seminarium magisterskie

NIEZAWODNOŚĆ SYSTEMÓW WSPÓŁBIEŻNYCH I OBIEKTOWYCH

Sławomir Lasota i Aleksy Schubert

www.mimuw.edu.pl/{~sl,~alx}

Niezawodność systemów współbieżnych i obiektowych

seminarium magisterskie

Termin

środy 12:15, sala 5060

Prowadzący

Sławomir Lasota
Aleksy Schubert

{sl,alx} małpa mimuw edu pl

USOSweb

strona seminarium

Seminarium poświęcone szeroko rozumianej niezawodności systemów komputerowych i metodom jej zapewniania, od praktycznych projektów weryfikacyjnych i narzędzi do modeli matematycznych i zagadnień teoretycznych z nimi związanych.

Ogłoszenia

- 10 X 2012: dzisiejsze seminarium odwołane

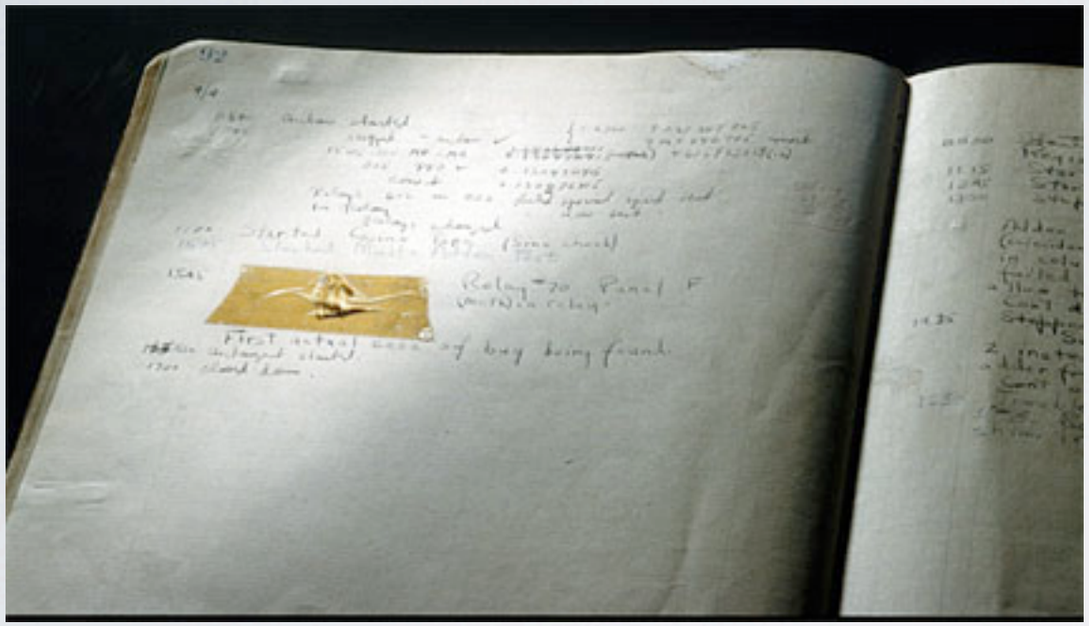
Referaty w semestrze zimowym 12/13

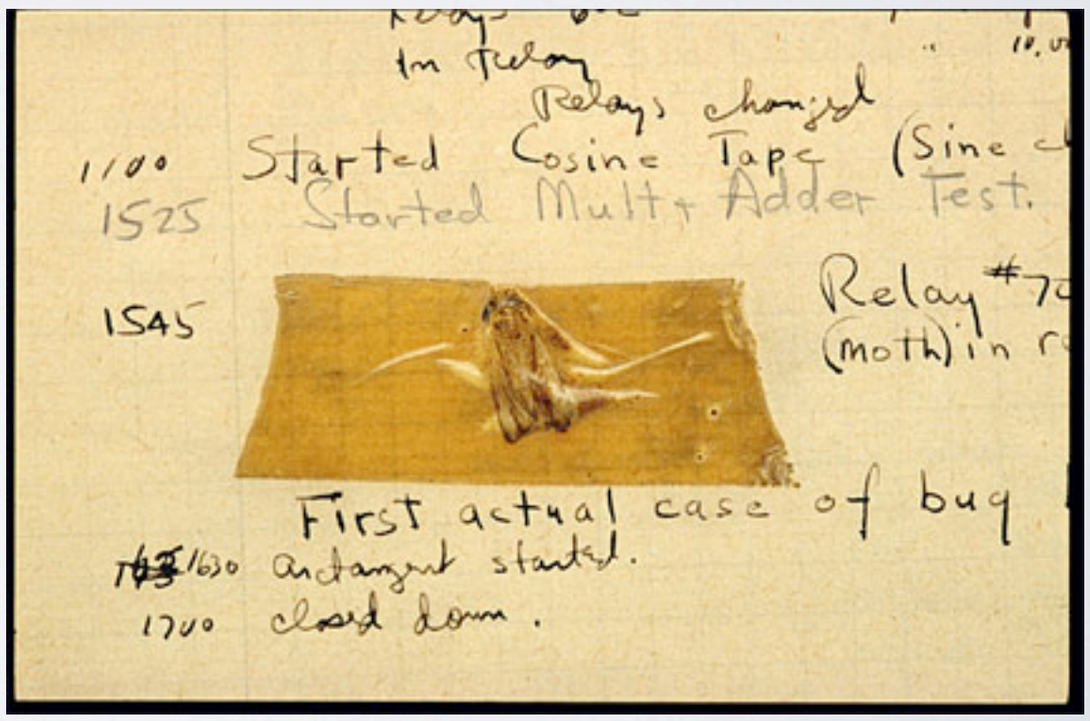
Temat referatu	Osoba referująca	Przybliżony termin	Materiały
?	M. Oniszczuk	12 XII 2012	
?	P. Iwaniuk	5 XII 2012	
Maszyny Turinga nad nieskończonymi alfabetami	S. Lasota	28 XI 2012	
Języki bezkontekstowe nad nieskończonymi alfabetami	M. Maciejewska	21 XI 2012	
Ograniczenia weryfikacji ograniczonej w narzędziu ESC/Java (cd)	Z. Chlebicki	14 XI 2012	
Ograniczenia weryfikacji ograniczonej w narzędziu ESC/Java	Z. Chlebicki	7 XI 2012	

Teoria i praktyka wspomaganych komputerowo metod analizy oprogramowania i układów sprzętowych.

