORCID: 0000-0003-3848-9852
Vasyl V. Tsurkan
v.v.tsurkan@gmail.com
ORCID: 0000-0003-1352-042X

Я. Ю. Дорогий,
канд. техн. наук, доцент
В. В. Цуркан,
канд. техн. наук, доцент

National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ

Abstract. In fact, at present there is no up-to-date overview of parametric model verification methods that would clearly answer the question of whether a parameterized model verification approach is appropriate. That is why the purpose of the study is to analyze the methods of verification of parameterized models. Methods of systematic analysis, comparison and analysis were used during the study.

Parametric model verification methods are analyzed, among which four groups of methods are investigated, such as analytical reduction methods, abstraction methods, symbolic methods and invariant search methods. In analytical methods of reduction, as a rule, the method is indicated displaying models of a family of parameterized models on one model of this family. For abstraction methods, the method for displaying parameterized models on one model is also often indicated, but this model does not belong to the parameterized family itself, but rather models the behavior of any number of homogeneous processes. The mapping method follows structural constraints on parameterized model families that can be verified in this way. Symbolic methods typically use modifications to the symbolic approach to solve the model verification problem. Invariant-based methods seek or construct an invariant process that simulates the behavior of multiple system processes.

The study identified their main characteristics, disadvantages and advantages. Symbolic methods are used mainly for checking security properties, while many invariant methods are applicable for checking survivability properties. To apply symbolic methods, the parameterized model description must be presented in a special method language, which is usually quite different from the model verification languages of the model verification tools. Invariant methods use process descriptions in the form of transition systems, marked with propositional variables, and are compatible with the model language of the model verification tools. Invariant methods are used to check the specification and search for counterexamples existing models verification tools, symbolic methods require implementation of special algorithms. As a result, the best group of methods has been identified that does not require the development of new description languages and special algorithms.

Key words: parameterized model; abstraction method; invariant search method; transition system; labeled transition system; context-free grammar.

Анотація. Фактично сьогодні відсутній сучасний огляд методів верифікації параметризованих моделей, який би чітко відповів на питання щодо доцільності використання того чи іншого підходу до верифікації параметризованих моделей. Саме тому метою дослідження є аналіз методів верифікації параметризованих моделей. Під час проведення дослідження застосовувалися методи системного аналізу, порівняння та аналізу.

Проведено аналіз методів верифікації параметризованих моделей, серед яких досліджено чотири групи методів, таких як аналітичні методи редукції, методи абстракції, символічні методи та методи, засновані на пошуку інваріантів. В аналітичних методах редукції, як правило, зазначається спосіб відображення моделей сімейства параметризованих моделей на одну модель цього сімейства. Для методів абстракції також часто зазначається спосіб відображення параметризованих моделей на одну модель, але ця модель не належить самому параметризованому сімейству, а моделює поведінку будь-якого числа однотипних процесів. Зі способу відображення впливають структурні обмеження на параметризовані сімейства моделей, які можуть бути верифіковані в такий спосіб. Символічні методи зазвичай використовують модифікації символічного підходу до розв'язання задачі верифікації моделей. У методах, заснованих на пошуку інваріанта, шукається або конструюється процес-інваріант, що моделює поведінку декількох процесів системи.

В ході дослідження визначено їх основні характеристики, недоліки та переваги. Символьні методи застосовуються переважно для перевірки властивостей безпеки, тоді як багато методів інваріантів застосовні також для перевірки властивостей живучості. Для застосування символьних методів опис параметризованої моделі потрібно представити спеціальною мовою методу, яка зазвичай досить сильно відрізняється від мов опису моделей засобів верифікації моделей. Методи інваріантів використовують описи процесів у вигляді транзитивних систем, розмічених пропозиціональними змінними, а також сумісні з мовами опису моделей засобів верифікації моделей. Методи інваріантів використовують для перевірки специфікації та пошуку контрприкладів наявні засоби верифікації моделей, символьні ж методи вимагають реалізації спеціальних алгоритмів.

Як наслідок, визначено найкращу групу методів, яка не потребує розроблення нових мов опису та спеціальних алгоритмів.

Ключові слова: параметризована модель; метод абстракції; метод пошуку інваріантів; транзитивна система; розмічена транзитивна система; контекстно-вільна граматика.

ПОСТАНОВКА ЗАДАЧІ

У науковій літературі зустрічаються різні формальні й неформальні описи розподілених систем (РС) [1; 2; 3]. Характерною властивістю РС є наявність взаємодіючих компонентів системи. Процеси виконуються на одному або декількох обчислювальних пристроях. Загалом обчислювальні пристрої можуть бути фізично віддалені один від одного.

Під час побудови моделі РС важливим є вибір модельованих аспектів системи. Зазвичай РС представляється у вигляді параметризованої моделі, яка характеризує якийсь аспект реальної системи. Саме тому задача побудови правильної моделі та її верифікації є актуальними.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Загалом задача верифікації параметризованих моделей (ВПМ) алгоритмічно нерозв'язна. Однак існують класи моделей, для яких ця задача може бути розв'язною.

Перші роботи, присвячені розв'язанню задачі ВПМ, з'явилися в другій половині 80-х років. За минулі тридцять років показана можливість розв'язання задачі для асинхронних односпрямованих кілець з маркером одного типу [4], кілець з довільним типом маркера й додатковими обмеженнями на РТС-процес [5–7], симетричних систем з глобальними передумовами переходів [7; 8], широкомовних синхронних систем [7; 9; 10], систем з довільною топологією і одним маркером [11]. Запропоновано різні часткові процедури, зокрема підходи, засновані на пошуку процесу-інваріанта [7; 12; 13; 14]; підходи, засновані на перевірці досяжних станів регулярних моделей [7; 15; 16; 17; 18; 19]; символьні методи верифікації [7; 20; 21]. Зазначено способи абстракції [7; 11; 22; 23; 24] для певних класів систем.

