Weryfikacja modelowa z zastosowaniem logiki temporalnej

Zagadnienie kierunkowe nr 8

Karolina Wiśniewska

Agenda

Logika temporalna

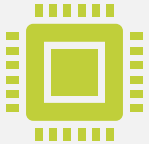
Co to jest weryfikacja modelowa

Weryfikacja modelowa z zastosowaniem logiki temporalnej

Podsumowanie

Logika

Logika to nauka argumentowania. Klasyczna logika zdań posługuje się zbiorem praw logicznych i nadaje logiczny stan dla każdego zdania.



Logika modalna jest matematycznie określonym odpowiednikiem logiki klasycznej.

Logika temporalna

- Logika temporalna pozwala wnioskować z uwzględnieniem czasu, przypisując wartości *prawda*/ *fałsz* do wyrażeń logiki modalnej
- Posiada operatory temporalne
- Opisuje jak statyczne stany zmieniają się w czasie
- Główne zastosowanie w modelowaniu i weryfikacji systemów

Podstawowe rodzaje logiki temporalnej

LTL (Linear Temporal Logic) - czas jest dyskretny, liniowy. Opisuje jedną możliwą linię czasu

CTL (Computation Tree Logic) - czas dyskretny, rozgałęziony. Opisuje wszystkie możliwe linie czasu

Operatory logiki temporalnej

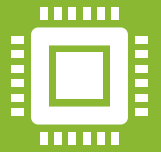
Operatory temporalne:

- **U** „dopóki”
- **X** „następnie”
- **G** „zawsze”
- **F** „kiedyś”

Operatory ścieżkowe

- A – dla każdej możliwej ścieżki
- E – dla pewnej możliwej ścieżki

Weryfikacja modelowa



Formalna technika weryfikacji skończonych systemów współbieżnych, automatów skończenie stanowych

Składa się z trzech podstawowych kroków: modelowania, specyfikacji oraz weryfikacji

