

OBLICZENIA SYMBOLICZNE

# GENERATOR PYTAŃ TESTOWYCH

Autorzy:

- Jedynak Mirosław
- Lewandowski Krzysztof

Obliczenia symboliczne

*CLISP*

**Prezentacja projektu**

# Zawartość projektu

- Konwerter pliku XML do formatu listy
- Trzy algorytmy wyboru optymalnego testu
- System zadawania pytań
- Przykładowe pytania oraz drzewo nauczania zapisane w pliku XML

Głównym plikiem projektu jest „main.lsp”, który wykorzystuje powyższe opcje

# Konwerter XML-a do listy

Przyjęto założenie poprawności pliku XML, który jest parsowany, tzn. brak nadmiarowych spacji w znacznikach, np. `<tag >`.

Konwersja odbywa się znak po znaku, z tym że niektóre znaczniki są traktowane w sposób szczególny, tj. jeżeli występuje właściwość `id` w znaczniku to jest ona traktowana jako nazwa znacznika. Dodatkowo jeżeli w kolejnych znacznikach tego samego typu nie podano tej właściwości, wartość `id` (jeżeli to możliwe) jest obliczana automatycznie.

Jako kolejne argumenty funkcji konwertującej można podać znaczniki które mają być ignorowane, tzn. treść znajdująca się w ignorowanym znaczniku jest przypisywana znacznikowi poprzedniemu, ma to zastosowanie na przykład w sytuacji parsowania drzewa nauczania (znacznik `<lower>`). Właściwości znacznika parsowane są tak, że po wczytaniu listy funkcjonują jako ‘dotted pair’.

Ponieważ niektóre znaki nie są normalnie interpretowane przez CLISP ( . , ? : itd.) zakładam, że takich znaków nie ma w znacznikach, jeżeli pojawiają się one między znacznikami dodawany jest apostrof, tak aby były one interpretowane poprawnie.

Wywołanie: `(XML2List xml_file lisp_file ignored_tags)`

# Algorytmy

Zaimplementowane algorytmy mają na celu ułożenie jednego z trzech rodzajów testów:

- 1) Zoptymalizowane pod kątem właściwej liczby pytań z danej dziedziny (oparte na danych dotyczących ważności danej dziedziny w całym drzewie nauczania [funkcja `znajdzTest1`])
- 2) Zoptymalizowanego pod kątem równomiernego pokrycia dziedzin nauczania (oparte na danych dotyczących pokrycia danej dziedziny przez dane pytanie) [funkcja `znajdzTest2`]
- 3) Zoptymalizowane pod kątem największego pokrycia dziedzin występujących w teście (algorytm jest podobny do alg. 2 ale nie uwzględnia stopnia trudności w zadawanych pytaniach) [`znajdzTest3`]

Algorytmy 1 i 2 przewidują możliwość ustawienia poziomu trudności (ilość dobrych odpowiedzi w stosunku do wszystkich udzielonych – dane na ten temat są zamieszczone w pliku z pytaniami). Współczynnik ten przyjmuje wartości od 0 do 1.

Przebieg pracy generatora może być wyświetlana po każdym kroku generacji (należy jako parametr podać 't' lub zapisywany do pliku–funkcja `verbose-search`). Oba algorytmy zakładają, że lista pytań oraz drzewo nauczania są w odpowiednio w plikach 'test.lsp' 'graph.lsp' (oba pliki mogą być wygenerowane przez parser załączony do projektu)

Dziedziny wybierane do testu nie muszą być na jednakowym poziomie, ponieważ każdej dziedzinie jest przypisane określone pokrycie w stosunku do dziedziny nadrzędnej (patrz schemat na końcu).

# Algorytmy wprowadzenie

Działanie obu algorytmów do generowania pytań zostało przedstawione na przykładowym teście.

## Algorytm 1:

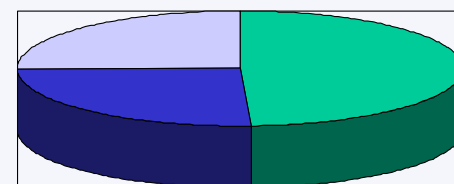
Test obejmuje dziedziny z „Systemów operacyjnych”:

Sygnały (0.2),

Kolejki komunikatów (IPC)( $0.5 \cdot 0.34 \cdot 0.6 = 0.1$ ) ,

Łącza komunikacyjne( $0.5 \cdot 0.2 = 0.1$ ).