Історично методи розв'язання задачі ВПМ розвивалися в таких напрямках:

- аналітичні методи редукції;
- методи абстракції;
- символьні методи;
- методи, засновані на пошуку інваріанта.

В аналітичних методах редукції, як правило, зазначається спосіб відображення моделей сімейства

параметризованих моделей на одну модель цього сімейства (cut-off model). У цьому разі точно відомі обмеження, яким повинні задовольняти моделі сімейства.

Для методів абстракції також часто зазначається спосіб відображення параметризованих моделей на одну модель, але ця модель не належить самому параметризованому сімейству, а моделює поведінку будь-якого числа однотипних процесів. Зі способу відображення випливають структурні обмеження на параметризовані сімейства моделей, які можуть бути верифіковані в такий спосіб.

Символьні методи зазвичай використовують модифікації символьного підходу до розв'язання задачі ВМ. Для опису параметризованого сімейства моделей використовуються системи переходів, в яких переходи задаються неявно, за допомогою символьних описів. Наприклад, множини переходів можуть представлятися регулярними виразами або логічними формулами з кванторами існування й загальності. Для заданої системи переходів шукається множина станів, досяжних з початкових станів. В разі символьних методів, як правило, немає гарантії завершеності.

У методах, заснованих на пошуку інваріанта, шукається або конструюється процес-інваріант, що моделює поведінку декількох процесів системи. Інваріант повинен володіти такою властивістю: система, що складається з інваріанта з додатковими однотипними процесами, також моделюється інваріантом. За допомогою індукції показується, що система з як завгодно великим числом однотипних процесів моделюється системою, що складається з фіксованого числа процесів та інваріанта. Основна відмінність від методів абстракції полягає в тому, що не потрібне явне відображення кожної моделі параметризованого сімейства на модель-інваріант.

ВІДОКРЕМЛЕННЯ НЕ ВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Фактично сьогодні відсутній сучасний огляд методів верифікації параметризованих моделей, який би чітко відповів на питання щодо доцільності використання того чи іншого підходу до верифікації параметризованих моделей.

МЕТА ДОСЛІДЖЕННЯ

Метою статті є аналіз методів верифікації параметризованих моделей.

Для досягнення поставленої мети дослідження потрібно виконати такі завдання:

- 1) аналіз методів верифікації;
- 2) характеристика їх особливостей;
- 3) визначення їх недоліків та переваг.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єктом дослідження є методи верифікації параметризованих моделей.

Предметом дослідження є аналіз методів верифікації параметризованих моделей.

Під час проведення дослідження застосовувалися такі методи, як метод системного аналізу (для дослідження формування математичних моделей методів верифікації), метод порівняння (для виявлення впливу окремих чинників на метод загалом), метод порівняльного аналізу (для визначення переваг і недоліків методів верифікації).

ОСНОВНИЙ МАТЕРІАЛ

Огляд наявних методів верифікації параметризованих моделей

Під параметризованим сімейством Φ розуміється нескінченна множина моделей $\Phi = \{M_n\}$, де кожна модель $M_n = Q \| P \|_{\dots} \| P$ складається з фіксованого процесу (компонента) Q та n ізоморфних процесів (компонент) P . Означимо за допомогою запису P^n паралельну композицію n процесів (компонент) типу P , тобто $P^n = P \|_{\dots} \| P$. Множина станів процесу (компонента) P позначається за допомогою запису S_P .

Методи редукції. В аналітичних методах редукції вибираються обмеження на структуру розмічених транзитивних систем (РТС) параметризованих моделей, що перевіряються. Після цього вибираються деяке відношення еквівалентності $\approx$ на транзитивній системі (ТС), яке має властивість консервативності відносно заданого класу специфікації Φ :

Якщо $M_1 \approx M_2$, то для будь-якої формули $\varphi \in \Phi$ є вірним $M_1 \models \varphi \Leftrightarrow M_2 \models \varphi$.

Після вибору відношення $\approx$ доводиться, що для будь-якого сімейства Φ , структура РТС процесів (компонент) якого задовольняють вибраним обмеженням, знайдеться таке число N , починаючи з якого є вірним $M_{N+i} \approx M_N$ для всіх натуральних i . В цьому випадку достатньо перевірити специфікацію на моделях $M_1, \dots, M_N$. Модель M_N називається граничною моделлю (cut-off), а число N – граничним значенням.

В роботі [4] розглядається клас асинхронних систем з топологією одно направленою кільця, які взаємодіють за допомогою синхронного обміну повідомленнями. В РТС процесу допускаються тільки локальні переходи та синхронна передача маркеру

одного типу. В якості відношення еквівалентності використовувалось відношення блочної бісимуляції, яке запропоновано авторами. Для специфікацій виду $\bigwedge_i f(i), \bigwedge_i f(i, i+1), \bigwedge_{i,j,i+j} f(i, i+1, j)$, в яких формули виду $f(k)$ є формулами логіки STL*-X відносно змінних процесу k , аналітично побудовано відношення блочної симуляції та знайдені граничні значення $N = 2, 3, 4, 5$ відповідно.

В роботі [5] розглядаються також асинхронні системи з топологією двонаправленого кільця процесів, що обмінюються типізованим маркером в одно направленою кільці. При цьому кожний процес може змінювати тип маркеру обмежене число раз (число b). Для цього випадку та специфікацій у вигляді формул логіки LTL-X визначається граничне значення $O(b)$, що залежить від ТС процесів.

