Pyt 2: Omówić semantyczną i syntaktyczną konsekwencję.

Semantyczna konsekwencja

α jest *semantyczną konsekwencją* zbioru formuł $\Phi \Leftrightarrow$ dowolny model zbioru Φ jest modelem formuły α, co zapisujemy

$$\Phi \models \alpha$$

Dla $\Phi = 0$ α jest zawsze prawdziwa i zwana *tautologią*

$$\models \alpha$$

α jest *równoważna* $\beta \Leftrightarrow \alpha \models \beta$ i $\beta \models \alpha$
α jest *modelowo równoważna* $\beta \Leftrightarrow \alpha \models \beta$

Pyt3: Omówić metody reprezentacji wiedzy (w podpunktach, tabelach, grafach itp.)

Metody reprezentacji wiedzy

- Zastosowania logiki (rachunek zdań, rachunek predykatów, syntaktyka, semantyka)
- Zapis twierdzeń, zapis reguł w systemach ekspertowych (schemat rezolucji na klauzulach Horna, wnioskowanie w przód i wstecz)
- Wiedza nieprecyzyjna (teoria Bayesa, współczynniki niepewności w systemie MYCIN, teoria Dempstera-Shafera)
- Teoria zbiorów przybliżonych (tablice warunkowo-działaniowe, relacje nierozróżnialności, klasyfikacje, aproksymacja dolna i górna, reguły pewne i możliwe)
- Teoria zbiorów rozmytych (funkcja przynależności, liczby rozmyte, relacje rozmyte)
- Sieci semantyczne
- Algorytmy genetyczne i sieci neuronowe

Pyt 4: Omówić zasadę wnioskowania w przód i wstecz. Do czego służy wnioskowanie w teorii zdefiniowanej m.in. poprzez aksjomaty i reguły?

Wnioskowanie wstecz

```
backward_chaining(g)
begin
    if g ∈ F then
      return TRUE;
    Q := φ;
    for each r ∈ R
        if match(r, g, s) then
          Q := Q ∪ {(r, s)};
    if Q = φ then return FALSE;
    repeat
        (r, s) := select(Q);
        Q := Q − {(r, s)};
        satisfied:=TRUE;
        for each c ∈ Cond(r)
            if not backward_chaining(c) then
              begin
                  satisfied:=FALSE; break;
              end
        if satisfied then
          begin
              F := F ∪ {g};
              return TRUE;
          end
    until Q = φ;
end
```

Wnioskowanie w przód

```
forward_chaining
begin
    Q := φ;
    repeat
        for each r ∈ R
            if match(r, F, s) then
              Q := Q ∪ {(r, s)};
        if Q = φ then
          break;
        (r, s) := select(Q);
        Q := Q − {(r, s)};
        F := F ∪ {fact(r, s)};
    until FALSE
end
```

Ogólnie o regułach wnioskowania:
Przystępując do dowodu **twierdzenia** T (a właściwie **zdania**, które będziemy mogli nazwać twierdzeniem, gdy zakończymy **dowód**) dysponujemy pewnym zbiorem zdań P_1, P_2,..., P_n uznanych za prawdziwe (mogą to być **aksjomaty** lub wcześniej udowodnione twierdzenia). Z prawdziwości zdań P_1, P_2,..., P_n chcemy wywnioskować prawdziwość zdania T. Zdania P_1, P_2,..., P_n nazywamy **przesłankami**, zaś zdanie T nazywamy **wnioskiem**.Dowody twierdzeń polegają więc na tym, by z faktu, że przesłanki są prawdziwe, tzn.

$$\mathbf{w(P_1) = w(P_2) = ... = w(P_n) = 1}$$

wywnioskować prawdziwość wniosku:

$$\mathbf{w(T) = 1}$$

Jeżeli przyjmiemy, że P_1, P_2,..., P_n, T mogą być zmiennymi zdaniowymi lub formułami

rachunku zdań, to możemy zbudować schematy zwane regułami wnioskowania.Reguły wnioskowania zapisujemy w postaci:

$$\mathbf{P_1, P_2,..., P_n|T}$$

Powyższy zapis rozumiemy następująco: "*Jeżeli prawdziwe są przesłanki $P_1, P_2,..., P_n$, to można wnioskować, że wniosek T jest prawdziwy.*" .

Wybrane rodzaje reguł wnioskowania:

1) Reguła odrywania oparta na prawie **"modus ponens"** zgodnie z którym jeśli uznany (prawdziwy) jest okres warunkowy (**implikacja***) i jego poprzednik, wolno zawsze uznać (za prawdziwy) jego następnik. "**Modus ponens**" polega na **wnioskowaniu w przód**, tzn. z przyczyny wnioskujemy o skutkach.

$$\mathbf{P, P => Q|Q}$$

Jako fakty przyjęte zostały przesłanki P=>Q oraz P, dlatego zostały one umieszczone ponad kreską. Na podstawie tych faktów i reguły **modus ponens** wywnioskowana została konkluzja (**wniosek**) Q.

Przykład z życia studentki:
a)
przesłanka 1: Studentka otrzymała w teście 20 punktów.
przesłanka 2: Jeżeli studentka otrzymała 20 punktów to studentka zaliczyła przedmiot.
wniosek: Studentka zaliczyła przedmiot (:)).

2) Reguła oparta na prawie "modus tollens" polegającego także na **wnioskowaniu wprzód.**

$$\mathbf{P => Q, \sim Q| \sim P}$$

Jako fakty przyjęto P=>Q oraz ~Q. Wnioskując zgodnie z regułą **modus tollens** otrzymano konkluzję (wniosek) ~P

Przykłady z życia studentki:
a)
przesłanka 1: Jeżeli studentka otrzymała w teście ponad 20 p unktów to studentka zaliczyła przedmiot.
przesłanka 2: Studentka nie zaliczyła przedmiotu (:( ).
wniosek: Studentka nie otrzymała na teście ponad 20 punktów (:( ).
b)
przesłanka1: Jeżeli rodzice studentki wysłali na jej konto 500 zł to studentka kupi sobie skórzane buty.
przesłanka 2: Studentka nie kupiła sobie skórzanych butów (:( ).
wniosek: Rodzice studentki nie wysłali na jej konto 500 zł (:( ) .

Inne reguły wnioskowania:
1) dylemat konstrukcyjny

P=>Q, ~P=>Q| Q

2) dylemat destrukcyjny

P=>Q, P=>~Q| ~P

3) prawo kompozycji dla koniunkcji

P=>Q, R=>S| (P && R) =>(Q && S)

4) prawo kompozycji dla alternatywy

P=>Q, R=>S| (P || R)=>(Q || S)

5) implikacja prosta

P=>Q

6) implikacja odwrotna

Q=>P

7) implikacja przeciwna

~P=>~Q

8) implikacja przeciwstawna

~Q=>~P

Pyt 5: Omówić rodzaje systemów ekspertowych i kryteria ich klasyfikacji (w podpunktach, tabelach, grafach itp.)

Określeniem "system ekspertowy" [2] można nazwać dowolny program komputerowy, który na podstawie szczegółowej wiedzy, tylko w jej granicach, może wyciągać wnioski, podejmować decyzję, działać w sposób zbliżony do procesu rozumowania człowieka. Konkretniej, systemy ekspertowe są programami komputerowymi przeznaczonymi do rozwiązywania specjalistycznych problemów wymagających profesjonalnej ekspertyzy.
Spotykane są następujące polskie i angielskie synonimy:

system ekspertowy = program regułowy = program z regułową bazą wiedzy
expert system = knowledge based system = rule based system

Z pojęciem "system ekspertowy" związane są nieodłącznie osoby inżyniera wiedzy i eksperta.
Inżynier wiedzy - zajmuje się:

- pozyskiwaniem i strukturalizacją wiedzy ekspertów,
- dopasowywaniem i wyborem odpowiednich metod wnioskowania i wyjaśniania rozwiązywanych problemów,
- projektowaniem odpowiednich układów pośredniczących między komputerem a użytkownikiem.

Ekspert - osoba posiadająca odpowiednią wiedzę i kompetencje do rozwiązywania problemów w danej dziedzinie.

Rodzaje systemów ekspertowych

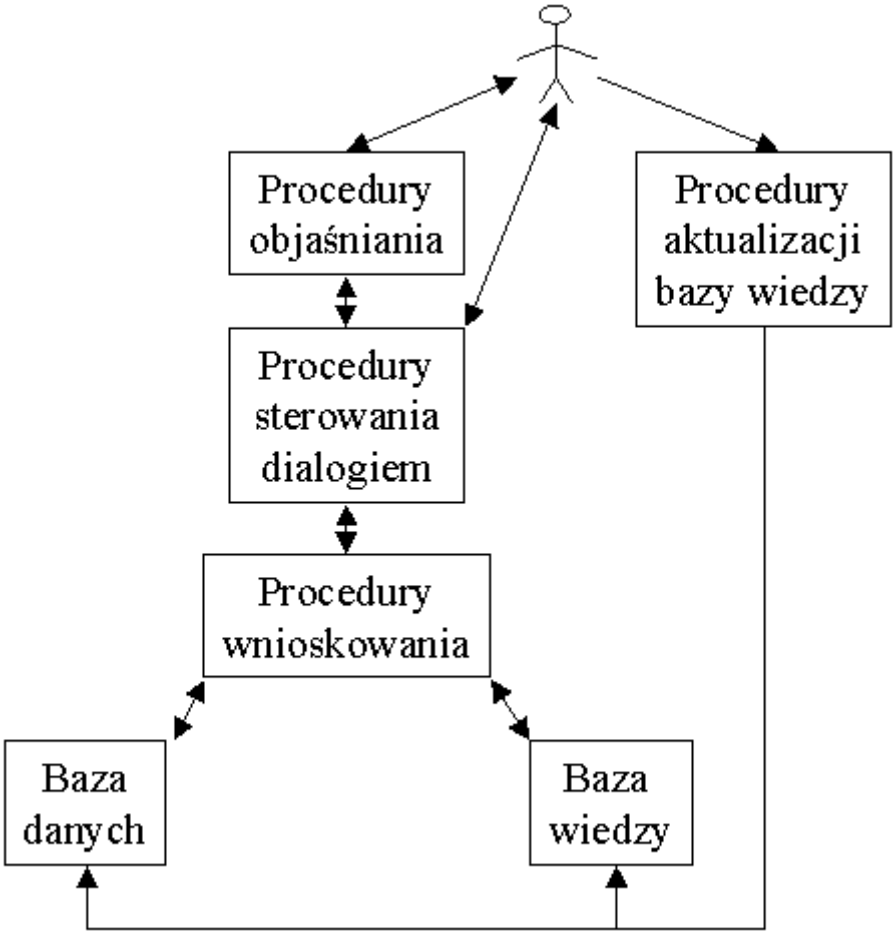
Systemy ekspertowe możemy podzielić na kilka sposobów m.in. ze względu na prezentację rozwiązania lub strategię ich tworzenia.
Pierwszy podział pozwala wyodrębnić następujące grupy:

- doradcze z kontrolą człowieka - prezentujące rozwiązania dla użytkownika, który jest w stanie ocenić ich jakość, zatwierdzić, lub zażądać innej propozycji,
- doradcze bez kontroli człowieka - system jest sam dla siebie końcowym autorytetem. Rozwiązanie takie jest wykorzystane m.in. w układzie sterowania promem kosmicznym. Układ 5 komputerów przygotowuje się do podjęcia decyzji. Następnie porównuje otrzymane wyniki i przy pełnej zgodności wykonuje odpowiednie działanie, w przeciwnym przypadku cały proces jest powtarzany,
- krytykujące - dokonujące analizy i komentujące uzyskane rozwiązanie.

Natomiast drugi:

- dedykowane - tworzone od podstaw przez inżyniera wiedzy współpracującego z informatykiem,
- szkieletowe - gotowy system z pustą bazą wiedzy, do wypełnienia przez inżyniera wiedzy i eksperta z danej dziedziny.

