Praktyczne metody weryfikacji

Sławomir Lasota

Uniwersytet Warszawski

semestr letni 08/09

I. Motywacja czyli po co?

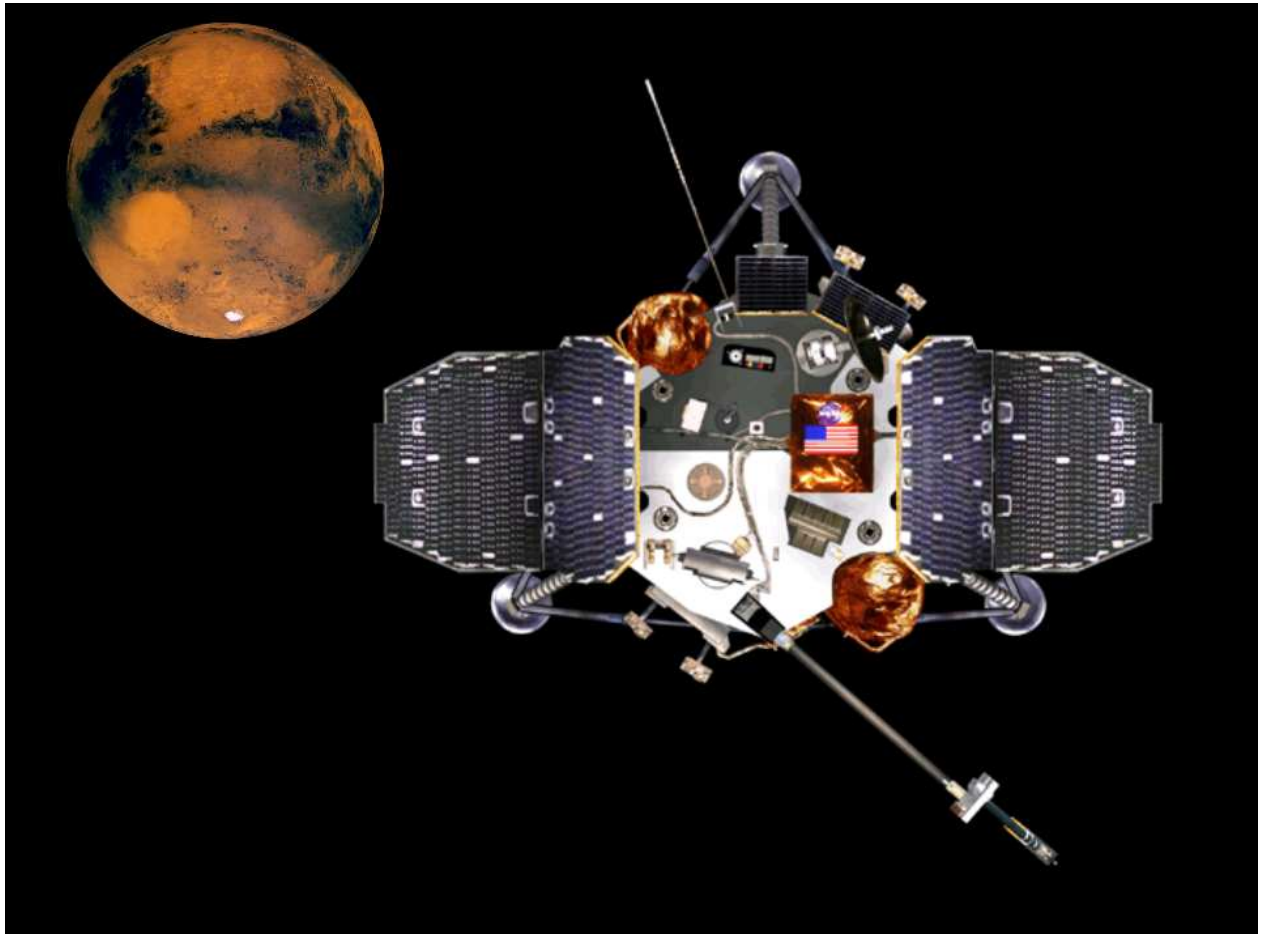
Arianne 5, czerwiec 1996

– nieobsłużony wyjątek

– szacunkowy koszt:

