

# OPRACOWANIE ZAGADNIENÍ - ISI

Hej :) pewnie pojawi się tu w części pytań więcej niż jedna wersja, więc proponuję żeby każda wersja była kolejnym podrozdziałem, tak żeby było wiadomo gdzie się jedno kończy, a zaczyna drugie :) Jak dodacie pierwsze opracowanie to wywalcie TODO z nazwy rozdziału, to od razu będzie widać co nam jeszcze zostało.

## [23. Organizacja systemów rozproszonych](#)

[Opracowanie Kuby - niech ktoś przeczyta i sprawdzi :\)](#)

## [24. Ocena systemów równoległych](#)

[Opracowanie Kuby - niech ktoś przeczyta i sprawdzi :\)](#)

## [25. Uczenie indukcyjne: metody, zastosowania](#)

[Opracowanie Agi](#)

## [26. Uczenie ze wzmocnieniem](#)

[Opracowanie Adama](#)

[Programowanie dynamiczne](#)

## [27. Modele obrazów: wektory cech, niezmienniki, relacje przestrzenne](#)

[Opracowanie Agi](#)

## [28. Klasyfikacja, opisywanie, interpretacja, rozumienie](#)

[Opracowanie Agi](#)

## [29. Obliczenia miękkie a obliczenia tradycyjne \(algorytmiczne\)](#)

[Opracowanie Agi](#)

## [30. Jakie techniki wchodzą w skład obliczeń miękkich. Charakterystyka każdej z nich](#)

[Opracowanie Agi](#)

## [31. Podstawy formalnego opisu języka naturalnego: założenia i stosowane metody](#)

[Opracowanie Adama P.](#)

## [32. Współczesna technologia językowa: narzędzia, zasoby językowe i ich zastosowania](#)

[Opracowanie Adama P.](#)

## [33. Proces przetwarzania języka naturalnego: typowe etapy, cele, stosowane metody](#)

[Opracowanie Adama P.](#)

## 23. Organizacja systemów rozproszonych

### Opracowanie Kuby - niech ktoś przeczyta i sprawdzi ;)

System rozproszony jest kolekcją niezależnych, autonomicznych komputerów, które dla użytkownika prezentują się jak jeden komputer. Architektury systemów rozproszonych mogą być klasyfikowane na kilku różnych poziomach:

1) ze względu na mechanizm sterowania

- SIMD

- MIMD

2) ze względu na organizację komunikacji i pamięci

- message-passing
- pamięć współdzielona
  - UMA
  - NUMA

3) ze względu na charakter sieci połączeniowej

- statyczne
- dynamiczne

4) ze względu na ziarnistość procesora

- coarse-grain
- medium-grain
- fine-grain

Ad 1)

Podział wg taksonomii Flynna:

- SISD – single instruction, single data – zwykły procesor sekwencyjny
- SIMD - single instruction, multiple data – komputery wektorowe – wiele procesorów przetwarza ten sam strumień instrukcji na różnych danych
- MISD – multiple instruction, single data - wiele równolegle wykonywanych programów przetwarza jednocześnie jeden wspólny strumień danych. W zasadzie jedynym zastosowaniem są systemy wykorzystujące redundancję (wielokrotne wykonywanie tych samych obliczeń) do minimalizacji błędów.
- MIMD – multiple instruction, multiple data – równolegle wykonywane jest wiele ciągów instrukcji na różnych danych

Ad 2)

Architektura message-passing – każdy procesor ma bezpośredni dostęp tylko do swojej pamięci lokalnej. Dostęp do informacji zawartych w innych bankach pamięci odbywa się z wykorzystaniem procesora, do którego bank ten należy (nie można bezpośrednio sięgnąć do pamięci), poprzez wysyłanie i odbiór komunikatów pomiędzy procesorami.

Pamięć współdzielona – dwie typowe organizacje dostępu do pamięci:zre

- UMA – jednolity dostęp do pamięci – każdy procesor łączy się z każdą pamięcią na tych samych zasadach przez sieć połączeń
- NUMA – każdy z procesorów jest bezpośrednio połączony ze swoim bankiem pamięci lokalnej, do reszty pamięci musi sięgać poprzez sieć połączeń. Dostęp do swojej pamięci przebiega znacznie szybciej niż dostęp do dowolnej innej pamięci.

Ad3)

Sieci statyczne:

- sieć pełna – połączenie każdy z każdym bezpośrednio
- gwiazda – jeden z procesorów znajduje się w centrum sieci, reszta komunikuje się ze sobą poprzez ten wyznaczony procesor
- magistrala – procesory połączone za pomocą pojedynczej szyny, bezpośrednia komunikacja tylko z najbliższymi sąsiadami

- ring – magistrala z dodatkowym bezpośrednim połączeniem pierwszego i ostatniego procesora na magistrali.
- mesh – krata, procesory ułożone w macierz
- mesh wrap-around – krata z zapętlzeniami
- drzewo binarne
- hypercube – procesory ułożone w n-wymiarową kostkę. Każdy procesor ma n sąsiadów, których etykiety w zapisie binarnym różnią się jednym bitem.

Sieci dynamiczne:

- architektura szynowa – procesory podłączone do wspólnej szyny komunikacyjnej, dowolne połączenie z pamięcią blokuje wszystkie inne połączenia.
- przełącznica krzyżowa – sieć krzyżujących się połączeń, na jednym boku mamy procesory, na innym pamięci, połączenie pomiędzy procesorem a pamięcią polega na przestawieniu stanu przełącznika. Każdy procesor może bezpośrednio połączyć się z każdą pamięcią, nie ma własności blokowania (to znaczy że dowolne połączenie nie blokuje połączeń pomiędzy dowolnym innym procesorem i dowolną inną pamięcią). Wadą jest duży koszt.
- sieć wielostanowa – pomiędzy procesorami a bankami pamięci znajdują się zestawy przełączników, z których każdy może być w pewnym stanie. Przykładem jest sieć Omega. W tej sieci przełącznik może znajdować się w dwóch stanach: przełączenie proste i krzyżowe. Sieć ma własność blokowania (ustanowienie jednego połączenia może zablokować możliwość ustanowienia drugiego połączenia pomiędzy innym procesorem i innym bankiem pamięci).

Ad 4)

- coarse-grain – gruboziarniste – system składa się z mniejszej ilości bardziej rozbudowanych komponentów
- fine-grain – drobnoziarniste – system składa się z dużej liczby niewielkich komponentów
- medium-grain – coś pośredniego pomiędzy ww.

## 24. Ocena systemów równoległych

### Opracowanie Kuby - niech ktoś przeczyta i sprawdzi ;)

Miary wydajności systemów równoległych:

Czas wykonania równoległego  $T_{par}$  – czas od rozpoczęcia obliczeń do momentu zakończenia obliczeń przez ostatni procesor. Na czas ten składa się czas wykonywania części sekwencyjnej kodu (która musi zostać wykonana na jednym procesorze) i czas wykonania części równoległej.

Przyspieszenie ( $S$ ) – stosunek czasu wykonania na jednym procesorze ( $T_{seq}$ ) do czasu potrzebnego na wykonanie tego samego zadania z wykorzystaniem systemu równoległego o  $p$  procesorach. Przyspieszenie można liczyć na kilka sposobów:

- względne –  $T_{seq}$  jest czasem wykonania alg. równoległego na jednym z procesorów systemu
- rzeczywiste –  $T_{seq}$  jest czasem wykonania najlepszego alg. sekwencyjnego na jednym z procesorów systemu

- bezwzględne – Tseq jest czasem wykonania najlepszego alg. sekwencyjnego na najlepszym komputerze sekwencyjnym

Efektywność (E) – stosunek przyspieszenia do liczby procesorów

Koszt – iloczyn równoległego czasu przetwarzania i liczny procesorów

Skalowalność – miara zdolności systemu do zwiększenia przyspieszenia proporcjonalnie do liczby procesorów. System jest skalowalny, jeśli utrzymuje stałą efektywność przy jednoczesnym zwiększaniu liczby procesorów i rozmiarze problemu.