Etapy weryfikacji modelowej

Modelowanie

Tworzenie formalnego modelu funkcji systemu w postaci automatu

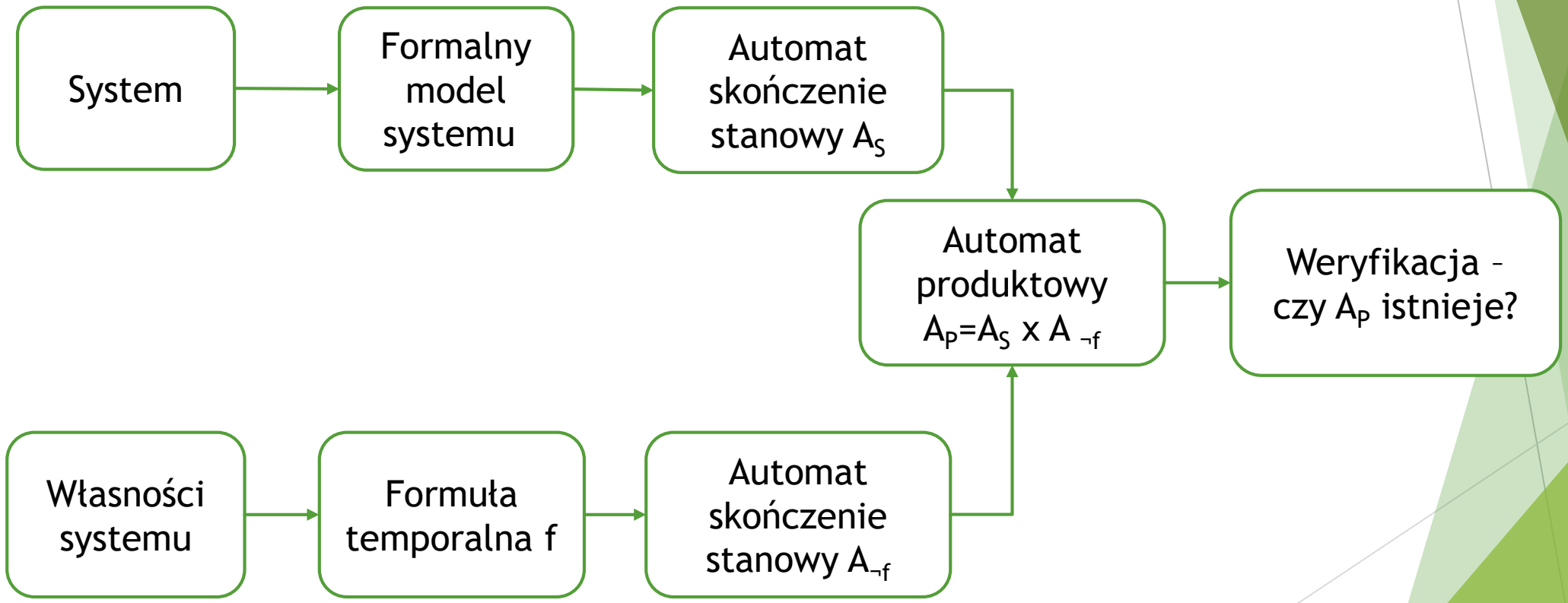
Specyfikacja

Określenie własności, które ma spełniać system w formie reguł logiki temporalnej

Weryfikacja

Sprawdzenie czy system spełnia własności

Algorytm modelowej weryfikacji systemu



Najważniejsze weryfikowane własności

- Osiągalność - czy "pożądany" stan *p* w końcu zostanie osiągnięty
  LTL: F*p*
  CTL: EF*p*
- Bezpieczeństwo- czy "niechciany" stan *q* systemu nigdy nie zostanie osiągnięty
  LTL: G ¬*q*
  CTL: AG ¬*q*
- Odpowiedź – czy *p* jest spełnione od czasu do czasu
  LTL: GF*p*
  CTL: EGF*p*
- Trwałość – od pewnego momentu *p* jest zawsze spełnione
  LTL: FG*p*
  CTL:EFG*p*
- Żywotność – *q* jest osiągalne jakoś skutek *p*
  LTL:G(*p* => F*q*)
  CTL: EG(*p* => F*q*)

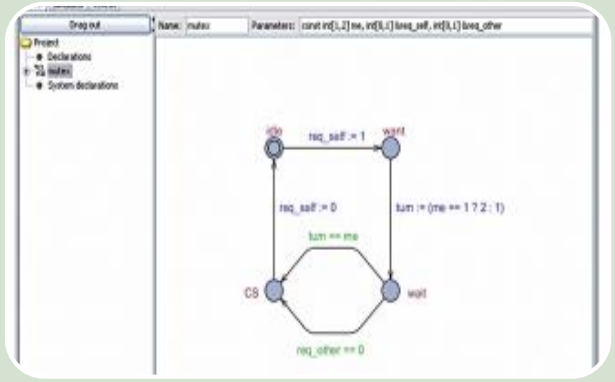
Narzędzia modelowej weryfikacji systemu - UPPAAL

SŁUŻY DO MODELOWANIA I ANALIZY SYSTEMÓW CZASU RZECZYWISTEGO, W TYM PROGRAMÓW WSPÓŁBIEŻNYCH.

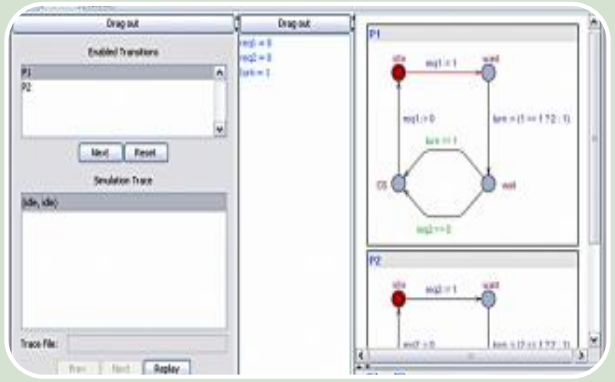
POZWALA NA GRAFICZNE MODELOWANIE JAKO SKOŃCZENIE STANOWE AUTOMATY ORAZ UŻYWANIE AUTOMATÓW CZASOWYCH

UDOSTĘPNIA GRAFICZNE SYMULOWANIE MOŻLIWYCH PRZEBIEGÓW AUTOMATÓW

POWALA NA SPECYFIKOWANIE WŁASNOŚCI SYSTEMU JAKO FORMUŁ CTL ORAZ NA WERYFIKOWANIE PEWNYCH WŁASNOŚCI MODELU.



Modelowanie — zbuduj model systemu jako automat lub automaty.



Symulowanie — krok po kroku sprawdź, czy model zachowuje się prawidłowo. Zapisz własności systemu jako formuły logiczne CTL.

Weryfikowanie — automatycznie zweryfikuj prawdziwość zapisanych formuł.

Narzędzia modelowej weryfikacji systemu - NuSMV

Służy do modelowania, symulowania i weryfikowania skończenie stanowych systemów czasu rzeczywistego

Posługuje się specjalnym językiem definicji automatów skończenie stanowych

Może być używany do badania systemów deterministycznych i niedeterministycznych oraz synchronicznych i niesynchronicznych

NuSMV

- Modelowanie - poprzez definicję modelu przez skrypt w języku NuSMV w postaci skończenie stanowego automatu.
- Symulowanie – pozwala na ręczną symulację możliwych przebiegów automatów, możliwość wyboru ścieżki stanów i jej długości
- Weryfikacja – jest automatyczna, dostępne są logiki LTL, CTL i ich modyfikacje oraz PSL. Formuły opisują specyfikację systemu. Weryfikacja formuły zwraca wartość true lub false. Dla negatywnego wyniku podawany jest kontrprzykład.

Zalety i ograniczenia

Zalety

- W pełni automatyczna
- Kontrprzykład, gdy uzyskujemy negatywną odpowiedź

Ograniczenia

- Złożoność obliczeniowa – eksplozja stanów
- Weryfikacja modelu a nie samego systemu
- Błędy w narzędziach

Źródła

- Wykłady dr inż. Pawła Głuchowskiego, http://pawel.gluchowski.staff.iiar.pwr.edu.pl/?page_id=1458 [26.04.2019]
- Edmund M. Clarke, E. Allen Emerson, Joseph Sifakis; *Model Checkin: Algorithmic Verification and Debugging*, Communications od the ACM [11.2009]
- Aftab Ali Haider, Aamer Nadeem, *A Survey of Model Checking Tools using LTL or CTL as temporal Logic and Generating Counterexamples*.

Dziękuję za uwagę