В роботі [8] розглянуто клас асинхронних параметризованих систем з широкомовним обміном повідомленнями. При цьому умови РТС процесів повинні мати спеціальний вигляд: умовою переходу є або диз'юнкцією $\vee(c \vee \dots \vee d)$ за станами $(c, \dots, d)$ всіх процесів, відмінних від активного; або кон'юнкцією $\wedge(i)$ за початковим станом всіх процесів, відмінних від активного.

Методи абстракції. Ідею методів абстракції можна описати таким чином. Для параметризованих сімейств Φ і кожного натурального числа n задається відображення α моделі P^n на абстрактну модель A . Після чого перевіряється модель $Q \| P^k \| A$ з фіксованим числом процесів P , одним процесом Q і одним процесом A . Вибір числа k , виду моделі A і відображення α залежить від методу. У деяких методах число k дорівнює 0. Варто відразу зауважити, що вид моделі A і відображення α визначається експертом.

Однією з перших робіт, в якій використовувався такий підхід, є робота [25]. У цій роботі розглядається клас систем асинхронно-взаємодіючих процесів з топологією «Зірка». У центрі знаходиться керуючий процес Q , з яким пов'язано n ізоморфних клієнтських процесів P . Абстрактна модель A будується у вигляді графа VASS (Vector Addition State System). Нехай множина $S_P = \{s_1, \dots, s_m\}$ – множина станів процесу P , впорядкована довільним чином. У РТС P^n кожен процес перебуває в одному з станів з множини S_P . Вершинами графа A є кортежі $(k_1, \dots, k_m)$.

Ці вершини відповідають станам РТС P^n для різних значень n , в яких для будь-якого $i, 1 \leq i \leq m$, число процесів P , що знаходяться в стані s_i , дорівнює k_i . Кількість вершин в отриманому графі є нескінченною, але автори показують, що для перевірки властивостей досить обмежитися переглядом кінцевих шляхів з вершини, що відповідає початкового стану.

Таким чином, метод гарантовано завершується. Однак складність перегляду двічі експоненційна

щодо числа станів клієнтських процесів, тому на практиці метод може виявитися непридатним.

Специфікації задаються за допомогою формул LTL для одного спостережуваного процесу.

В роботі [26] для побудови абстрактної моделі A асинхронної системи застосовується схожа ідея. Однак замість повного графа VASS використовується його абстракція: в вершинах $(k_1, \dots, k_m)$ графа, відповідного моделі A , елементи k_i приймають значення з множини $\{0, 1, +\}$. Елементи 0, 1, + відповідають числу процесів 0, 1 і більше одного.

Таким чином, будь-яка модель сімейства Φ відображається на модель з кінцевим числом станів, в будь-якому стані якої зберігаються абстракції числа процесів для кожного вихідного стану процесу P . За допомогою зазначеного відображення абстракції успішно перевірялися параметризовані моделі протоколів когерентності кешів (MESI, ICCP і DCCP) в системі верифікації моделей Murφ. Метод завжди завершується, але специфікація може не виконуватися на абстрактній моделі A . У такому разі не можна зробити висновок про можливість виконання специфікації на моделях сімейства Φ .

Специфікації задаються за допомогою формул логіки ACTL* над глобальними змінними системи.

Модифікація графа VASS використовується для перевірки асинхронних систем з змінними, що розділяються і в роботі [27]. За описами процесів за допомогою САДТ PVS [28] проводиться булева абстракція систем переходів і неявно будується абстрактна модель A , система переходів якої представлена системою лінійних обмежень. Для системи лінійних обмежень за допомогою стандартних засобів шукається рішення системи, в якому більше одного процесу P знаходиться в заданому стані $s_{cs} \in S_p$. Стан s_{cs} відповідає критичній секції. Якщо рішення знайдене, то в критичній секції знаходиться більше одного процесу і властивість взаємного виключення процесів в критичній секції порушується. Ключовим моментом методу є побудова булевої абстракції графа за допомогою САДТ. При побудові абстракції користувач повинен вказати предикати над змінними ТС, що використовуються при абстракції. Автори продемонстрували застосовність методу для алгоритмів взаємного виключення в критичній секції.

В роботі [11] розглядаються параметризовані моделі розподілених систем з асинхронним паралелізмом і довільною топологією, в яких процеси обмінюються одним маркером. Для процесів P вказується спосіб побудови абстрактної РТС A і відображення РТС P^n на РТС A . Після чого доводиться, що для перевірки формули виконуваності φ логіки LTL-X на сімействі Φ досить перевірити здійснимість формули φ на РТС A . Алгоритм побудови абстрактної моделі A завжди завершується.

В роботі [29] автори досліджували параметризовану модель протоколу Берна, що використовується для взаємного виключення процесів в критичній секції. Для РТС процесів цього протоколу визначається абстрактна модель $A = P^2$ і відображення α моделі P^n на модель A . Абстрактна модель і відображення α будувалися авторами вручну. Перевірка коректності відображення α проводилася з використанням САДТ. Перевірка якості критичної секції на моделі A , заданій у вигляді формули логіки LTL здійснювалася за допомогою верифікатора моделей SPIN.

В роботі [30] автори перевірили модель змагання процесів в протоколі Firewire. Для цього автори застосували спосіб побудови абстрактної моделі A , аналогічний описаному вище відповідно до роботи [26]. Автори розглядали спеціальний клас моделей, в яких кожне обчислення веде в фінальний стан. Такі моделі автори назвали деградуючими (degenerating). Абстракція застосовувалася авторами і для перевірки моделей телекомунікаційних протоколів [31].

В роботі [32] розглядається параметризована асинхронна модель, яка використовує дві абстракції. Перша абстракція предикатна – стан пам'яті системи апроксимується як кортеж булевих змінних, які визначають, чи перебуває пам'ять в визначеному стані чи ні. Друга використовувана абстракція – лічильник. Для кожного можливого стану окремого процесу абстрактний стан містить лічильник, який вказує скільки процесів знаходиться в відповідному стані.