Elementy składowe systemów ekspertowych



Rysunek 1.1: Elementy składowe systemu ekspertowego

Zasadnicze elementy składowe i relacje pomiędzy nimi dla typowego systemu ekspertowego pokazane są na rysunku 1.1, gdzie:

- baza wiedzy - zbiór reguł; pewne uogólnienie informacji o grupie obiektów, w najprostszym przypadku:
  IF warunek THEN wniosek
- baza danych - dane o obiektach, szczegółowe informacje o dostępnych rozwiązaniach,
- procedury wnioskowania - maszyna wnioskująca, algorytm sterowania procesem dochodzenia do odpowiedzi przez system,
- procedury objaśniania - opis otrzymanego rozwiązania i ewentualnie sposobu jego osiągnięcia,
- procedury sterowania dialogiem - formatowanie danych WE/WY systemu, dialog z użytkownikiem.

Istnieje jeszcze inny podział systemu ekspertowego na składowe, proponowany przez Antoniego Niederlińskiego [3], w którym baza wiedzy zawiera reguły, procedury objaśniania jako bazę rad i pliki rad, oraz tzw. bazę ograniczeń dbającą o nie powstawanie sprzeczności podczas wnioskowania.

Tabela 8. Rodzaje systemów ekspertowych w zależności od realizowanych przez te systemy zadań

Kategoria	**Zadania zrealizowane przez systemy ekspertowe**
INTERPPRETACYJNE	dedukują opisy sytuacji z obserwacji lub stanu czujników, np.

	rozpoznawanie mowy, obrazów, struktur danych.
PREDYKCYJNE	wnioskują o przyszłości na podstawie danej sytuacji, np. prognoza pogody, rozwój choroby.
DIAGNOSTYCZNE	określają wady systemu na podstawie obserwacji, np. medycyna, elektronika, mechanika.
KOMPLETOWANIA	konfigurują obiekty w warunkach ograniczeń, np. konfigurowanie systemu komputerowego.
PLANOWANIA	podejmują działania, aby osiągnąć cel, np. ruchy robota.
MONITOROWANIA	porównują obserwacje z ograniczeniami, np. w elektrowniach atomowych, medycynie, w ruchu ulicznym.
STEROWANIA	kierują zachowaniem systemu; obejmują interpretowanie, predykcję, naprawę i monitorowanie zachowania się obiektu.
POPRAWIANIA	podają sposób postępowania w przypadku złego funkcjonowania obiektu, którego te systemy dotyczą.
NAPRAWY	harmonogramują czynności przy dokonywaniu napraw uszkodzonych obiektów.
INSTRUOWANIA	systemy doskonalenia zawodowego dla studentów.

Pyt 6: Jak można zdefiniować sztuczną inteligencję i w jakim celu rozwijane są badania w tej dziedzinie

Sztuczna inteligencja (ang. *Artificial Intelligence – AI*) to technologia i kierunek badań na styku informatyki, neurologii i psychologii. Jego zadaniem jest konstruowanie maszyn i oprogramowania zdolnego rozwiązywać problemy nie poddające się algorytmizacji w sposób efektywny, w oparciu o modelowanie wiedzy (inaczej: zajmuje się konstruowaniem maszyn, które robią to, co obecnie ludzie robią lepiej). Problemy takie bywają nazywane AI-trudnymi i zalicza się do nich między innymi analiza (i synteza) języka naturalnego, rozumowanie logiczne, dowodzenie twierdzeń, gry logiczne (szachy, warcaby, go) i manipulacja wiedzą - systemy doradcze, diagnostyczne.

Istnieją dwa różne podejścia do pracy nad AI. Pierwsze to tworzenie całościowych modeli matematycznych analizowanych problemów i implementowanie ich w formie programów komputerowych, mających realizować konkretne cele. Drugie to próby tworzenia struktur i programów "samouczących się", takich jak modele sieci neuronowych oraz opracowywania procedur rozwiązywania problemów poprzez "uczenie" takich programów, a następnie uzyskiwanie od nich odpowiedzi na "pytania".

W trakcie wieloletniej pracy laboratoriów AI stosujących oba podejścia do problemu, okazało się, że postęp w tej dziedzinie jest i będzie bardzo trudny i powolny. Często mimo niepowodzeń w osiąganiu zaplanowanych celów, laboratoria te wypracowywały nowe techniki informatyczne, które okazywały się użyteczne do zupełnie innych celów.

Przykładami takich technik są np. języki programowania LISP i Prolog. Laboratoria AI stały się też "rozsadnikiem" kultury hakerskiej, szczególnie laboratorium AI w MIT.

Praca ta przyniosła też konkretne rezultaty, które znalazły już praktyczne i powszechne zastosowania.

[Edytuj]

Współczesne praktyczne zastosowania AI

- Technologie oparte na logice rozmytej – powszechnie stosowane do np. sterowania przebiegiem procesów technologicznych w fabrykach w warunkach "braku wszystkich danych".
- Systemy ekspertowe – czyli rozbudowane bazy danych z wszczepioną "sztuczną inteligencją" umożliwiającą zadawanie im pytań w języku naturalnym i uzyskiwanie w tym samym języku odpowiedzi. Systemy takie stosowane są już w farmacji i medycynie.
- maszynowe tłumaczenie tekstów – systemy takie jak SYSTRANS, jakkolwiek wciąż bardzo ułomne, robią szybkie postępy i zaczynają się nadawać do tłumaczenia tekstów technicznych.
- Sieci neuronowe – stosowane z powodzeniem w wielu zastosowaniach łącznie z programowaniem "inteligentnych przeciwników" w grach komputerowych
- Eksploracja danych - omawia obszary, powiązanie z potrzebami informacyjnymi, pozyskiwaniem wiedzy, stosowane techniki analizy, oczekiwane rezultaty
- Rozpoznawanie optyczne – stosowane są już programy rozpoznające osoby na podstawie zdjęcia twarzy lub rozpoznające automatycznie zadane obiekty na zdjęciach satelitarnych.
- Rozpoznawanie mowy (identyfikacja treści wypowiedzi) i rozpoznawanie mówców (identyfikacja osób) – stosowane już powszechnie na skalę komercyjną
- Rozpoznawanie ręcznego pisma – stosowane już masowo np. do automatycznego sortowania listów, oraz w elektronicznych notatnikach.
- Sztuczna twórczość – istnieją programy automatycznie generujące krótkie formy poetyckie, komponujące, aranżujące i interpretujące utwory muzyczne, które są w stanie skutecznie "zmylić" nawet profesjonalnych artystów, w sensie, że nie rozpoznają oni tych utworów jako sztucznie wygenerowanych.
- W ekonomii, powszechnie stosuje się systemy automatycznie oceniające m.in. zdolność kredytową, profil najlepszych klientów, czy planujące kampanie medialne. Systemy te poddawane są wcześniej automatycznemu uczeniu na podstawie posiadanych danych (np. klientów banku, którzy regularnie spłacali kredyt i klientów, którzy mieli z tym problemy).

[Edytuj]

Czego nie udało się dotąd osiągnąć mimo wielu wysiłków

- *Programów skutecznie wygrywających w niektórych grach.* Jak dotąd nie ma programów skutecznie wygrywających w go, brydża sportowego, i polskie warcaby,

mimo że podejmowano próby ich pisania. Trzeba jednak przyznać, że programy do gry w szachy, w które zainwestowano jak dotąd najwięcej wysiłku i czasu spośród wszystkich tego rodzaju programów, osiągnęły bardzo wysoki poziom, ogrywając nawet obecnego mistrza świata.

- *Programu, który skutecznie potrafiłby naśladować ludzką konwersację.* Są programy "udające" konwersowanie, ale niemal każdy człowiek po kilku-kilkunastu minutach takiej konwersacji jest w stanie zorientować się, że rozmawia z maszyną, a nie człowiekiem. Najsłynniejszym tego rodzaju programem jest ELIZA, a obecnie najskuteczniejszym w teście Turinga jest cały czas rozwijany program-projekt ALICE.
- *Programu, który potrafiłby skutecznie generować zysk, grając na giełdzie.* Problemem jest masa informacji, którą taki program musiałby przetworzyć i sposób jej kodowania przy wprowadzaniu do komputera. Mimo wielu prób podejmowanych w tym kierunku (zarówno w Polsce jak i na całym świecie), z użyciem sztucznej inteligencji nie da się nawet odpowiedzieć na pytanie, czy jest możliwe zarabianie na giełdzie (bez podawania samego przepisu jak to zrobić). Prawdziwym problemem w tym przypadku może być fakt, że nie istnieje żadna zależność między danymi historycznymi, a przyszłymi cenami na giełdzie (taką tezę stawia Hipoteza Rynku Efektywnego). Gdyby hipoteza ta była prawdziwa, wtedy nawet najlepiej przetworzone dane wejściowe nie byłyby w stanie wygenerować skutecznych i powtarzalnych zysków.
- *Programu skutecznie tłumaczącego teksty literackie i mowę potoczną.* Istnieją programy do automatycznego tłumaczenia, ale sprawdzają się one tylko w bardzo ograniczonym stopniu. Podstawową trudnością jest tu złożoność i niejasność języków naturalnych, a w szczególności brak zrozumienia przez program znaczenia tekstu.

Pyt 7: Omówić metody reprezentacji wiedzy (w podpunktach, tabelach, grafach itp.)

Prosty podział metod Reprezentacji Wiedzy:
- oparte na pomysłach i koncepcjach wymyślonych przez człowieka
- oparte na rozwiązaniach wytworzonych przez „matkę naturę” w drodze ewolucji

Kilka metod reprezentacji wiedzy:
- język naturalny
- metody stosowane w obszarze baz danych
- logika matematyczna (klasyczna, niestandardowa)
- reguły produkcji (production rules)
- sieci semantyczne (semantic networks)
- grafy koncepcji (concept graphs)
- ontologie
- ramy, scenariusze (frames, scripts)
- zbiory przybliżone (rough sets)
- XML
- Sieci neuronowe (neural nets)
- Algorytmy genetyczne (genetic algorithms)
- ...

Pyt 8: Omówić budowę i zasadę konstruowania systemów ekspertowych

System ekspertowy (SE) – program komputerowy, przeznaczony do rozwiązywania specjalistycznych problemów, które wymagają profesjonalnej ekspertyzy na poziomie trudności pokonywanych przez ludzkiego eksperta.

Dowolny program komputerowy może być systemem ekspertowym o ile na podstawie szczegółowej wiedzy „potrafi" wyciągać wnioski i używać ich podejmując decyzję, tak jak człowiek. Bardzo często zdarza się jednak, iż taki system, pracujący w czasie rzeczywistym, pełni swoją rolę lepiej niż człowiek (ekspert). Główną przewagą systemu ekspertowego nad człowiekiem jest szybkość oraz brak zmęczenia.

Systemy ekspertowe, ze względu na zastosowanie, możemy podzielić na trzy ogólne kategorie:

- systemy doradcze (advisory systems),
- systemy krytykujące (criticizing systems),
- systemy podejmujące decyzje bez kontroli człowieka.

Pierwszy rodzaj - systemy doradcze, zajmują się doradzaniem, tj. wynikiem ich działania jest metoda rozwiązania jakiegoś problemu. Jeżeli rozwiązanie to nie odpowiada użytkownikowi, może on zażądać przedstawienia przez system innego rozwiązania, aż do wyczerpania się możliwych rozwiązań.

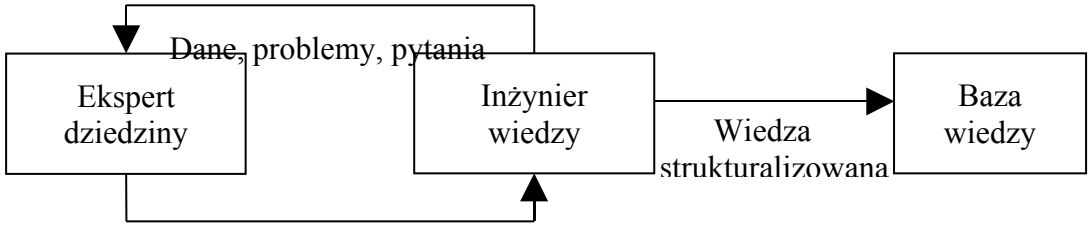
Odwrotnym działaniem do systemów doradczych charakteryzują się systemy krytykujące. Ich zadaniem jest ocena rozwiązania (danego problemu) podanego przez użytkownika systemowi. System dokonuje analizy tego rozwiązania i przedstawia wyniki w postaci opinii.