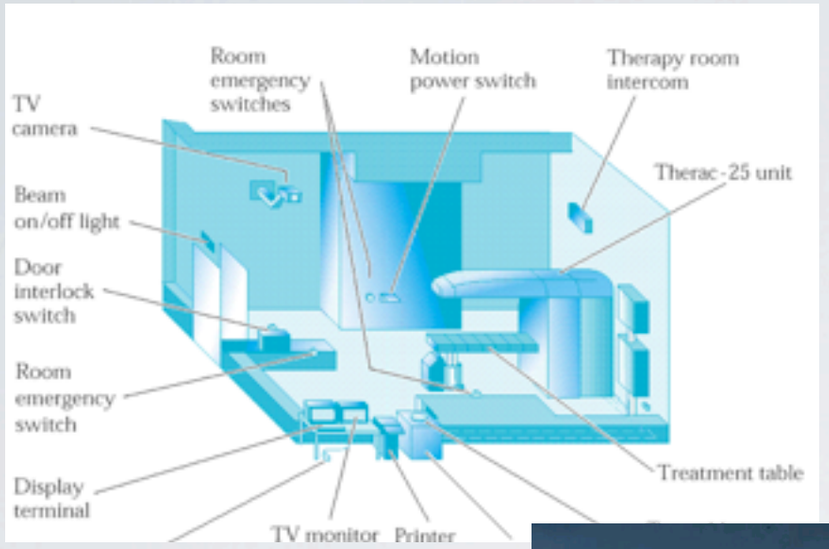
Weryfikacja wspomagana komputerowo

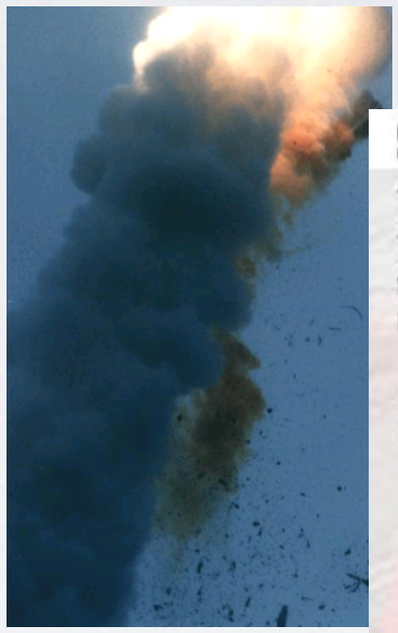
Motywacja

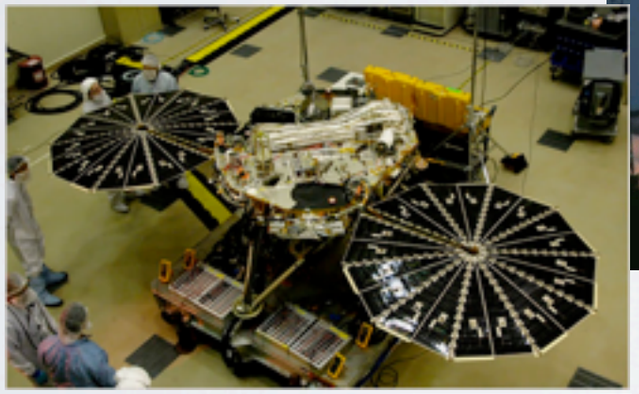




1947 Harward

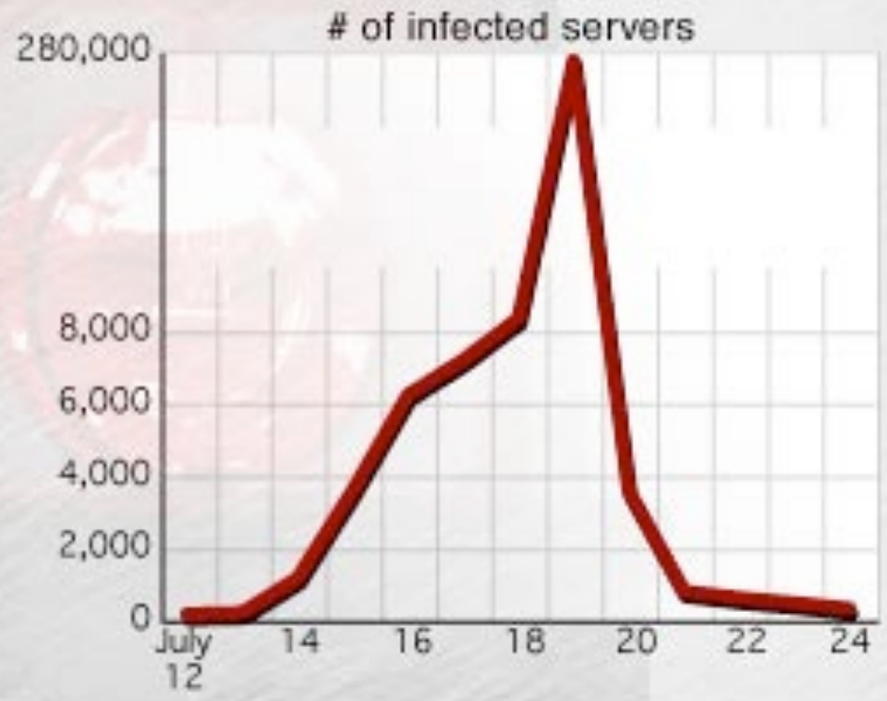






Spreading fast

The worm slowly spread until July 19, when the number of computers attacking networks skyrocketed. Now, the worm is hibernating, ready to re-infect Aug. 1.