*(w nawiasach pokrycie dziedzin względem korzenia)*

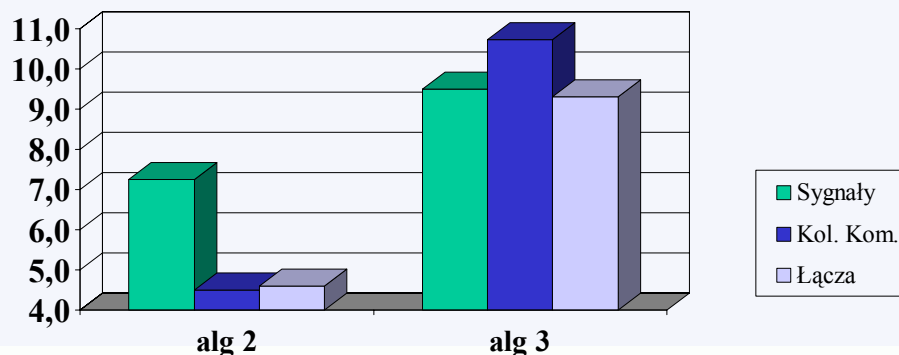


*Procentowy udział danej  
dziedziny w teście*

## Algorytm 2, 3:

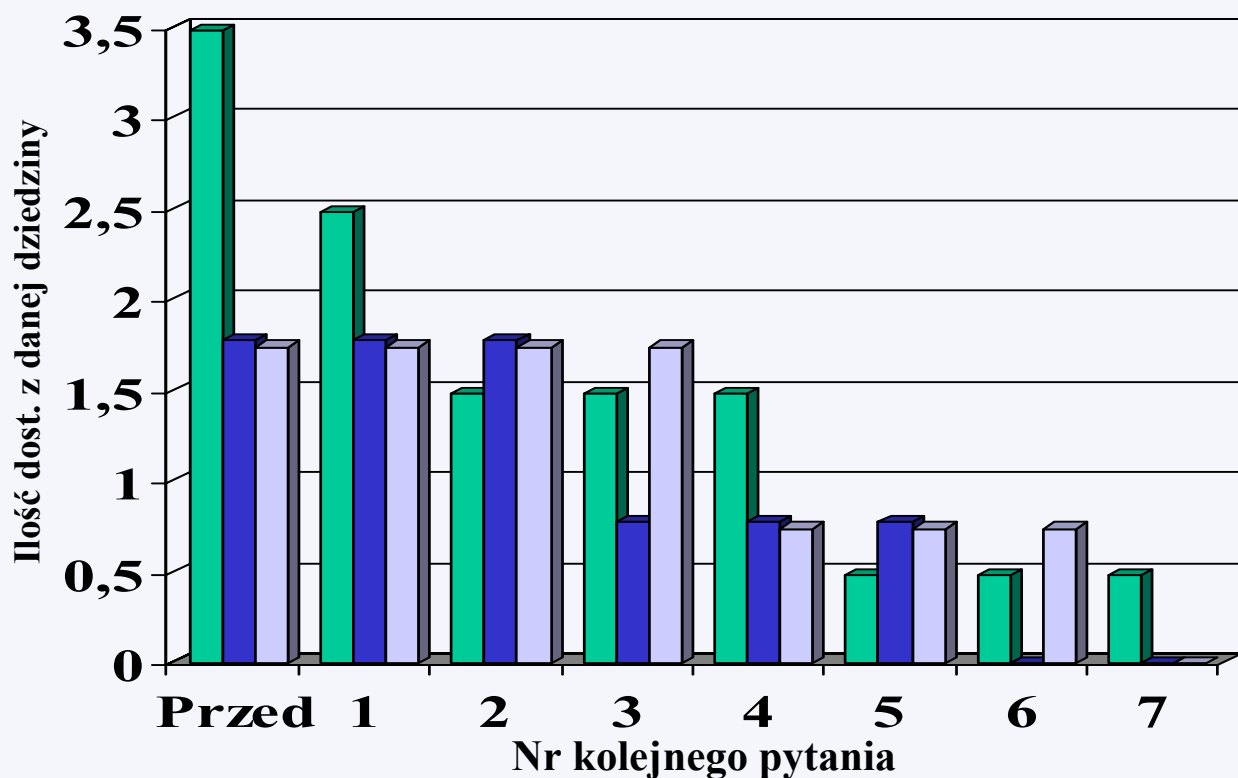
Ponieważ test ten bazuje na równomiernym pokryciu wszystkich dziedzin, więc na początku wszystkie dziedziny mają pokrycie równe 0.