Ostatnim rodzajem systemów ekspertowych są systemy podejmujące decyzje bez kontroli człowieka. Działają one niezależnie. Pracują najczęściej tam gdzie udział człowieka byłby niemożliwy, same dla siebie są autorytetem.

Najszersze i najliczniejsze zastosowanie wśród systemów ekspertowych mają systemy doradcze. Budowane dziś systemy doradcze wykorzystują różne metody reprezentacji wiedzy: reguły, ramy, sieci semantyczne, rachunek predykatów, scenariusze. Najbardziej powszechną metodą jest reprezentacja wiedzy w formie reguł i przeważnie wielkość systemu określa liczba wpisanych reguł. Przyjęto, że system, który posiada

poniżej 1000 reguł nazywany jest zazwyczaj małym lub średnim systemem ekspertowym, zaś powyżej - systemem dużym.

Aby zbudować inteligentny program będący systemem ekspertowym należy go wyposażyć w dużą ilość prawdziwej i dokładnej wiedzy z dziedziny, jaką będzie zajmował się dany system. Ogólnie mówiąc wiedza jest informacją, która umożliwia ekspertowi podjęcie decyzji. Zasadniczym celem przy realizacji systemu ekspertowego jest pozyskanie wiedzy od ekspertów, jej strukturalizacja i przetwarzanie. Proces pozyskiwania wiedzy obrazuje poniższy schemat.

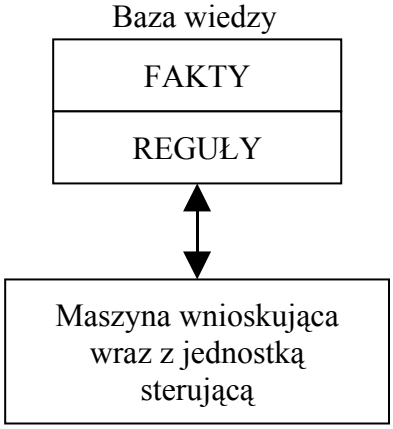


Rys 2. Proces pozyskiwania wiedzy.

Jak widać na schemacie, wiedza jest pobierana przez inżyniera wiedzy od eksperta z danej dziedziny, w razie niejasności inżynier zwraca się z pomocą do eksperta. Następnie jest strukturalizowana do Bazy wiedzy, skąd może być przetwarzana.

Następnym krokiem przy realizacji systemu ekspertowego jest dopasowanie i wybór odpowiednich metod wnioskowania i wyjaśniania rozwiązywanych problemów. Na zakończenie należy jeszcze zaprojektować odpowiednio przyjazny i naturalny interfejs między użytkownikiem a maszyną.

Systemy ekspertowe nazywane są inaczej systemami z baza wiedzy, bowiem w systemach takich baza wiedzy odseparowana jest od reszty. Oprócz bazy wiedzy na system składa się również mechanizm wnioskowania zwany maszyną wnioskującą. Podstawowe bloki systemu ekspertowego obrazuje poniższy schemat.



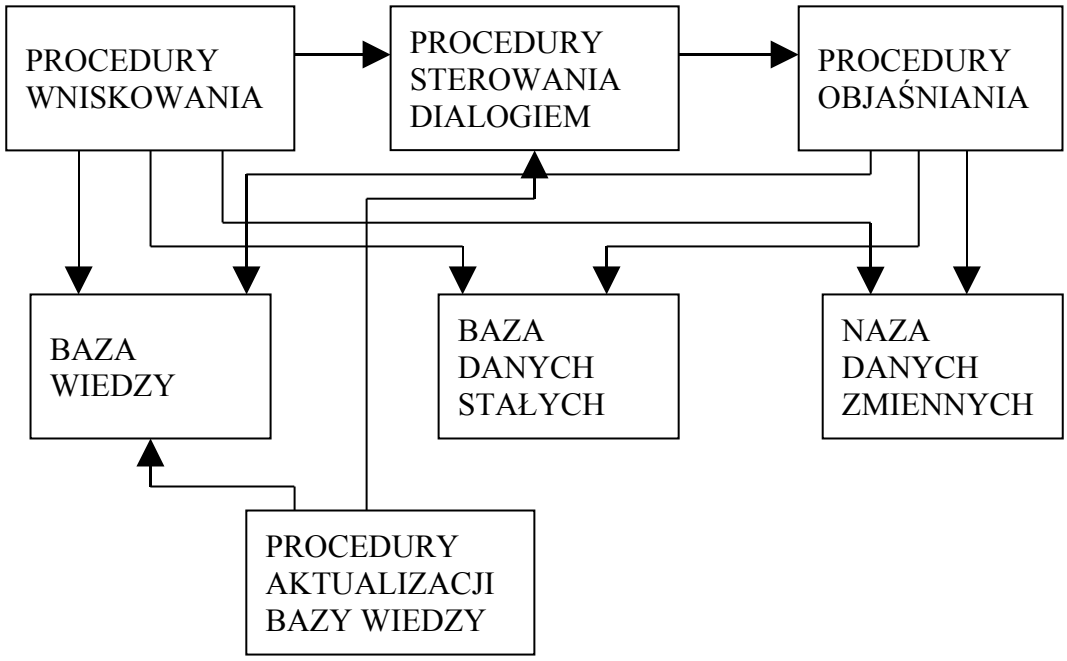
Rys 3. Podstawowe bloki systemu ekspertowego.

Baza wiedzy są to reguły opisujące relacje między faktami, opisują one jak system ma się w danym momencie działania zachować. Maszyna wnioskująca zaś, dopasowuje fakty do przesłanek i uaktywnia reguły.

Zagłębiając się bardziej szczegółowo w strukturę należy zaznaczyć, iż system ekspertowy posiadać musi takie elementy jak:

- bazę wiedzy,
- bazę danych stałych (raz zapisane nie zmieniają się),
- bazę danych zmiennych (zmieniają się w czasie działania systemu),
- maszynę wnioskującą (czyli procedury wnioskowania),
- elementy objaśniające strategię (procedury objaśniania),
- interfejs z użytkownikiem (procedury wejścia/wyjścia do formułowania zapytań przez użytkownika maszynie oraz procedury umożliwiające pobranie wyników od systemów),
- procedury aktualizacji bazy wiedzy.

Łącząc te elementy można przedstawić system jako bardziej skomplikowaną strukturę ukazaną na schemacie poniżej.



Rys 4. Struktura systemu ekspertowego.

4. Rodzaje problemów rozwiązywanych przez systemy ekspertowe.

System ekspertowy mają szerokie zastosowanie w niemal każdej dziedzinie. Oto wybrane problemy i zagadnienia, którymi zajmują się systemy ekspertowe:

- nadzór sieci telefonicznej na podstawie raportów o uszkodzeniach i zgłoszeń abonenckich (ACE),
- systemy diagnozy medycznej (CASNET)
- wyznaczanie relacji przyczynowo – skutkowej w diagnostyce medycznej (ABEL),
- systemy interpretacyjne dla nadzoru,
- rozpoznawania mowy,
- interpretacji sygnałów (np. z czujników alarmowych),
- interpretowanie postaci elektrokardiogramów (CAA),
- identyfikacja struktur cząstek białka (CRYSTALIS),
- diagnostyka maszyn cyfrowych (DART),

- prognozowanie pogody,
- interpretacja wyników spektrografii masowej (DENDRAL),
- interpretacja wyników badań geologicznych przy poszukiwaniu ropy naftowej (DIPMETER ADVISOR),
- diagnostyka chorób,
- diagnostyka komputerów (FAULTFINDER, IDT),
- interpretacja wyników pomiarów dla potrzeb chemii (GAL),
- identyfikacja związków chemicznych metodą emisyjną (GAMMA),
- wspomaganie badań geologicznych (LITHO),
- konfiguracja systemu komputerowego,
- diagnostyka chorób nowotworowych (ONCOCIN),
- analiza rynku,
- planowanie projektu np. w handlu,
- poszukiwanie złóż minerałów (PROSPECTOR),
- diagnostyka siłowni jądrowych (REACTOR),
- nadzorują i planują czynności przy dokonywaniu napraw uszkodzonych obiektów,
- pełnią rolę nauczania (np. przy szkoleniu studentów),
- planowanie eksperymentów genetycznych (MOLGEN, GENESIS, SPEX),
- nadzorowanie eksploatacji sprzętu do wiercenia szybów naftowych,
- kompletowanie sprzętu komputerowego (CONAD, R1, XCON),
- diagnostyka lokomotyw spalinowych (DELTA),
- kształcenie lekarzy (Gwidon),
- szkolenie operatorów siłowni jądrowych (STEAMER),
- analiza obwodów cyfrowych (CRITTER),
- analiza układów elektrycznych (EL),
- analityczne rozwiązanie zadań w zakresie algebry i równań różniczkowych (MAKSYMA),

- planowanie ruchów robota,
- monitorowanie (np. w elektrowniach, medycynie)
- sterowanie układami mechanicznymi i elektronicznymi,
- modelowanie układów mechanicznych (SACON),
- prowadzenie dialogu z maszyną cyfrową w języku naturalnym (INTELLECT),
- projektowanie komputerów,
- doradztwo (np. dla rolnictwa).

Pyt 9: Podać definicję języka logiki I rzędu tzw. Językiem predykantów np. omówić termy, reguły, predykanty. Wyjaśnić dlaczego $\exists x\ x \Rightarrow y$ jest sformułowane nieprawidłowo

Język logiki I rzędu: syntaktyka

Wyrażenia języka składają się z term i formuł.

1. termy - encje, obiekty
   - symbole stałych (zwykle z początku alfabetu): $a, b, \ldots$
   - symbole zmiennych
   - n-argumentowe symbole funkcyjne $f(t_1, \ldots, t_n)$, gdzie $t_1, \ldots, t_n$ to termy

   np. $a, f(g(x, b), c)$ to termy zamknięty (bez zmiennych) oraz otwarty (ze zmiennymi), termy mogą być z indeksami: a_1, f_3^n, gdzie n to ilość argumentów funkcji

2. formuły - fakty zachodzące w świecie
   - formuły atomowe - symbole relacji 0-argumentowych zwanych stałymi zdaniowymi oraz relacje n-argumentowe oznaczane P tzn. $P(t_1, \ldots, t_n)$, gdzie t_n to termy dla $n \geqslant 1$
   - formuły - z formuł atomowych, $\neg$, $\Rightarrow$, $\forall$

     i) α, β

     ii) $(\neg\alpha)$

     iii) $(\alpha \Rightarrow \beta)$

     iv) $(\forall x\ \alpha)$, gdzie x jest zmienną

Relacja n-argumentowa oznaczana literą P jest zwana predykatem np. predykat P związany jest z pojęciem jakiejś konkretnej rzeczy tzn. $P(a)$ np. jest symbolem rzeczy osoby oznaczonej termem a np: Joanny.
Literałem pozytywnym jest α, a negatywnym $\neg\alpha$.
Korzystając z podanych definicji tworzenia formuł rozszerza się zbiór spójników i kwantyfikatorów języka poprzez następujące definicje:

- $(\alpha \vee \beta) = ((\neg\alpha) \Rightarrow \beta)$
- $(\alpha \wedge \beta) = (\neg((\neg\alpha) \vee (\neg\beta)))$
- $(\exists x\ \alpha) = (\neg(\forall x(\neg\alpha)))$

Symbol $\exists$ jest kwantyfikatorem szczegółowym (egzystencjalnym), $\alpha \vee \beta$ - alternatywą formuł, $\alpha \wedge \beta$ - koniunkcją formuł.

Formuła zamknięta zwana także zdaniem lub formułą zdaniową jest formułą bez zmiennych wolnych (zmiennych nie związanych z kwantyfikatorem $\forall$ lub $\exists$) w przeciwieństwie do formuły otwartej np. $P(x,y) \Rightarrow \exists x \forall z P(x,y)$ jest formułą otwartą.
W celu poprawienia czytelności można pomijać także nawiasy kierując się następującą listą - od najmocniej do najsłabiej wiążących - spójników i kwantyfikatorów: $\neg$ $\forall$ $\exists$ $\wedge$ $\vee$ $\Rightarrow$

Pyt 10: Wyjaśnić pojęcie selektora.

