Pytanie: K8
Temat: Weryfikacja modelowa z wykorzystaniem logiki temporalnej
Autor opracowania: Karolina Wiśniewska

1. Czas

Informacje o czasie:

- Kontinuum nieprzestrzenne, gdzie zdarzenia zachodzą w nieodwracalnej kolejności
- Jeden z wymiarów kontinuum czaso- przestrzennego w fizyce
- Czas ciągły (dla każdych dwóch momentów istnieje pewien moment między nimi
- Czas dyskretny (istnieją dwa takie momenty, między którymi nie istnieje żaden moment
- Czas skończony (przeszłość lub przyszłość jest ograniczona do pewnego momentu)
- Czas obustronnie skończony (przeszłość i przyszłość są ograniczone do pewnych momentów)
- Czas nieskończony (przyszłość i przeszłość nie są ograniczone)
- Czas liniowy (istnieje tylko jeden wariant przepływu czasu)
- Czas rozgałęziony (istnieją różne warianty przepływu czasu, mające wspólną część ze sobą
- Czas równoległy (istnieją różne warianty przepływu czasu, niemające ze sobą żadnej części wspólnej

2. 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.

2.1. Logika temporalna

Pozwala wnioskować z uwzględnieniem czasu, przypisując wartości *prawda/ fałsz* do wyrażeń logiki modalnej, umieszczając je w strukturze czasowej. Posiada operatory temporalne. Opisuje jak statyczne stany zmieniają się w czasie.

Główne zastosowanie logiki znajduje się w modelowaniu i weryfikacji systemów.

Rodzaje logiki temporalnej

LTL (Linear Temporal Logic)

- **Czas: dysskretny, lewostronnie skończony, liniowy, punktowy**

CTL (Computation Tree Logic)

- **Czas: dyskretny, lewostronnie skończony, rozgałęziony, punktowy**

RTCTL (Real-Time Computation Tree Logic)

- Odmiana CTL, gdzie wartości czasu dane są ilościowo jako stałe

PRTCTL (Parametrised Real-Time Computation Tree Logic)

- Wersja CTL, gdzie wartości czasu dane są ilościowo jako zmienne

ITL (Interval Temoporal Logic)

- Czas: dyskretny, skończony lub nieskończony, liniowy, przedziałowy

Operatory temporalne

Podstawowe operatory temporalne • U "dopóki" • X "następnie" • G "zawsze" • F "kiedyś	W logice CTL pojawiają się dwa operatory ścieżkowe A - dla **każdej** możliwej ścieżki E - dla **pewnej** możliwej ścieżki

3. Weryfikacja modelowa

Automatyczna i wyczerpująca weryfikacja danego modelu systemu pod kątem spełnienia przez niego, zadanych specyfikacji. Jest to metoda automatycznego sprawdzania poprawności właściwości automatów skończonych. Zawiera wytyczne dotyczące bezpieczeństwa i osiągalności.

Jest to formalna technika weryfikacji skończonych systemów współbieżnych, automatów skończenie stanowych. Składa się z trzech podstawowych kroków: modelowania, specyfikacji i weryfikacji.

Co to jest skończony system współbieżny?
To program składający się z procesów, które:

- Są wykonywane w tym samym czasie
- Mogą współdzielić pewne zasoby, np. zmienne
- Mogą wzajemnie na siebie oddziaływać na siebie

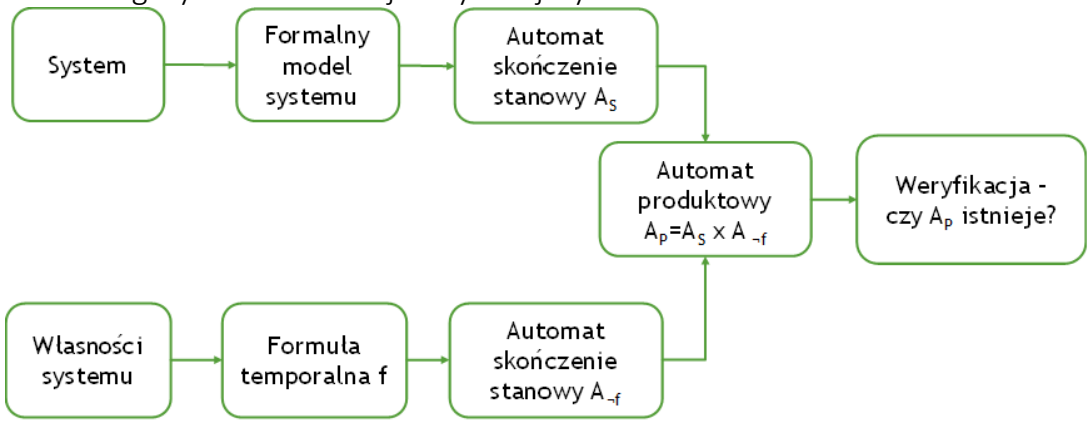
Co to jest automat skończenie stanowy?
To abstrakcyjna maszyna stanowa, która:

- składa się ze skończonej liczby stanów i przejść między nimi
- ma stany początkowe i może mieć stany końcowe
- jest automatem, w którym przejścia między stanami są jednoznacznie opisane funkcją przejścia

3.1. Weryfikowane własności modelu:

- ✓ Osiągalność - czy "pożądany" stan p w końcu zostanie osiągnięty
  LTL: Fp
  CTL: EFp
- ✓ 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: GFp
  CTL: EGFp
- ✓ Trwałość – od pewnego momentu p jest zawsze spełnione
  LTL: FGp
  CTL:EFGp
- ✓ Żywotność – q jest osiągalne jakoś skutek p
  LTL: G(p => Fq)
  CTL: EG(p => Fq)

3.2. Algorytm modelowej weryfikacji systemu



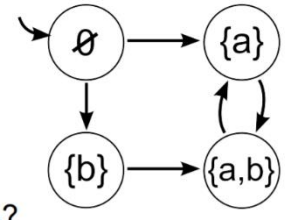
1. Zbuduj skończenie stanowy automat A_S dla modelu systemu S
2. Zapisz własności systemu jako formułę f
3. Zbuduj skończenie stanowy automat $A_{\neg f}$ dla formuły ¬f
4. Zbuduj automat produktowy dla $A_P = A_S \times A_{\neg f}$
5. Zweryfikuj, czy istnieje A_P

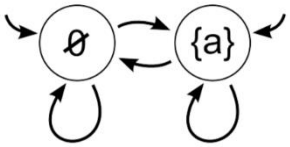
- $A_{\neg f}$ powinien akceptować tylko takie sekwencje stanów programu S, które spełniają formułę ¬f
- Jeśli istniej taka sekwencja stanów w A_S, która odpowiada formule ¬f, możliwe jest takie wykonanie programu, które nie spełnia podanych własności.
-

Przykład z wykładu 4 (strona 23) dr Głuchowskiego:

Prosty przykład

- Automat A_S modeluje system, gdzie każdy stan etykietowany jest tymi wszystkimi wyrażeniami, które są w nim prawdziwe: *a* i *b*. (∅ – *a* i *b* są fałszywe)
- Czy własność $f \equiv EFb$ jest spełniona dla A_S?
- Automat $A_{\neg f}$ modeluje formułę $\neg f \equiv \neg EFb \equiv AG\neg b$.
- W A_S nie istnieje żaden nieskończony przebieg $A_{\neg f}$ — własność f jest spełniona.



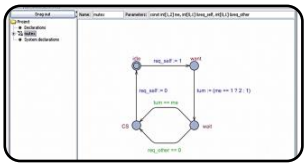


Zauważ, że własność AF*b* *też jest spełniona.*

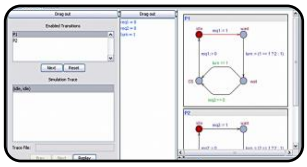
3.3. Narzędzia modelowej weryfikacji

3.3.1. UPPAAL

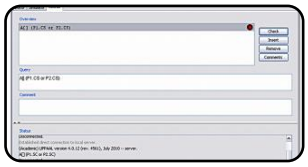
- Służy do modelowania i analizy systemów czasu rzeczywistego, w tym programów współbieżnych
- Pozwala na graficzne modelowanie jako skończenie stanowe automaty oraz używanie automatów czasowych
- Udostępnia graficzne symulowanie możliwych przebiegów automatów
- Powala na specyfikowanie własności systemu 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.

3.3.2. NuSVM

- 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

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.

3.4. Zalety i ograniczenia

3.4.1. Zalety

- W pełni automatyczna
- Kontrprzykład, gdy uzyskujemy negatywną odpowiedź

3.4.2. Ograniczenia

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