

**REGUŁY PRZEPROWADZENIA EGZAMINU KIERUNKOWEGO  
KATEDRA INFORMATYKI AGH  
ROK 2006**

**Ustalenia ogólne**

1. Zgodnie z zatwierdzonym przez Radę Wydziału EAIiE programem studiów dla kierunku Informatyka wprowadza się niniejszym zasady przeprowadzenia egzaminu kierunkowego na studiach dziennych.
2. Egzamin odbywa się w trakcie ostatniego semestru studiów przed złożeniem pracy magisterskiej. Ponieważ egzamin stanowi integralną część ostatniego semestru studiów jest **obowiązkowy** i jest warunkiem zaliczenia ostatniego semestru.
3. Obowiązują dla niego zasady egzaminów określone w regulaminie studiów w tym (terminy podane są orientacyjnie możliwe przesunięcia maksymalnie o parę dni):
  - a. termin 0 – egzamin próbny (tylko dla rocznika aktualnego). Oceny z tego egzaminu przepisywane są do indeksu na życzenie studenta.  
Termin – 8 maja 2006
  - b. I termin podstawowy - 29 maja 2006
  - c. II termin podstawowy – 5 lipca 2006 (pierwszy dla stypendystów)
  - d. I termin poprawkowy – 1 września dla tych co nie zdali w terminie podstawowym (drugi podstawowy dla stypendystów).
  - e. II termin poprawkowy – 22 września dla tych co nie zdali w terminie podstawowym (poprawka dla stypendystów)
  - f. Studenci wyjeżdżający na praktyki w ramach programów europejskich, mający zwolnienia lekarskie i inne wypadki losowe indywidualnie traktowane przez Dziekana traktują termin I poprawkowy jako termin podstawowy i II poprawkowy jako I poprawkowy. Dodatkowy termin II poprawkowy dla tych studentów to koniec września. Termin ten także jest ostatnim terminem dla studentów, którzy nie byli w stanie z przyczyn losowych przystąpić do któregoś z wcześniejszych egzaminów (indywidualnie traktowanych przez Dziekana)
  - g. W pozostałych przypadkach niezdany egzamin traktowany jest tak jak inny egzamin i obowiązuje procedura opisana w Regulaminie Studiów.
4. Konsekwencją powyższego jest fakt, że pozytywna ocena uzyskana na egzaminie kierunkowym jest warunkiem koniecznym dopuszczenia studenta do obrony pracy (*tak jak każdy inny egzamin*).

**Warunki i tryb przeprowadzania egzaminu:**

5. Egzamin przygotowuje i przeprowadza Komisja d/s Egzaminu Kierunkowego powołana przez Kierownika Katedry Informatyki spośród pracowników Katedry na początku roku dyplomowego. Przewodniczącym Komisji z urzędu jest z-ca kierownika Katedry d/s dydaktycznych.
6. Członkowie komisji – 5 osób - są wyznaczeni przez Kierownika nauczyciele akademicy odpowiedzialni za grupy przedmiotów kierunkowych, które stanowią przedmiot egzaminu.

7. Egzamin przygotowuje Komisja na podstawie pytań (wraz z wariantami odpowiedzi) przygotowanych przez osoby odpowiedzialne za poszczególne przedmioty i moderowane przez członka KEKIRu.
8. Egzamin przeprowadzany jest w formie **testu wielokrotnego wyboru** realizowanego pisemnie lub za pomocą komputera. Pytania są jednakowo punktowane, ocena ustalana jest w oparciu o regulamin studiów.
9. Pytania obejmować będą zakres przedmiotów **kierunkowych jedynie tych o treści informatycznej** zgodnie z programem studiów obowiązującym dla aktualnego rocznika dyplomowego. Lista przedmiotów kierunkowych na rok 2006 została podana poniżej w ustaleniach szczegółowych.
10. Zakres egzaminu w formie zagadnień z każdego przedmiotu został podany do wiadomości studentów.
11. Propozycja szczegółowa.