Reguly: selektory

Atrybuty symboliczne:

- ♢ Selektor równościowy $X = v$
- ♢ Selektor wykluczający $X \neq v$
- ♢ Selektor ogólny $X \in \{v_1, \ldots, v_k\}$

Atrybuty numeryczne:

- ♢ Selektor przedziałowy $X \in (a, b)$

Przedział może być jednostronnie nieograniczony,
może też być jedno- lub obustronnie domknięty

Pyt 11: Eyjaśnić pojęcie kompleksu.

Kompleks jest zwięzłym opisem hipotezy, czyli w tym znaczeniu proponowanego przez algorytm klastra. Możemy wyobrazić sobie klaster stworzony przez dwa przykłady: („biały", „okrągły", „wysoki") i („czarny", „okrągły", „wysoki"). Jedną z możliwych hipotez, czyli opisów takiego klastra, jest to, że zawiera on wszystkie przykłady „okrągłe" i „wysokie", natomiast w zależności od tego czy atrybut kolor ma dwie, czy więcej wartości, może być on nieistotny (wtedy hipoteza o nim nie wspomni), albo mający dwie możliwe wartości – wtedy klaster opisany jest jako „biały" lub „czarny", „okrągły" i „wysoki".

Kompleks nadaje takim hipotezom formę zwięzłą i dającą się przetwarzać komputerowo. Każdy z atrybutów ma odpowiadający ***selektor***, opisujący możliwe jego wartości. Istnieją cztery rodzaje warunków nakładanych na atrybut:

selektor pojedynczy: spełniony, jeśli odpowiadający atrybut ma wartość wskazaną przez selektor;

selektor dysjunkcyjny: określający zbiór możliwych wartości danego atrybutu;

selektor uniwersalny: dopuszczający wszystkie wartości danego atrybutu;

selektor pusty: nie dopuszczający żadnej wartości;

Reprezentacja poprzez kompleksy jest szczególnie skuteczna w zastosowaniach, w których występują atrybuty o stosunkowo małym zbiorze wartości, gdyż zbiory możliwych wartości w selektorze dysjunkcyjnym zazwyczaj określa się poprzez wyliczenie wszystkich elementów. Jest to reprezentacja chętnie używana w grupowaniu pojęciowym, części maszynowego uczenia się.

23

Pyt 12: Co to oznacz, że jeden kompleks jest bardziej szczegółowy od drugiego kompleksu?

Pyt 13: Wyjaśnić zjawisko pokrywania przykładów ze zbioru treningowego przez kompleks i podać jakim symbolem jest oznaczane?

k - komplex

- $S \rhd k$ to dokładniej $(\exists k \in S) k \rhd x$ - zbiór wszystkich x pokrywanych przez $k \in S$

Algorytmy indukcji reguł przeszukują przestrzeń hipotez w poszukiwaniu takich reguł dla każdej kategorii, które pokrywają możliwie wiele przykładów należących do tej kategorii (w przypadku dokładnych reguł wszystkie) i możliwie mało przykładów należących do innych kategorii (w przypadku dokładnych reguł żadnego). Algorytmy te są na ogół realizacjami ogólnego schematu sekwencyjnego pokrywania, przedstawionego w maksymalnie uproszczonej postaci w tablicy 1 dla zbioru trenującego T_c.

Tablica: Ogólny schemat algorytmu sekwencyjnego pokrywania.

Powtarzaj jak długo w T są przykłady nie pokryte przez pokrycie żadnej reguły

1. ***znajdź kompleks p, który pokrywa pewną liczbę przykładów z T z dużą dokładnością;***
2. ***dodaj do zbioru reguł regułę***

IF p THEN d,

gdzie $d \in C$ jest większościową kategorią w zbiorze przykładów pokrywanych przez kompleks p.

Różnice pomiędzy poszczególnymi realizacjami schematu sekwencyjnego pokrywania polegają przede wszystkim na sposobie znajdowania kompleksu dodawanego do pokrycia reguły i kryteriach, które musi on spełniać. Przedstawimy dwa algorytmy indukcji reguł oparte na tym schemacie: AQ Michalskiego i innych (wykorzystywany w serii systemów AQ n, ostatnio chyba AQ18?) i CN2 Clarka i Nibletta. Ściśle rzecz biorąc, nie będzie to dokładna prezentacja oryginalnych algorytmów, lecz raczej pewna rekonstrukcja, pomijająca niektóre szczegóły i drobne różnice w stosunku do ogólnego algorytmu skwencyjnego pokrywania, zaś uwypuklająca to, czym te dwa pokrewne algorytmy się różnią od siebie w sposób istotny.

Pyt 14: Wyjaśnić pojęcie entropii. Zilustrować wzorem stosowanym na przykład przy konstrukcji drzew decyzyjnych .

Z kolei **entropię** zbioru przykładów P **ze względu na wartość r** atrybutu t :

$$E_{tr}(P) = \sum_{d \in C} -\frac{\left|P_{tr}^{d}\right|}{\left|P_{tr}\right|} \log \frac{\left|P_{tr}^{d}\right|}{\left|P_{tr}\right|}$$

Natomiast **entropia** zbioru przykładów P **ze względu na atrybut t** :

$$E_{t}(P) = \sum_{r \in R_t} \frac{\left|P_{tr}\right|}{\left|P\right|} E_{tr}(P)$$

Kolejne kroki konstrukcji drzewa

1. Pierwsze wywołanie funkcji:
   buduj-drzewo(T,1,{*aura,temperatura,wilgotność,wiatr*}).
2. Kryterium stopu dla zbioru $P = T$ nie jest spełnione.
3. Tworzony jest nowy węzeł, dla którego na podstawie obliczonych wcześniej współczynników przyrostu informacji wybierany jest test tożsamościowy atrybutu *aura* o największym współczynniku.
4. Większościową etykietą w zbiorze P jest 1 i dalej jest przekazywana jako etykieta.
5. Następuje wywołanie rekurencyjne dla wyniku *słoneczna* testu *aura*:
   - *buduj-drzewo*(P,1,{*temperatura,wilgotność,wiatr*}), gdzie $P = \{1, 2, 8, 9, 11\}$ i nie jest spełnione kryterium stopu.
   - Tworzony jest nowy węzeł dla którego wybierany jest test o najmniejszej entropii (w przypadku wątpliwości o największym współczynniku przyrostu informacji) tzn.: atrybut *wilgotność*:

$$E_{temp,zimna}(P) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$$

$$E_{temp,ciepła}(P) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{0}{2}\log_2\frac{0}{2} = 0$$

$$E_{wilg,duża}(P) = -\frac{0}{3}\log_2\frac{0}{3} - \frac{3}{3}\log_2\frac{3}{3} = 0$$

$$E_{wiatr,silny}(P) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$E_{temp,umiarkowana}(P) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$E_{wilg,normalna}(P) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2} = 0$$

$$E_{wiatr,słaby}(P) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0,918$$

$$E_{temp}(T) = 0,4;\ E_{wilgotność}(T) = 0;\ E_{wiatr}(T) = 0,951$$

Kolejne kroki konstrukcji drzewa c.d

5. Ciąg dalszy rekurencyjnego wykonania głównej funkcji dla wyniku *słoneczna* testu *aura* czyli punktu 5:
   - Większościową etykietą kategorii w zbiorze P jest 0 i będzie ona przekazana dalej.
   - Dla wyniku *normalna* testu *wilgotność* następuje wykonanie rekurencyjne:
     buduj-drzewo(P,0,{*temperatura,wiatr*}), gdzie $P = \{9, 11\}$ i jest spełnione kryterium stopu, gdyż zbiór P ma jedną etykietę 1. Jest tworzony liść z etykietą 1 i zwracany jako wynik funkcji.
   - Dla wyniku *duża* testu *wilgotność* następuje wykonanie rekurencyjne:
     buduj-drzewo(P,0,{*temperatura,wiatr*}), gdzie $P = \{1, 2, 8\}$ i jest spełnione kryterium stopu, gdyż zbiór P ma jedną etykietę 0. Jest tworzony liść z etykietą 0 i zwracany jako wynik funkcji.
   - Zwracany jest jako wynik węzeł z testem *wilgotność*.
6. Następuje wywołanie rekurencyjne dla wyniku *pochmurna* testu *aura* dla $P = \{3, 7, 12, 13\}$ w wyniku czego powstaje liść z etykietą 1.
7. Następuje wywołanie rekurencyjne dla wyniku *deszczowa* testu *aura* dla $P = \{4, 5, 6, 10, 14\}$ w wyniku czego powstaje węzeł w testem *wiatr*, a następnie po dwóch rekurencyjnych wywołaniach powstają liście z etykietą 1 dla wyniku *słaby* przy czym $P = \{4, 5, 10\}$ oraz z etykietą 0 dla wyniku *silny* przy czym $P = \{6, 14\}$.

Skonstruowane drzewo decyzyjne

aura=słoneczna:		$P = \{1, 2, 8, 9, 11\}$
	wilgotność=normalna:= 1	*dla* $P = \{9, 11\}$
	wilgotność=duża:= 0	*dla* $P = \{1, 2, 8\}$
aura=pochmurna: 1		*dla* $P = \{3, 7, 12, 13\}$
aura=deszczowa:		$P = \{4, 5, 6, 10, 14\}$
	wiatr=słaby:= 1	*dla* $P = \{4, 5, 10\}$
	wiatr=silny:= 0	*dla* $P = \{6, 14\}$

Pyt 15: Opisać ogólnie algorytm zstępującego konstruowania drzewa decyzyjnego:

Zstępujące konstruowanie drzewa

funkcja *buduj-drzewo*(P, d, S)
argumenty wejściowe:

- P - zbiór przykładów etykietowanych pojęcia c,
- d - domyślna etykieta kategorii,
- S - zbiór możliwych testów;

zwraca: drzewo decyzyjne jako hipotezę przybliżającą c na zbiorze P;

```
jeśli kryterium-stopu(P, S) to
        utwórz liść l;
        d_l := kategoria(P, d);
        zwróć l;
    koniec jeśli
utwórz węzeł n;
t_n := wybierz-test(P, S);
d := kategoria(P, d);
dla wszystkich r ∈ R_{t_n} wykonaj
        n[r] := buduj-drzewo(P_{t_n,r}, d, S − {t_n});
    koniec dla
zwróć n
```

Pyt 16: Opisać wybór testu dla największego przyrostu informacji dla algorytmu zstępującego konstruowania drzewa decyzyjnego:

Wybór testu dla największego przyrostu informacji

Wybór testu tworzącego węzeł lub liść zależy od przyrostu informacji $v_t(P)$ dla danego zbioru P i atrybutu t. *Informację* zawartą w zbiorze etykietowanych przykładów P można wyrazić następująco:

$$I(P) = \sum_{d \in C} -\frac{|P^d|}{|P|} log \frac{|P^d|}{|P|}$$

Z kolei *entropię* zbioru przykładów P ze względu na wynik r testu t określa się jako:

$$E_{tr}(P) = \sum_{d \in C} -\frac{|P^d_{tr}|}{|P_{tr}|} log \frac{|P^d_{tr}|}{|P_{tr}|}$$

$$E_t(P) = \sum_{r \in R_t} \frac{|P_{tr}|}{|P|} E_{tr}(P)$$

Przyrost informacji wynikający z zastosowania testu t do zbioru przykładów etykietowanych P jest określony jako różnica:

$$g_t(P) = I(P) - E_t(P)$$

Jeśli *przyrost informacji* podzielimy przez *wartość informacyjną* $IV_t(P)$ testu t dla zbioru przykładów P, to otrzymamy *współczynnik przyrostu informacji* zdefiniowany jako:

$$v_t(P) = \frac{g_t(P)}{IV_t(P)}, \text{ gdzie } IV_t(P) = \sum_{r \in R_t} -\frac{|P_{tr}|}{|P|} log \frac{|P_{tr}|}{|P|}$$

Pyt 17: Opisać ogólnie kryterium stopu i wyboru kategorii dla algorytmu zstępującego konstruowania drzewa decyzyjnego:

Sprawdza czy wywołanie rekurencyjne algorytmu tworzącego drzewo należy zakończyć. Określa ono czy dany węzeł drzewa powinien być traktowany jako końcowy liść drzewa (decyzję). Musi on zwrócić wartość etykiety w dwóch przypadkach:

a) Kiedy w trakcie wywołań rekurencyjnych w zestawie przykładów znajdują się już tylko przykłady opisujące tylko jedną klasę-decyzji.
b) Kiedy zestaw atrybutów argumentów osiągnie zero wtedy kryterium stopu powinno zgłosić jedną z możliwości:
   - Błąd, gdyż na podstawie przykładów nie można jednoznacznie ustalić odpowiednią klasę-odpowiedź.
   - Zwrócić etykietę klasy-decyzji, która najliczniej występuje w zestawie przykładów.

