

# Zastosowanie metod formalnych

Karol Kozłowski  
Wydział Elektryczny  
Politechnika Warszawska

Katarzyna Mielęcka  
Wydział Elektryczny  
Politechnika Warszawska

Jan Gawroński  
Wydział Elektryczny  
Politechnika Warszawska

Piotr Głowacki  
Wydział Elektryczny  
Politechnika Warszawska

## I. WPROWADZENIE DO METOD FORMALNYCH

Metody formalne to matematyczne techniki wspomagające projektowanie i analizę systemów informatycznych. Ich istotę stanowią trzy kluczowe elementy:

- **Specyfikacja** - precyzyjny opis systemów za pomocą języków matematycznych (np. logika temporalna, rachunek procesów), eliminujący niejednoznaczności typowe dla dokumentacji tekstowej.
- **Weryfikacja** - formalne dowodzenie poprawności systemów poprzez:
  - model checking (np. narzędzie SPIN do weryfikacji protokołów).
  - dowody twierdzeń (np. w systemach Coq, Isabelle).
- **Automatyzacja analizy** - wykrywanie sprzeczności i luk na etapie modelowania (np. weryfikacja struktur wskaźnikowych w PVS).

Główną zaletą metod formalnych są **matematyczne gwarancje poprawności**, szczególnie istotne w systemach krytycznych (medycznych, transportowych). Pozwalają one identyfikować złożone błędy, takie jak zakleszczenia czy warunki wyścigu, które często wymykają się tradycyjnym metodom testowania.

## II. SYSTEM PVS

System PVS (Prototype Verification System) to zaawansowane narzędzie do opracowywania i weryfikacji formalnych specyfikacji, które pozwala na modelowanie i dowodzenie poprawności złożonych systemów. PVS wyróżnia się rozbudowaną składnią i możliwością operowania w logice wyższego rzędu, co umożliwia definiowanie własnych typów, podtypów oraz tworzenie teorii parametryzowanych. Dzięki temu system jest szczególnie przydatny do weryfikacji struktur dynamicznych, takich jak układy wskaźnikowe, gdzie często dochodzi do wycieków pamięci, nieprawidłowych dereferencji czy zakleszczeń w systemach współbieżnych. PVS umożliwia formalne określenie niezmienników i warunków spójności, a następnie ich automatyczną lub półautomatyczną weryfikację.

## III. FORMALNA WERYFIKACJA Z UŻYCIEM TLA+

TLA+ (Temporal Logic of Actions) to język specyfikacji formalnej zaprojektowany przez Lesliego Lamporta do opisu i analizy systemów współbieżnych i rozproszonych. Umożliwia modelowanie systemu jako zbioru zmiennych oraz akcji – przejść między stanami – z wykorzystaniem logiki temporalnej

i matematyki zbiorów. Język ten pozwala na tworzenie deklaratywnych specyfikacji, które opisują zamierzone zachowanie systemu niezależnie od jego implementacji. Weryfikacja poprawności odbywa się za pomocą narzędzia TLC, które stosuje technikę model checking, przeszukując przestrzeń stanów w celu sprawdzenia, czy system spełnia zadane własności.

TLA+ wykorzystuje trzy główne podejścia formalne. Pierwszym z nich jest matematyczne modelowanie systemu – precyzyjne określenie przestrzeni stanów i możliwych akcji, z użyciem obiektów takich jak zbiory, funkcje czy sekwencje. Drugim jest weryfikacja inwariantów, czyli logicznych warunków, które muszą być spełnione w każdym stanie systemu, co pozwala wykrywać istotne błędy, np. naruszenia spójności lub niedozwolone operacje. Trzecim podejściem jest analiza własności temporalnych w logice – umożliwiająca weryfikację zachowań systemu w czasie, takich jak osiągalność pewnych stanów, brak zakleszczeń czy deterministyczne następstwo zdarzeń. TLA+ znajduje zastosowanie w projektowaniu i weryfikacji złożonych systemów rozproszonych w środowiskach przemysłowych, m.in. w Amazonie, Microsoftzie i Google, gdzie wykorzystywany jest do analizy protokołów, systemów bazodanowych i usług chmurowych.