- Egzamin dla rocznika 2006 odbędzie się w tradycyjny sposób.
- Każda osoba dostanie zestaw pytań oraz tabelkę w której zaznaczać będzie prawidłowe odpowiedzi.
- Egzamin składa się z 50 pytań testowych średnio po 2 z każdego przedmiotu.
- Pytanie testu wielokrotnego wyboru posiada 4 podpunkty. Wybranie **wszystkich** odpowiedzi prawidłowych i tylko odpowiedzi prawidłowych to 2 punkty. Wybranie **tylko** prawidłowych odpowiedzi w liczbie o jeden mniej niż maksymalna to 1 punkt. W przeciwnym wypadku odpowiedź oceniana jest na 0.
- Czas egzaminu nie powinien przekraczać 2 godzin lekcyjnych.
- Egzamin uważać się będzie za zdany, jeżeli student uzyska 50 punktów.
- Skala pozostałych ocen zgodna z regulaminem Studiów (&21.1)
- Egzamin w przyszłości przeprowadzony będzie w środowisku e-learningowym (Moodle, Blackboard).
- Lista przedmiotów obejmuje podstawowe przedmioty informatyczne:

Przedmiot
Wstęp do informatyki
Algorytmy i struktury danych
Teoria obliczeń i złożoności obliczeniowej
Metody obliczeniowe w nauce i technice
Wprowadzenie do systemu UNIX
Systemy operacyjne
Teoria współbieżności
Techniki programowania równoległego
Sieci komputerowe
Systemy rozproszone
Bazy danych
Projektowanie systemów informatycznych
Technologie obiektowe
Inżynieria oprogramowania
Metody sztucznej inteligencji
Techniki i języki programowania
Teoria automatów i języków formalnych
Teoria kompilacji
Asemblery
Technika cyfrowa
Technika mikroprocesorowa
Architektury komputerów

# LISTA ZAGADNIENÍ

## 1. BAZY DANYCH

1. Podstawy technologii baz danych.
2. Elementy budowy modeli relacyjnych baz danych.
3. Konstrukcja poprawnych schematów relacyjnych baz danych
4. Integralność w relacyjnych bazach danych
5. Proces tworzenia systemów baz danych
6. Języki baz danych ze szczególnym uwzględnieniem SQL
7. Architektury systemów baz danych i ich realizacje
8. Podstawy założeń i realizacji systemów OLAP
9. Systemy typu OLTP vs OLAP.

## 2. ASEMBLERY

Materiał dotyczy asemblera procesorów zgodnych z architekturą 8086.

1. Budowa i działanie procesora 8086 (główne moduły).
2. Dostępne programowo rejestry i ich przeznaczenie.
3. Lista rozkazów procesora 8086 i ich sposób działania.
4. Dyrektywy asemblera.
5. Pojęcie adresu logicznego i fizycznego.
6. Tryby adresowania pamięci.
7. Reprezentacja liczb w różnych systemach liczbowych (zapis: szesnastkowy, dziesiętny, ósemkowy, dwójkowy, ...)

## 3. SZTUCZNA INTELIGENCJA

1. Podstawowe zagadnienia z zakresu algorytmów ewolucyjnych i sieci neuronowych.
2. Systemy regułowe i zasady wnioskowania.
3. Podstawowe pojęcia logiki temporalnej, operatory modalne.
4. Paradygmaty systemów agentowych.
5. Logika rozmyta - pojęcia, operatory, relacje.

## 4. WSTĘP DO INFORMATYKI

1. Pojęcia podstawowe: algorytm, informacja – pomiar informacji (wzór Shannona).
2. Aspekty języka programowania – składnia i semantyka;
3. Sposoby opisu składni – BNF, EBNF, diagramy syntaktyczne;
4. Składnia i semantyka instrukcji języka Pascal.
5. Typy danych w językach programowania.
6. Przekazywanie parametrów do procedur i funkcji.
7. Procedury i funkcje rekurencyjne.