*Na wykresie jest przedstawione pokrycie po zakończonym teście*



# Algorytm 1

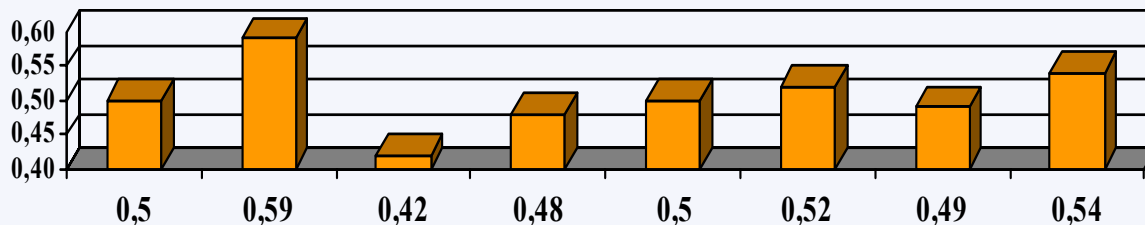
(wg ilości możliwych do zadania pytań)



■ Sygnały  
■ Kolejki kom.  
■ Łącza

Na podstawie przykładowego testu:

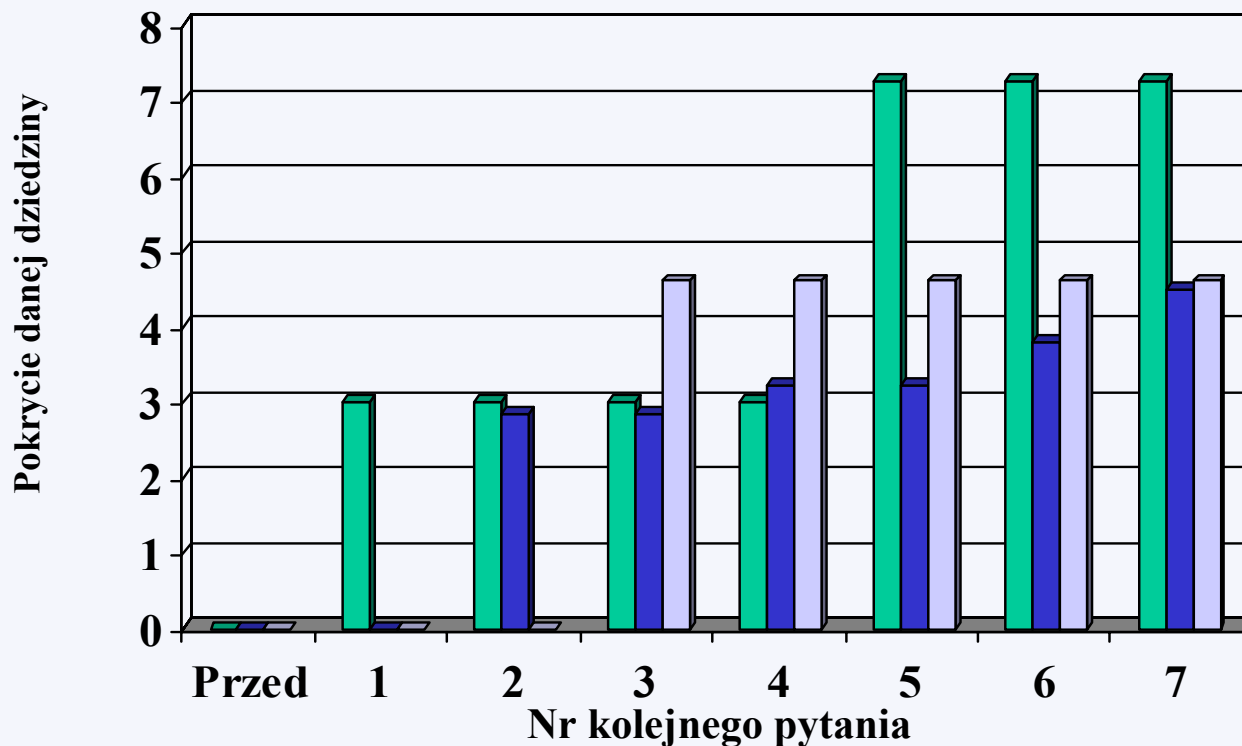
- z dziedzin: Sygnały  
IPCKolKom ŁaczaKomun;
- z poziomem trudności 0.5
- możliwością zadania 7 pytań