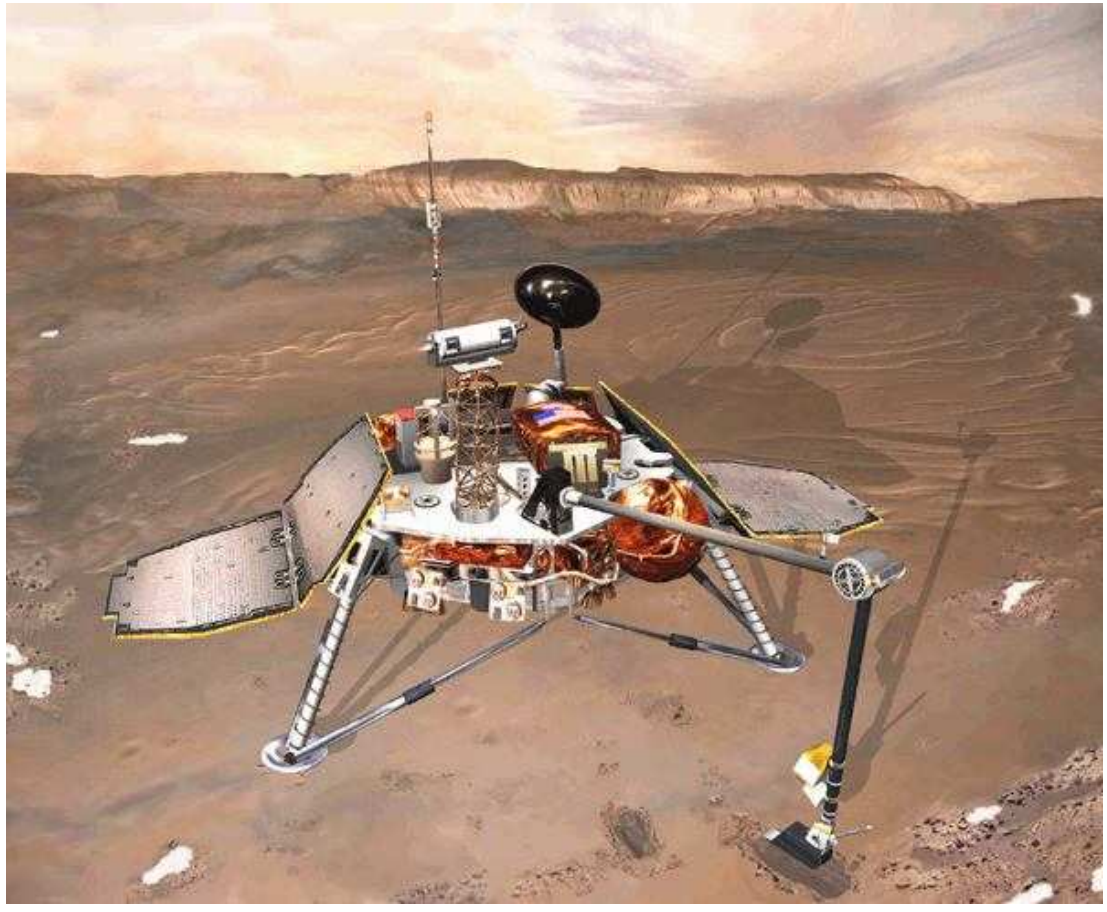
600 mln euro

Mars Polar Lander, styczeń 1999



nasa.gov

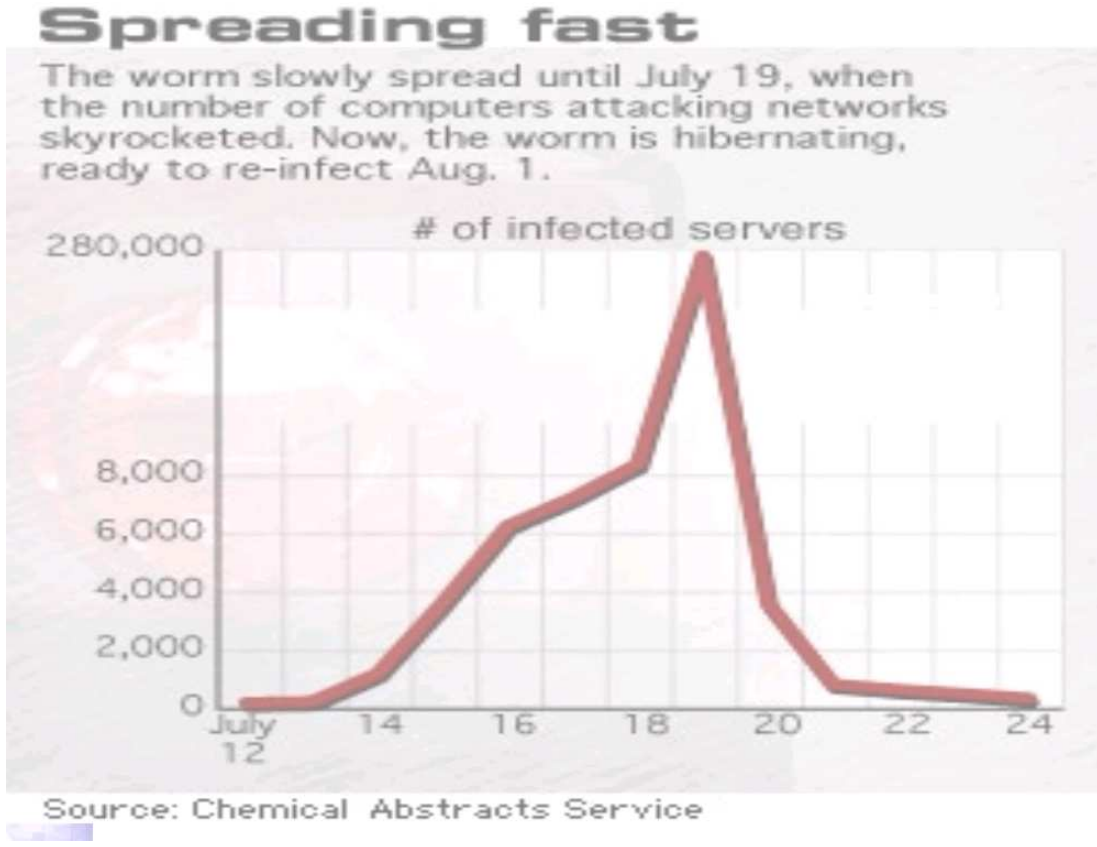
grudzień 1999



wikipedia.org

– awaria z powodu niezainicjowanej zmiennej

Code Red, lipiec 2001