8. Rodzaje przydziału pamięci dla zmiennych;
9. Niedziesiątkowe systemy zapisu liczb.
10. Arytmetyka maszynowa, własności, zakres i dokładność liczb;
11. Reprezentacja liczb stałoprzecinkowych – kod U2, działania, nadmiar stałoprzecinkowy;
12. Reprezentacja liczb zmiennoprzecinkowych – zakres, dokładność, działania;
13. Języki imperatywne, aplikatywne i deklaratywne – przykłady.
14. Architektura prostego komputera, jednostki funkcjonalne prostego komputera;
15. Pamięć podręczna, układ DMA, jednostka ALU, CPU;
16. Podstawowe pojęcia ze złożoności obliczeniowej.

## **5. ALGORYTMY I STRUKTURY DANYCH**

1. Klasyfikacja struktur danych.
2. Struktury mnogościowe, liniowe, pierścieniowe, drzewiaste i grafowe.
3. Algorytmy wyszukiwania.
4. Algorytmy sortowania, klasyfikacja metod sortowania. Pojęcie mediany.
5. Rozmieszczanie algorytmiczne.
6. Struktury drzewiaste, drzewa BST i AVL.
7. Struktury grafowe i ich implementacja.

## **6. METODY OBLICZENIOWE W NAUCE I TECHNICIE**

1. Własności numerycznej reprezentacji liczb rzeczywistych i arytmetyki zmiennoprzecinkowej
2. Zadanie, algorytm, realizacja zmiennoprzecinkowa algorytmu.
3. Uwarunkowanie zadania, poprawność numeryczna algorytmu, stabilność numeryczna algorytmu.
4. Interpolacja Lagrange'a, interpolacja Hermite'a, efekt Rungego.
5. Funkcje sklejjane: definicja, zastosowanie.
6. Aproksymacja średniokwadratowa i jednostajna.
7. Podstawowe własności wielomianów Czebyszewa oraz ich zastosowania.
8. Kwadratury Newtona-Cotesa i Gaussa.
9. Algorytmy rozwiązywania układów równań liniowych: eliminacji Gaussa, dekompozycji LU, blokowe.
10. BLAS, LAPACK, PBLAS, BLACS, ScaLAPACK.
11. Metody iteracyjne Jacobiego, S-R, SOR, Czebyszewa.
12. Metody znajdowania rozwiązań równań nieliniowych.
13. Wyznaczanie całek metodami Monte Carlo.
14. FFT - szybka transformata Fouriera.
15. Podstawowe metody minimalizacji lokalnej i globalnej.
16. Optymalizacja kombinatoryczna.
17. Przedstaw metoda simulated annealing.
18. Podstawowe metody rozwiązywania równań różniczkowych zwyczajnych.
19. Podstawowe metody rozwiązywania równań różniczkowych cząstkowych.

20. Konstruowanie wydajnych algorytmów obliczeniowych a znajomość architektur komputerowych.
21. Aktualne tendencje w obliczeniach dużej skali.

## **7. TECHNOLOGIE OBIEKTOWE**

1. Zasady modelowania obiektowego, jednostki abstrakcji, sposoby organizacji modelu.
2. Podstawy notacji UML (diagramy klas, przypadków użycia, interakcji).
3. Wymagania – modelowanie przypadków użycia. Współdziałanie obiektów jako realizacja przypadków użycia.
4. Struktura logiczna modelu. Modelowanie powiązań, agregacja a kompozycja. Generalizacja a dziedziczenie interfejsu i implementacji. Pakiety i relacje między nimi.
5. Wsparcie wielokrotnego użycia w modelowaniu obiektowym. Wzorce projektowe.

## **8. WPROWADZENIE DO SYSTEMU UNIX**

1. Użytkownicy w systemie UNIX, konfiguracja zarządzanie,
2. Systemy plików, organizacja, administrowanie,
3. Zarządca przestrzeni logicznej (LVM),
4. Procesy w systemie UNIX, podstawy zarządzani, sygnały, zmiana wartości NICE,
5. Kolejowanie zadań w wersjach SystemV i BSD,
6. Badanie wydajności systemu (vmstat, sar)
7. Interpretery poleceń, zmienne konfiguracyjne, środowiska programowe, podstawowe pliki konfiguracyjne,
8. Programowanie w języku interpretera poleceń bash,
9. Podstawy protokołu TCP/IP.