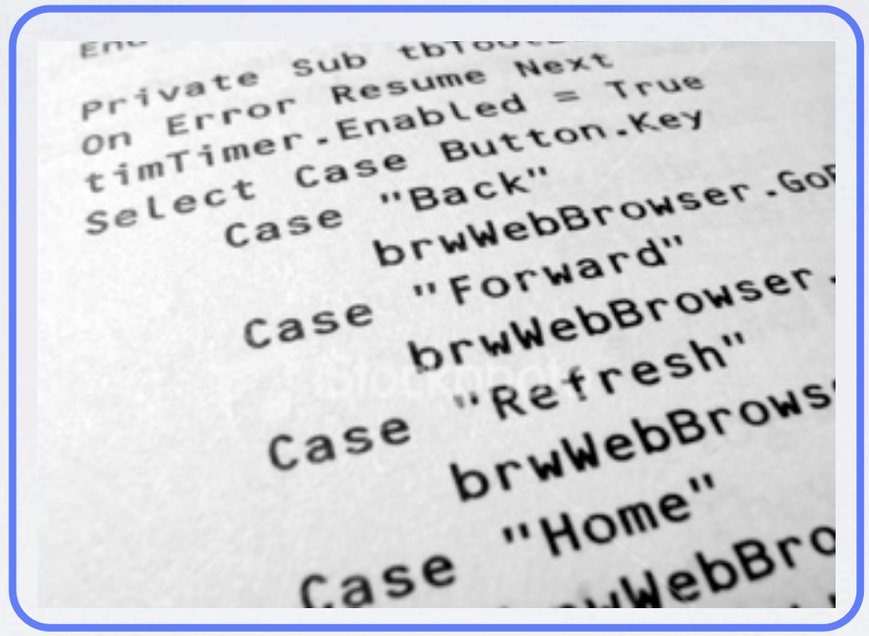


Source: Chemical Abstracts Service

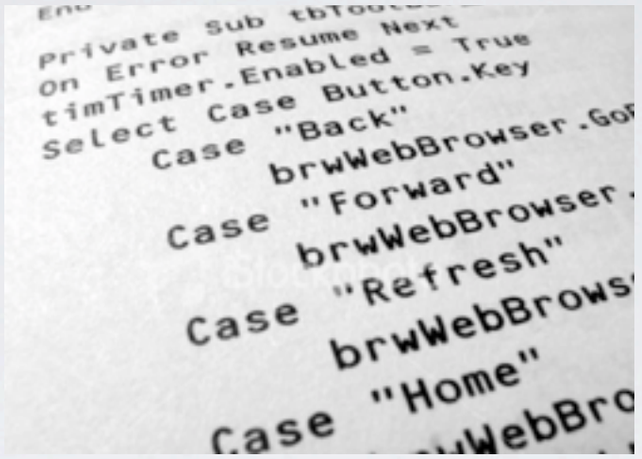
Weryfikacja formalna

OGRANICZENIE

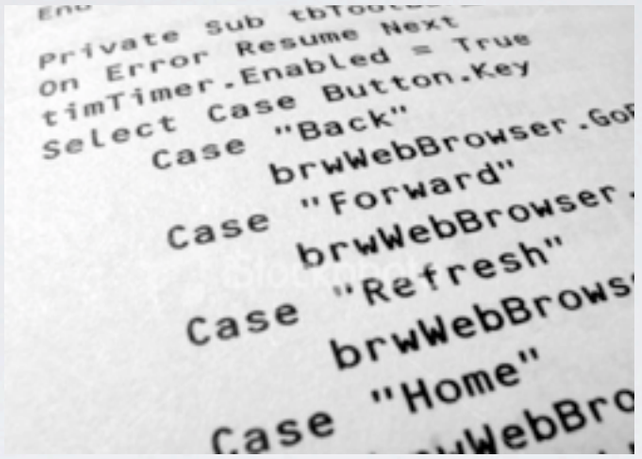
każde nietrywialne pytanie jest nierozstrzygalne !