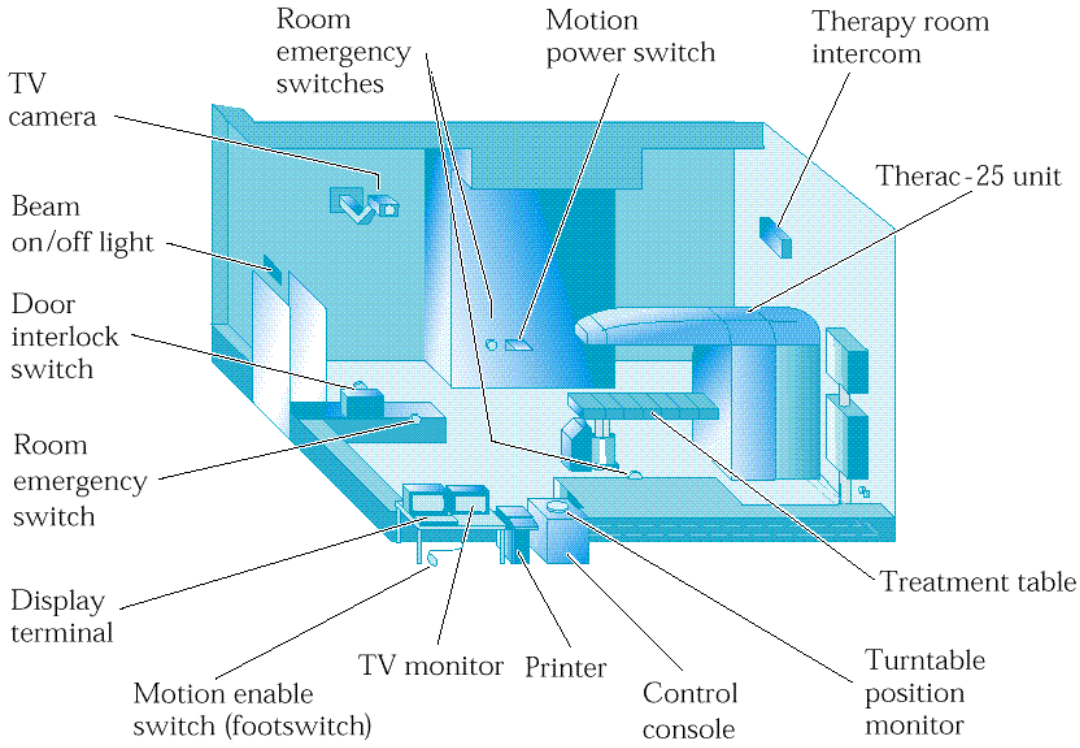


news.cnet.com

– usterka w serwerze Microsoft Internet Information Server (przepełnienie bufora)

– szacunkowy koszt: 2.5 miliarda USD

Therac-25, 1985-87



www.cs.jhu.edu/~pari

– „race condition”