## **9. ARCHITEKTURY KOMPUTERÓW**

1. Geneza zapotrzebowania na dużą moc obliczeniową,
2. miary oceny wydajności systemów komputerowych,
3. budowa i własności procesorów CISC, RISC, EPIC,
4. organizacja komputera,
5. taksonomie komputerów o wielu procesorach,
6. własności poszczególnych rozwiązań,
7. systemy klastrowe i gridowe,
8. podstawowe własności modeli programowania systemów o wielu procesorach.

## **10. TECHNIKA MIKROPROCESOROWA**

1. Elementy składowe mikroprocesora i systemu. Pojęcia podstawowe: instrukcja, długość słowa, przestrzeń adresowa, sygnały sterujące, magistrala. Organizacja prostego komputera: przesłania międzyrejstrowe i mikrooperacje, funkcje sterujące, mikrorozkazy, pamięć sterująca.

2. Od przetwarzania sekwencyjnego do architektury superskalarnej. Architektura mikroprocesora ze stałą listą instrukcji, procesor CISC. Przetwarzanie sekwencyjne. Podstawowe podzespoły funkcjonalne: cykle, sieci działań, przebiegi czasowe, sposoby adresowania, organizacja przestrzeni adresowej. Przetwarzanie potokowe. Wielozadaniowość, praca w trybie wirtualnym. Zarządzanie i ochrona pamięci: praca w trybie chronionym. Segmentacja i stronicowanie. Pamięć typu „cache” poziomu pierwszego i drugiego. Przetwarzanie wielopotokowe. Konflikt zasobów, argumentów i sterowania. Mechanizm przewidywania skoków warunkowych i zmiany kolejności wykonywanych instrukcji, jednostka alokacji rejestrów. Przetwarzanie wielowątkowe. Rozwiązania wieloprocessorowe. Procesor RISC.
3. Układy wejścia-wyjścia i zasady ich przyłączania do systemu. Programowane układy szeregowego i równoległego transferu informacji, układy czasowe. Kontrolery przerwań i sterowniki bezpośredniego dostępu do pamięci. Układy specjalizowane. Zasady współpracy i przyłączania urządzeń zewnętrznych do systemów mikroprocesorowych: standardy przesyłania informacji, magistrale systemowe – równoległe i szeregowe.

## **11. TECHNIKA CYFROWA**

1. Pojęcia podstawowe - podział układów cyfrowych, parametry układu cyfrowego. Układy kombinacyjne i sekwencyjne. Elementarne układy przełączające. Techniki realizacyjne układów cyfrowych - układy TTL, MOS. Ogólne charakterystyki i porównanie wymienionych technik. Sposoby łączenia układów różnych klas. Podstawowe własności algebry Boole'a. Bramki logiczne, tablice prawdy. Realizacje złożonych funkcji logicznych i minimalizacja projektu. Siatki Karnaugh'a. Przerzutniki: bistabilne, monostabilne i astabilne, sposoby sterowania, własności dynamiczne. Liczniki: asynchroniczne i synchroniczne, binarne, dziesiętne, zliczające do dowolnego n, liczniki rewersyjne. Projektowanie liczników o określonej długości. Automaty. Rejestry: równoległe i szeregowe. Układy komutacyjne: multipleksery, demultipleksery. Kodery, dekodery, konwertery kodów.
2. Układy arytmetyczne: sumatory szeregowo i równoległe, komparatory, arytmometry, jednostki arytmetyczno-logiczne. Układy przyspieszenia przeniesienia.
3. Programowalne układy logiczne. Na styku techniki cyfrowej i analogowej: przetworniki A/C i C/A. Przykłady rozwiązania wybranych układów cyfrowych. Zasady projektowania systemów cyfrowych.