Kryterium stopu i wyboru kategorii

Kryterium stopu przyjmuje następującą postać:

$$P = \phi \vee S = \phi \vee |\{d' \in C | (\exists x \in P) \quad c(x) = d'\}| = 1$$

Operacja *wyboru kategorii* liścia natomiast taką:

$$kategoria(P, d) == \begin{cases} d & \text{jeśli } P = \phi, \\ \text{argmax}_{d'} |P^{d'}| & \text{w przeciwnym przypadku} \end{cases}$$

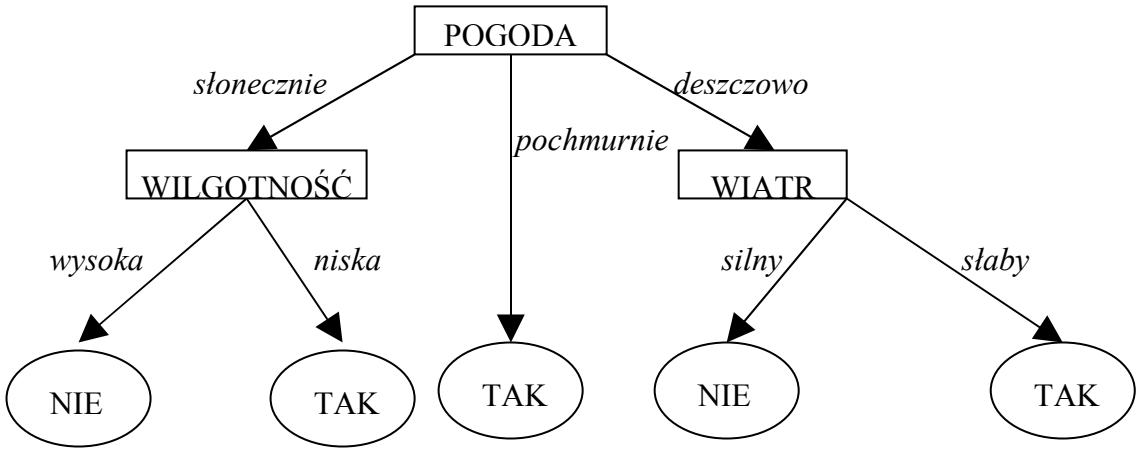
Pyt 18: W jakim celu konstruuje się drzewa decyzyjne?:

Sposób zapisywania wiedzy służącej do podejmowania decyzji przy pomocy drzew, jest bardzo starty i nie wywodzi się ani z systemów ekspertowych ani z sztucznej inteligencji. Dzisiaj jednak drzewa decyzyjne stanową podstawową metodę indukcyjnego uczenia się maszyn, spowodowane jest to dużą efektywnością, możliwością prostej programowej implementacji, jak i intuicyjną oczywistość dla człowieka. Ta metoda pozyskiwania wiedzy opiera się na analizie przykładów, przy czym każdy przykład musi być opisany przez zestaw atrybutów, gdzie każdy atrybut może przyjmować różne wartości. Wartości te powinny być dyskretne, w przypadku ciągłości dokonuje się zwykle dyskretyzacji na podstawie kilku przedziałów.

Dopuszcza się możliwość, iż ciąg przykładów może zawierać błędy, jak również może zawierać atrybuty nieposiadające określonej wartości.

Formalnie drzewem decyzyjnym jest graf-drzewo, którego korzeń jest tworzony przez wybrany atrybut, natomiast poszczególne gałęzie reprezentują wartości tego atrybutu. Węzły drzewa w następnych poziomach będą przyporządkowane do kolejnym atrybutom, natomiast na najniższym poziomie otrzymujemy węzły charakteryzujące poszczególne klasy-decyzje.

Przykładowe drzewo decyzyjne może wyglądać tak:



Czasami ze względów czysto technicznych odchodzi się od realizacji algorytmów rekurencyjnych szczególnie, gdy operujemy na dużej ilości przykładów i atrybutów - wtedy każde wywołanie procedury pociąga za sobą duże ilości danych magazynowanych na stosie programowym. Dlatego w takich przypadkach korzysta się z tak zwanego „jawnego stosu" bądź też z wykorzystuje się metodę konstruowania drzewa strategią „wszerz" jednak działanie samego algorytmu jest w praktyce identyczne.

Pyt 19: Wyjaśnić pojęcie przecięcia dwóch zbiorów kompleksów:

Przecięcie zbioru kompleksów A i B: $\{p \cap q | p \in A, q \in B\}$

Pyt20: Opisać tworzenie gwiazdy częściowej w algorytmie sekwencyjnego pokrywania AQ.

Indukcja reguł - częściowa gwiazda

funkcja *częściowa-gwiazda*(x_s, x_n)
argumenty wejściowe:

- x_s – ziarno-pozytywne,
- x_n – ziarno-negatywne

zwraca: zbiór maksymalnie ogólnych kompleksów pokrywających x_s i nie pokrywających x_n

$S' := 0$

dla wszystkich atrybutów a_i określonych na dziedzinie wykonaj

$k :=< ? >$; - kompleks
$V := A_i - \{a_i(x_n)\}$;
jeśli $a_i(x_s) \in V$ to
umieść selektor s_V w k na pozycji i;
$S' := S' \cup \{k\}$;
koniec jeśli
koniec dla
zwróć S'

Indukcja reguł - algorytm AQ

funkcja *znajdź-kompleks-aq*(T, P)
argumenty wejściowe:

- T - zbiór trenujący dla pojęcia c,
- P - podzbiór zbioru T zawierający przykłady nie pokryte przez wygenerowane wcześniej reguły

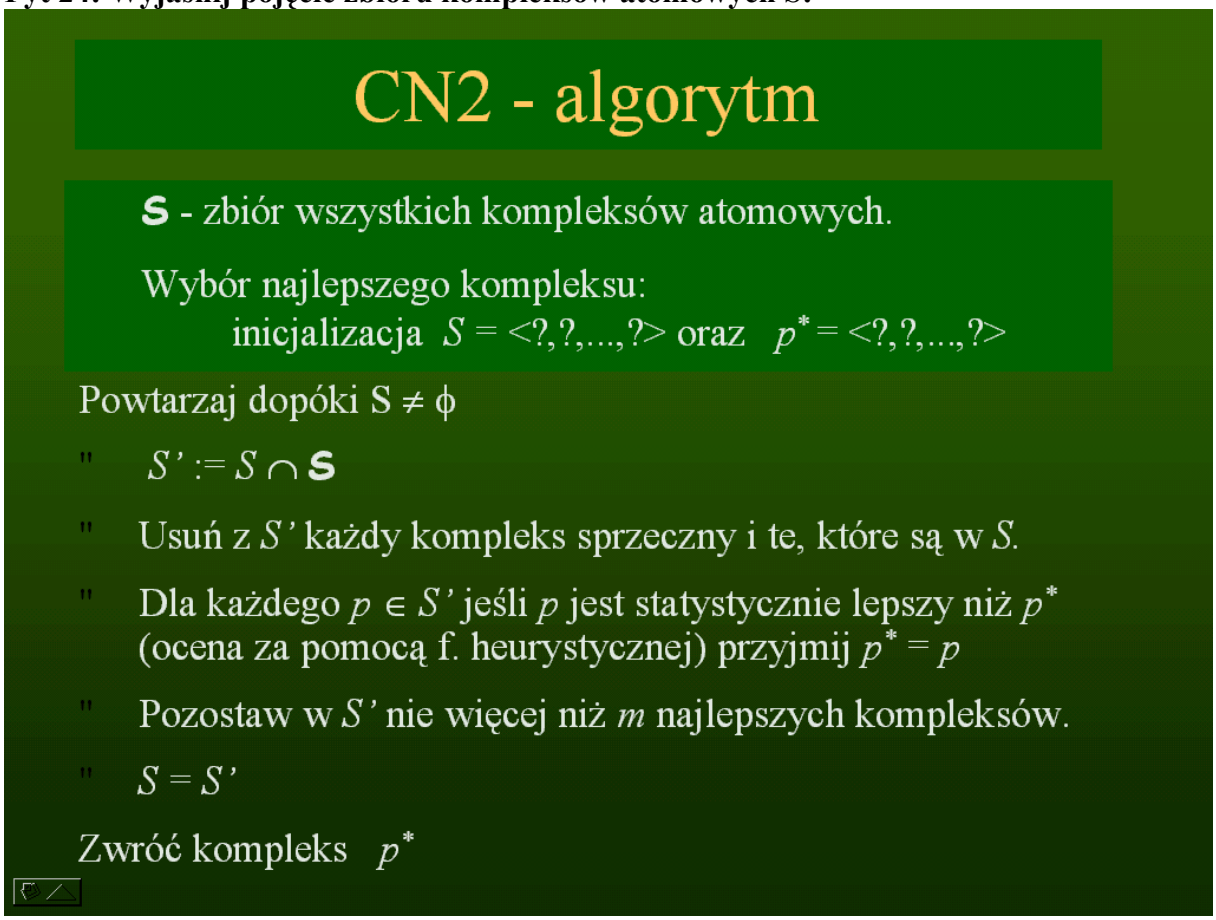
zwraca: kompleks pokrywający pewną liczbę przykładów z P należących do jednej kategorii;

$x_s :=$ ziarno – pozytywne(P);
$S := \{< ? >\}$;
jak długo $(\exists x \in T) S \rhd x \wedge c(x) \neq c(x_s)$ wykonaj
$x_n :=$ ziarno-negatywne(T, S, x_s);
$S' :=$ częściowa-gwiazda(x_s, x_n);
jeśli $S' = 0$ to zwróć $< 0 >$;
koniec jeśli
$S := S \cap S'$
$S := S - \{k \in S | (\exists k' \in S) k \prec k'\}$
$S :=$ Arg $\max^m_{k \in S} v_k(x_s, T, P)$
koniec jak długo
zwróć $\arg\max_{k \in S} v_k(x_s, T, P)$

Pyt21: Wyjaśnij pojęcie reguły zdaniowej (skonstruowanej z jednego kompleksu)

Pyt 22: Wyjaśnij pojęcie reguły asocjacyjnej (skonstruowanej z dwóch kompleksów)

Pyt 24: Wyjaśnij pojęcie zbioru kompleksów atomowych S:



Pyt25: Opisać ogólnie algorytm sekwencyjnego pokrywania AQ

ALGORYTM AQ

Algorytm oparty na pokrywaniu sekwencyjnym, według schematu:

- wybieramy obiekt niepasujący do żadnej dotychczas znalezionej reguły,
- znajdujemy najlepszą regułę, do której pasuje,
- itd., aż pokryjemy wszystkie reguły.

Kryterium oceny reguły decyzyjnej jest jej wysokie wsparcie (liczba pasujących obiektów) przy założonym progu dokładności reguły (np. nie więcej, niż 1% kontrprzykładów).

W przypadku, gdy algorytm daje kilka alternatywnych (zbliżonych pod względem jakości) ścieżek – stosujemy przeszukiwanie wiązkowe.

ALGORYTM AQ

Czasem wygodnie jest przedstawiać lewe strony reguł w postaci **wzorców**. Jeśli mamy 5 atrybutów $\mathbf{a_1 ... a_5}$, to regułę: $a_1=2 \wedge a_2=7$ zapisujemy jako (2, 7, *, *, *)

Wejście: cały zbiór danych.

1. Znajdź niepokryty obiekt x*.
2. Niech reguła r=(*,*,...,*).
3. Znajdź obiekt y z innej klasy decyzyjnej, niż x*, pasujący do r.
4. Jeśli y nie istnieje – idź do 7.
5. Znajdź taką najlepszą (ze względu na wsparcie) specjalizację r, aby y nie był pokrywany przez r.
   (Specjalizacja polega na zamianie pewnej „*" na wartość pochodzącą z obiektu x*.)
6. Jeśli reguła r ma nadal zbyt dużo kontrprzykładów, wróć do 3.
7. Zapamiętaj r, oznacz pokryte obiekty, wróć do 1.