Systemy równoległe opisuje się za pomocą praw:

- prawo Amdhala – przyspieszenie nie może być większe niż odwrotność części sekwencyjnej programu ( $S \leq 1/P_{seq}$ )
- prawo Gustafsona – przyspieszenie jest większe równe iloczynowi liczby procesorów i części równoległej programu ( $S \geq p \cdot P_{par}$ )

## 25. Uczenie indukcyjne: metody, zastosowania

### Opracowanie Agi

#### Indukcja

Nie jest poprawną formą wnioskowania logicznego. Jest wnioskowanie bottom-up czyli od szczegółu do ogółu. W indukcji nie znamy ogólnej reguły lub reguł. Znamy tylko zbiór faktów – przykładów. Musimy domyślać się czegoś sensownego, indukować ogólną regułę, która będzie również prawdziwa dla innych przykładów, niewykorzystanych w procesie indukcji. Np. obserwuje kilka owoców i stwierdzam, że jak coś jest miękkie, ma skórkę i dowolny kształt to owoc. Przykładem wnioskowania indukcyjnego jest rozpoznawanie obiektów za pomocą klasyfikatorów. EDIT Łukasz: Problem klasyfikacji wykorzystuje dedukcję wnioskując z wiedzy, którą przechowuje model (jedynie w kNN można się zastanawiać nad indukcją). Na podstawie wcześniej opisanych reguł klasyfikator podejmuje decyzję na temat wyboru klasy np. wnioskowanie na podstawie teorii prawdopodobieństwa. Natomiast uczenie klasyfikatorów wykorzystuje indukcję.

#### Tryby uczenia:

- inkrementacyjny - przykłady dostarczane pojedynczo, udoskonalenie hipotezy po każdym z nich
- epokowy - cykl uczenia podzielony na epoki, w którym hipotezę udoskonala się po przetworzeniu danej liczby przykładów

- wsadowy - uczeń otrzymuje wszystkie przykłady, po przetworzeniu których podaje gotową hipotezę, nie można odczytać hipotez w trakcie procesu uczenia. W przypadku zmiany zbioru przykładów uczenie przeprowadza się od nowa.
- korekcyjny - odmiana inkrementacyjnego, w którym uczeń najpierw przedstawia swoją odpowiedź na podany przykład, a następnie otrzymuje informację korygującą, powstałą na podstawie różnicy wartości otrzymanej od oczekiwanej

Metody uczenia indukcyjnego można podzielić na:

- **uczenie się pojęć (uczenie z nadzorem)**
  - Pojęcia służą do klasyfikacji obiektów na grupy (kategorie)
  - Podstawowa wersja: podział na dwie grupy – obiekty należące do danego pojęcia (pozytywne przykłady) i nie należące do niego (negatywne przykłady)
  - uczenie się funkcji odwzorowującej przykłady na skończony i niewielki zbiór kategorii
  - przykładowe metody: kNN, naiwny bayes, SVM, sieci neuronowe, drzewa decyzyjne EDIT Łukasz: kNN nie jest uczone - dla danych wejściowych od razu jest gotowe do klasyfikacji.
- **tworzenie pojęć (uczenie bez nadzoru)**
  - Nie zawsze kategorie pojęć są znane – uczeń obserwuje nieetykietowane przykłady (opisy obiektów)
  - Uczeń, na podstawie obserwacji, grupuje obiekty w kategorie zgodnie z pewnymi kryteriami podobieństwa
  - Tworzenie pojęć łączy dwa podzadania: podział przykładów trenujących na grupy, które odpowiadają kategoriom i nauczanie się pojęć odpowiadających tworzonemu kategoriom, aby było możliwe klasyfikowanie nowych przykładów. EDIT (Łukasz): uczenie bez nadzoru nie ma bezpośrednio nic wspólnego z klasyfikacją i dla Waszego dobra radzę nie mówić nic takiego przy komisji :P (Adam): można przyjąć, że klasyfikacja jest nadzorowana a grupowanie (klasteryzacja) jest nienadzorowana.
  - Przykładowa metoda - k-średnich
- **aproksymacji funkcji (uczenie z nadzorem)**
  - zbiór wartości uczonej funkcji to zbiór liczb rzeczywistych
  - Przykłady – pary składające się z argumentu funkcji, reprezentowanego zazwyczaj przez wektor liczb rzeczywistych i jej wartości dla tego argumentu
  - Uczeń (system) ma wygenerować funkcję dobrze przybliżającą funkcję docelową
  - Wartość funkcji ma być obliczona z dużą dokładnością nie tylko dla przykładów, ale – indukcyjne uogólnienie – dla dowolnych innych argumentów z dziedziny
  - Przykładowe metody: ewolucyjne, inspirowane naturą

## SVM

Proces uczenia tego klasyfikatora polega na znalezieniu hiperpłaszczyzny dzielącej obiekty z dwóch. W przypadku gdy niemożliwe jest znalezienie takiej hiperpłaszczyzny stosuje się funkcję jądra, która tak przekształca przestrzeń, że obiekty są separowalne liniowo. Funkcję jądra stosuje się wtedy w obliczeniach zamiast iloczynu skalarnego (stosowanego w przypadku

liniowo separowalnych danych) . EDIT Łukasz: SVM transformuje dane funkcją jądra do przestrzeni separowalnej liniowo. Linia rozdzielająca klasy wyznaczana jest w taki sposób, aby pas (linia + margines) był jak największy. Czasami operowanie na wielowymiarowych danych jest niewygodne i kosztowne, dlatego wymyślono coś takiego jak Kernel Trick, polega to na nieobliczaniu bezpośrednio wektorów reprezentujących dane, ale na obliczaniu od razu wartości ich iloczynów skalarnych -- pozwala to uniknąć wielu problemów i zmniejszyć złożoność obliczeniową.

Dokładny opis SVM: [http://156.17.130.153/~agonczarek/Estymator/Estymator\\_5.pdf](http://156.17.130.153/~agonczarek/Estymator/Estymator_5.pdf)

## **Drzewa decyzyjne**

- Popularna metoda wnioskowania indukcyjnego
- Służy do aproksymacji docelowej funkcji o wartościach dyskretnych
- Uczona funkcja jest reprezentowana w postaci drzewa
- Nauczone drzewo może być przedstawione w postaci reguł if-then (czytelność)
- Odpowiednia dla zaszumionych danych
- Ma możliwości uczenia wyrażeń dysjunkcyjnych (alternatyw).
- Do tej rodziny należą ID3, ASSISTANT, C4.5.
- Ich obciążenie to preferencja małych drzew.

Uczenie się drzewa decyzyjnego z danych – zasady ogólne

1. Wstęp – dobór zbioru atrybutów, aby były istotne dla klasyfikacji.
2. Najpierw tworzymy jednowęzłowe drzewo skierowane i przypisujemy mu zbiór wszystkich zaobserwowanych obiektów.
3. Dla każdego, jeszcze nie rozpatrzonego liścia konstruowanego drzewa sprawdzamy, czy przypisany mu zbiór obiektów należy do jednej klasy. Jeśli tak – etykietujemy ten liść nazwą tej klasy. Więcej tego liścia nie rozpatrujemy.
4. Po zakończeniu powyższego postępowania zwykle następuje krok ‘obcinania’ (pruning) – nadmiernie rozrośniętych gałęzi drzewa

ID3 wybiera atrybuty o największej entropii, C4.5 jest modyfikacją ID3, która dodaje możliwość korzystania z danych ciągłych, generowanie reguł “mniejszy/większy od”, włączenie przycinania drzewa, umożliwienie klasyfikacji mimo brakujących danych.

## **26. Uczenie ze wzmocnieniem**

### **Opracowanie Adama**

Podejście do uczenia usytuowane pomiędzy uczeniem z nauczycielem oraz nauczeniem bez nadzoru. Wykorzystuje informację zwrotną w postaci zmiany stanu środowiska oraz numerycznej wartości nagrody/wzmocnienia. W sumie najbardziej jest zbliżone do biologicznego, w większości przypadków nauczyciel nie może nam dokładnie przekazać wiedzy. Częściej spotyka się przypadek, w którym nauczyciel tylko pokazuje czy w dobrą stronę zmierzamy. Zadaniem ucznia jest wyciągnąć z tego poprawne wnioski. W literaturze zamiennie jest używana nazwa nagrody/wzmocnienia. Na tym wykładzie będę stosował wzmocnienia.