■ Poziom trudności  
przed zadaniem

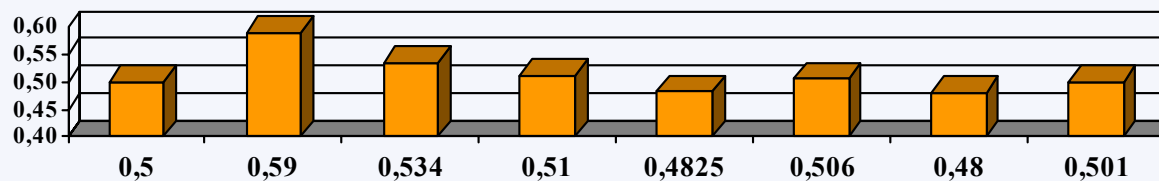
# Algorytm 2

(wg pokrycia dziedzin i st. trudn.)



Na podstawie przykładowego testu:

- z dziedzin: Sygnały  
IPCKolKom LaczaKomun;
- z poziomem trudności 0.5
- możliwością zadania 7 pytań

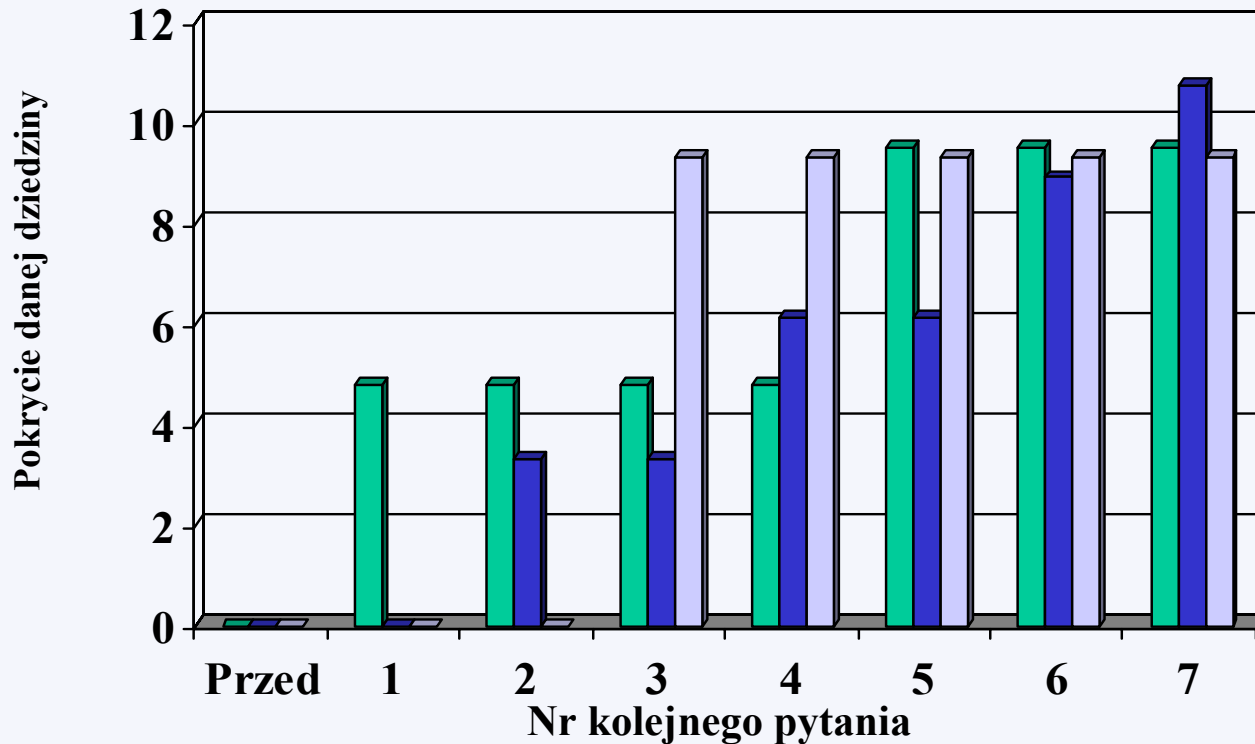


Poziom trudn.