## **12. SYTEMY OPERACYJNE**

1. Rodzaje systemów operacyjnych.
2. Systemy czasu rzeczywistego.
3. Stany procesu.
4. Budowa procesu.
5. Blok kontrolny procesu.
6. Algorytmy szeregowania procesów.
7. Wątki.

8. Elementy współdzielone przez wątki wchodzące w skład tego samego procesu.  
Poziomy realizacji wątków..
9. Definicja zakleszczenia.
10. Warunki zajęcia zakleszczenia w systemie.
11. Graf przydziału zasobów.
12. Zapobieganie i unikanie zakleszczeń.
13. Algorytm bankiera.
14. Ciągły przydział pamięci.
15. Stronicowanie.
16. Segmentacja.
17. Pamięć wirtualna.
18. Algorytmy wyboru stron do wymiany.
19. Problem sekcji krytycznej.
20. Warunki poprawnego rozwiązania problemu sekcji krytycznej.
21. Mechanizmy synchronizacyjne.
22. Semafor. Monitory.
23. Systemy plików.
24. Urządzenia.

### **13. SIECI KOMPUTEROWE**

Modele warstwowe: OSI/ISO i TCP/IP

Warstwa fizyczna modelu OSI/ISO

- Charakterystyka mediów transmisyjnych
- Kodowanie informacji
- Zakłócenia transmisji
- Urządzenia warstwy fizycznej
- Okablowanie strukturalne i inne sposoby budowy sieci lokalnych

Warstwa łącza danych modelu OSI/ISO

- Zadania warstwy łącza danych
- Protokół dostępu do medium
- Zasada działania protokołu CSMA/CD
- Rozszerzenia standardu Ethernet
- Adresacja fizyczna
- Zasada działania mostów w sieci Ethernet
- Sieci wirtualne
- Protokół budowy drzewa rozpinającego

Warstwa sieciowa modelu OSI/ISO

- Zadania warstwy sieciowej
- Adresacja logiczna
- Cechy protokołu IP
- Protokół ARP
- Protokół ICMP
- Metody uzyskiwania adresu IP

- Routing statyczny i dynamiczny w sieciach IP
- Tablica routingu
- Metryka i zbieżność w protokołach routingu dynamicznego
- Programy traceroute i ping

#### Warstwa transportowa modelu OSI/ISO

- Zadania warstwy transportowej
- Kontrola przepływu
- Osiąganie niezawodności w protokole sieciowym
- Protokół TCP
- Protokół UDP
- Zastosowania protokołów TCP i UDP

## **14. SYSTEMY ROZPROSZONE**

1. Definicja aplikacji rozproszonej, systemu rozproszonego oraz rozproszonego systemu operacyjnego.
2. Definicja i rola warstwy pośredniczącej (middleware).
3. Czas i jego synchronizacja w systemach rozproszonych (zegary wektorowe, Lamporta).
4. Algorytmy elekcji w systemach rozproszonych.
5. Komunikacja przy użyciu gniazd (klasyfikacja i rodzaje portów).
6. Wykorzystanie gniazd BSD do komunikacji między procesowej.
7. Wykorzystanie RPC (pojęcia: PortMapper, XDR, procedura idempotentna).
8. Serializacja i technologia RMI (pojęcia: zdalny obiekt, serwer nazw, stub i skeleton).
9. Mechanizm RMI Activation.
10. Własności technologii CORBA.
11. Elementy architektury CORBA (język IDL, GIOP, IIOP, DII, DSI, BOA, POA, IOR).
12. Usługi CORBA.

## **15. TECHNIKI PROGRAMOWANIA RÓWNOLEGŁEGO.**

1. Obliczenia typu HPC, HTC i gridowe.
2. Efektywność czasowa i pamięciowa obliczeń równoległych.
3. Wpływ architektury komputera i środowiska programowania na algorytm.
4. Analiza przyspieszanie i efektywności obliczeniowej programu równoległego przy stałym i zmiennym rozmiarze problemu.
5. Modele CSP, BSP, PRAM. Model PCAM.
6. Modele obliczeń równoległych. Środowiska PVM i MPI.
7. Języki i środowiska programowania równoległego: przegląd własności HPF i OpenMP.
8. Odwzorowanie struktur danych na architekturę.