Podstawowym założeniem nauczania ze wzmocnieniem jest maksymalizacja funkcji oceny agenta, podejmującego decyzje w określonym środowisku. Agent jest niepewny swojego otoczenia. Dopuszczalna jest możliwość nieznajomości środowiska oraz decyzje najczęściej są podejmowane w dyskretnych krokach czasu. Niepewność środowiska oznacza, iż generowane pod wpływem akcji wzmocnienia oraz zmiany środowiska mogą być stochastyczne. Nieznajomość natomiast oznacza, że rozkłady prawdopodobieństwa mogą być nieznane agentowi. Ponadto agent nie ma wpływu na te rozkłady prawdopodobieństwa.

Ogólny scenariusz działania:

- – Dla każdego kroku czasu  $t$ :
  - – Obserwuj aktualny stan  $x_t$
  - – Wybierz akcję  $a_t$  do wykonania w stanie  $x_t$
  - – Wykonaj akcję  $a_t$
  - – Obserwuj wzmocnienie  $r_t$  i następny stan  $x_{t+1}$
  - – Ucz się na podstawie doświadczenia  $\langle x_t, a_t, r_t, x_{t+1} \rangle$

Agent obserwuje stan  $x_t$ . Na podstawie obserwacji i własnych doświadczeń (strategii) wybiera akcję. Po wykonaniu akcji agent otrzymuje rzeczywistoliczbowe wzmocnienie oraz ew. zmianę stanu środowiska (stanowi to pewną miarę, jakości podjętego działania). Uczenie się polega na zmianie strategii podejmowania akcji na podstawie obserwowanych zmian i wzmocnień. Większość znanych algorytmów uczenia można przedstawić, jako konkretyzację tego schematu.

TODO Procesy Decyzyjne Markova  
TODO Programowanie dynamiczne

Łukasz:

## Procesy decyzyjne Markowa

Dla procesu decyzyjnego mamy określony zbiór stanów, zbiór akcji, funkcje oceny i funkcje przejścia stanów. Proces decyzyjny jest procesem Markowa jeżeli:

- 1) Wybór akcji jest zależny tylko od aktualnego stanu
- 2) Ocena zależna od aktualnego stanu i wybranej akcji jest zmienną losową
- 3) Następny stan jest realizacją zmiennej losowej

Proces decyzyjny Markowa uczony jest ze wzmocnieniem.

Rozważmy rzeczywisty przykład ;):

Wg teorii ekonomicznej zwiększenie stóp procentowych wzmacnia wartość waluty, a zmniejszenie osłabia tę wartość. Załóżmy, że celem jest utrzymanie wartości waluty na stałym poziomie względem dolara, a stany czasowe zmieniane są co kwartał.

- Możliwymi akcjami NBP są: zwiększenie stóp procentowych albo pozostawienie stóp na tym samym poziomie.
- Aktualnym stanem są dane statystyczne dotyczące gospodarki GUS (niepewność środowiska – dane statystyczne zawierają większość, ale nie wszystkie istotne informacje)
- NBP w aktualnym stanie podejmuje jedną z 3 decyzji: zwiększyć stopy procentowe, zostawić je na aktualnym poziomie albo je zmniejszyć. Załóżmy, że NBP zostawia stopy procentowe na aktualnym poziomie – jednak USA znajduje złoża ropy w zachodniej Adżakarcie co wpływa na wartość dolara a więc na ocenę wyznaczonej decyzji (losowość oceny podjętej akcji)
- W następnym kwartale wartość PLN spada względem USD (następny stan jest realizacją zmiennej losowej).

Strategią nazywamy zbiór akcji podejmowanych dla każdego ze stanów – funkcja akcji od stanu. Ocena strategii w zależności od stanu  $x$  jest średnią z ocen dla całego ciągu akcji i stanów począwszy od  $x$ . Mówimy, że strategia  $s_1$  jest lepsza od  $s_2$  jeżeli dla każdego  $x$  ocena strategii  $s_1$  jest równa bądź wyższa  $s_2$  i co najmniej dla jednego  $x$  ocena  $s_1$  jest wyższa. Strategia optymalna to strategia, dla której nie da się znaleźć lepszej strategii (może istnieć wiele optymalnych strategii).

## Programowanie dynamiczne

Założeniem rozwiązania problemu przez programowanie dynamiczne jest własność optymalnych podstruktur. Tzn. że jeżeli w danym stanie  $x+1$  mamy wyznaczoną optymalną strategię (optymalny ciąg przejść stanów do stanu końcowego) to optymalną decyzją dla stanu  $x$  jest wybór akcji prowadzącej do stanu  $x+1$  i dalej postępowanie według tej strategii. Idea programowania dynamicznego opiera się na wyznaczaniu optymalnych decyzji 'od końca'. Dla wszystkich możliwych stanów końcowych  $x_n$  wyznaczane są optymalne strategie przejść z  $x_{n-1}$  do  $x_n$ . Następnie dla stanu  $x_{n-2}$  wyznaczane są wszystkie możliwe przejścia do stanów  $x_{n-1}$  i łączone ze strategią przejść z  $x_{n-1}$  do  $x_n$  wyznaczonych w poprzednim kroku, wybierane są tylko optymalne strategie. Ten proces analogicznie powtarzany jest aż do analizy strategii od stanu  $x_0$ . W ten sposób wyselekcjonowana jest optymalna strategia przejścia od  $x_0$  do  $x_n$ .

Programowanie dynamiczne gwarantuje znalezienie optymalnej strategii. Każdy krok pozwala na odrzucenie lokalnie nieoptymalnych strategii, przez co liczba możliwych kombinacji przejść jest redukowana, co w rezultacie powoduje efektywne i szybkie działanie. Jest to alternatywa dla metod zachłanych.

Programowanie dynamiczne może być stosowane jedynie kiedy występuje własność optymalnych podstruktur. W rzeczywistości wymaga to znalezienie pewnych rekurencyjnych zależności w funkcji celu - co bywa kłopotliwe.

Wiem, że nie jest to najczytelniej wytłumaczone, ale jedyny łatwy sposób wytłumaczenia to narysowanie i omówienie grafu decyzji i stanów ;)



## 27. Modele obrazów: wektory cech, niezmienniki, relacje przestrzenne

### Opracowanie Agi

**Cechą wizualną** nazywa się daną dotyczącą obrazu lub fragmentu obrazu, która dostarcza podstawową informację potrzebną do dalszego przetwarzania i stanowi kluczowy element metod analizy obrazów. Wybór cechy i metody ekstrakcji powinien być dostosowany do rozwiązywanego problemu.

Istnieje wiele różnych metod wyznaczających cechy wizualne obrazów. Można je podzielić na trzy grupy: cechy globalne, semi-lokalne i lokalne. **Cechy globalne** generują pojedyncze cechy lub wektor cech dla całego obrazu. Metody generujące **cechy semi-lokalne** dzielą obraz na segmenty, które następnie oddzielnie opisują, generując w ten sposób wektor cech lub zbiór wektorów. Segmenty mogą być wyznaczone w sposób dynamiczny lub sztywny: dynamiczny wykorzystuje jedną z metod segmentacji i przybliża granice rzeczywistych obiektów, natomiast sztywny dzieli obraz na stałe fragmenty (górze, środek dół, równe n elementów). Najbardziej złożoną pamięciowo metodą

jest ta generująca **cechy lokalne**. Takie metody najpierw wyznaczają punkty na obrazie, często wraz z pewnym otoczeniem, a następnie każdy z tych punktów opisują cechą lub wektorem cech.

Podstawowymi **niskopoziomowymi cechami** są cechy opisujące kolor, teksturę lub kształt (w przypadku segmentacji dynamicznej). Między innymi na ich podstawie powstają **cechy wysokopoziomowe**. Są to cechy, które wydobywają specjalizowaną informację, są dostosowane do konkretnej dziedziny obrazów i mają bezpośrednie przełożenie na semantykę obrazu. Przykładami takich cech są: liczba wystąpień postaci ludzkich, lokalizacja oczu, występowanie napisów, ich treść.

**Niezmienniczość** to stałość cechy przy określonych przekształceniach obrazu. Jest bardzo porządną właściwością cech. Ważne jest, aby wybór cechy był oparty o jej niezmienniczość potrzebną do konkretnego zadania. Przykładowe podstawowe przekształcenia to przesunięcie, skalowanie, obrót.