## IV. KOLEJNICTWO

Metody formalne odgrywają kluczową rolę w rozwoju systemów krytycznych, gdzie błędy mogą prowadzić do katastrofalnych konsekwencji. W niniejszym rozdziale omówiono zastosowanie metod formalnych na przykładzie systemów kolejowych. Metody formalne w kolejnictwie pozwalają zweryfikować pozwalają zweryfikować, czy sygnalizacja **uniemożliwi kolizję pociągów** w dowolnych warunkach, poprawność działania jednostek **DCU (Door Control Unit)** czy **TCU (Traction Control Unit)**. W praktyce stosuje się różne techniki formalne, dopasowane do specyfiki problemów:

- Model checking - np. narzędzia UPPAAL, NuSMV czy SPIN. Są one szczególnie przydatne w analizie systemów czasu rzeczywistego, gdzie istotne są ograniczenia czasowe i współbieżność. W kolejowym systemie ERTMS/ETCS Level 3 wykorzystano UPPAAL SMC do weryfikacji ruchu pociągów w trybie „moving block”, uwzględniając stochastyczne czynniki, takie jak opóźnienia komunikacji. Wadą tego podejścia jest natomiast rozmiar modelu, który może nie zmieścić się w pamięci komputera w przypadku modelowania dużych systemów.
- Theorem proving - np. Isabelle/HOL czy Coq. Wymagają one głębszej wiedzy matematycznej, ale pozwalają na

analizę systemów o nieskończonej przestrzeni stanów. W przypadku chińskiego systemu CTCS zastosowano Isabelle/HOL do formalnej weryfikacji algorytmów kontroli ruchu. Z kolei na 14. linii metra paryskiego zastosowanie metody B pozwoliło na modernizację systemu sygnalizacji dla zwiększenia częstotliwości kursowania pociągów przy zachowaniu bezpieczeństwa.

- Statyczna analiza i abstrakcyjna interpretacja - np. narzędzia oparte na technikach przybliżonych, które balansują między precyzją a wydajnością.

Badania systematyczne (ter Beek 2025, przegląd 328 publikacji z lat 1989-2020) wykazały, że 65% prac w dziedzinie kolejnictwa wykorzystuje ściśle formalne metody, podczas gdy tylko 9% ogranicza się do półformalnych. Co istotne, w projektach przemysłowych odsetek metod półformalnych rośnie (51% vs. 49%), co wynika z konieczności komunikacji między inżynierami a interesariuszami.

#### A. Metoda B

Metoda formalna B to podejście do projektowania i weryfikacji systemów komputerowych, oparte na matematycznych podstawach. Metoda ta została wprowadzona przez Jeana-Raymonda Abrial'a i opiera się głównie na teorii zbiorów oraz logice predykatów. Podstawowym elementem metody B jest tzw. „**abstrakcyjna maszyna stanowa**”, która stanowi formalny model systemu. Abstrakcyjna maszyna stanowa składa się z opisu stanu oraz operacji, które mogą zmieniać ten stan. Stan systemu definiowany jest przy użyciu zmiennych, których wartości są elementami określonych zbiorów. Operacje są opisywane za pomocą precyzyjnych reguł, określających warunki, jakie muszą być spełnione przed ich wykonaniem (tzw. prewarunki) oraz efekt, jaki zostanie osiągnięty po ich wykonaniu (tzw. postwarunki).

Jednym z głównych atutów metody B jest możliwość stopniowego rozwoju systemu poprzez proces tzw. „**udoskonalania**”. Udoskonalanie polega na stopniowym przechodzeniu od wysokopoziomowej, abstrakcyjnej specyfikacji do bardziej szczegółowego, implementacyjnego opisu systemu. W każdej fazie udoskonalania wprowadzane są kolejne szczegóły, przy jednoczesnym zachowaniu dowodzonej poprawności poprzednich etapów. Taki proces umożliwia wykrywanie potencjalnych błędów już na etapie specyfikacji, co znacząco zmniejsza ryzyko pojawienia się krytycznych usterek podczas późniejszych faz rozwoju. Dopiero na samym końcu następuje **generacja kodu** w języku docelowym.