– przynajmniej 6 ofiar

Błędy

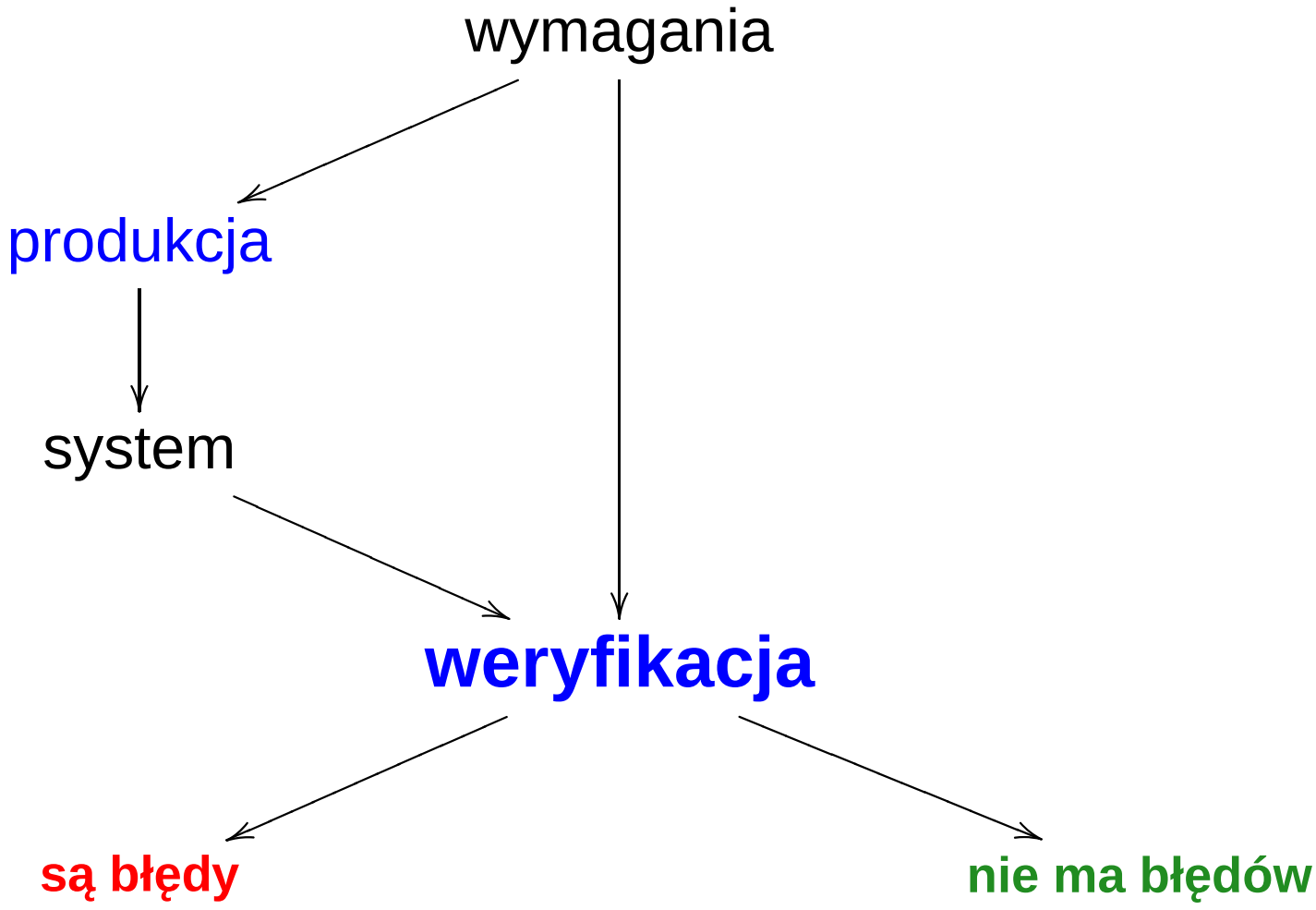
- kosztowne
- nieakceptowalne (ang. *safety-critical*)

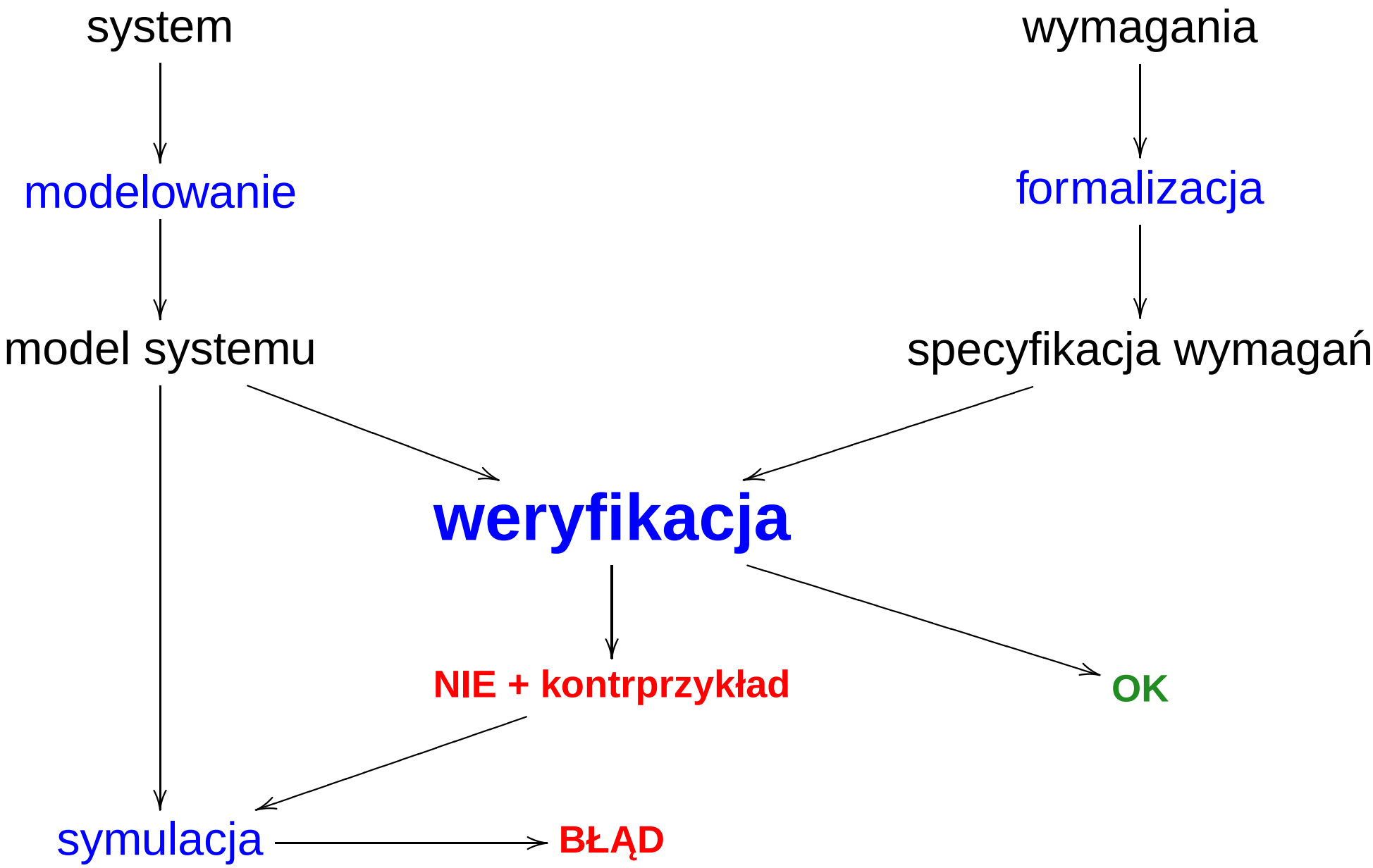
Rozwiązanie:

formalna weryfikacja = dowód poprawności

II. Weryfikacja, ale jak?

Weryfikacja a posteriori




system
wymagania
modelowanie
formalizacja
model systemu
specyfikacja wymagań
weryfikacja
NIE + kontrprzykład
OK
symulacja
BŁĄD

Weryfikacja modelowa (ang. *model checking*)

– model M – możliwe zachowania

(abstrakcja rzeczywistego systemu)

– specyfikacja własności ϕ – dopuszczalne zachowania

– automatycznie sprawdzamy, czy

$$M \models \phi$$

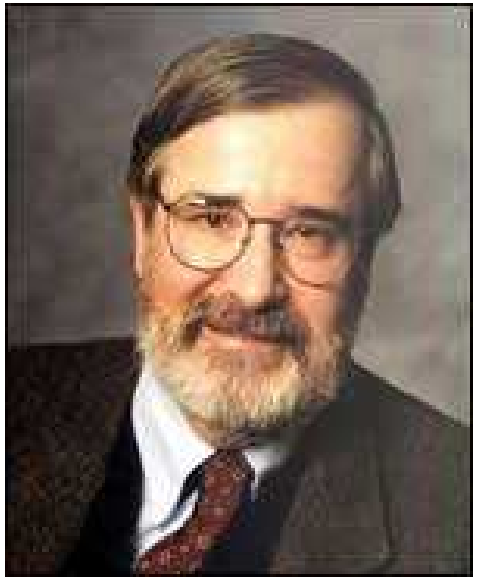
Przykładowe własności ϕ:

- bezpieczeństwo: wszystkie stany osiągalne spełniają dany wymóg
- żywotność: zawsze osiągniemy stan, w którym dany wymóg jest spełniony
- sprawiedliwość (ang. *fairness*): pożądana własność będzie zachodzić nieskończenie wiele razy
- …

Cechy charakterystyczne

- model formalny (stany i przejścia)
- analiza dynamiczna (przeszukiwanie stanów osiągalnych)
- wymagania = formuła temporalna
- metoda w pełni automatyczna
- kontrprzykład (gdy odpowiedź negatywna)

Nagroda Turinga 2008



Ed Clarke

Allen Emerson

Joseph Sifakis

- [Clarke, Emerson 1981]
- [Queille, Sifakis 1982]

Inne podejścia

- testowanie
- dowodzenie poprawności (proof checker)
- audyt kodu źródłowego
- analiza statyczna
- …

Weryfikacja modelowa

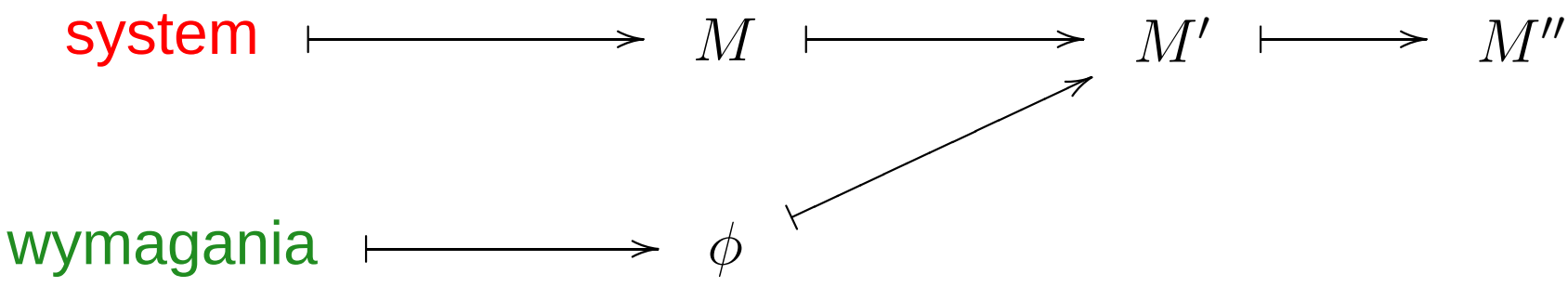
- dowodzenie twierdzeń: $M \models \phi$
- w pełni automatyczne
- nie wymaga pracy eksperta

vs

Weryfikacja interakcyjna

- dowodzenie twierdzeń
- narzędzie wspomagające (ang. *proof checker*)
- wymaga pracy eksperta

Od rzeczywistości do modelu



$M'' \models \phi$?

– odpowiedni poziom abstrakcji

– proces tylko częściowo automatyczny

Weryfikujemy nie system, lecz jego model!