**Przykładowe cechy** - na pamięć nie ma co się uczyć, ale coś kojarzyć warto :)

**Przykładowe cechy - Kolor:**

możliwość zastosowania różnych modeli barw

- podstawowe statystyki: średnia, odchylenie standardowe:  
bardzo niska wymiarowość, szybkość;  
niezmiennicze: przesunięcie, skalowanie, obrót
- histogramy i dystrybuanty:  
zwykły histogram występowania poszczególnych kolorów, pominięcie informacji przestrzennej, można zastosować grupowanie kolorów  
niezmiennicze: przesunięcie, skalowanie, obrót
- korelogram:  
rozkład odległości pomiędzy punktami o poszczególnych kolorach, rozkład ma trzy wymiary: kolor1, kolor2 i odległość; ze względu na dużą wymiarowość powinno się stosować grupowanie kolorów  
niezmiennicze: przesunięcie, obrót, skala (jeśli mamy znormalizowaną odległość)

- autokorelogram:  
to samo tylko dla przypadku kiedy rozważamy tylko odległości punktów z tego samego koloru, mamy dwa wymiary: kolor i odległość

### Przykładowe cechy - kształt

charakteryzuje geometrię regionów zainteresowania, ma znaczenie tylko wtedy gdy regiony zainteresowania odzwierciedlają obiekty widoczne na obrazie

- moment zwykły:  
suma po współrzędnych podniesionych do potęgi  
niezmienniczość: brak
- moment centralny:  
suma  $(x-x_{sr})^p(y-y_{sr})^q$   
niezmienniczość: przesunięcie
- znormalizowany moment centralny  
niezmienniczość: przesunięcie i skala
- moment Hu:  
kombinacja znormalizowanych momentów centralnych  
niezmienniczość: przesunięcie, skala, obrót, odbicie lustrzane
- transformata Hough:  
Metoda estymuje parametry konkretnego, podstawowego elementu geometrycznego (okrąg, kwadrat, prosta). Transformata przekształca punkty znalezione na obrazie w rozkład przestrzeni paramentów. Wykrywanie elementów polega na znajdowaniu maksima w tej przestrzeni. Dla prostej przestrzeń ma dwa wymiary: odległość od środka układu współrzędnych i kąt nachylenia.  
Niezmienniczość: skala (dla znormalizowanych współrzędnych punktów)

### Przykładowe cechy tekstury

- Momenty  
analogicznie jak wyżej
- Macierze współwystępowania i cechy wyznaczone na jej podstawie (energia, entropia, kontrast, homogeniczność):  
określają lokalne, wzajemne współwystępowanie kolorów.  
Niezmienniczość: przesunięcie
- Siła i kierunek krawędzi:  
Po wykryciu krawędzi można obliczyć siłę, kierunek, ale też proste statystyki takie jak dla koloru, histogramy, korelogramy  
Niezmienniczość: skala, przesunięcie, obrót (nie dla kierunku oczywiście)

**Podsumowanie:** można cechy wybierać ręcznie, automatycznie z jakiegoś zbioru, liczba cech powinna być mała i dostosowana do rozwiązywanego problemu.

### Relacje przestrzenne

Dzięki metodom analizy obrazu można zlokalizować więcej niż jeden obiekt na obrazie. W niektórych zadaniach wzajemna lokalizacja obrazów jest istotna, np. w medycynie. **Relacje przestrzenne określają wzajemną lokalizację obiektów na obrazie.**

- **Histogram kątów**  
Zawiera informację o kątach pomiędzy wszystkimi pikselami z dwóch obiektów. Metoda liczy kąt, który tworzy prosta obliczona na podstawie każdej pary pikseli z dwóch obiektów. Kąty te trafiają do histogramu, a wzajemne położenie jest określone jako maksimum w tym histogramie

## 28. Klasyfikacja, opisywanie, interpretacja, rozumienie

## Opracowanie Agi

(to jest pytanie od Paradowskiego, przynajmniej tak wygląda, stąd definicje od niego z slajdów)  
Schemat działania systemu wizyjnego:



**Rozpoznawanie obrazu (klasyfikacja)** jest to przypisanie do obrazu, na podstawie jego cech, jednej klasy ze ściśle określonego zbioru klas (słownika).

**Opisywanie obrazu** jest to przypisanie do obrazu, na podstawie jego cech, podzbioru klas ze ściśle określonego zbioru klas (słownika).

(to ze slajdów z roku wyżej, ale ma sens)

**Interpretacja** to określenie zależności pomiędzy obiektami.

**Rozumienie** to wyniesienie wiedzy na podstawie zebranych informacji.

**Luka semantyczna** jest brakiem zgodności pomiędzy informacją, którą można wydobyć z danych wizualnych a interpretacją użytkownika tych danych w zadanej sytuacji.

## 29. Obliczenia miękkie a obliczenia tradycyjne (algorytmiczne)

### Opracowanie Agi

Obliczenia tradycyjne polegają na wnioskowaniu logicznym (dedukcji) i tradycyjnym numerycznym modelowaniu i przeszukiwaniu. Obliczenia miękkie natomiast wykorzystują podejścia bazujące na przybliżonych modelach, przybliżenia, losowe przeszukiwanie i przybliżone wnioskowanie. Metody te charakteryzują się nieprecyzyjnością i niepewnością. Modele używane w metodach z dziedziny obliczeń miękkich charakteryzują się:

- **aproksymacją**: cechy modelu są podobne do rzeczywistych, ale nie dokładnie takie same
- **niepewnością**: nie ma przekonania czy cechy modelu są takie same jak cechy obiektu
- **nieprecyzyjnością**: wartości modelu są nieprawidłowe ilościowo (??)

Jest kilka definicji obliczeń miękkich:

- każdy proces , który zawiera nieprecyzyjność w obliczeniach na jednym lub wielu poziomach i pozwala tę nieprecyzyjność zmniejszyć lub uczynić cel optymalizacji bardziej miękkim
- rozwiązywanie problemów metodami wzorowanymi na naturze
- gałąź nauki rozwiązująca problemy dla których nie ma efektywnych metod obliczeniowych

W skład metod obliczeń miękkich wchodzi metody **uczenia maszynowego**, czyli metody które na podstawie danych empirycznych potrafią uczyć się rozpoznawania i predykcji.

Przyjmując, że dane są reprezentatywne i w wystarczającej liczbie, budowane metody potrafią generalizować wiedzę.

Obliczenia miękkie różnią się od tradycyjnych tym, że tolerują nieprecyzyjność, niepewność, częściową prawdę i przybliżenia. Metody z dziedziny obliczeń miękkich wzorują się na bytach inteligentnych, które charakteryzują się:

- percepcją - łączenie informacji, przetwarzanie, interpretacja
- podejmowaniem akcji - koordynacja, sterowanie
- wnioskowaniem - dedukcja, abdukcja, indukcja, wnioskowanie przez analogie
- adaptacją - uczenie - z nauczycielem, bez nauczyciela, ze wzmocnieniem
- planowaniem i rozwiązywaniem problemów - sterowanie celem
- komunikacją
- kreatywnością
- organizacją - grupy społeczne bazujące na dzielonych celach

## 30. Jakie techniki wchodzi w skład obliczeń miękkich. Charakterystyka każdej z nich

### Opracowanie Agi

#### 1. 31. Podstawy formaln **Podejście ewolucyjnymi**

##### a. **Algorytmy genetyczne**

Najczęściej działanie algorytmu przebiega następująco: Losowana jest pewna populacja początkowa. Populacja poddawana jest ocenie (selekcja). Wybrane osobniki biorą udział w procesie reprodukcji. Genotypy wybranych osobników poddawane są operatorom ewolucyjnym: są ze sobą kojarzone poprzez złączanie genotypów rodziców (krzyżowanie), przeprowadzana jest mutacja, czyli wprowadzenie drobnych losowych zmian. Rodzi się drugie (kolejne) pokolenie. Aby utrzymać stałą liczbę osobników w populacji te najlepsze (według funkcji oceniającej fenotyp) są powielane, a najgorsze usuwane. Jeżeli nie znaleziono dostatecznie dobrego rozwiązania, algorytm powraca do kroku drugiego. W przeciwnym wypadku wybieramy najlepszego osobnika z populacji - jego genotyp to uzyskany wynik.