Metoda B jest wykorzystywana przede wszystkim w systemach wymagających wysokiego poziomu bezpieczeństwa i niezawodności, takich jak **systemy sterowania ruchem kolejowym, systemy bankowe, oprogramowanie dla lotnictwa czy systemy wbudowane**.

### V. SYSTEMY WBUDOWANE

Metody formalne odgrywają istotną rolę w zapewnieniu niezawodności oraz bezpieczeństwa systemów wbudowanych. Jest to szczególnie ważne, gdyż systemy te są powszechnie stosowane w krytycznych sektorach takich jak motoryzacja,

lotnictwo czy medycyna, gdzie każda awaria może mieć poważne konsekwencje. Dzięki zastosowaniu metod formalnych możliwe staje się precyzyjne modelowanie działań systemów, co umożliwia szczegółową analizę ich działania w różnych warunkach. Co więcej, metody te pozwalają na weryfikację zgodności systemów z rygorystycznymi normami i standardami bezpieczeństwa, minimalizując ryzyko wystąpienia błędów. Ważnym aspektem jest również walidacja poprawności działania systemów w różnorodnych scenariuszach eksploatacyjnych. W efekcie, wdrożenie metod formalnych w procesie projektowania i testowania systemów wbudowanych znacząco redukuje ryzyko wystąpienia awarii, chroniąc użytkowników przed potencjalnie katastrofalnymi skutkami błędów systemowych.

Rozszerzony Automat Skończony z Czasem Rzeczywistym (RT-EFSM) jest efektywną metodą formalną stosowaną w testowaniu oprogramowania wbudowanego działającego w czasie rzeczywistym, szczególnie w sektorach o kluczowym znaczeniu, takich jak lotnictwo, medycyna czy transport. RT-EFSM, w przeciwieństwie do tradycyjnych automatów FSM czy EFSM, pozwala na szczegółowe modelowanie ograniczeń czasowych oraz obsługę złożonych interakcji współbieżnych i wejściowo-wyjściowych. Metoda ta opiera się na precyzyjnej definicji stanów, zdarzeń oraz przejść, które uwzględniają zarówno zmienne środowiskowe, jak i czasowe warunki strażnicze. Dzięki wykorzystaniu RT-EFSM możliwe jest tworzenie klas równoważności przejść z ograniczeniami czasowymi, na podstawie których generowane są szczegółowe sekwencje i przypadki testowe. Praktyczne zastosowanie metody RT-EFSM na przykładzie testowania systemu nawigacji bezwładnościowo-GPS wykazało jej wysoką skuteczność, prowadząc do wykrycia istotnych błędów logicznych, funkcjonalnych i czasowych. Połączenie formalnego modelowania RT-EFSM z automatyzacją procesu testowania znacząco zwiększa dokładność i efektywność weryfikacji krytycznych systemów wbudowanych.

Sieci Petriego z Czasem i Kolorami (TCPN) są efektywną metodą formalną wykorzystywaną w procesie weryfikacji systemów wbudowanych działających współbieżnie oraz zależnych od precyzyjnych ograniczeń czasowych. Ze względu na swoją formalną semantykę matematyczną i intuicyjną graficzną reprezentację, TCPN pozwalają na precyzyjne odwzorowanie zachowań systemów, których poprawność jest kluczowa, na przykład w sektorach transportu czy automatyki przemysłowej. Modelowe podejście wykorzystujące TCPN polega na transformacji diagramów stanów UML (które są językiem półformalnym) do formy sieci Petriego z czasem i kolorami, umożliwiając tym samym dokładną analizę dynamicznych i czasowych aspektów działania systemu. Metoda ta została z powodzeniem zweryfikowana w praktyce, na przykładzie kontrolera świateł drogowych, gdzie przy użyciu narzędzia CPN Tools potwierdzono spełnienie krytycznych wymagań bezpieczeństwa, takich jak wykluczenie jednoczesnego świecenia świateł w kolizyjnych kierunkach ruchu. Dzięki dalszym pracom, polegającym na rozwijaniu reguł transformacji i tworzeniu specjalistycznych narzędzi,