METODA 1: INTERAKCYJNA



METODA 1: INTERAKCYJNA

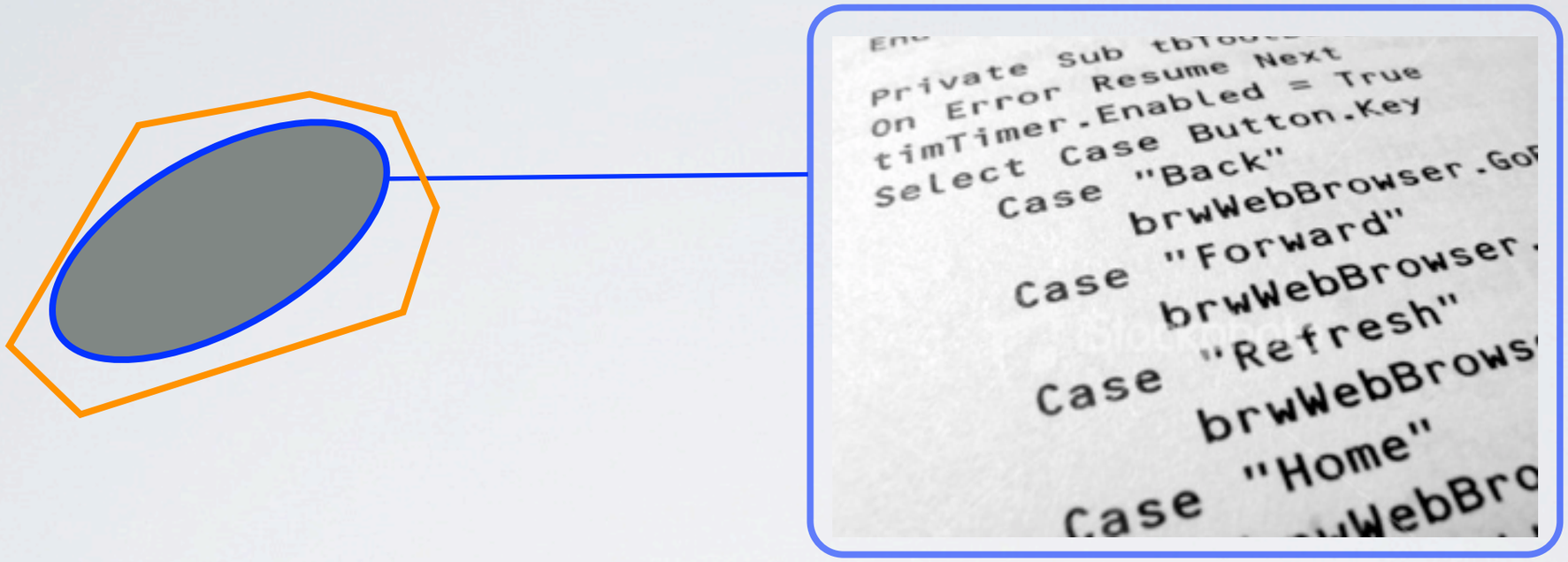


✔

✘

stosowana w dowodzeniu poprawności programów

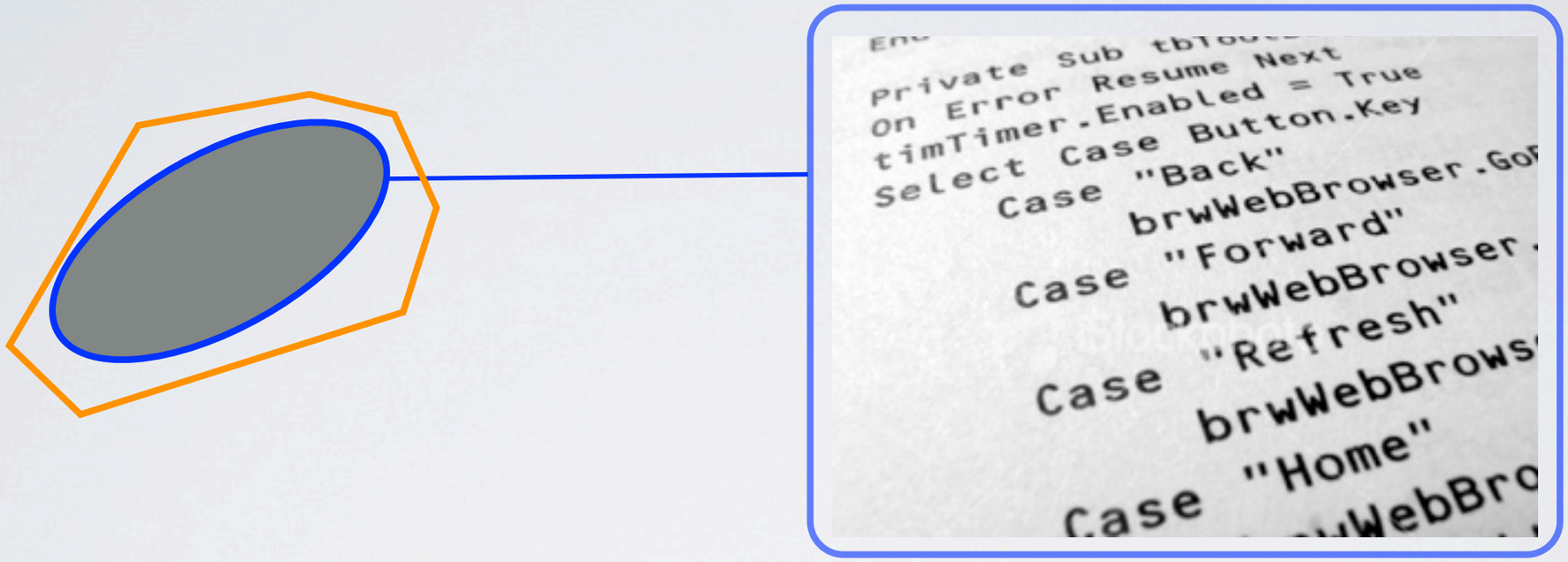
METODA 2: PRZYBLIŻENIE



na pewno ✔