##### b. **Ewolucja różnicowa**

###### i. inicjalizacja populacji

- ii. mutacja każdego osobnika (czyli do każdego osobnika dodajemy różnicę dwóch innych osobników przemnożoną przez parametr F)
- iii. krzyżowanie (wybieramy część wartości z osobnika pierwotnego i część z mutantą)
- iv. selekcja wybieramy osobnik o lepszej wartości funkcji przystosowania (z osobnika pierwotnego i tego powstałego w wyniku krzyżowania)

## 2. Systemy immunologiczne

Sztuczne systemy immunologiczne to, w dużym uproszczeniu, takie systemy informatyczne, które wykorzystują podstawowe zasady funkcjonowania naturalnych systemów immunologicznych istniejących w przyrodzie. Ważną ich cechą jest to, że nie odwzorowują one całkowicie, w niemalże niezmienionej formie, swoich naturalnych pierwowzorów, a jedynie wykorzystują najważniejsze, najbardziej przydatne w rozwiązaniu konkretnego problemu właściwości. Uściślijmy kilka podstawowych pojęć związanych z systemami immunologicznymi:

- a. **Limfocyty (typu T, typu B)** – mają zdolność rozpoznawania antygenów oraz możliwość reagowania na nie. Limfocyty typu T są odpowiedzialne za podejmowanie działań i odpowiedzi na atak. Limfocyty typu B to komórki „wykonawcze”, podległe limfocytom typu T. Ich zadaniem jest produkcja przeciwciał oraz komórek pamięciowych.
- b. **Receptor** – element limfocyty, ma za zadanie rozpoznawać antygeny.
- c. **Patogen** – ciało obce, które system immunologiczny ma za zadanie wyeliminować.

Przykład algorytmu:

**in:**

S → zbiór wzorców do rozpoznania  
n → ilość najgorszych elementów do usunięcia

**out:**

M → zbiór detektorów rozpoznających nienapotkane wzorce

**begin**

Wytwórz losowo zbiór początkowy antyciał → A;

**forall** wzorców z S **do**

Określ dopasowanie z każdym antyciałem w A;

Wygeneruj klony podzbioru antyciał z największym dopasowaniem (proporcjonalnie do dopasowania);

Zmutuj atrybuty uzyskanych klonów (dodając je do A) i umieść kopie najlepszych z nich w pamięci → M;

Zastąp n najgorzej dopasowanych antyciał losowo wygenerowanymi;

**done**

**end**

Jak widać metoda jest bardzo podobna do standardowego algorytmu genetycznego i wykorzystuje tylko inne pojęcia i analogie.

## 3. Logika rozmyta

Jest to podejście proste i intuicyjne, dosyć dobrze działające, popularne i powszechnie stosowane. Logika rozmyta:

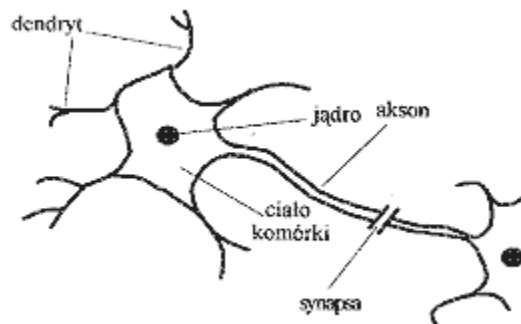
- a. bazuje na logice wielowartościowej

- b. każdy fakt może należeć z pewną miarą rozmytości do różnych zbiorów rozmytych
  - c. fakty mogą być różnie postrzegane przez różne osoby (np. wysoki, niski – odczucia subiektywne)
- łatwiejsza do lingwistycznego określenia przez eksperta (łatwiej powiedzieć „wysoka temperatura” niż podać konkretną liczbę stopni)

Schemat przetwarzania:

1. Pomiar wartości wejściowych:  $t$  – wartość liczbowa,  $w$  – wartość liczbowa
2. Rozmywanie zmiennych wejściowych
3. Rozmyte wnioskowanie – jeśli z kilku reguł otrzymujemy różne wartości na wyjściu to wybieramy tę minimalną
4. Wyostrenie zmiennych wyjściowych

#### 4. Sieci neuronowe



#### Ogólny zarys

Biologiczny neuron składa się z dendrytów, jądra, aksonu i synapsy. W trakcie pobudzenia neuronu przez dendryty przechodzą neurotransmitery z synapsy do dendrytu. Po przekroczeniu progu neuron zostaje wzbudzony i aniony/kationy zaczynają dryfować w aksonie w kierunku synapsy. Istnieją neurotransmitery, które hamują ruch.

#### I generacja sieci neuronowych

- a. model Pittsa, McCulloca - jeden neuron
- b. przybliżenia: aktywność neuronu jest procesem zero-jedynkowym, pewna liczba synaps musi być pobudzona, żeby neuron był pobudzony, jedyne opóźnienie na drodze przetwarzania to opóźnienie synaptyczne, struktura wzajemnych połączeń jest stała
- c. neuron uczy się wag na podstawie wzorców
- d. sieć nie rozwiązuje problemów nieseparowalnych liniowo

#### II generacja sieci neuronowych

- a. więcej warstw neuronów
- b. funkcja aktywacji musi być różniczkowalna
- c. druga generacja sieci dążyła do rozwiązywania problemów nieliniowych, nie dążyła do wierniejszego naśladowania swojego biologicznego pierwowzoru
- d. uczenie: korekcja wag w celu zmniejszenia błędu na wyjściu
- e. metoda propagacji wstecznej, SOM

#### III generacja sieci neuronowych

- a. dążenie do wierniejszego odwzorowania modelu biologicznego

- b. szybkie, akcje w czasie rzeczywistym
- c. za pomocą mniejszej liczby neuronów potrafią rozwiązywać skomplikowane problemy
- d. sieci impulsowe, Neurocognitron, Pulse-Coupled NN
- e. projektowanie sieci specyficznych do konkretnego problemu - Neurocognitron i model oka

## 5. Podejście probabilistyczne

- a. Naiwny Bayes

$$p(c | x) = \frac{p(c)p(x | c)}{p(x)}$$

$p(c|x)$  – prawdopodobieństwo klasy  $c$  pod warunkiem obserwacji  $x$ , nazywa się również prawdopodobieństwem a posteriori.

$p(c)$  – prawdopodobieństwo klasy  $c$ , nazywa się również prawdopodobieństwem a priori.

$p(x|c)$  – prawdopodobieństwo obserwacji  $x$  pod warunkiem klasy  $c$ , nazywa się również funkcją gęstości klasy  $c$ .

$p(x)$  – prawdopodobieństwo obserwacji  $x$ .

Powyższe prawdopodobieństwo oblicza się dla każdej z klas i wybiera spośród nich największe - będzie ono odpowiadało rozpoznanej klasie:

$$c^* = \arg \max_{c \in C} p(c | x) = \arg \max_{c \in C} \frac{p(c)p(x | c)}{p(x)} = \arg \max_{c \in C} p(c)p(x | c)$$

Prawdopodobieństwo wystąpienia klasy  $c$  (prawdopodobieństwo a priori) to stosunek liczby elementów klasy  $c$  do liczby wszystkich elementów zbiorze uczącym:

$$p(c) = \frac{|T_c|}{|T|} \quad T_c - \text{podzbiór } T \text{ zawierający elementy klasy } c$$

## 6. Algorytmy mrówkowe

Obserwacja wyszukiwania pożywienia mrówek i optymalizacja ścieżki do niego dała początek algorytmom mrówkowym. Robotnice podczas ruchu wydzielają substancję zwaną **feromonem**, którym znaczą przebytą przez siebie ścieżkę. Poruszając się mrówki zawsze wybierają drogę, na której feromonu jest najwięcej.