Символьні методи. Символьні методи розв'язання задачі ВПМ беруть початок від символьних методів розв'язання задачі ВМ. Ідея символьних методів ВМ полягає в поданні відношень переходів РТС у вигляді формул логіки висловлювань.

Операції пошуку досяжних станів, що використовуються в ВМ, замінюються операціями логічного виведення. Часто формули логіки висловлювань представляються за допомогою бінарних діаграм рішень (БДР) [33].

У випадку задачі ВПМ формул логіки висловлювань виявляється недостатньо для застосування символьних методів. З цієї причини для опису ТС використовуються інші мови: регулярні вирази, арифметика Пресбургера, логіки з квантором за однією змінною і т.п. За допомогою вибраної мови відразу для всіх моделей сімейства Φ описується: множина початкових станів I моделей сімейства Φ ; перетворювач T множини станів системи, відповідний відношенню переходів; специфікація φ , яка повинна виконуватися на множині досяжних станів.

У символьних методах задача ВПМ зводиться до перевірки досяжності стану, в якому порушується специфікація φ , з множини початкових станів I за допомогою переходів, заданих перетворювачем T . Символьні методи розв'язання задачі ВПМ використовувалися, в основному, для перевірки алгоритмів

взаємного виключення процесів в критичній секції. Найчастіше розглядаються системи з асинхронним паралелізмом, в яких процеси взаємодіють за допомогою поділюваних змінних. Топологія комунікаційної мережі, як правило, відповідає повнозв'язному графу.

У роботах [34, 15] перетворювач T задається у вигляді регулярного виразу над алфавітом S_p . За перетворювачем T ітеративно будується його транзитивне замикання T^+ , яке відповідає транзитивному замиканню відношення переходів. Після чого перевіряється порожнеча перетину мов $\neg\phi$ і $T(I)$. Якщо перетин мов є порожнім, то специфікація ϕ вірна в станах, досяжних з множини, заданої описом I . Даний підхід отримав назву «Верифікація регулярних моделей» (Regular Model Checking). При застосуванні методу виникає кілька проблем. Ітеративне обчислення T^+ не має властивості завершуваності, так як перетворювач T не обмежений зверху. Тому часто будується апроксимація T_A транзитивного замикання T^+ за допомогою підказок експерта. Як і у випадку методів абстракції, з порушення специфікації ϕ на множині станів, обчислених за допомогою апроксимації T_A , не впливає порушення специфікації на множині досяжних станів.

Схожий підхід використовується і в роботі [35]. Крім регулярних виразів над алфавітом S_p використовуються автомати над деревами, в вершинах яких знаходяться стани процесу P . Аналогічно для деревовидної топології обчислюється апроксимація множини досяжних станів. Даний підхід детально викладено в роботі [36], там же запропоновані способи перевірки не тільки властивостей безпеки, а й живучості.

В роботі [37] для представлення перетворювача T використовується одномісна логіка другого порядку (M2L). Описи на мові M2L транслюються в БДР і перевіряються за допомогою засобу MONA.

Робота [10] присвячена перевірці параметризованих моделей, в яких перетворювач T описаний за допомогою арифметики Пресбургера. Процеси взаємодіють за допомогою поділюваної пам'яті або широкомовної розсилки повідомлень. Специфікації, що перевіряються, задаються за допомогою формул лінійної логіки відносно змінних однотипних процесів. Показано, що за допомогою використовуваного представлення і алгоритмів перевірки формул арифметики Пресбургера задачу ВПМ можна вирішити для систем з кільцевою топологією і широкомовних систем. Отримані результати про розв'язність задачі ВПМ схожі з результатами робіт [8, 38].

В роботі [39] авторами запропонована методика символічного кодування динаміки моделей PADE, тобто моделей, заданих за допомогою кусково-афінними диференціальними рівняннями для дослідження мереж регуляторів генів. Методика дозволила виключити з процесу явну генерацію простору станів та перерахування можливих параметрів моделі. Методика

може бути застосована до неповністю параметризованих моделей та може обробляти будь-які динамічні характеристики, які можна визначити засобами темпоральної логіки.

Методи пошуку інваріантів. Перший метод, який використовували для розв'язання задачі ВПМ, заснований на пошуку РТС, яка називається інваріантом, для параметризованого сімейства Φ і індуктивному доказі про зведення задачі верифікації специфікації на всіх моделях сімейства до задачі верифікації моделі, що містить процеси моделей параметризованого сімейства і інваріант. Далі всі методи, засновані на пошуку інваріанта, називаються коротко методами інваріантів. Огляд методів інваріантів наведено в роботі [40].

Ідея методу інваріантів полягає в наступному. На множині РТС вибирається відношення еквівалентності або часткового порядку $\leq$. Обране відношення порядку $\leq$ має бути монотонним відносно операції паралельної композиції $\parallel$, тобто для будь-яких РТС $\{M_i\}_{i=1}^4$ з співвідношень $M_1 \leq M_2$ і $M_3 \leq M_4$ має слідувати співвідношення $M_1 \parallel M_3 \leq M_2 \parallel M_4$. Відношення $\leq$ також має задовольняти властивості консервативності для заданого класу специфікацій Φ , тобто для будь-яких РТС M_1 і M_2 та будь-якої формули $\phi \in \Phi$ з співвідношень $M_1 \leq M_2$ і $M_2 \models \phi$ слідує співвідношення $M_1 \models \phi$.