## **16. PROJEKTOWANIE SYSTEMÓW INFORMATYCZNYCH**

1. Analiza systemowa – podstawowe pojęcia, cele, zadania, metody; analityk systemowy: charakterystyka cech, ról, zadań, czynności, metod pracy.
2. Modele cyklu tworzenia systemu informatycznego (kaskadowy, przyrostowy, prototypowy, spiralny) – opis, zalety, wady, użyteczność, zakres stosowania.
3. Faza strategiczna i faza określania wymagań w modelu kaskadowym tworzenia systemów informatycznych – szczegóły (czynności, metody, dokumentowanie prac, produkty wynikowe).
4. Metodyka strukturalna analizy i projektowania systemów informatycznych (notacje i ich interpretacja, proces tworzenia pełnego modelu strukturalnego: trzy aspekty (DFD, ERD, STD), specyfikacje projektowe.
5. Podstawowe elementy projektowania interfejsu.

## **17. INŻYNIERIA OPROGRAMOWANIA**

1. Zakres inżynierii oprogramowania
  - 1.1. Inżynieria systemów komputerowych
  - 1.2. Proces tworzenia oprogramowania
  - 1.3. Zarządzanie przedsięwzięciami
2. Wymagania
  - 2.1. Wymagania stawiane oprogramowaniu
  - 2.2. Proces inżynierii wymagań
  - 2.3. Modele systemów
  - 2.4. Prototypowanie oprogramowania
  - 2.5. Specyfikacja formalna
3. Projektowanie
  - 3.1. Projektowanie architektoniczne
  - 3.2. Architektury systemów
  - 3.3. Projektowanie z użyciem wielokrotnym
  - 3.4. Specjalne zagadnienia projektowania (RTS, interfejs użytkownika)
4. Weryfikacja i walidacja oprogramowania
  - 4.1. Weryfikacja i walidacja systemów
  - 4.2. Testowanie
5. Zarządzanie projektami
  - 5.1. Zarządzanie personelem
  - 5.2. Zarządzanie jakością
  - 5.3. Ulepszanie procesu
6. Ewolucja
  - 6.1. Systemy odziedziczone
  - 6.2. Modyfikacja oprogramowania
  - 6.3. Restrukturyzacja oprogramowania
  - 6.4. Zarządzanie konfiguracjami

## **18. JĘZYKI I TECHNIKI PROGRAMOWANIA**

1. Zakres obowiązywania deklaracji w C i C++ (wskazać i omówić odpowiednie przypadki).
2. Realizacja postulatów programowania obiektowego w C i C++ (wskazać i omówić odpowiednie napisy).
3. Konwersje niejawne w C i C++ (cel i przypadki występowania).
4. Mechanizm polimorfizmu.
5. Przeciążanie operatorów w C i C++.

## **19. TEORIA KOMPILACJI**

1. Translacja, kompilacja, interpretacja, podobieństwa, różnice, zalety, wady.
2. Podstawowe moduły funkcjonalne translatora i ich zadania.
3. Analiza leksykalna, syntaktyczna, semantyczna; błędy leksykalne, syntaktyczne, semantyczne.
4. Narzędzia i metody analizy leksykalnej. Konstruowanie i implementacja skanerów.
5. Narzędzia i metody analizy składniowej, rodzaje i własności wykorzystywanych gramatyk. Zależności między klasami gramatyk wykorzystywanych do projektowania parserów. Niejednoznaczność gramatyki. Konstruowanie i implementacja parserów.
6. Generatory skanerów i parserów. Dane wejściowe i wyjściowe tych generatorów. Zasady, na których oparte jest ich działanie.
7. Istota mechanizmu gramatyk atrybutywnych, rodzaje atrybutów, klasy gramatyk atrybutywnych, metody analizy semantycznej.
8. Kontrola typów.
9. Problemy organizacji pamięci. Obsługa podprogramów. Ramka stosu. Przekazywanie parametrów do podprogramów, l-wartości i r-wartości. Struktury danych tworzone i obsługiwane przez kompilator w czasie kompilacji, struktury danych wykorzystywane przez skompilowany program podczas wykonania.
10. Generacja kodu pośredniego. Języki pośrednie.
11. Metody optymalizacji kodu pośredniego. Analiza przepływu. Wykorzystanie informacji zebranych podczas analizy przepływu w transformacjach usprawniających kod. Optymalizacja „przez szparkę”.
12. Generacja kodu ostatecznego, przydział rejestrów. Zastosowanie metod analizy przepływu przy przydziale rejestrów.