## 7. Zbiory przybliżone

Teoria zbiorów przybliżonych umożliwia:

- a. opis klas decyzyjnych za pomocą reguł
- b. selekcję cech (redukty)

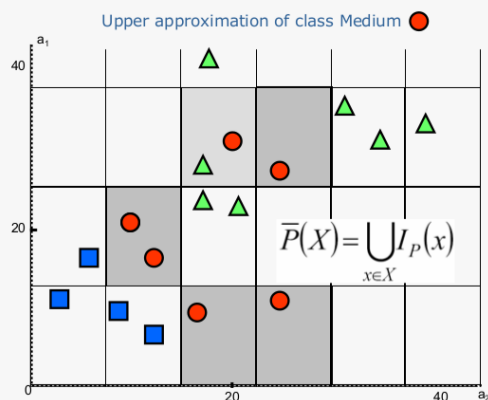
Zakłada się, że każdy obiekt jest opisany zbiorem atrybutów warunkowych i atrybutem (lub zbiorem atrybutów) decyzyjnych. Zbiór atrybutów jest oznaczony literą A, zbiór obiektów literą U.

**Klasą nierozróżnialności**  $IND(A)$  nazywany zbiór obiektów o takich samych wartościach atrybutów warunkowych. **Klasą decyzyjną** natomiast wszystkie obiekty o tej samej wartości atrybutu decyzyjnego. Jeśli  $B \subset A$  i  $X \subset U$  to **B-dolnym przybliżeniem** zbioru X jest zbiór tych obiektów dla których klasa nierozróżnialności zawiera się w X. B-górnym przybliżeniem nazywamy zbiór obiektów dla których conajmniej jeden obiekt z klasy nierozróżnialności zawiera się w X:

**BX**

(BX górne podkreślenie)

**Brzegiem** nazywamy różnicę powyższych zbiorów. Aby było łatwiej:



Zbiorem X są zwykle obiekty z jednej klasy decyzyjnej (na obrazie wyżej kółka). Klasami nierozróżnialności są przedziały w przestrzeni (np. te zaznaczone na szaro). Dolnym przybliżeniem są obiekty z tych klas dla których wszystkie obiekty z tych klas są ze zbioru X (czyli kółek) - czyli te obiekty, które są położone na ciemnoszarym obszarze. Górnym przybliżeniem są te obiekty, dla których w klasie nierozróżnialności jest chociaż jeden obiekt ze zbioru X, czyli obiekty z dolnego przybliżenia plus dwa obiekty leżące na jasnoszarym obszarze.

**Reduktem** nazywamy taki zbiór atrybutów, który nie zmienia podziału na klasy nierozróżnialności. Znajdując redukt dokonuje się selekcji cech.

### Generowanie reguł (Metoda LEM):

- wyberamy obiekty z dolnego, górnego przybliżenia lub granicy (generujemy wtedy odpowiednio reguły: pewne, możliwe i przybliżone)
- znajdujemy wszystkie pojedyncze pary (atrybut, wartość), które pokrywają wybrane obiekty
- wyberamy atrybut, który pokrywa najwięcej wybranych obiektów i jednocześnie najmniej innych obiektów
- mamy pojedynczy warunek, sprawdzamy czy pokrywa on tylko wybrane obiekty, jeśli nie to musimy wybrać kolejny warunek



- e. jeśli wybrano już warunki, które po złożeniu w koniunkcję pokrywają tylko wybrane obiekty dodajemy regułę
- f. jeśli zostały jeszcze jakieś niepokryte obiekty to idziemy do punktu a)

## 8. Algorytmy rojowe

### a. PSO

Pojedyncze elementy nazywamy cząsteczkami, które tworzą rój; początkowy rój tworzy się losowo, położenie jednej cząsteczki = jedno rozwiązanie, każda cząsteczka ma swoje położenie i prędkość i zna swoją prędkość i swoje położenie, wartość funkcji oceny dla tego położenia, najlepsze (globalnie i lokalnie) dotychczas osiągnięte położenie i przypisaną jej wartość funkcji oceny.

Jeśli lokalne położenie to  $x(t)$ , prędkość cząsteczki to  $v(t)$ , najlepsze lokalnie  $p_l(t)$  i najlepsze globalnie  $p_g(x)$ , a  $q_1(t)$  i  $q_2(t)$  to nowe położenie cząsteczki to:

$$x(t+1) = v(t) + q_1(p_l(t) - x(t)) + q_2(p_g(t) - x(t))$$

W kolejnych iteracjach znajdujemy najlepsze rozwiązania i obliczamy nowe położenia cząsteczek, aż zostanie osiągnięty warunek stopu. Problemem jest rosnąca prędkość cząsteczek, stąd wprowadza się różne modyfikacje rozwiązujące ten problem.

### b. świetlikowe

Wiele gatunków świetlikowatych przywabia partnera błyskami światła, charakterystycznymi dla danego gatunku. Badania doprowadziły do sformułowania wniosku, że owady błyskające najdłużej, odnoszą największe sukcesy godowe. Algorytm świetlikowy wykorzystuje pomysł świecenia owadów. Zasady świecenia są charakteryzowane przez trzy reguły:

- i. Świetliki są jednej płci a to oznacza, że jeden świetlik jest przyciągany przez dowolny inny niezależnie od płci.
- ii. Atrakcyjność świetlika jest proporcjonalna do jego jasności, a więc spośród dwóch owadów, mniej jasny będzie przyciągany przez świecącego intensywniej. Atrakcyjność świetlika jest proporcjonalna do jasności i obie te wielkości maleją, jeżeli ich odległość wzrasta. Jeśli nie ma jaśniejszego świetlika niż ten który rozpatrujemy, wówczas porusza się on losowo.
- iii. Jasność albo inaczej intensywność światła wynika z funkcji celu, której optimum jest poszukiwane.

Dla problemu optymalizacyjnego, jakim jest poszukiwanie maksimum, jasność może być wprost proporcjonalna do funkcji celu, ale znane są inne sposoby jej definiowania, podobnie jak to jest w algorytmach genetycznych.

### c. pszczoły

Algorytm rozpoczyna się wylosowaniem populacji  $n$  zwiadowców i w pierwszym kroku ich natychmiastową oceną. Następnie w pętli, dopóki nie zostanie osiągnięty warunek stopu, wykonywane są następujące kroki. W pierwszej kolejności wybieranych jest  $m$  miejsc do sprawdzenia jakości pożywienia ( $m$  – rozwiązań). Dla każdego z tych rozwiązań określany jest promień sąsiedztwa. Do tych rozwiązań losowo, ale w określonym promieniu sąsiedztwa rekrutowane są pszczoły z grupy  $(n-m)$ , przy czym obowiązuje zasada więcej pszczoł

do lepszych e miejsc. Dla każdego sąsiedztwa wybierana jest najlepsza pszczoła (najlepsze rozwiązanie). Utwórz następną populację pszczoł. Te kroki powtarzane są w pętli, aż do uzyskania satysfakcjonującego rozwiązania.

d. **nietoperzowe**

8. **Systemy agentowe**

- a. Rozproszone podejście do rozwiązywania problemów
- b. Wysoki poziom abstrakcji postrzegania procesów zachodzących w systemie
- c. Metoda formalnego opisu modelu systemu opartego o autonomiczne byty zwane agentami

Agent:

- a. Reprezentowany przez fizyczny lub wirtualny byt
- b. Oddziałuje na środowisko
- c. Bezpośrednio komunikuje się z innymi agentami
- d. Działa realizując wyznaczone cele
- e. Uzyskuje dostęp oraz dysponuje zasobami
- f. Odbiera sygnały z otoczenia, lecz posiada ograniczoną percepcję
- g. Posiada częściową wiedzę na temat środowiska (lub dopiero ją gromadzi)
- h. Włada zbiorem umiejętności i oferuje zakres usług
- i. W pewnych przypadkach może się rozmnażać
- j. Konkretnie cechy agenta są uzależnione od środowiska, w którym operuje.

System agentowy to system obliczeniowy, w którym luźno zgrupowane autonomiczne agenty wchodzi w interakcję z otoczeniem w celu rozwiązania zadanego problemu.

9. **Teoria chaosu**

**System chaotyczny** to taki który przy niewielkich zmianach parametrów lub warunków początkowych potrafi generować rozbieżne wyniki.

**System dynamiczny** jest to zestaw reguł określających zachowanie się układu w czasie. Przez zachowanie to rozumie się zmianę wektora liczbowego określanego mianem wektora stanu. **Wektor stanu** określa jednoznacznie zachowanie systemu, co oznacza, że znajomość wektora stanu w pewnej chwili czasu pozwala wyznaczyć wektor stanu w każdej innej chwili czasu. Zbiór wszystkich wektorów stanów nazywa się **przestrzenią stanów lub przestrzenią fazową**. Ciąg wartości wektora stanu systemu w zależności od upływu czasu nazywa się **trajektorią lub orbitą**.