Для заданого сімейства Φ шукається РТС I , яка називається інваріантом і задовольняє властивостям: P і $P \parallel I \leq I$. Після чого індукцією за числом процесів за допомогою властивості монотонності показується, що $P^n \leq I$. З властивості консервативності слідує, що специфікацію $\phi \in \Phi$ досить перевірити на моделі $Q \parallel I$. Якщо специфікація виконується на моделі $Q \parallel I$, то вона виконується і на всіх моделях $M_n, n \geq 1$ сімейства Φ .

В роботі [41] в якості відношення $\leq$ використовується відношення еквівалентності, назване авторами відношенням відповідності. У наступній роботі авторів [42] на підставі цього відношення визначено відношення прорідженої еквівалентності або прорідженої бісімуляції (stuttering bisimulation). Специфікації задаються за допомогою формул індексованої логіки CTL^*-X , тобто формул виду $\bigvee_{i=1}^n f(i)$, в яких підформули $f(i)$ є еквівалентними формулами логіки CTL^*-X з одною вільною змінною i , а всі пропозиціональні $f(i)$ є змінними процесу з номером i . В роботі вручну будується відношення відповідності для алгоритму взаємного виключення в критичній секції з топологією кільця. Як інваріанта використовується процес $I = P$.

В роботі [43] авторами явно формулюється теорема про індукції в методі інваріантів. Відношення $\leq$ задається як відношення мовного включення трас процесів. Процеси об'єднуються в лінійну топологію за допомогою синхронної паралельної композиції.

Пропонований підхід працює автоматично. Як інваріанта використовується процес $I = P$.

В роботі [44] описано метод інваріантів для випадку $I \neq P$. Процеси об'єднуються в лінійну або кільцеву топологію за допомогою асинхронної паралельної композиції. В якості відношення $\leq$ використовується відношення трасового включення зі спостереженням відмов і дивергенції («зациклення») процесів. У прикладах інваріант I будується вручну.

До появи роботи [45] розв'язання задачі ВПМ пропонувалися для двох простих топологій розподілених систем: лінійної та кільцевої. В цій роботі топологія параметризованого сімейства описується за допомогою мережових (графових) граматики. Терміналами мережової граматики є РТС, в правилах виведення задаються способи породження екземплярів процесів і їх синхронізації. Кожному слову граматики відповідає модель з сімейства Φ . Для опису специфікації використовується індексована логіка LTL, аналогічна індексованій логіці CTL*-X. Процеси взаємодіють за допомогою обміну повідомленнями, композиція процесів асинхронна. Для перевірки якості $P \parallel I \leq I$ будуються абстрактні графи комунікаційної поведінки РТС (Communication behaviour), які перевіряються на ізоморфізм.

Підхід, описаний в роботі [45], розвивається далі в роботах [46, 47]. В роботі [46] в якості відношення $\leq$ вже використовується відношення суворої симуляції. Для процесів, що беруть участь в правилах виведення, будуються абстрактні РТС, в яких використовуються регулярні вирази над станами процесів. У моделі застосовується синхронний паралелізм. В роботі [47] результати так само застосовуються для асинхронного паралелізму з використанням відношення суворої симуляції, проте це відношення на практиці може бути застосовано в методі тільки в разі досить сильною абстракції.

В роботі [48] розглядаються параметризовані моделі з лінійною і деревовидною топологіями і синхронним паралелізмом. Відношення $\leq$ є відношенням трасового включення. Специфікації, що перевіряються, виражаються за допомогою додаткових процесів, які називаються спостерігачами (observers). Інваріант шукається автоматично. Для цього вводиться монотонний оператор F додавання додаткових процесів, діючий на множинах трас. Для оператора F будується нерухома точка. Оператор F необмежений, тому процедура пошуку, в загальному випадку, може не завершитися. Як і в символьних методах, експерт може модифікувати ТС, застосовуючи техніки акселерації і розширення.

В роботі [22] автори застосовують комбінацію методів абстракції і інваріантів. Відношення $\leq$ визначається як відношення модульної абстракції, що поєднує побудову абстрактного графа поведінки і трасового включення. За допомогою відношення модульної абстракції шукається інваріант. Застосування методу демонструється на прикладах алгоритму взаємного виключення і об'їдаючих філософів, інваріанти для яких будуються вручну.

Важливим досягненням роботи є поєднання методів інваріантів і абстракції, в тому числі, для процесів з цілочисельними лічильниками.

В роботі [30] автори використовували індукцію і відношення прорідженої бісимуляції для перевірки моделі змагання процесів в протоколі Firewire. Фактично, обґрунтування коректності редукції проводилося вручну.

В роботі [49] автор досліджує асинхронну параметризовану модель, визначену за допомогою РТС з накладеними на нею обмеженнями, які визначені засобами контекстно-вільних граматики. Автор наводить два нових означення щодо квазіблочної симуляції та полублочної симуляції, які використовуються для спрощеного пошуку інваріантів. Для перевірки наведеного методу автор застосовує розроблений власноруч фреймворк CheAPS.

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

В ході дослідження проведено аналіз основних методів верифікації параметризованих моделей, визначено їх основні недоліки та переваги (табл. 1).

Як випливає з наведеного опису переваг і недоліків представлених методів розв'язання задачі ВПМ, більшість аналітичних методів редукції та абстракції має властивості алгоритмізації та завершуваності. Однак область застосування обох груп методів невелика. За необхідності перевірки моделей з іншою топологією потрібне розроблення нових методів, можливо, аналогічних уже відомим.

Символьні методи й методи інваріантів застосовуються до параметризованих моделей з різною топологією, а саме кільцевою, лінійною, деревовидною. Символьні методи застосовуються переважно для перевірки властивостей безпеки, тоді як багато методів інваріантів застосовні також для перевірки властивостей живучості. Обидві групи методів використовують полурозв'язувальні процедури, тому загалом не завершуються.