# Algorytm 3

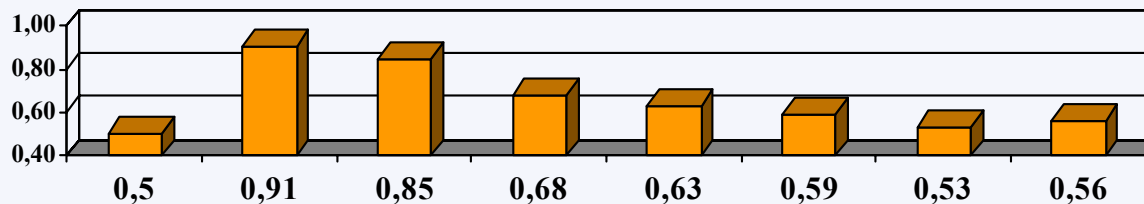
(wg max. pokrycia dziedzin)



D1  
D2  
D3

Na podstawie przykładowego testu:

- z dziedzin: Sygnały  
IPCKolKom LaczaKomun;
- z poziomem trudności 0.5
- możliwością zadania 7 pytań



Poziom trudn.

# System zadawania pytań

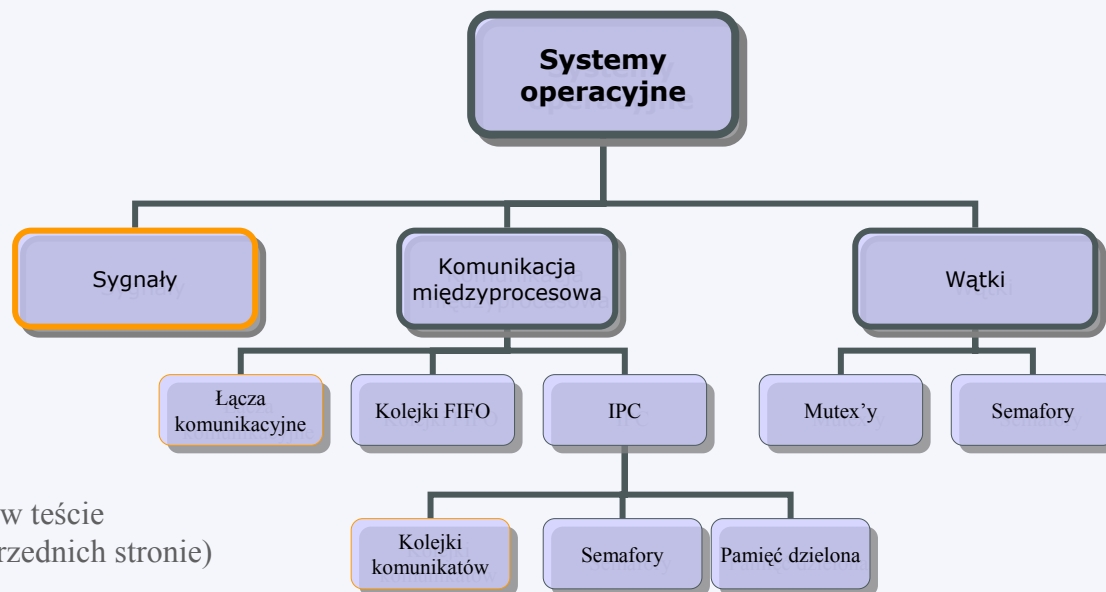
System oferuje następujące możliwości:

- Przeprowadzenie testu
- Zapisanie pytań do określonego strumienia
- Zapisanie statystyk dotyczących testu do określonego strumienia

Przykładowy test oparty jest na pytaniach z kolokwium z systemów operacyjnych. Jest to test wielokrotnego wyboru. Odpowiedzi są mieszane, chyba że jest to nielogiczne. Ocena wyliczana jest na podstawie procentu uzyskanych punktów.

W systemie jest uwzględniony czas w którym trzeba odpowiedzieć, aby punkty za pytanie zostały uwzględnione (dane te są zdefiniowane w pliku z testem).

# Przykładowy test



Dziedziny użyte w teście  
(wykresy na poprzednich stronach)

Schemat dziedzin nauczania, które obejmuje przykładowy test obejmujący 12 dziedzin nauczania oraz prawie 100 pytań z materiału „Systemy operacyjne” (3 semestr informatyki).

Pokrycie dziedzin przez pytanie oraz trudność pytania zostały ustalone losowo (w zakresie dopuszczalnych wartości).