Pyt 26: Opisać ogólnie algorytm sekwencyjnego pokrywania CN2:

ALGORYTM CN2

Ten algorytm podchodzi do problemu szukania reguł od strony najkrótszych z nich:

- na początku bierzemy regułę pustą, czyli wzorzec (*,*,...,*),
- dokonujemy wszystkich możliwych uszczegółowień,
- dla każdego z nich liczymy wartość przeważającej decyzji i jakość reguły (według przyjętej miary jakości),
- te wzorce, które jeszcze mogą być uszczegółowiane, uszczegółowiamy dalej.

Przykładowe kryteria jakości reguł:
- entropia ze względu na rozkład klas decyzyjnych,
- dokładność reguły.

ALGORYTM CN2

Wejście: cały zbiór danych, minimalny próg wsparcia t, zbiór wzorców atomowych (jednodeskryptorowych) S.

1. Niech zbiór wzorców R={(*,*,...,*)}, r*=(*,*,...,*).
2. Rozszerz każdy wzorzec z R na wszystkie możliwe sposoby, tzn. R=R∩S, usuwając powtórzenia.
3. Dla każdego r∈R policz wsparcie. Jeśli supp(r)<t, to R=R-{r}.
4. Oceń każdy r∈R wybraną miarą jakości (zamień r na regułę wyliczając dominującą decyzję w r). Jeśli r jest lepszy od r*, to r*=r.
5. Usuń z R wszystkie wzorce, z wyjątkiem *m* najlepszych.
6. Jeśli R niepuste, idź do 2.
7. Zwróć r* jako najlepszy wzorzec. Znajdź dominującą decyzję i zbuduj z r* regułę decyzyjną.

Pyt 27: Opisać pojęcie statystycznej istotności stosowane w algorytmie CN2

CN2 jest udoskonaleniem idei konstruowania reguł „od ogółu do szczegółu” .

- Główne własności CN2:
- Stosuje *przeszukiwanie wiązkowe*. W każdym kroku wybieranych jest „wiązka” złożona z m najbardziej obiecujących kandydatów (a nie wszyscy).
- Sprzeczne i niepoprawne wyniki (reguły) są automatycznie eliminowane w trakcie działania algorytmu.
- Reguły statystycznie nieistotne nie są dalej rozważane.
- Ustala decyzję (następnik) dla reguły za pomocą głosowania większościowego.

```
find-rule(P,m, ε)
k*:=< ?, ... ,? >; K:={ k* } ;
while  K ≠ ∅ do
   K_new:=add-selectors(K,S(P));
   K_new:=K_new \ ( K ∪ { < ∅ > } );
   forall k ∈ K_new  do
      if ( ψ_k(P) > ε ∧ θ_k(P)> θ_k*(P))
      then  k*:=k;
   end;
   K:= arg max_{k ∈ K^m_new} (θ_k(P_k));
end;
r:= k* → decyzja(P_k*,d)
return r;
```

- decyzja(S,d) zwraca najczęściej występującą w zbiorze przykładów S decyzję (gdy $S \neq \emptyset$) lub wartość domyślną d.
- $\psi_k(S)$ miara statystycznej istotności dla poprzednika reguły na zbiorze przykładów S. ε stanowi wartość graniczną dla (miary) istotności. Zwykle $\varepsilon \in [0.01, 0.05]$.
- $\theta_k(S)$ miara jakości (części warunkowej **k**) reguły.

Pyt 28: Opisać statystyki stosowane w algorytmie CN2:

Algorytm CN2 - statystyka χ

Niech f_i oznacza *zaobserwowaną częstość* (liczbę wystąpień) i-tej wartości atrybutu y_i dla $i = 1, 2, 3, \ldots, v_1$ i odpowiednio f_j dla y_j dla $j = 1, 2, 3, \ldots, v_2$, f_{ij} liczbę (częstość) jednoczesnych wystąpień i-tej i j-tej wartości atrybutów y_i i y_j, a e_{ij} to wartość oczekiwana jednoczesnego wystąpienia przy założeniu niezależności y_1 i y_2 i $(v_1 - 1)(v_2 - 1)$ stopniach swobody.

$$\chi^2 = \sum_{i=1}^{v_1} \sum_{j=1}^{v_2} \frac{(f_{ij} - e_{ij})^2}{e_{ij}},$$

$$\text{gdzie } e_{ij} = \frac{f_i^1 f_j^2}{n}$$

Im większa wartość statystyki tym bardziej atrybuty są zależne od siebie.

Algorytm CN2 - statystyka χ

$$\chi_k^2(P) = \sum_{d \in C} \frac{(|P_k^d| - e_k^d(P))^2}{e_k^d(P)},$$

$$\text{gdzie } e_k^d(P) = |P_k| \frac{|P^d|}{|P|}$$

Pyt 29: Wyjaśnić mechanizm uzyskiwania uporządkowanego zbioru reguł przez algorytm AQ.

x	wiek	samochód	ryzyko
1	18	maluch	duże
2	35	maluch	małe
3	50	sportowy	duże
4	66	minivan	duże
5	18	sportowy	duże
6	35	minivan	małe
7	60	maluch	małe
8	70	sportowy	duże
9	25	minivan	małe

ROZWIĄZANIE:
Atrybut wiek otrzymuje po dyskretyzacji trzy wartości:

- w_1: wiek < 30,
- w_2: wiek $\geqslant 30 \wedge$ wiek < 65,
- w_3: wiek $\geqslant 65$.

Kolejne kroki algorytmu AQ

(a) Początkowo $R = 0, P = T = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

(b) Następuje wywołanie *znajdź-kompleks*(T, P).

- $x_s = 1, c(x_s) =$ duże, $x_n = 2, c(x_n) =$ małe, $S = \{<?>\}$
- powstaje częściowa gwiazda S': $S = S \cap S' = \{< w_1 \vee w_3, ? >\}$;
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii małe, wybór następnego ziarna negatywnego $x_n = 6$
- $S' = \{< w_1 \vee w_3, ? >, <?, \text{maluch} \vee \text{sportowy} >\}$
- $S = S \cap S' = \{< w_1 \vee w_3, ? >, < w_1 \vee w_3, \text{maluch} \vee \text{sportowy} >\}$
- $S = \{k_1, k_2\}$, $v_{k_1} = |T_{k_1}^{\text{duże}}| + (|T^{\text{małe}}| - |T_{k_1}^{\text{małe}}|) = 4 + (4 - 1) = 7$, $v_{k_2} = 3 + 4 = 7$
  Wartości funkcji oceny dla dwóch uzyskanych kompleksów ze zbioru S są takie same, ale k_2 pokrywa wyłącznie przykłady o jednej etykiecie duże, stąd on wchodzi w skład nowej reguły:

(c) $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} > \rightarrow \text{duże}\}$

(d) $P = \{2, 3, 4, 6, 7, 9\}$, dla $P \neq 0$ *znajdź-kompleks*(P, P)

- $x_s = 2, c(x_s) =$ małe, $x_n = 3, c(x_n) =$ duże, $S = \{<?>\}$
- powstaje częściowa gwiazda S': $S = S \cap S' = \{<?, \text{maluch} \vee \text{minivan} >\}$;
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii duże, wybór następnego ziarna negatywnego $x_n = 4$
- $S' = \{< w_1 \vee w_2, ? >, <?, \text{maluch} \vee \text{sportowy} >\}$
- $S = S \cap S' = \{< w_1 \vee w_2, \text{maluch} \vee \text{minivan} >, <?, \text{maluch} >\}$

- $S = \{k_1, k_2\}$, $v_{k_1} = |I_{k_1}^{\text{małe}}| + (|I^{\text{duże}}| - |I_{k_1}^{\text{duże}}|) = 4 + 2 = 6$, $v_{k_2} = 2 + 2 = 4$
  Kompleks k_1 nie dosyć, że ma lepszą wartość funkcji oceny, to jeszcze pokrywa wyłącznie przykłady o jednej etykiecie małe (ze zbioru P), stąd on wchodzi w skład nowej reguły:

(e) $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} > \rightarrow \text{duże}, < w_1 \vee w_2, \text{maluch} \vee \text{minivan} > \rightarrow \text{małe}\}$

(f) $P = \{3, 4\}$, dla $P \neq 0$ *znajdź-kompleks*(P, P)

- $x_s = 3, c(x_s) = \text{duże}$, $S = \{<? >\}$ Gwiazda S pokrywa przykłady o jednej etykiecie duży i kompleks $<? >$ wchodzi w skład nowej reguły:
- $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} > \rightarrow \text{duże}, < w_1 \vee w_2, \text{maluch} \vee \text{minivan} > \rightarrow \text{małe}, <? > \rightarrow \text{duże}\}$
- ewentualnie, gdy $x_n = 9$, to
- $S = S \cap S' = \{< w_2 \vee w_3, ? >, <?, \text{maluch} \vee \text{sportowy} >\}$
  Kompleks k_1 pokrywa wszystkie przykłady ze zbioru P i wchodzi w skład nowej reguły:

(g) Ostatecznie

$$R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} > \rightarrow \text{duże},$$
$$< w_1 \vee w_2, \text{maluch} \vee \text{minivan} > \rightarrow \text{małe},$$
$$< w_2 \vee w_3, ? > \rightarrow \text{duże}\}$$

W uzyskanym zbiorze reguł NIE można reguł zamieniać miejscami, gdyż jest to zbiór uporządkowany. Najpierw nowe przykłady klasyfikuje reguła pierwsza, jak ona zawiedzie to druga itd.

Pyt 30: Wyjaśnić mechanizm uzyskiwania nieuporządkowanego zbioru reguł przez algorytm AQ

x	wiek	samochod	ryzyko
1	18	maluch	duże
2	35	maluch	małe
3	50	sportowy	duże
4	66	minivan	duże
5	18	sportowy	duże
6	35	minivan	małe
7	60	maluch	małe
8	70	sportowy	duże
9	25	minivan	małe

ROZWIĄZANIE:
Atrybut wiek otrzymuje po dyskretyzacji trzy wartości:

- w_1: wiek < 30,
- w_2: wiek $\geqslant 30 \wedge$ wiek < 65,
- w_3: wiek $\geqslant 65$.

Kolejne kroki algorytmu AQ

(a) Początkowo $R = 0, P = T = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

(b) Następuje wywołanie *znajdź-kompleks*(T, P).