chyba ✘

METODA 2: PRZYBLIŻENIE

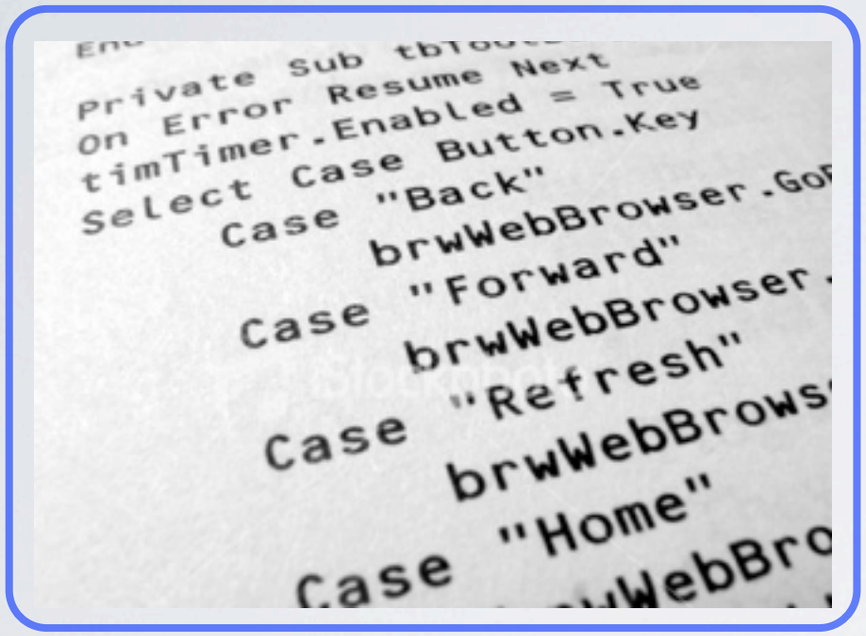


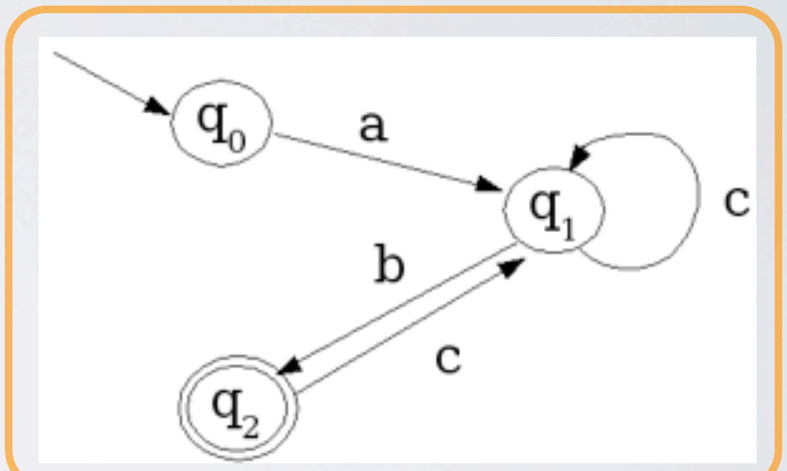
na pewno ✔

chyba ✘

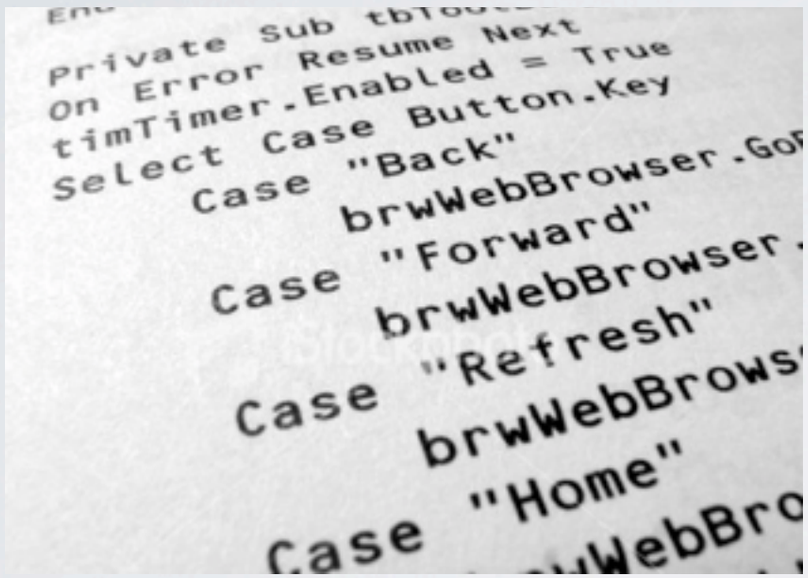
stosowana w statycznej analizie programów