Для застосування символьних методів опис параметризованої моделі потрібно представити спеціальною мовою методу, яка зазвичай досить сильно відрізняється від мов опису моделей засобів ВМ. Методи інваріантів використовують описи процесів у вигляді РТС, розмічених пропозиціональними змінними, та сумісні з мовами опису моделей засобів ВМ. Методи інваріантів використовують для перевірки специфікації та пошуку контрприкладів наявні засоби ВМ, символьні ж методи вимагають реалізації спеціальних алгоритмів.

ВИСНОВКИ

В роботі проведено дослідження методів верифікації параметризованих моделей, наведено їх основні недоліки та переваги.

Як наслідок, встановлено, що найзручнішим методом верифікації параметризованих моделей, який не потребує додаткових витрат на розроблення мов опису та спеціальних алгоритмів, згідно з вищенаведеним дослідженням є метод інваріантів.

Таблиця 1. Основні недоліки та переваги методів верифікації параметризованих моделей

Методи ВПМ	Недоліки	Переваги
Методи абстракції	Навіть в тих випадках, коли експертам вдається зазначити вид абстрактної моделі A і спосіб абстракції α для класу моделей, область застосовності методів обмежується системами, процеси яких обмінюються маркером [11]; широко-мовними системами [26]; системами з топологією «зірка» [25]. Інакше методи абстракції вимагають підказок від експерта. Наприклад, потрібно явне визначення змінних і предикатів у ТС. Більш того, коли абстракція побудована, специфікація може не виконуватися на абстрактній моделі. З цього не випливає, що специфікація порушується також на моделях параметризованого сімейства Φ . В такому разі потрібна участь експерта для уточнення абстракції, проте уточнення є можливим не для всіх зазначених методів.	Основна перевага методів абстракції полягає в тому, що після знаходження виду абстрактної моделі та відображення моделей сімейства Φ на абстрактну модель перевірка властивостей здійснюється автоматично для будь-якої параметризованої моделі (за винятком методу з [27], де булева абстракція будується за допомогою САДТ). Методи верифікації параметризованих моделей за допомогою абстракції спираються на апарат побудови мулевих абстракцій та абстрактної інтерпретації, який постійно вдосконалюється, тому область застосування методів абстракції до задач ВПМ поступово розширюється.
Символьні методи	Оскільки символьні методи засновані на представленні ТС параметризованого сімейства Φ загалом за допомогою більш виразних мов, ніж РТС, розмічених пропозиціональними змінними, завершувальність методів загалом не гарантується. Іноді завершувальність може бути досягнута модифікацією перетворювача T , наприклад, за допомогою застосування технік додавання додаткових переходів (акселерація, acceleration) і пошуку апроксимацій (розширення, widening). Однак в цьому разі потрібна участь експерта задля зміни перетворювача. Іншим недоліком, що є істотним з точки зору цілей цієї роботи, є вимога повторного опису моделі в параметризованому вигляді мовами, які сильно відрізняються від мов опису моделей, що використовуються в засобах ВМ. Після того як модель розподіленої системи побудована й верифікована для фіксованого числа процесів за допомогою стандартного засобу ВМ, бажано використовувати вибраний рівень абстракції та перевести модель у параметризований вигляд з мінімальним втручанням експерта.	Символьні методи можуть застосовуватися для перевірки властивостей безпеки параметризованих моделей асинхронних розподілених систем з такими топологіями, як кільцева, деревоподібна, повнозв'язна. Позитивною властивістю методів є використання добре відомих у математичній логіці мов і готових засобів виведення для цих мов. Виразні можливості вживаних мов дають змогу описувати не тільки протоколи, параметризовані за числом процесів, але й такі протоколи, що використовують нескінченні типи даних, наприклад цілочисельні лічильники.
Методи редукції	Серед недоліків підходу можна назвати такі. По-перше, аналітичні методи редукції застосовні тільки до двох видів систем, а саме асинхронних систем з кільцевою топологією, процеси яких обмінюються маркером, та асинхронних широкомовних систем з переходами спеціального виду. По-друге, теоретичні оцінки граничного значення для деяких методів залежать від числа станів процесів, тому число станів граничної моделі може бути настільки великим, що в результаті цього перевірка граничної моделі буде практично неможливою.	Безперечними перевагами аналітичних методів редукції є точний опис області застосовності методів, алгоритмізація та завершувальність. За допомогою засобів ВМ досить перевірити всі моделі $M_1, \dots, M_N$ до граничного значення N , яке заздалегідь відоме.
Методи пошуку інваріантів	Тоді, коли не вдається знайти інваріант серед моделей з малим числом процесів, необхідно додатково застосовувати методи абстракції, що вимагають знань експерта. Оскільки автоматичний пошук інваріанта проводиться серед моделей сімейства, загалом для процедури пошуку немає гарантії завершувальності. Однак на практиці інваріанти необхідно шукати серед моделей з досить невеликим числом процесів, оскільки розмір множин станів у моделях параметризованого сімейства збільшується експоненціально зі збільшенням кількості процесів.	У методах інваріантів використовуються спосіб індуктивного доведення й досить простий критерій перевірки інваріанта, тому методи можуть бути автоматизовані. Пошук інваріанта серед моделей сімейства Φ з малим числом процесів відповідає інтуїтивному уявленню інженерів, що розробляють модель, про поведінку моделі. За допомогою методів інваріантів можуть перевірятися параметризовані моделі з нетривіальною топологією, що визначаються мережевими граматами, та різними прототипами процесів. Істотною перевагою є також те, що після отримання інваріанта (або декількох інваріантів в разі мережевої граматики) для перевірки моделі, що містить інваріант, досить застосувати розвинені засоби методу верифікації моделей.