WERYFIKACJA = sprawdzenie, czy wytwarzamy to, co trzeba.

WALIDACJA = sprawdzenie, czy weryfikujemy to, co trzeba.

Dziedziny zastosowań:

- sprzęt (SMV, NuSMV, Murphi)
- protokoły, oprogramowanie systemowe, sterowniki (Spin)
- oprogramowanie (BLAST, PathFinder, CBMC)
- systemy zależne od czasu (Uppaal, Kronos)
- systemy stochastyczne (PRISM)

Czym zajmuje się teoretyk?

- formalny model i specyfikacja
- złożoność obliczeniowa
- algorytmy, czasem heurystyczne lub częściowe
- podejścia symboliczne
- metody abstrakcji
- …

Dziedziny

- teoria języków i automatów
- logiki temporalne, rozstrzygalność i złożoność
- matematyczne modele systemów współbieżnych
- algorytmy grafowe

Czym zajmuje się praktyk?

- dostosowanie ogólnej metodologii do specyfiki zastosowania
- heurystyki dla wydajności
- zastosowanie do rzeczywistych, dużych systemów
- włączenie weryfikacji do procesu projektowania sprzętu i oprogramowania
- zarządzanie procesem weryfikacji
- …

III. Jaki model?

Jaki model?

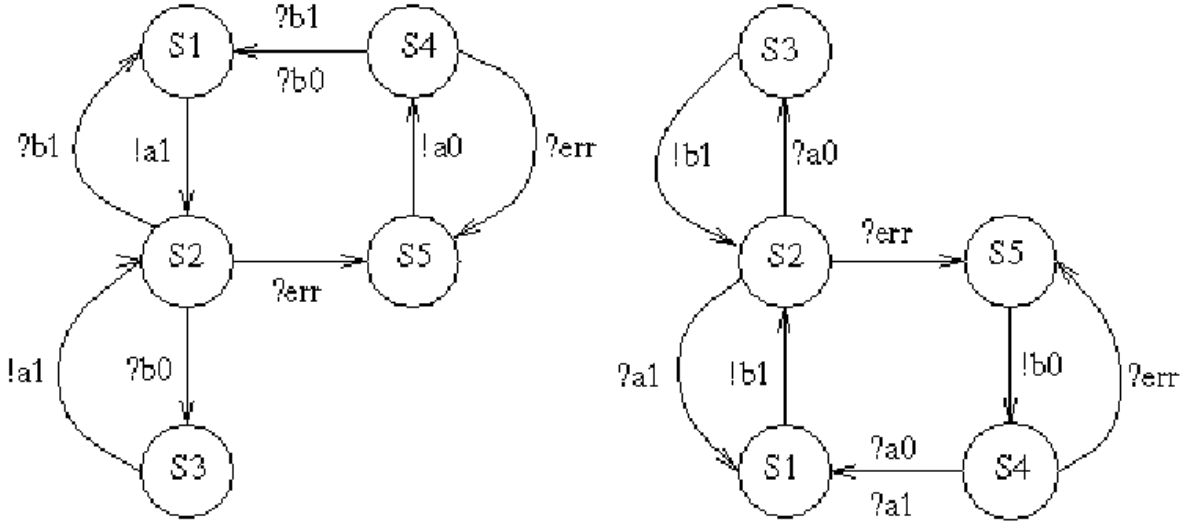
- funkcyjny: dane $\mapsto$ wynik
- reaktywny:
  - interakcja z otoczeniem
  - działanie może się nie kończyć
  - wymagania = własności temporalne
  - przykłady: system operacyjny, bankomat, serwer WWW, ...

Model = sterowanie + interakcja

- brak skomplikowanych danych i

  skomplikowanych obliczeń na danych
- kompozycjonalność
- dopuszczamy niedeterminizm
- ścisłość matematyczna

np. ABP


S1
S4
?b1
?b0
?b1
!a1
!a0
?err
S2
S5
?err
!a1
?b0
S3
S3
!b1
?a0
?err
S2
S5
?a1
!b1
!b0
?err
?a0
S1
S4
?a1

Własności modeli

- modele wysokiego poziomu (abstrakcyjne),

  niedospecyfikowane (częściowe) $\longrightarrow$ niedeterminizm

- interakcja pomiędzy składowymi $\longrightarrow$ współbieżność
- prototypowanie/symulacja $\longrightarrow$ model wykonywalny
- weryfikacja formalna $\longrightarrow$ semantyka matematyczna

Typowy model: maszyna skończeniestanowa (stany i przejścia)

+ zmienne (np. liczniki)
+ komunikacja (kanały komunikacyjne)
+ zegary
+ …

Stan = punkt sterowania

+ wartości zmiennych
+ zawartość kanałów komunikacyjnych
+ …

Opis kompozycjonalny

- lokalne składowe $\mapsto$ globalny system
- złożenie równoległe
- przemianowanie (kopie modułu)
- …