## **20. TEORIA AUTOMATÓW I JĘZYKÓW FORMALNYCH**

1. Symbol, łańcuch, zbiór słownikowy, język formalny, podstawowe operacje na łańcuchach i na językach. Własności języków. Zależności między językami.
2. Sposoby definiowania języków formalnych, pojęcie automatu, gramatyki.
3. Hierarchia Chomsky’ego dla gramatyk i języków. Hierarchia Chomsky’ego a rodzaje automatów. Języki nieobjęte hierarchią Chomsky’ego.
4. Własności zamkniętości klas języków.
5. Automaty deterministyczne i nondeterministyczne. Rodzaje automatów. Równoważność deterministycznych i nondeterministycznych automatów różnych

rodzajów. Własności automatów. Języki akceptowane przez automaty poszczególnych rodzajów.

6. Jednoznaczność gramatyki. Determinizm automatu ze stosem. Bezkontekstowe języki jednoznaczne, Bezkontekstowe języki deterministyczne.

7. Maszyna Turinga jako automat lingwistyczny oraz jako model procesu obliczeniowego.

8. Problem, problem rozstrzygalny i nierozstrzygalny. Przykłady lingwistycznych i obliczeniowych problemów rozstrzygalnych i nierozstrzygalnych. Problem przynależności słowa do języka generowanego przez gramatykę.

9. Wyrażenie regularne a automat skończony.

10. Tożsamości dla wyrażeń regularnych, równoważność dwóch wyrażeń regularnych.

11. Niepustość, skończoność, nieskończoność języków regularnych a liczba stanów automatu skończonego.

12. Sposoby akceptacji języków przez automat ze stosem.

13. Automat skończony niedeterministyczny, deterministyczny, minimalny;  $\epsilon$ -przejścia w automacie skończonym.

14. Spełnialność tezy lematu o pompowaniu dla języków regularnych i bezkontekstowych

## **21. TEORIA OBLICZEŃ I ZŁOŻONOŚCI OBLICZENIOWEJ**

1. Jakie problemy zaliczamy do klasy problemów NP-zupełnych?

2. Co to znaczy, że funkcja  $f$ , przekształcająca egzemplarze problemu  $P_{i1}$  w egzemplarze problemu  $P_{i2}$ , zachowuje problem? ( $D_{P_i}$  - zbiór zadań dla problemu  $P_i$ ,  $Y_{P_i}$  - zbiór zadań dla problemu  $P_i$  z rozwiązaniem TAK)

3. Jakie problemy zawiera klasa LOG-Space?

4. Na czym polega metoda dowodzenia faktu, że  $P_i$  należy do NP-com?

5. Jaka zależność zachodzi pomiędzy klasami co-NP i NP przy założeniu P różne od NP?

6. Jakie byłyby konsekwencje znalezienia wielomianowego deterministycznego algorytmu dla problemu NP-zupełnego?

7. Od czego zależy następny krok maszyny Turinga?

8. Funkcje rekursywne definiowane są przy pomocy funkcji pierwotnych. Jakie funkcje zaliczamy do funkcji pierwotnych?

9. Co można powiedzieć o algorytmie obliczenia dla problemu STOPu?

10. Co można powiedzieć o efektywności obliczenia równoległego?

11. Jakie parametry opisują złożoność obliczenia równoległego?

12. Modele rejestrowe: RAM i RASP.

13. Pojęcia wielomianowej redukowalności problemów, zupełności.

14. Modele obliczenia z wyrocznią.