metoda ta ma potencjał znacząco zwiększyć niezawodność oraz skalowalność weryfikacji systemów wbudowanych.

Oprócz opisanych metod, w systemach wbudowanych szeroko stosuje się również inne podejścia formalne, takie jak modelowanie z wykorzystaniem metody B. Dzięki swojej precyzji i możliwości dowodzenia poprawności, metoda ta znajduje zastosowanie szczególnie w projektowaniu komponentów krytycznych dla bezpieczeństwa.

## VI. BIBLIOGRAFIA

### LITERATURA

- [1] Leslie Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*, Addison-Wesley, 2002.
- [2] Chris Newcombe et al., *How Amazon Web Services Uses Formal Methods*, Communications of the ACM, 2015.
- [3] Igor Konnov, Jure Kukovec, Thanh-Hai Tran, *TLA+ Model Checking Made Symbolic*, CAV 2019.
- [4] Hillel Wayne, *Practical TLA+: Planning Driven Development*, Lospinato Books, 2018.
- [5] S. Poreda, *Wykorzystanie metod formalnych do specyfikacji struktur wskaźnikowych*, Uniwersytet Warszawski, 2023.
- [6] Sławomir Lasota, *Weryfikacja protokołu Needhama-Schroedera przy użyciu narzędzi SPIN i UPPAAL*, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski,
- [7] Igor Wojnicki, *Weryfikacja własności systemów współbieżnych z użyciem metod formalnych*, Praca doktorska, Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, 2019,
- [8] Abrial, J.R., Lee, M.K.O., Neilson, D.S., Scharbach, P.N., Sørensen, I.H. *The B-method*. VDM '91 Formal Software Development Methods, 1991. <https://doi.org/10.1007/BFb0020001>
- [9] Dominique Cansell, Dominique Méry. *Tutorial on the event-based B method*. IFIP FORTE 2006, Paris, 2006.
- [10] A. Abdullah, *CENELEC EN 50128: Railway Applications - Communication, signaling and processing systems, Software for railway control and protection systems*, Frankfurt University of Applied Sciences, 2020.
- [11] Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, Michael Deardeuff, *How Amazon Web Services Uses Formal Methods*, Communications of the ACM, Vol. 58, No. 4, pp. 66–73, 2015.
- [12] Y. Yin, B. Liu and H. Ni, *Real-time embedded software testing method based on extended finite state machine*, in Journal of Systems Engineering and Electronics, vol. 23, no. 2, pp. 276-285, April 2012, doi: 10.1109/JSEE.2012.00035.
- [13] F. Moin, F. Azam and M. W. Anwar, *A Model-driven Approach for Formal Verification of Embedded Systems Using Timed Colored Petri Nets*, 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 2018, pp. 2580-2584, doi: 10.1109/CompComm.2018.8780731.
- [14] J. R. Abrial, *Modeling in Event-B: System and Software Engineering*, Cambridge University Press, 2010. Noguchi, Kenichiro. *Application Of Formal Methods For Designing A Separation Kernel For Embedded Systems*. Zenodo, 2010.
- [15] M. H. ter Beek, *Models for formal methods and tools: the case of railway systems*, Springer, 2025. <https://doi.org/10.1007/s10270-025-01276-3>
- [16] A. Ferrari, M. H. ter Beek, *Formal Methods in Railways: A Systematic Mapping Study*, ACM Computing Surveys, Volume 55, Issue 4, 2022, <https://doi.org/10.1145/3520480>