Złożenie równoległe

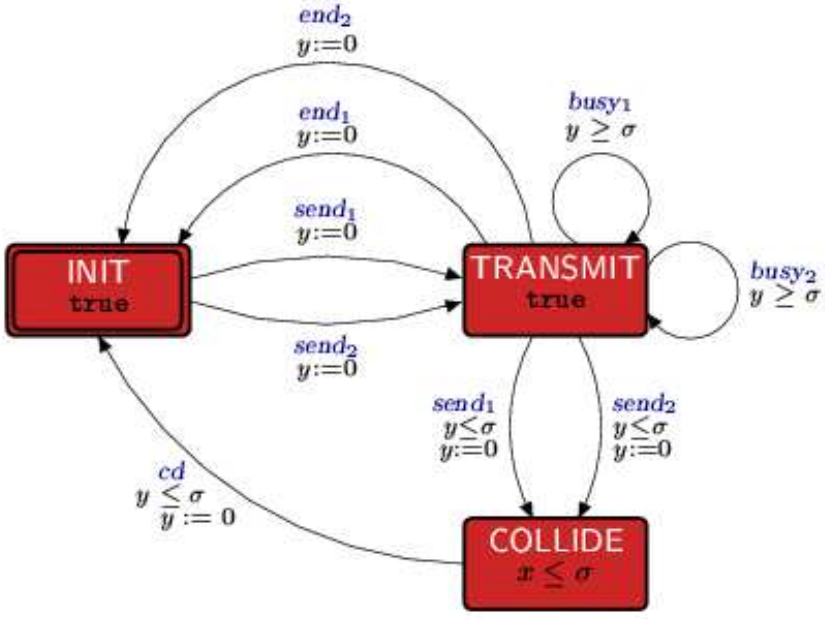
- synchroniczne (hardware)
- asynchroniczne (software lub anynchr. hardware)

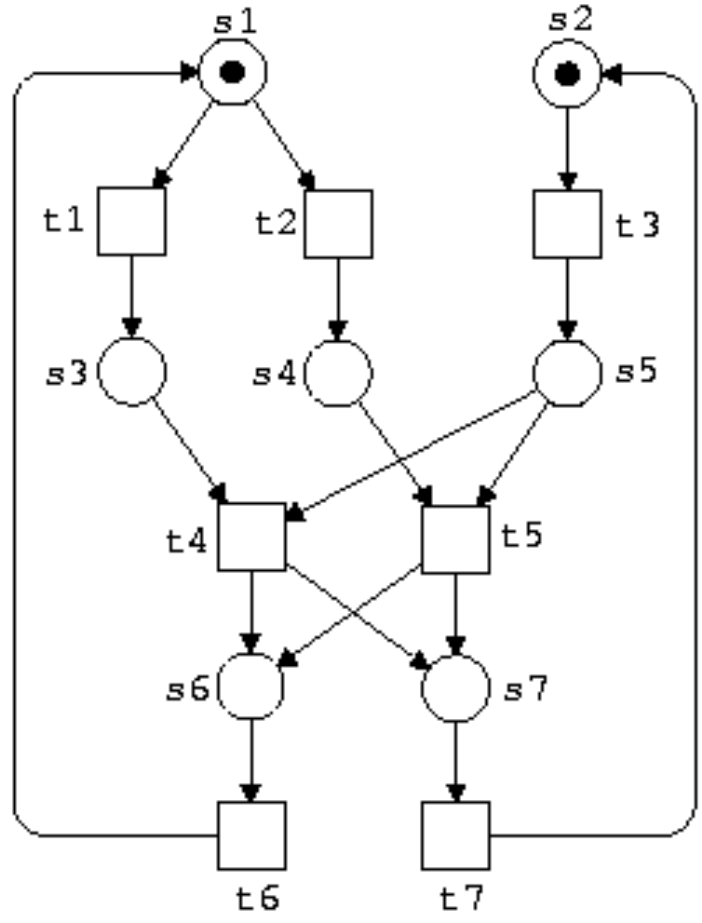
Warianty

- pojęcie stanu
- atomowy krok obliczeń
- interakcja
- równoważność semantyczna

Inne modele

- różne rozszerzenia automatów
- algebra procesów:
  - asynchroniczne: CSP, CCS, ACP, rachunek Π, ...
  - synchronniczne: SCCS, Esterel, Lustre
- sieci Petriego