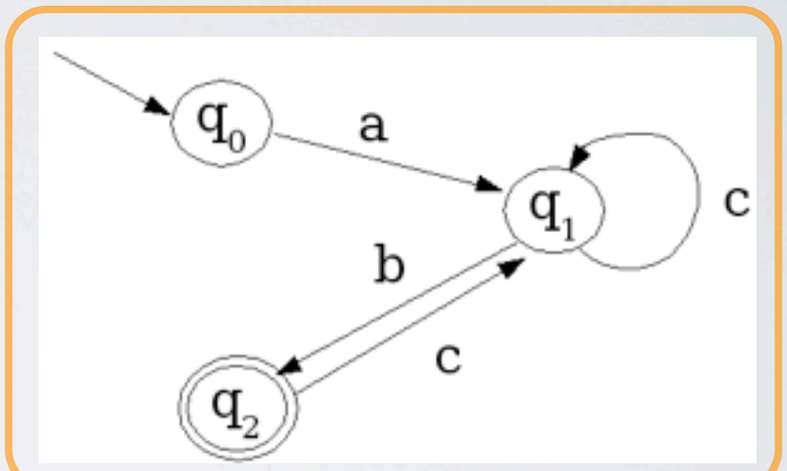
METODA 3: ABSTRAKCJA





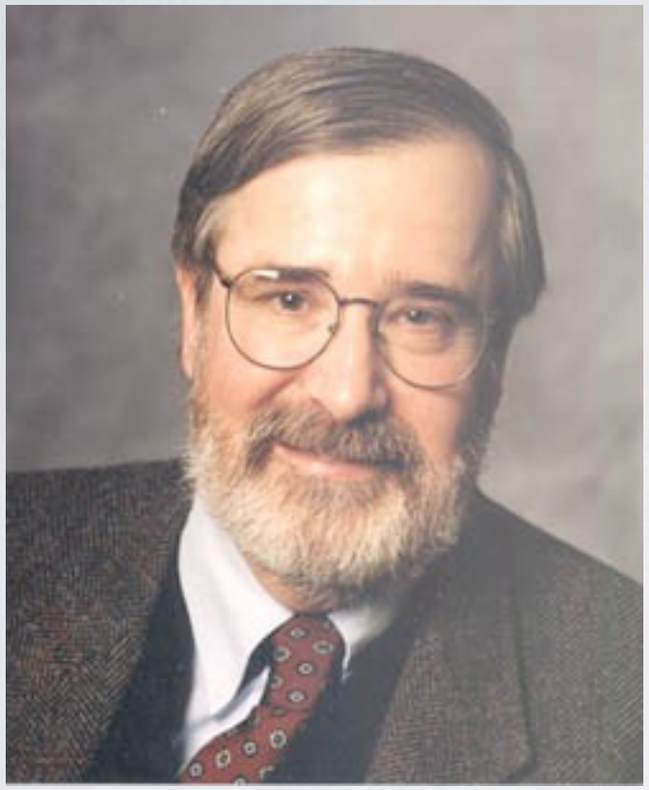
METODA 3: ABSTRAKCJA



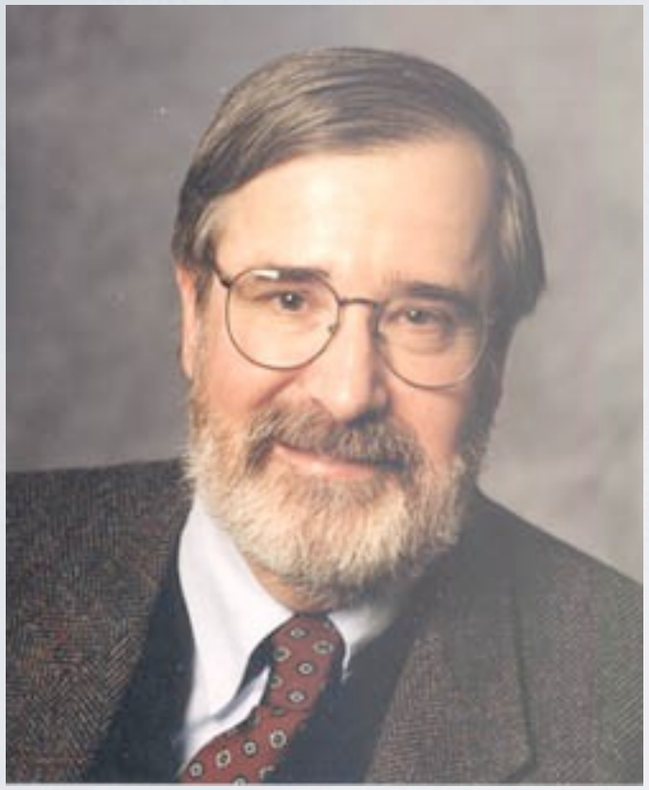


stosowana w weryfikacji modelowej (ang. model checking)

NAGRODA TURINGA 2007



NAGRODA TURINGA 2007

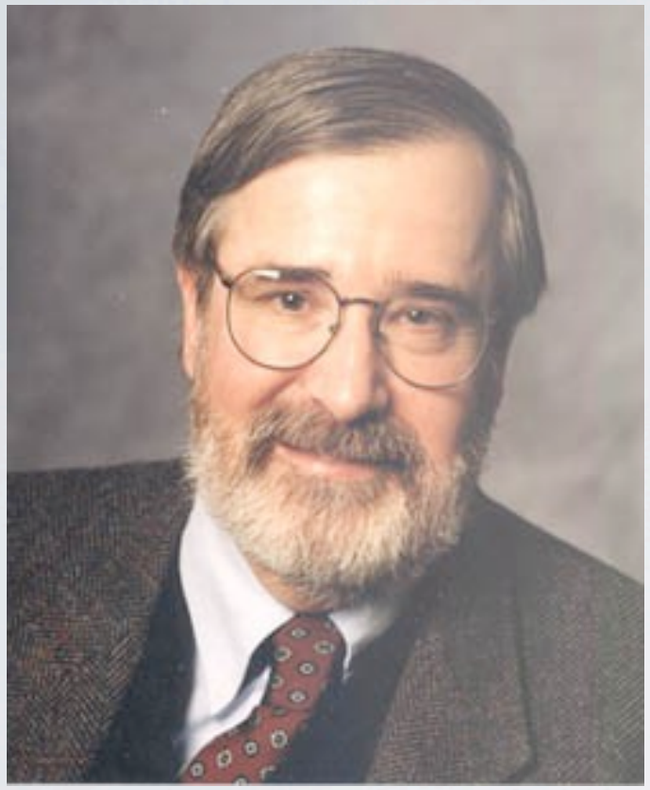


Ed Clarke

Allen Emerson

Joseph Sifakis

NAGRODA TURINGA 2007



Ed Clarke

Allen Emerson

Joseph Sifakis

1972, 1978, 1980

Wymagania?

Prace magisterskie cz. I

PRACE MGR DOTYCHCZAS

PRACE MGR DOTYCHCZAS

- Weryfikacja programów binarnych

PRACE MGR DOTYCHCZAS

- Weryfikacja programów binarnych
- Zastosowanie agregacji w stochastycznym model-checking'u

PRACE MGR DOTYCHCZAS

- Weryfikacja programów binarnych
- Zastosowanie agregacji w stochastycznym model-checking'u
- Modelowanie sygnalizacji świetlnej

PRACE MGR DOTYCHCZAS

- Weryfikacja programów binarnych
- Zastosowanie agregacji w stochastycznym model-checking'u
- Modelowanie sygnalizacji świetlnej
- Sprawdzanie równoważności programów przy użyciu ograniczonego model-checking'u

PRACE MGR WKRÓTCE

PRACE MGR WKRÓTCE

- weryfikacja programistów

PRACE MGR WKRÓTCE

- weryfikacja programistów
- nieskończone alfabety w weryfikacji symbolicznej

- weryfikacja programistów

weryfikacja programistów

weryfikacja programistów

we model-check coders

Zadanie: oblicz punkt równowagi

$$k \text{ jest punktem równowagi} \iff \sum_{i=0}^{k-1} a_i = \sum_{i=k+1}^{n-1} a_i$$

Rozwiązanie:

```
int equi(int *A, int n) {
  int i;
  long long d = 0;

  // Obliczenie wartosci d_0
  for (i = 0; i < n; ++i)
    d += A[i];

  // Poszukiwanie punktu równowagi
  for (i = 0; i < n; ++i) {
    if (d == A[i])
      return i;
    d -= 2 * A[i];
  }

  return -1;
}
```

Rozwiązanie:

```
int equi(int *A, int n) {
  int i;
  long long d = 0;

  // Obliczenie wartosci d_0
  for (i = 0; i < n; ++i)
    d += A[i];

  // Poszukiwanie punktu równowagi
  for (i = 0; i < n; ++i) {
    if (d == A[i])
      return i;
    d -= 2 * A[i];
  }

  return -1;
}
```