- $x_s = 1, c(x_s) =$ duże, $x_n = 2, c(x_n) =$ małe, $S = \{<? >\}$
- powstaje częściowa gwiazda S': $S = S \cap S' = \{< w_1 \vee w_3, ? >\}$;
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii małe, wybór następnego ziarna negatywnego $x_n = 6$
- $S' = \{< w_1 \vee w_3, ? >, <?, \text{maluch} \vee \text{sportowy} >\}$
- $S = S \cap S' = \{< w_1 \vee w_3, ? >, < w_1 \vee w_3, \text{maluch} \vee \text{sportowy} >\}$
- $S = \{k_1, k_2\}$, $v_{k_1} = |T_{k_1}^{\text{duże}}| + (|T^{\text{małe}}| - |T_{k_1}^{\text{małe}}|) = 4 + (4 - 1) = 7$, $v_{k_2} = 3 + 4 = 7$
  Wartości funkcji oceny dla dwóch uzyskanych kompleksów ze zbioru S są takie same, ale k_2 pokrywa wyłącznie przykłady o jednej etykiecie duże, stąd on wchodzi w skład nowej reguły:

(c) $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} > \rightarrow \text{duże}\}$

(d) $P = \{2, 3, 4, 6, 7, 9\}$, dla $P \neq 0$ *znajdź-kompleks*(T, P)

- $x_s = 2, c(x_s) =$ małe, $x_n = 3, c(x_n) =$ duże, $S = \{<? >\}$
- powstaje częściowa gwiazda S': $S = S \cap S' = \{<?, \text{maluch} \vee \text{minivan} >\}$;
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii duże, wybór następnego ziarna negatywnego $x_n = 4$
- $S' = \{< w_1 \vee w_2, ? >, <?, \text{maluch} \vee \text{sportowy} >\}$
- $S = S \cap S' = \{< w_1 \vee w_2, \text{maluch} \vee \text{minivan} >, <?, \text{maluch} >\}$
- $S = \{k_1, k_2\}$, $v_{k_1} = |T_{k_1}^{\text{małe}}| + (|T^{\text{duże}}| - |T_{k_1}^{\text{duże}}|) = 4 + 5 = 9$, $v_{k_2} = 2 + (5 - 1) = 6$
  Kompleks k_1 ma lepszą wartość funkcji oceny, stąd pozostaje w składzie gwiazdy (jej parametr $m = 1$).
  $S = \{< w_1 \vee w_2, \text{maluch} \vee \text{minivan} >\}$.
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii duże (ze zbioru T), wybór następnego ziarna negatywnego $x_n = 5$
- $S' = \{< w_2 \vee w_3, ? >, <?, \text{maluch} \vee \text{minivan} >\}$
- $S = S \cap S' = \{< w_2, \text{maluch} \vee \text{minivan} >, < w_1 \vee w_2, \text{maluch} \vee \text{minivan} >\}$

- $S = \{\kappa_1, \kappa_2\}$, $v_{k_1} = |I_{k_1}^{\text{małe}}| + (|I^{\text{duże}}| - |I_{k_1}^{\text{duże}}|) = 3 + 5 = 8$, $v_{k_2} = 4 + (5 - 2) = 7$
  Kompleks k_1 nie dosyć, że ma lepszą wartość funkcji oceny, to jeszcze pokrywa wyłącznie przykłady o jednej etykiecie małe (ze zbioru T), stąd on wchodzi w skład nowej reguły:

(e) $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} >\rightarrow \text{duże}, < w_2, \text{maluch} \vee \text{minivan} >\rightarrow \text{małe}\}$

(f) $P = \{3, 4, 9\}$, dla $P \neq 0$ *znajdź-kompleks*(T, P)

- $x_s = 3, c(x_s) = \text{duże}$, $S = \{<? >\}$, $x_n = 6$
- $S = S \cap S' = \{<?, \text{maluch} \vee \text{sportowy} >\}$
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii małe ze zbioru T, wybór następnego ziarna negatywnego $x_n = 7$
- $S' = \{<?, \text{sportowy} \vee \text{minivan} >\}$
- $S = S \cap S' = \{<?, \text{sportowy} >\}$ Kompleks z S pokrywa wyłącznie przykłady o jednej etykiecie duże (ze zbioru T), stąd on wchodzi w skład nowej reguły:

(g) $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} >\rightarrow \text{duże}, < w_2, \text{maluch} \vee \text{minivan} >\rightarrow \text{małe}, < ?, \text{sportowy} >\rightarrow \text{duże}\}$

(h) $P = \{4, 9\}$, dla $P \neq 0$ *znajdź-kompleks*(T, P)

- $x_s = 4, c(x_s) = \text{duże}$, $S = \{<? >\}$, $x_n = 9$
- $S = S \cap S' = \{< w2 \vee w3, ? >\}$
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii małe ze zbioru T, wybór następnego ziarna negatywnego $x_n = 6$
- $S' = \{< w1 \vee w3, ? >\}$
- $S = S \cap S' = \{< w3, ? >\}$
  Kompleks z S pokrywa wyłącznie przykłady o jednej etykiecie duże (ze zbioru T), stąd on wchodzi w skład nowej reguły:

(i) $R = \{< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} >\rightarrow \text{duże}, < w_2, \text{maluch} \vee \text{minivan} >\rightarrow \text{małe}, < ?, \text{sportowy} >\rightarrow \text{duże}, < w3, ? >\rightarrow \text{duże}\}$

(j) $P = \{9\}$, dla $P \neq 0$ *znajdź-kompleks*(T, P)

- $x_s = 9, c(x_s) = \text{duże}$, $S = \{<? >\}$, $x_n = 4$
- $S = S \cap S' = \{< w1 \vee w2, ? >\}$
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii duże ze zbioru T, wybór następnego ziarna negatywnego $x_n = 1$
- $S' = \{<?, \text{minivan} \vee \text{sportowy} >\}$
- $S = S \cap S' = \{< w1 \vee w2, \text{minivan} \vee \text{sportowy} >\}$
- gwiazda w dalszym ciągu pokrywa przykłady z T o kategorii duże ze zbioru T, wybór następnego ziarna negatywnego $x_n = 5$
- $S' = \{<?, \text{minivan} \vee \text{maluch} >\}$
- $S = S \cap S' = \{< w1 \vee w2, \text{minivan} >\}$
  Kompleks z S pokrywa wyłącznie przykłady o jednej etykiecie małe (ze zbioru T), stąd on wchodzi w skład nowej reguły:

(k) Ostatecznie

$$
\begin{aligned}
R = \{&< w_1 \vee w_3, \text{maluch} \vee \text{sportowy} >\rightarrow \text{duże},\\
&< w_2, \text{maluch} \vee \text{minivan} >\rightarrow \text{małe},\\
&<?, \text{sportowy} >\rightarrow \text{duże},\\
&< w3, ? >\rightarrow \text{duże},\\
&< w1 \vee w2, \text{minivan} >\rightarrow \text{małe}\}
\end{aligned}
$$

W uzyskanym zbiorze reguł można reguły zamieniać miejscami, gdyż jest to zbiór

Pyt 31: Wyjaśnić mechanizm uzyskiwania uporządkowanego zbioru reguł przez algorytm CN2.

x	wiek	samochód	ryzyko
1	18	maluch	duże
2	35	maluch	małe
3	50	sportowy	duże
4	66	minivan	duże
5	18	sportowy	duże
6	35	minivan	małe
7	60	maluch	małe
8	70	sportowy	duże
9	25	minivan	małe

ROZWIĄZANIE:
Atrybut wiek otrzymuje po dyskretyzacji trzy wartości:

- w_1: wiek < 30,
- w_2: wiek $\geqslant 30 \wedge$ wiek < 65,
- w_3: wiek $\geqslant 65$.

Zbiór $\mathbb{S}$ kompleksów atomowych (czyli tylko z jednym selektorem nieuniwersalnym) ($\mathbb{S} = \{\mathbb{K}_1, \mathbb{K}_2, \mathbb{K}_3, \mathbb{K}_4, \mathbb{K}_5, \mathbb{K}_6, \mathbb{K}_7, \mathbb{K}_8, \mathbb{K}_9, \mathbb{K}_{10}, \mathbb{K}_{11}, \mathbb{K}_{12}\}$) jest następujący:

	$\mathbb{S} = \{$
$\mathbb{K}_1$	$< w_1, ? >$,
$\mathbb{K}_2$	$< w_2, ? >$,
$\mathbb{K}_3$	$< w_3, ? >$,
$\mathbb{K}_4$	$< w_1 \vee w_2, ? >$,
$\mathbb{K}_5$	$< w_2 \vee w_3, ? >$,
$\mathbb{K}_6$	$< w_1 \vee w_3, ? >$,
$\mathbb{K}_7$	$<?,$ maluch $>$,
$\mathbb{K}_8$	$<?,$ minivan $>$,
$\mathbb{K}_9$	$<?,$ sportowy $>$,
$\mathbb{K}_{10}$	$<?,$ maluch $\vee$ minivan $>$,
$\mathbb{K}_{11}$	$<?,$ minivan $\vee$ sportowy $>$,
$\mathbb{K}_{12}$	$<?,$ maluch $\vee$ sportowy $>\}$

Kolejne kroki algorytmu CN2

(a) Początkowo $R = \phi, P = T = \{1,2,3,4,5,6,7,8,9\}, \mathbb{S}$

(b) Następuje wywołanie *znajdź-kompleks*(T, P).

- $S = \{<? >\} \neq \phi, k_* = <? >$
  $\vartheta_{k_*}(P) = -E_{k_*}(P) = \frac{|P^{małe}|}{|P|}\log_2(\frac{|P^{małe}|}{|P|}) + \frac{|P^{duże}|}{|P|}\log_2(\frac{|P^{duże}|}{|P|}) = \frac{5}{9}\log_2(\frac{5}{9}) + \frac{4}{9}\log_2(\frac{4}{9}) = -0.991,$
- $S' = \mathbb{S} = S \cap \mathbb{S},$
  $\vartheta_{\mathbb{K}_1}(P) = -E_{\mathbb{K}_1}(P) = \frac{|P^{małe}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}) + \frac{|P^{duże}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}) = \frac{1}{3}\log_2(\frac{1}{3}) +$

$\frac{2}{3}\log_2(\frac{2}{3}) = -0.918,$

$\vartheta_{\mathbb{K}_2}(P) = -E_{\mathbb{K}_2}(P) = \frac{|P^{małe}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}) + \frac{|P^{duże}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}) = \frac{3}{4}\log_2(\frac{3}{4}) +$
$\frac{1}{4}\log_2(\frac{1}{4}) = -0.811,$

$\vartheta_{\mathbb{K}_3}(P) = -E_{\mathbb{K}_3}(P) = \frac{|P^{małe}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}) + \frac{|P^{duże}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}) = \frac{0}{3}\log_2(\frac{0}{3}) +$
$\frac{3}{3}\log_2(\frac{3}{3}) = 0,$

$\vartheta_{\mathbb{K}_4}(P) = -E_{\mathbb{K}_4}(P) = \frac{|P^{małe}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}) + \frac{|P^{duże}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}) = \frac{4}{7}\log_2(\frac{4}{7}) +$
$\frac{3}{7}\log_2(\frac{3}{7}) = -0.985,$

$\vartheta_{\mathbb{K}_5}(P) = -E_{\mathbb{K}_5}(P) = \frac{|P^{małe}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}) + \frac{|P^{duże}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}) = \frac{3}{6}\log_2(\frac{3}{6}) +$
$\frac{3}{6}\log_2(\frac{3}{6}) = -1,$

$\vartheta_{\mathbb{K}_6}(P) = -E_{\mathbb{K}_6}(P) = \frac{|P^{małe}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}) + \frac{|P^{duże}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}) = \frac{1}{5}\log_2(\frac{1}{5}) +$
$\frac{4}{5}\log_2(\frac{4}{5}) = -0.721,$

$\vartheta_{\mathbb{K}_7}(P) = -E_{\mathbb{K}_7}(P) = \frac{|P^{małe}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}) + \frac{|P^{duże}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}) = \frac{2}{3}\log_2(\frac{2}{3}) +$
$\frac{1}{3}\log_2(\frac{1}{3}) = -0.918,$

$\vartheta_{\mathbb{K}_8}(P) = -E_{\mathbb{K}_8}(P) = \frac{|P^{małe}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}) + \frac{|P^{duże}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}) = \frac{2}{3}\log_2(\frac{2}{3}) +$
$\frac{1}{3}\log_2(\frac{1}{3}) = -0.918,$

$\vartheta_{\mathbb{K}_9}(P) = -E_{\mathbb{K}_9}(P) = \frac{|P^{małe}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}) + \frac{|P^{duże}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}) = \frac{0}{3}\log_2(\frac{0}{3}) +$
$\frac{3}{3}\log_2(\frac{3}{3}) = 0,$

$\vartheta_{\mathbb{K}_{10}}(P) = -E_{\mathbb{K}_{10}}(P) = \frac{|P^{małe}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}) + \frac{|P^{duże}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}) = \frac{4}{6}\log_2(\frac{4}{6}) +$
$\frac{2}{6}\log_2(\frac{2}{6}) = -0.918,$

$\vartheta_{\mathbb{K}_{11}}(P) = -E_{\mathbb{K}_{11}}(P) = \frac{|P^{małe}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}) + \frac{|P^{duże}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}) = \frac{2}{6}\log_2(\frac{2}{6}) +$
$\frac{4}{6}\log_2(\frac{4}{6}) = -0.918,$

$\vartheta_{\mathbb{K}_{12}}(P) = -E_{\mathbb{K}_{12}}(P) = \frac{|P^{małe}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}) + \frac{|P^{duże}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}) = \frac{2}{6}\log_2(\frac{2}{6}) +$
$\frac{4}{6}\log_2(\frac{4}{6}) = -0.918$

- $\mathbb{K}_9 = <?, \text{sportowy} >$ ma największą wartość $\vartheta = 0$ w zbiorze $\mathbb{S}$ razem z $\mathbb{K}_3$, ale więcej przykładów pokrywa; $S = \{\mathbb{K}_9\}, k_* = \mathbb{K}_9,$