Systemy chaotyczne posiadają zwykle zbiory określane mianem **atraktorów**. Zbiór A jest definiowany jako atraktor, jeżeli jest domkniętym i ograniczonym zbiorem niezmienniczym, który posiada takie otoczenie, że wszystkie trajektorie startujące z tego otoczenia dążą do A. Obszar, dla którego trajektorie z niego startujące dążą do atraktora nazywa się obszarem lub **basenem przyciągania**. W sposób analogiczny do atraktora wprowadza się pojęcie **repelera**, jako domkniętego i ograniczonego zbioru niezmienniczego, dla którego istnieje takie otoczenie, że dla każdego stanu należącego do tego otoczenia, a nie należącego do repelera, trajektoria startująca z tego punktu opuszcza je.

To co jest istotne to świadomość zachowania się niektórych systemów w ten sposób. Tworząc model powinno się mieć świadomość zachodzenia takiego zjawiska i powinno się znać zakres parametrów dla których projektowany system może zachowywać się

chaotycznie.

#### 10. Systemy hybrydowe

Systemem hybrydowym nazywamy system będący połączeniem dwóch innych systemów. Tak jak np. robiliśmy na projektach semestr temu: metody ewolucyjne znajdowały wagi do sieci neuronowej.

## 31. Podstawy formalnego opisu języka naturalnego: założenia i stosowane metody

### Opracowanie Adama P.

Dla przypomnienia gramatyką nazywany czwórkę  $\langle V, T, S, P \rangle$ , gdzie:

- $V$  - to dowolny niepusty zbiór symboli,
- $T \subseteq V$  - to zbiór symboli terminalnych,
- $V \setminus T$  - zbiór symboli nieterminalnych,
- $S$  - wyróżniony symbol nieterminalny, który jest symbolem początkowym,
- $P$  - zbiór odzworowań (produkcji, reguł)  $X \rightarrow Y$ , gdzie  $X, Y \in V^*$

Mówiąc o podstawach formalnego opisu języka warto wspomnieć o hierarchii Chomsky'ego, która jest hierarchią języków formalnych. Hierarchia ta składa się z 4 klas:

- **języki typu 3 - regularne**

Generowane są przez gramatyki regularne. W gramatyki prawostronnie regularnych lewa strona produkcji składa się z jednego symbolu nieterminalnego, a prawa strona składa się z jednego symbolu terminalnego, po którym może wystąpić jeden symbol nieterminalny. W gramatykach lewostronnie regularnych jest na odwrót. To znaczy, że symbol nieterminalny znajduje się zawsze po lewej stronie symbolu terminalnego.

Przykład:

(R1)  $S \rightarrow aP$

(R2)  $P \rightarrow aS$

(R3)  $P \rightarrow a$

Wygenerowanie słowa 'aa' polega na uruchomieniu kolejno reguł R1 i R3.

- **języki typu 2 - bezkontekstowe**

To języki generowane przez gramatyki bezkontekstowe. W gramatykach takich po stronie produkcji występuje jeden nieterminalny, natomiast po prawej stronie skończony ciąg symboli terminalnych i nieterminalnych. Prawa strona może być też słowem pustym  $\epsilon$ .

Przykład:

$A \rightarrow aAb \mid \epsilon$

Często mimo, że dany język należy do języków niższego poziomu, łatwiej jest go opisać za pomocą gramatyk bezkontekstowych. Są one po prostu czytelniejsze dla człowieka. Trzeba jednak pamiętać, że gramatyki regularne są bardziej efektywne obliczeniowo niż gramatyki bezkontekstowe.

- **języki typu 1 - kontekstowe**

Jest to uogólnienie gramatyk bezkontekstowych, pozwalające na wstawienie po lewej stronie produkcji więcej niż jednego symbolu.

Przykład takiej gramatyki:

$S \rightarrow abc \mid aSXc$

$cX \rightarrow Xc$

$bX \rightarrow bb$

- **języki typu 0 - rekurencyjnie przeliczalne**

Są to języki opisywane przez gramatyki kombinatoryczne. W takich gramatykach nie ma żadnych ograniczeń na postać produkcji. Za prof. Agnieszką Mykowiecką powtarzam, że istnieją gramatyki, które nie należą do żadnej z wymienionych klas, jednak takie gramatyki nie zostały wymienione.

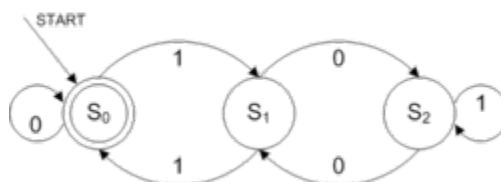
Mimo tak określonej klasyfikacji języków trudno jest odpowiedzieć na pytanie jakiego typu są języki formalne. Chomsky podał dowód, że język angielski nie jest językiem regularnym. Obecny stan wiedzy o języku angielskim pozwala na postawienie twierdzeń:

- składnia i morfologia angielskiego jest bezkontekstowa,
- szwajcarski dialekt niemieckiego ma gramatykę kontekstową,
- morfologia języka Bambara (Mali) ma gramatykę kontekstową.

Ważniejsze od stwierdzenia jakiego typu są języki naturalne wydaje się znalezienie takiej gramatyki, która w łatwy sposób pozwoli użyć jej przez człowieka.

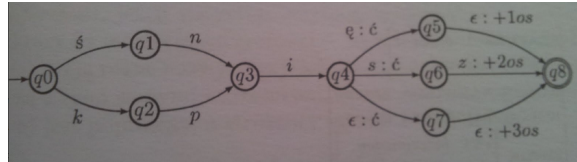
**Automaty skończone** - jest to urządzenie sekwencyjne mogące znajdować się w skończonej liczbie stanów, czytające symbol po symbolu dane z taśmy wejściowej. Przeczytanie symbolu zmienia stan urządzenia (na inny lub ten sam) i przesuwają taśmę wejściową o jeden symbol.

Przykład automatu skończonego:



Przy czym strzałka start zaznacza miejsce, w którym rozpoczyna się działanie automatu, a stan zaznaczony dwoma okrągami oznacza dopuszczalny stan końcowy. Oznacza to, że słowo zostanie zaakceptowane, kiedy automat po odczytaniu całego ciągu symboli zakończy swoją pracę w którymś ze stanów końcowych.

**Transduktory** - to automaty skończone, które każdemu symbolowi z wejścia dokonują pewnego przyporządkowania. Transduktor na zdjęciu poniżej dokonuje przypisania form: 'śnieć' z 'śnieć+1os', 'śniesz' z 'śnieć+2os', 'śni' z 'śnieć+3os' oraz 'kpię' z 'kpić+1os', 'kpisz' z 'kpić+2os' i 'kpi' z 'kpić+3os'



## 32. Współczesna technologia językowa: narzędzia, zasoby językowe i ich zastosowania

### Opracowanie Adama P.

#### Narzędzia

Podstawowe:

1. Segmenter (np. Toki dla j. pol)
2. Analizator morfologiczny (np. Maca dla j. pol)
3. Tager (ujednoznacznianie morfosyntaktyczne np. WCRFT dla j. pol)
4. Chunker (rozpoznawanie fraz np. lobber dla j. pol)
5. Parser zależnościowy (rozpoznawanie zależności pomiędzy wyrazami)
6. Program do maszynowego tłumaczenia
7. Program do rozpoznawania mowy

Inne, które znam:

1. Rozpoznawacz nazw własnych (np. Liner2)
2. Rozpoznawacz relacji pomiędzy chunkami
3. Rozpoznawacz anafory
4. Ujednoznacznianie sensu słów (ujednoznacznianie znaczeń leksykalnych).