Rozwiązanie:

```
int equi(int *A, int n) {
  int i;
  long long d = 0;

  // Obliczenie wartosci d_0
  for (i = 0; i < n; ++i)
    d += A[i];

  // Poszukiwanie punktu równowagi
  for (i = 0; i < n; ++i) {
    if (d == A[i])
      return i;
    d -= 2 * A[i];
  }

  return -1;
}
```

kontrprzykład : $\{2^{30}, 0, 2^{30}\}$

```
int equi(int *A, int n) {
  int i;
  long long d = 0;

  // Obliczenie wartosci d_0
  for (i = 0; i < n; ++i)
    d += A[i];

  // Poszukiwanie punktu równowagi
  for (i = 0; i < n; ++i) {
    if (d == A[i])
      return i;
    d -= 2 * A[i];
  }

  return -1;
}
```

```
int equi(int *A, int n) {
  int i;
  long long d = 0;

  // Obliczenie wartosci d_0
  for (i = 0; i < n; ++i)
    d += A[i];

  // Poszukiwanie punktu równowagi
  for (i = 0; i < n; ++i) {
    if (d == A[i])
      return i;
    d -= A[i];
    d -= A[i];
  }

  return -1;
}
```

- nieskończone alfabety w weryfikacji symbolicznej

automaty nad nieskończonym alfabetem

```
<show name="Law &amp; Order">
  <episode>
    <season>13</season>
    <number>22</number>
    <title>Sheltered</title>
    <airdate>2003-05-14-05:00</airdate>
    <synopsis>
          The NYPD detectives hunt down a sniper who kills his
          victims in broad daylight.
       </synopsis>
  </episode>
  <episode>
    <season>13</season>
    <number>23</number>
    <title>Couples</title>
    <airdate>2003-05-21-05:00</airdate>
    <synopsis>
          The detectives catch four murders and a kidnapping on
          the same day.
       </synopsis>
  </episode>
  <episode>
    <season>13</season>
    <number>24</number>
    <title>Smoke</title>
    <airdate>2003-05-21-05:00</airdate>
    <synopsis>
          An eccentric comedian is under suspicion of murdering
          his baby son by dangling him over a ledge.
       </synopsis>
  </episode>
</show>
```

automaty nad nieskończonym alfabetem

```
<show name="Law &amp;amp; Order">
  <episode>
    <season>13</season>
    <number>22</number>
    <title>Sheltered</title>
    <airdate>2003-05-14-05:00</airdate>
    <synopsis>
          The NYPD detectives hunt down a sniper who kills his
          victims in broad daylight.
       </synopsis>
  </episode>
  <episode>
    <season>13</season>
    <number>23</number>
    <title>Couples</title>
    <airdate>2003-05-21-05:00</airdate>
    <synopsis>
          The detectives catch four murders and a kidnapping on
          the same day.
       </synopsis>
  </episode>
  <episode>
    <season>13</season>
    <number>24</number>
    <title>Smoke</title>
    <airdate>2003-05-21-05:00</airdate>
    <synopsis>
          An eccentric comedian is under suspicion of murdering
          his baby son by dangling him over a ledge.
       </synopsis>
  </episode>
</show>
```

$$\text{alfabet} = A \times D$$

automaty nad nieskończonym alfabetem

nowe pojęcie skończoności

automaty nad nieskończonym alfabetem

nowe pojęcie skończoności

M. Bojańczyk, B. Klin, S. Lasota, ***Automata with group actions.*** Proc. LICS'11, 2011.

M. Bojańczyk, L. Braud, B. Klin, S. Lasota, ***Towards nominal computation.*** Proc. POPL'12, 2012.

M. Bojańczyk, S. Lasota, ***A machine-independent characterization of timed languages.*** Proc. ICALP'12, 2012.

po co nieskończoność?

po co nieskończoność?

- dane

po co nieskończoność?

- dane
- stemple czasowe

po co nieskończoność?

- dane
- stemple czasowe
- identyfikatory procesów

po co nieskończoność?

- dane
- stemple czasowe
- identyfikatory procesów
- ...

nieskończone alfabety w weryfikacji symbolicznej

nieskończone alfabety w weryfikacji symbolicznej

- symboliczna weryfikacja osiągalności

nieskończone alfabety w weryfikacji symbolicznej

- symboliczna weryfikacja osiągalności
- systemy zależne od czasu

nieskończone alfabety w weryfikacji symbolicznej

- symboliczna weryfikacja osiągalności
- systemy zależne od czasu
- osiągalność w sieciach Petriego z danymi

Prace magisterskie cz. 2

PRACE MGR DOTYCHCZAS

PRACE MGR DOTYCHCZAS

- Wykorzystanie kart graficznych w obliczeniach równoległych

PRACE MGR DOTYCHCZAS

- Wykorzystanie kart graficznych w obliczeniach równoległych
- Weryfikacja niemutowalności obiektów w Javie

PRACE MGR DOTYCHCZAS

- Wykorzystanie kart graficznych w obliczeniach równoległych
- Weryfikacja niemutowalności obiektów w Javie
- Analiza bezpieczeństwa kodu OpenSSH

PRACE MGR WKRÓTCE

PRACE MGR WKRÓTCE

- Formalizacja wybranego protokołu komunikacyjnego

PRACE MGR WKRÓTCE

- Formalizacja wybranego protokołu komunikacyjnego
- Program do znajdowania błędów w Pythonie

PRACE MGR WKRÓTCE

- Formalizacja wybranego protokołu komunikacyjnego
- Program do znajdowania błędów w Pythonie
- Program do oglądania bajtkodu Javy