REFERENCES

- [1] Tel, G. (2000). *Introduction to Distributed Algorithms*. Cambridge University.
- [2] Tanenbaum, A., & van Steen, M. (2016). *Distributed Systems: Principles and Paradigms*. CreateSpace Independent Publishing Platform.
- [3] van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing*, 98 (10), 967–1009. DOI: 10.1007/s00607-016-0508-7.
- [4] Emerson, E., & Namjoshi, K. (1995). Reasoning about rings. *22th ACM Conf. on Principles of Programming Languages*, pp. 85–94. DOI: 10.1145/199448.199468.
- [5] Emerson, E. A., & Kahlon, V. (2004). Parameterized model checking of ring-based message passing systems. *Computer Science Logic, 3210/2004*, 325–339. DOI: 10.1007/978-3-540-30124-0_26.
- [6] Thistle, J., & Nazari, S. (2007). Structural conditions for model-checking of parameterized Networks. In *7th International Conference on Application of Concurrency*. Bratislava, Slovak Republic: 2007. DOI: 10.1109/ACSD.2007.32.
- [7] Bloem, R., Jacobs, S., Khalimov, A., & Konnov, I. R. (2015). Decidability of Parameterized Verification. In *Synthesis Lectures on Distributed Computing Theory*. San Rafael: Morgan & Claypool. DOI: 10.2200/S00658ED1V01Y201508DCT013.
- [8] Emerson, E. A., & Kahlon, V. (2003). Exact and efficient verification of parameterized cache coherence protocols. In *12th IFIP Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME)* (pp. 247–262). L'Aquila, Italy. DOI: 10.1007/978-3-540-39724-3_22.
- [9] Chen, X., Yang, Y., DeLisi, M., & Gopalakrishnan, G. (2007). Hierarchical cache coherence protocol verification one level at a time through assume guarantee. In *High Level Design Validation and Test Workshop (HLDVT)*. Piscataway. DOI: 10.1109/HLDVT.2007.4392796.
- [10] Maidl, M. (2001). A unifying model checking approach for safety properties of parameterized systems. In *CAV '01: Proceedings of the 13th International Conference* (pp. 311–323). London, UK: Springer-Verlag. DOI: 10.1007/3-540-44585-4_29.
- [11] Clarke, E., Talupur, M., & Touili, T. (2004). Verification by network decomposition. *CONCUR 2004*, 3170, 276–291. DOI: 10.1007/978-3-540-28644-8_18.
- [12] Pnueli, A., Ruah, S., & Zuck, L. D. (2001). Automatic deductive verification with invisible invariants. *Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2031, 82–97. DOI: 10.1007/3-540-45319-9_7.
- [13] Sistla, A. P., & Gyrus, V. (1999). Parameterized verification of linear networks using automata as invariants. *Formal Aspects of Computing*, 11, 402–425. DOI: 10.1007/s001650050040.
- [14] Abdulla, P. A., Sistla, A. P., & Talupur, M. (2018). Model Checking Parameterized Systems. In *Handbook of Model Checking* (pp. 685–725). Springer, Cham. DOI: 10.1007/978-3-319-10575-8_21.
- [15] Abdulla, P., Jonsson, B., & Nilsson, M. (2004). A survey of regular model checking. *Proc. 15th Int. Conf. on Concurrency Theory*, 3170, 35–48. DOI: 10.1007/978-3-540-28644-8_3.
- [16] Abdulla, P. A., Delzanno, G., Henda, N. B., & Rezine, A. (2007). Regular model checking without transducers (on efficient verification of parameterized systems). *Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 4424, 721–736. DOI: 10.1007/978-3-540-71209-1_56.
- [17] Touili, T. (2001). Regular model checking using widening techniques. *Electronic Notes in Theoretical Computer Science*, 50 (4), 342–356. DOI: 10.1016/S1571-0661(04)00187-2.
- [18] Boigelot, B., Legay, A., & Wolper, P. (2004). Omega-regular model checking. *Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2988, 561–575. DOI: 10.1007/978-3-540-24730-2_41.
- [19] Bouajjani, A., Habermehl, P., & Rogalewicz, A. (2012). Abstract regular (tree) model checking. *International Journal on Software Tools for Technology Transfer*, 14, 167–191. DOI: 10.1007/s10009-011-0205-y.
- [20] Pnueli, A., & Zuck, L. D. (2003). Model-checking and abstraction to the aid of parameterized systems (a survey). *VMCAI 2003: Proceedings of the 4th International*, 4. DOI: 10.1016/j.cl.2004.02.006.
- [21] Delzanno, G. (2000). Verification of consistency protocols via infinite-state symbolic model checking. In *Formal Methods for Distributed System Development (FORTE)* (pp. 171–186). Heidelberg: Springer. DOI: 10.1007/978-0-387-35533-7_11.
- [22] Kesten, Y., & Pnueli, A. (2000). Control and data abstraction: The cornerstones of practical formal verification. *International Journal on Software Tools for Technology Transfer*, 2 (4), 328–342. DOI: 10.1007/s100090050040.
- [23] Krstic, S. (2005). Parameterized system verification with guard strengthening and parameter abstraction. *Automated Verification of Infinite State Systems*.
- [24] Abdulla, P. A., Haziza, F., & Hol'ik, L. (2013). All for the price of few (parameterized verification through view abstraction). In *Intl. Conf. on Verification, Model Checking and Abstract Interpretation (VMCAI)* (Vol. 7737, pp. 476–495). Heidelberg: Springer. DOI: 10.1007/978-3-642-35873-9_28.
- [25] German, S. M., & Sistla, A. P. (1992). Reasoning about systems with many processes. *Journal of the ACM*, 39 (3), 675–735. DOI: 10.1145/146637.146681.
- [26] Ip, C. N., & Dill, D. L. (1997). Verifying systems with replicated components in murphi. *Formal Methods in System Design*, 14 (3), 273–310. DOI: 10.1007/3-540-61474-5_65.
- [27] Lesens, D., & Saidi, H. (1997). Abstraction of parameterized networks. *Electronic Notes in Theoretical Computer Science*, 9, 42. DOI: 10.1016/S1571-0661(05)80429-3.