#### Zasoby językowe:

1. Anotowane dane językowe:
  - Korpus (anotowany morfologicznie, składniowo, sensami słów, jednostkami nazewniczymi – w zasadzie może być anotowany wszystkim, co próbuje rozpoznawać IJN)  
np. NKJP - anotowany morfosyntaktycznie, składniowo, sensami słów, jednostkami nazewniczymi
  - Korpusy równoległe – używane do uczenia programów maszynowego tłumaczenia.
2. Bazy wiedzy:
  - Gramatyka formalna opisująca podzbiór języka
  - Wordnet
  - Model językowy określający prawdopodobieństwo wystąpienia ciągów wyrazów (np. n-gramowy)

Cele ogólne:

- wydobywanie wiedzy z tekstu,
- wyszukiwanie informacji,
- systemy automatycznego odpowiadania na pytania,
- rozumienie języka naturalnego.

Cele szczegółowe są dobrze opisane w następnym punkcie.

### 33. Proces przetwarzania języka naturalnego: typowe etapy, cele, stosowane metody

Opracowanie Adama P.

Typowe etapy:

1. **Wyodrębnianie tekstu z dokumentu**

Celem tego etapu jest wydobywanie tekstu z np. zeskanowanego dokumentu. Czasem zachowuje się informacje o tym, w którym miejscu znajdował się tekst. Jednym z podstawowych zapamiętywanych informacji jest podział na paragrafy.

2. **Segmentacja (podział na zdania)**

Celem tego etapu jest podzielenie tekstu na zdania. Jest trudny ze względu na problem haploglogi kropki (sytuacja, w której kropka oznacza jednocześnie elementem skrótu oraz oznacza koniec zdania). Metodą dla języka polskiego to użycie odpowiednich reguł. Reguły takie mówią przykładowo, że jeżeli po kropce znajduje się słowo, które jest z napisane z małych liter, to jest to przesłanka do tego, że kropka jest częścią skrótu. W przypadku wielkiej litery jest to przesłanka za tym, że kropka oznacza koniec zdania.

3. **Tokenizacja**

Celem tego etapu jest podzielenie zdań na tokeny (w przybliżeniu słowa). Najczęściej wyróżniane klasy segmentów to:

- ciąg małych liter poprzedzonych wielką literą, np. Kraków,
- ciąg składający się tylko z wielkich liter, np. PZU,
- ciąg małych liter, np. dom,
- ciąg liter małych i wielkich, np. PeKaO,
- ciąg cyfr,
- ciąg cyfr z wewnętrzną kropką lub przecinkiem,
- znak interpunkcyjny

Najczęściej do rozpoznawania granic tokenów używana jest technika automatów skończonych.

4. **Analiza morfologiczna**

Celem tego etapu jest „znajdowanie możliwych opisów morfoskładniowych dla słów, czyli ich interpretacja jako form wyrazowych”.

Metody wspierające ten etap to stemming oraz lematyzacja.

Na ogół polega na sprawdzeniu, czy analizowany token występuje w tablicy odmian (np. Morfeusz) i określenie czy jest formą jakiegoś z lematów znajdujących się w słowniku i jakie zostały im przypisane tagi (dla j. polskiego tagi składniowe).

5. **Ujednoznacznianie morfosyntaktyczne (desambiguacja morfosyntaktyczna)**

Jest to prosto rzecz ujmując wybranie jednego (lub kilku, jeżeli inaczej się nie da) prawidłowych interpretacji danego tokenu. Oznacza to wybranie lematu oraz tagów określających cechy morfosyntaktyczne (klasę gramatyczną (rzeczownik) i kategorie gramatyczne (rodzaj, liczba itp.)) danego tokenu. Zadanie to nazywa się zwykle tagowaniem. Bardzo często, gdy mówi się o tagowaniu ma się na myśli tak naprawdę etapy 2-5.

6. **Ujednoznacznianie sensu słów (WSD)**

Celem tego zadania jest rozstrzygnięcie problemów homonimi oraz polisemi (inaczej wieloznaczności wyrazów). Polega na przypisaniu konkretnej jednostki leksykalnej do danego tokenu. Jednostka leksykalna w Słownosieci jest reprezentowana jako lemat oraz numer znaczenia. Przykładowo dla słowa zamek w Słownosieci jest 7 znaczeń, a zadanie WSD sprowadza się do wybrania, czy w zdaniu autor mówi o zamku błyskawicznym, czy zamku – budowli. Jedną z ostatnio chwalonych metod przez dr. Piaseckiego jest budowanie sieci semantycznej dla całego tekstu. Następnie na podstawie analizy relacji pomiędzy wyrazami w tej sieci oraz Słownosieci wybranie jednostki leksykalnej, która odpowiada użytemu słowu w tekście.

## 7. Płytki parsing (opcjonalnie)

Problem ten polega na częściowym rozpoznaniu składniowej struktury zdania. Głównie stosowane są dwa podejścia: parsing zależnościowy oraz parsing frazowy.

**Zależnościowy** polega na rozpoznaniu zależności między wyrazami. Polega na wybraniu korzenia zdania oraz rozpoznaniu hierarchicznych relacji pomiędzy wyrazami. Jego głównymi cechami są konieczność rozpoznania relacji pomiędzy wszystkimi tokenami oraz konieczność przypisania relacji do tylko jednego wyrazu.

**Frazowy** polega na rozpoznawaniu fraz, składających się z wielu tokenów, takich jak: rzeczownikowa, czasownikowa, przymiotnikowa lub rzadziej spotykane przymiennikowa lub uzgodnione rzeczownikowe.

Są też dwa podejścia rozwiązania tego problemu: ręczne pisanie reguł (np. Spejd) lub wykorzystanie maszynowego uczenia (np. lobber).

## 8. Rozpoznawanie wyrażen wielowyrazowych

(np. nazwy własne, byty nazwane)

Celem tego etapu jest oznaczenie tokenów, które wspólnie tworzą wyrażenia wielowyrazowe.

Bardzo często wykorzystywaną metodą jest metoda słownikowa. Trzeba pamiętać, że w języku polskim musi ona być wspierana lematyzacją.

## 9. Rozpoznawanie związków w tekście

(np. anafory, koreferencji, relacji semantycznych, sytuacji)

Polega na rozpoznawaniu relacji pomiędzy tokenami (w przypadku anafory) oraz frazami (w przypadku rozpoznawania relacji typu: orzeczenie – podmiot, orzeczenie – dopełnienie)

W przypadku rozpoznawania relacji między frazami chodzi o to by zbliżyć się bardziej do parsingu głębokiego, w którym są zaznaczone wszystkie relacje. Trzeba jednak zwrócić uwagę, że w parsingu głębokim mamy do czynienia z pełnym rozbiorem zdania, natomiast w przypadku płytkiego parsingu mamy do czynienia z frazami.

## 10. Głęboki parsing

Polega na znalezieniu pełnego rozbioru składniowego zdania. Dla przykładowego zdania:

“Czekam na kobietę w kapeluszu”

Zadaniem jest rozpoznać, że albo autor zdania może czekać w kapeluszu, albo kobieta, na którą czeka może być w kapeluszu. Następnie w zależności od użytego kontekstu wybrać, który rozbiór zdania jest prawidłowy. Jest to problem bardzo czasochłonny i złożony, przez co w rozwiązaniach inżynierskich stosuje się raczej płytki parsing. W celu rozwiązania tego problemu

można stosować reguły pisane ręcznie, gramatyki bezkontekstowe atrybutowe, gramatyki bezkontekstowe, gramatyki HPSG.

Do tych dwóch ostatnich etapów raczej nikt nie dojdzie mam nadzieję po tak długim opracowaniu poprzednich etapów.

### **11. Analiza semantyczna**

Można wyróżnić dwa podejścia do tego zadania:

- Rozszerzenie algorytmu parsowania, by dla całkowicie rozpoznanych składników tworzyła się interpretacja semantyczna
- Przypisanie interpretacji po zbudowaniu całego drzewa rozbioru

W praktyce stosuje się uproszczone podejście do semantyki - role semantyczne.

Przykładowo w zdaniu:

„Tomek dokładnie opowiedział przebieg ślubu Marty i Marka Sławkowi”

możemy rozpoznać zdarzenie opowiadania, w którym autorem informacji był Tomek, odbiorcą Sławek, a treścią było zdanie relacji z innego zdarzenia, którym był ślub Marty i Marka.

### **12. Analiza pragmatyczna (w tym struktury dyskursu)**

Celem analizy pragmatycznej jest więc „praktyczne” określenie znaczenia wypowiedzi w zależności od kontekstu dyskursu (wypowiedzi).