IV. Prace doktorskie

WOJTEK CZERWIŃSKI

złożoność obliczeniowa bisymulacji

PIOTREK HOFMAN

automaty na danych, bisymulacja

JOANNA OCHREMIAK

nieskończone alfabety

TADEUSZ SZNUK

własny system weryfikacji programów

JĘDREK FULARA

analiza statyczna, generowanie JML'a

KRZYSZTOF JAKUBCZYK

analiza statyczna, generowanie JML'a

VI. Wykłady

WERYFIKACJA WSPOMAGANA KOMPUTEROWO

WYKŁAD: @SŁAWOMIR LASOTA LAB: @DARIA WALUKIEWICZ-CHRZĄSZCZ @GRZEGORZ MAJ

Rankiem 4 czerwca 1996 rakieta Arianne-5 wybuchła w powietrzu, 36 sekund po starcie z kosmodromu w Gujanie Francuskiej. Przyczyną był nieobsłużony wyjątek, spowodowany nieudaną konwersją z 64-bitowej liczby zmiennopozycyjnej na 16-bitową liczbę całkowitą. Awarii spowodowanych wadliwym działaniem oprogramowania było więcej. Ulegały im m.in. lądownik marsjański Mars Pathfinder, samolot Airbus czy centrala telefoniczna AT&T. Niektóre z nich pociągnęły ofiary, np. awaria urządzenia Therac-25 stosowanego w radioterapii.

WYKŁAD

pon., 10:15 - 11:45, s. 4060

LAB

pon., 12:15 - 13:45, s. 2043
piątki, 8:30 - 10:00, s. 2042

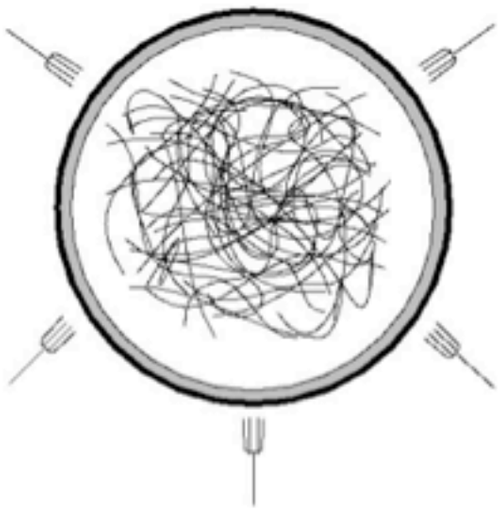
USOS

opis wykładu w USOSie

Wykład poświęcony jest metodom weryfikacji pozwalającym na zwiększenie niezawodności systemów komputerowych. Ograniczymy się do tzw. *metod formalnych*, tzn. opartych na ścisłych podstawach matematycznych, i na ogół w dużym stopniu automatycznych. Omówimy trzy najważniejsze grupy podejść: *weryfikacja modelowa* (ang. *model-checking*), *analiza statyczna* (ang. *static analysis*) i dowodzenie poprawności programów. Pierwsza z nich odniosła już sukces przemysłowy w weryfikacji układów sprzętowych; połączenie dwóch pozostałych pozwala odnaleźć rozsądny kompromis pomiędzy siłą metody a jej efektywnością, i wydaje się być najlepsze w weryfikacji programów. W ramach wykładu omówione zostaną podstawy teoretyczne i algorytmy. W czasie laboratorium studenci zapoznają się z kilkoma narzędziami. Wykonają też własne projekty, co pozwoli lepiej zrozumieć treść wykładu.

WSPÓŁBIEŻNOŚĆ

WYKŁAD: SŁAWOMIR LASOTA ĆWICZENIA: PIOTR HOFMAN



WYKŁAD

środy 8:45 - 10:13, s. 4070

ĆWICZENIA

poniedziałki 8:30 - 10:00, s. 4050

USOS

strona w USOSie

OGŁOSZENIA:

- Wykład zaczynamy o 8:45 (3.10.2012).

WYKŁADY:

- **wykład 1:** wprowadzenie, definicja sieci Petriego
- **wykład 2:** współbieżność i konflikt; problemy analizy
- **wykład 3:** dolna granica dla analizy sieci ogólnych

Wspomagane komputerowo ulepszanie programów w Javie

AN

Informacje ogólne

Kod przedmiotu:	1000-2M11WKU	Kod Erasmus:	11.3 i
Nazwa przedmiotu:	Wspomagane komputerowo ulepszanie programów w Javie		
Jednostka:	Wydział Matematyki, Informatyki i Mechaniki		
Grupy:	Przedmioty monograficzne dla III - V roku informatyki Przedmioty obieralne dla informatyki		
Punkty ECTS i inne:	6.00 i → zobacz reguły punktacji		
Język prowadzenia:	angielski		
Rodzaj przedmiotu:	monograficzne		
Założenia (lista przedmiotów):	Języki, automaty i obliczenia 1000-214bJAO Podstawy matematyki 1000-211bPM Semantyka i weryfikacja programów 1000-215bSWP		
Skrócony opis:	Metody organizowania i porządkowania kodu oraz polepszania jego struktury oraz niezawodności z użyciem narzędzi składniowego wykrywania niedoskonałości, narzędzi obliczania miar niedoskonałości kodu, narzędzi analizy statycznej i weryfikacji systemów komputerowych.		
Pełny opis:	1. Wprowadzenie do zapewniania jakości kodu (1 wykład) 2. Metody podstawowego porządkowania kodu o charakterze syntaktycznym (np. Checkstyle, 1 wykład) 3. Metody podstawowego porządkowania kodu o charakterze syntaktycznym sięgające do reprezentacji w bajtkodzie (np. PMD, 1 wykład) 4. Metody bardziej rozbudowanego znajdowania błędów z użyciem rozbudowanych metod o charakterze syntaktycznym (np. FindBugs, 1 wykład) 5. Metody ilościowej analizy kodu (np. Metrics, 1 wykład) 6. JUnit i narzędzia analizy pokrycia kodu testami (np. EMMA, 1-2 wykłady) 7. Analiza abstrakcyjna (np. COSTA, 3 wykłady) 8. Rozszerzona analiza statyczna (np. ESC/Java2, 3 wykłady)		

Dziękujemy !