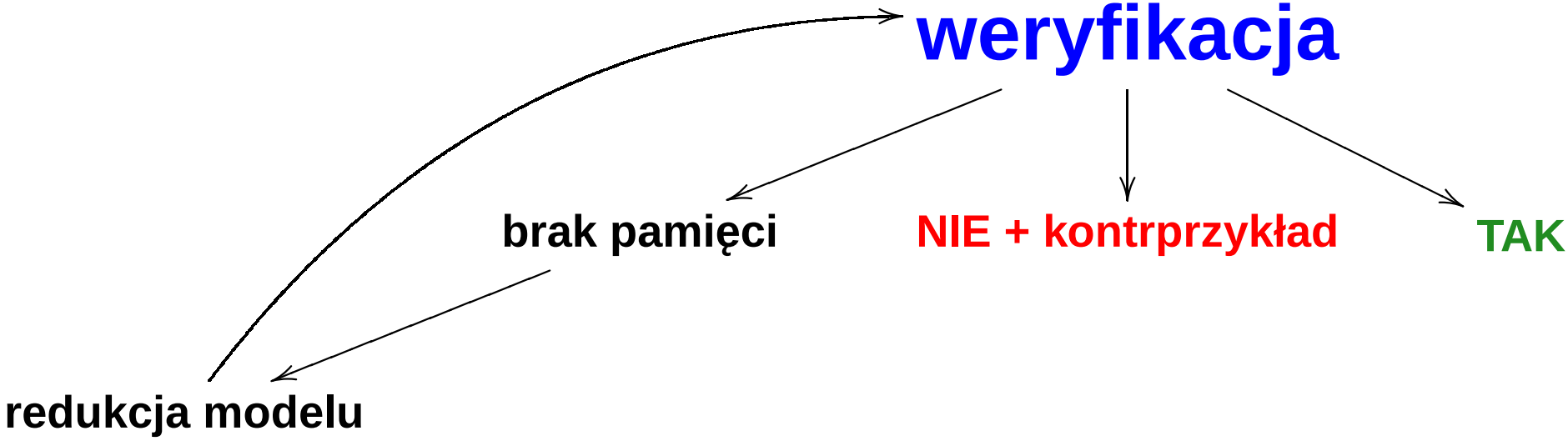


$$
\begin{array}{rcl}
x + y & = & y + x \\
(x + y) + z & = & x + (y + z) \\
x + x & = & x \\
(x + y) \cdot z & = & (x \cdot z) + (y \cdot z) \\
(x \cdot y) \cdot z & = & x \cdot (y \cdot z)
\end{array}
$$

Eksplozja stanów!

$$M = M_1 || \ldots || M_n$$

$$S \approx S_1 \times \ldots \times S_n$$


weryfikacja
brak pamięci
NIE + kontrprzykład
TAK
redukcja modelu

Metody walki z eksplozją

- podejście symboliczne
  - symboliczna weryfikacja modelowa
  - ograniczona weryfikacja modelowa

    (ang. *bounded model checking*)
- niezależność modułów
  - redukcje częściowo-porządkowe

IV. Za i przeciw

Weryfikacja modelowa – zalety

- uniwersalna
- pozwala na częściową weryfikację
- skuteczność niezależna od prawdopodobieństwa błędu
- w pełni automatyczna
- informacja diagnostyczna – kontrprzykład
- solidne matematyczne podstawy
- sukces przemysłowy
- łatwa integracja z cyklem produkcyjnym

Weryfikacja modelowa – ograniczenia

- złożoność obliczeniowa, eksplozja stanów
- źle sobie radzi z danymi
- nie weryfikacja, tylko tropienie błędów
  - weryfikujemy model a nie sam system
  - sprawdzamy tylko sformułowane wymaganie ϕ
- wykonanie abstrakcji wymaga jednak pracy eksperta
- błędy w narzędziach weryfikujących
- brak generalizacji/parametryzacji

Motto:

celem formalnej weryfikacji nie jest tworzenie poprawnego oprogramowania, lecz dostarczenie metodologii, która pozwoliłaby zwiększyć niezawodność (zmiejszyć liczbę błędów).

Wybrane osiągnięcia weryfikacji modelowej

- powszechnie stosowana w cyklu produkcji sprzętu
- NASA weryfikuje sondy: Deep Space 1, PathFinder, ...
- znaleziony atak na protokół Needhama-Schroedera

  [Lowe 1995]
- 10^{476} stanów
- ...

V. Historia

Dowodzenie poprawności programów

- [Dijkstra]
- [Floyd]
- [Hoare] podejście strukturalne

– Boyer–Moore prover: automatyczne dowodzenie poprawności prostych programów w LISPie
– [Owicki-Gries] rozszerzenie Hoare’a dla programów współbieżnych
– [Pnueli] logika temporalna dla systemów reaktywnych
– weryfikacje protokołów, zjawisko eksplozji stanów

Model checking

- [Clarke, Emerson], [Queille, Sifakis]
- model explicite (skończony system przejść, automat)
- specyfikacja = formuła logiki temporalnej

  (później też automat)
- weryfikacja automatyczna
- kontrprzykład gdy odpowiedź negatywna
- zweryfikowano małe układy cyfrowe, znaleziono błędy

Metody walki z eksplozją:

- metody abstrakcji
- symboliczna weryfikacja modelowa
- redukcje częściowo-porządkowe

- systemy nieskończeniestanowe, np.
  - zależne od czasu
  - parametryczne
- narzędzia (SMV, Murphi, . . .)
- zastosowania przemysłowe
- zastosowanie SAT-solverów:

  ograniczona weryfikacja modelowa

- narzędzia dla rozszerzonych modeli:
  - zależnych od czasu
  - stochastycznych
  - …
- zastosowania przemysłowe cd
- nowe dziedziny zastosowań, np. bioinformatyka
- weryfikacja oprogramowania

  (ang. *software model checking*)

Plan

wykład		laboratorium	
LTL, translacja do ω-automatów	(3x)	SPIN	(4x)
redukcje częśc.-porz.	(1x)		
CTL, OBDD, symboliczna w.m.	(3x)	NuSMV	(4x)
ograniczona w.m., redukcja do SAT	(1x)		
abstrakcje	(2x)	CBMC	(2x)
automaty czasowe	(2x)	Uppaal	(3x)
weryfikacja stochastyczna	(1x)		

W przyszłym tyg. nie ma labu!