- [28] Owre, S., Rushby, J., & Shankar, N. (1992). PVS: A prototype verification system. *11th International Conference on Automated Deduction (CADE)*, 607, 748–752. DOI: 10.1007/3-540-55602-8_217.
- [29] Jensen, H. E., & Lynch, N. A. (1998). A proof of Burns' n-process mutual exclusion algorithm using abstraction. *TACAS '98: Proceedings of the 4th International Conference on Tools and Algorithms for Construction and Analysis of Systems*, 409–423. DOI: 10.1007/BFb0054186.
- [30] Calder, M., & Miller, A. (2003). Using spin to analyze the tree identification phase of the IEEE 1394 high performance serial bus (firewire) protocol. *Formal Aspects of Computing*, 247–266. DOI: 10.1007/s001650300004.
- [31] Calder, M., & Miller, A. (2004). Detecting feature interactions: how many components do we need? *Objects, Agents and Features*, 2975, 45–66. DOI: 10.1007/978-3-540-25930-5_4.
- [32] Clarke, E., Talupur, M., & Veith, H. (2006). Environment Abstraction for Parameterized Verification. *VMCAI 2006*, 3855, 126–141. DOI: 10.1007/11609773_9.
- [33] Drechsler, R., & Becker, B. (1998). *Binary Decision Diagrams – Theory and Implementation*. Kluwer Academic Publishers. DOI: 10.1007/978-1-4757-2892-7.
- [34] Nilsson, M. (2005). Structural Symmetry and Model Checking: Ph.D. thesis. Uppsala, Sweden: Uppsala University.
- [35] Kesten, Y., Maler, O., & Marcus, M. (1997). Symbolic model checking with rich assertional languages. In *CAV '97: Proceedings of the 9th International Conference on Computer Aided Verification* (pp. 424–435). London, UK: Springer-Verlag.
- [36] Shahar, E. (2001). Tools and Techniques for Verifying Parameterized Systems: Ph.D. thesis. Israel: Weizmann Institute of Science.
- [37] Henriksen, J. G., Jensen, J. L., & Jorgensen, M. E. (1995). Mona: Monadic second-order logic in practice. In *TACAS 95: Proceedings of the First International Workshop on Tools and Algorithms for Construction and Analysis of Systems* (pp. 89–110). London, UK: Springer-Verlag. DOI: 10.1007/3-540-60630-0_5.
- [38] Emerson, E. A., & Kahlon, V. (2003). Model checking guarded protocols. In *LICS* (pp. 361–370). IEEE Computer Society. DOI: 10.1109/LICS.2003.1210076.
- [39] Batt, G., Page, M., Cantone, I., Goessler, G., Monteiro, P., & de Jong, H. (2010). Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26 (18), 603–610. DOI: 10.1093/bioinformatics/btq387.
- [40] Calder, M., & Miller, A. (2002). Five ways to use induction and symmetry in the verification of networks of processes by model-checking. In *Automated Verification of Critical Systems (AvoCS 2002)* (pp. 29–42).
- [41] Clarke, E., Grumberg, O., & Browne, M. (1986). Reasoning about networks with many identical finite-state processes. In *PODC '86: Proceedings of the fifth annual ACM symposium on Principles of distributed computing* (pp. 240–248). New York, NY, USA: ACM. DOI: 10.1016/0890-5401(89)90026-6.
- [42] Browne, M. C., Clarke, E. M., & Grumberg, O. (1988, July). Characterizing finite kripke structures in propositional temporal logic. DOI: 10.1016/0304-3975(88)90098-9.
- [43] Kurshan, R. P., & McMillan, K. (1989, June). A structural induction theorem for processes. DOI: 10.1145/72981.72998.
- [44] Wolper, P., & Lovinfosse, V. (1989). Verifying properties of large sets of processes with Network Invariants. Access date 02.20.2019, available at https://link.springer.com/content/pdf/10.1007/3-540-52148-8_6.pdf. DOI: 10.1007/3-540-52148-8_6.
- [45] Shtadler, Z., & Grumberg, O. (1990). Network grammars, communication behaviors and automatic verification. In *Proceedings of the International Workshop on Automatic Verification Methods for Finite State Systems* (pp. 151–165). London, UK: Springer-Verlag. DOI: 10.1007/3-540-52148-8_13.
- [46] Clarke, E., Grumberg, O., & Jha, S. (1995). Verifying parameterized networks using abstraction and regular languages. In *6th International Conference on Concurrency Theory* (pp. 395–407). DOI: 10.1007/3-540-60218-6_30.
- [47] Clarke, E., Grumberg, O., & Jha, S. (1997). Verifying parameterized networks. *ACM Transactions on Programming Languages and Systems*, 19 (5), 726–750. DOI: 10.1145/265943.265960.
- [48] Lesens, D., Halbwachs, N., & Raymond, P. (2001). Automatic verification of parameterized linear networks of processes. In *Theoretical Computer Science*, 256 (1–2). DOI: 10.1145/263699.263747.
- [49] Konnov, I. (2012). Parameterized Model Checking by Network Invariants: the Asynchronous Case. *LICS Workshop Algorithmics on Infinite State Systems*. Dubrovnik, Croatia.

© Я. Ю. Дорогий, В. В. Цуркан

Дата надходження статті до редакції: 19.03.2020

Дата затвердження статті до друку: 17.04.2020