(c) $R = \{<?, \text{sportowy} > \rightarrow \text{duże}\},\ P = \{1, 2, 4, 6, 7, 9\},$

(d) $P \neq \phi \Rightarrow$ *znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* = <? >$ i $\vartheta_{k_*}(P) = -0.918,$
- $S' = \mathbb{S} - S \cap \mathbb{S}$

sportowy już w zbiorze P nie ma.
Dla zbioru uporządkowanego trzeba wartość funkcji oceny kompleksów atomowych obliczać przed każdym wyborem najlepszego kompleksu.
$\vartheta_{\mathbb{K}_1}(P) = -1$, $\vartheta_{\mathbb{K}_2}(P) = 0$, $\vartheta_{\mathbb{K}_3}(P) = 0$, $\vartheta_{\mathbb{K}_4}(P) = -0,721$, $\vartheta_{\mathbb{K}_5}(P) = -0,811$, $\vartheta_{\mathbb{K}_6}(P) = -0.918$, $\vartheta_{\mathbb{K}_7}(P) = -0.918$, $\vartheta_{\mathbb{K}_8}(P) = -0.918$, $\vartheta_{\mathbb{K}_{10}}(P) = -0.918$,

- $\mathbb{K}_2 =< w_2, ? >$ ma największą wartość $\vartheta = 0$ razem z $\mathbb{K}_3$, ale więcej przykładów pokrywa; $S = \{\mathbb{K}_2\}, k_* = \mathbb{K}_2$,

(e) $R = \{<?, \text{sportowy} > \rightarrow \text{duże}, < w2, ? > \rightarrow \text{małe}\}$, $P = \{1, 4, 9\}$,

(f) $P \neq \phi \Rightarrow$*znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* =<? >$ i $\vartheta_{k_*}(P) = -0.918$,
ze względu na użycie $\mathbb{K}_2$ wyklucza się wszystkie kompleksy atomowe z wartością atrybutu wiek $= w_2$ czyli $\mathbb{K}_2$,$\mathbb{K}_4$,$\mathbb{K}_5$, bo takich przykładów z wartością w_2 już w zbiorze P nie ma.
$\vartheta_{\mathbb{K}_1}(P) = -1$, $\vartheta_{\mathbb{K}_3}(P) = 0$, $\vartheta_{\mathbb{K}_6}(P) = -0.918$, $\vartheta_{\mathbb{K}_7}(P) = 0$, $\vartheta_{\mathbb{K}_8}(P) = -1$, $\vartheta_{\mathbb{K}_{10}}(P) = -0.918$,
- $\mathbb{K}_3 =< w_3, ? >$ ma największą wartość $\vartheta = 0$ razem z $\mathbb{K}_7$ i tyle samo przykładów pokrywa, ale trzeba wybrać i można zauważyć, że w zbiorze T pokrywa tylko przykłady o jednej etykiecie; $S = \{\mathbb{K}_3\}, k_* = \mathbb{K}_3$,

(g) $R = \{<?, \text{sportowy} > \rightarrow \text{duże}, < w2, ? > \rightarrow \text{małe}, < w3, ? > \rightarrow \text{duże}\}$, $P = \{1, 9\}$,

(h) $P \neq \phi \Rightarrow$*znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* =<? >$ i $\vartheta_{k_*}(P) = -1$,
- $\mathbb{K}_8 =<?, \text{minivan} >$ ma największą wartość $\vartheta = 0$ razem z $\mathbb{K}_7$ i tyle samo przykładów pokrywa, ale trzeba wybrać go wybrać, aby ostatni przykład miał etykietę duże; $S = \{\mathbb{K}_8\}, k_* = \mathbb{K}_8$,

(i) $R = \{<?, \text{sportowy} > \rightarrow \text{duże}, < w2, ? > \rightarrow \text{małe}, < w3, ? > \rightarrow \text{duże}, <?, \text{minivan} > \rightarrow \text{małe}\}$, $P = \{1\}$,

(j) $P \neq \phi \Rightarrow$*znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* =<? >$ i $\vartheta_{k_*}(P) = 0$,
Kompleks k_* tym razem ma największą wartość funkcji oceny i zostaje częścią reguły.

(k) Ostatecznie
$R = \{<?, \text{sportowy} > \rightarrow \text{duże},$
$< w2, ? > \rightarrow \text{małe},$
$< w3, ? > \rightarrow \text{duże},$
$<?, \text{minivan} > \rightarrow \text{małe},$
$<? > \rightarrow \text{duże}\}$

W uzyskanym zbiorze reguł NIE można reguł zamieniać miejscami, gdyż jest to zbiór uporządkowany. Najpierw nowe przykłady klasyfikuje reguła pierwsza, jak ona zawiedzie to druga itd.

Pyt 32: Wyjaśnić mechanizm uzyskiwania nieuporządkowanego zbioru reguł przez algorytm CN2

3. Za pomocą algorytmu sekwencyjnego pokrywania CN2 uzyskać uporządkowany zbiór zdaniowych reguł ze zbioru treningowego podanego w tabeli poniżej. Opisać dokładnie kolejne kroki algorytmu. Atrybut wiek zdyskretyzować korzystając z dwóch progów 30 i 65 lat. Atrybut ryzyko będzie kategorią. Dla ułatwienia założyć, że wszystkie kompleksy są istotne statystycznie oraz że kompleks warunkujący z reguły zdaniowej musi pokrywać przykłady tylko z jedną etykietą - jedną wartością kategorii.

x	wiek	samochód	ryzyko
1	18	maluch	duże
2	35	maluch	małe
3	50	sportowy	duże
4	66	minivan	duże
5	18	sportowy	duże
6	35	minivan	małe
7	60	maluch	małe
8	70	sportowy	duże
9	25	minivan	małe

ROZWIĄZANIE:
Atrybut wiek otrzymuje po dyskretyzacji trzy wartości:

- w_1: wiek < 30,
- w_2: wiek $\geqslant 30 \wedge$ wiek < 65,
- w_3: wiek $\geqslant 65$.

Zbiór $\mathbb{S}$ kompleksów atomowych (czyli tylko z jednym selektorem nieuniwersalnym) ($\mathbb{S} = \{\mathbb{K}_1, \mathbb{K}_2, \mathbb{K}_3, \mathbb{K}_4, \mathbb{K}_5, \mathbb{K}_6, \mathbb{K}_7, \mathbb{K}_8, \mathbb{K}_9, \mathbb{K}_{10}, \mathbb{K}_{11}, \mathbb{K}_{12}\}$) jest następujący:

	$\mathbb{S} = \{$
$\mathbb{K}_1$	$< w_1, ? >$,
$\mathbb{K}_2$	$< w_2, ? >$,
$\mathbb{K}_3$	$< w_3, ? >$,
$\mathbb{K}_4$	$< w_1 \vee w_2, ? >$,
$\mathbb{K}_5$	$< w_2 \vee w_3, ? >$,
$\mathbb{K}_6$	$< w_1 \vee w_3, ? >$,
$\mathbb{K}_7$	$<?,$ maluch $>$,
$\mathbb{K}_8$	$<?,$ minivan $>$,
$\mathbb{K}_9$	$<?,$ sportowy $>$,
$\mathbb{K}_{10}$	$<?,$ maluch $\vee$ minivan $>$,
$\mathbb{K}_{11}$	$<?,$ minivan $\vee$ sportowy $>$,
$\mathbb{K}_{12}$	$<?,$ maluch $\vee$ sportowy $>\}$

Kolejne kroki algorytmu CN2

(a) Początkowo $R = \phi, P = T = \{1,2,3,4,5,6,7,8,9\}, \mathbb{S}$

(b) Następuje wywołanie *znajdź-kompleks*(T, P).

- $S = \{<? >\} \neq \phi, k_* = <? >$
  $\vartheta_{k_*}(P) = -E_{k_*}(P) = \frac{|P^{małe}|}{|P|}\log_2(\frac{|P^{małe}|}{|P|}) + \frac{|P^{duże}|}{|P|}\log_2(\frac{|P^{duże}|}{|P|}) = \frac{5}{9}\log_2(\frac{5}{9}) + \frac{4}{9}\log_2(\frac{4}{9}) = -0.991,$
- $S' = \mathbb{S} = S \cap \mathbb{S},$
  $\vartheta_{\mathbb{K}_1}(P) = -E_{\mathbb{K}_1}(P) = \frac{|P^{małe}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}) + \frac{|P^{duże}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_1}|}{|P_{\mathbb{K}_1}|}) = \frac{1}{3}\log_2(\frac{1}{3}) +$

$\frac{2}{3}\log_2(\frac{2}{3}) = -0.918,$

$\vartheta_{\mathbb{K}_2}(P) = -E_{\mathbb{K}_2}(P) = \frac{|P^{małe}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}) + \frac{|P^{duże}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_2}|}{|P_{\mathbb{K}_2}|}) = \frac{3}{4}\log_2(\frac{3}{4}) +$
$\frac{1}{4}\log_2(\frac{1}{4}) = -0.811,$

$\vartheta_{\mathbb{K}_3}(P) = -E_{\mathbb{K}_3}(P) = \frac{|P^{małe}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}) + \frac{|P^{duże}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_3}|}{|P_{\mathbb{K}_3}|}) = \frac{0}{3}\log_2(\frac{0}{3}) +$
$\frac{3}{3}\log_2(\frac{3}{3}) = 0,$

$\vartheta_{\mathbb{K}_4}(P) = -E_{\mathbb{K}_4}(P) = \frac{|P^{małe}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}) + \frac{|P^{duże}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_4}|}{|P_{\mathbb{K}_4}|}) = \frac{4}{7}\log_2(\frac{4}{7}) +$
$\frac{3}{7}\log_2(\frac{3}{7}) = -0.985,$

$\vartheta_{\mathbb{K}_5}(P) = -E_{\mathbb{K}_5}(P) = \frac{|P^{małe}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}) + \frac{|P^{duże}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_5}|}{|P_{\mathbb{K}_5}|}) = \frac{3}{6}\log_2(\frac{3}{6}) +$
$\frac{3}{6}\log_2(\frac{3}{6}) = -1,$

$\vartheta_{\mathbb{K}_6}(P) = -E_{\mathbb{K}_6}(P) = \frac{|P^{małe}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}) + \frac{|P^{duże}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_6}|}{|P_{\mathbb{K}_6}|}) = \frac{1}{5}\log_2(\frac{1}{5}) +$
$\frac{4}{5}\log_2(\frac{4}{5}) = -0.721,$

$\vartheta_{\mathbb{K}_7}(P) = -E_{\mathbb{K}_7}(P) = \frac{|P^{małe}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}) + \frac{|P^{duże}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_7}|}{|P_{\mathbb{K}_7}|}) = \frac{2}{3}\log_2(\frac{2}{3}) +$
$\frac{1}{3}\log_2(\frac{1}{3}) = -0.918,$

$\vartheta_{\mathbb{K}_8}(P) = -E_{\mathbb{K}_8}(P) = \frac{|P^{małe}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}) + \frac{|P^{duże}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_8}|}{|P_{\mathbb{K}_8}|}) = \frac{2}{3}\log_2(\frac{2}{3}) +$
$\frac{1}{3}\log_2(\frac{1}{3}) = -0.918,$

$\vartheta_{\mathbb{K}_9}(P) = -E_{\mathbb{K}_9}(P) = \frac{|P^{małe}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}) + \frac{|P^{duże}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_9}|}{|P_{\mathbb{K}_9}|}) = \frac{0}{3}\log_2(\frac{0}{3}) +$
$\frac{3}{3}\log_2(\frac{3}{3}) = 0,$

$\vartheta_{\mathbb{K}_{10}}(P) = -E_{\mathbb{K}_{10}}(P) = \frac{|P^{małe}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}) + \frac{|P^{duże}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_{10}}|}{|P_{\mathbb{K}_{10}}|}) = \frac{4}{6}\log_2(\frac{4}{6}) +$
$\frac{2}{6}\log_2(\frac{2}{6}) = -0.918,$

$\vartheta_{\mathbb{K}_{11}}(P) = -E_{\mathbb{K}_{11}}(P) = \frac{|P^{małe}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}) + \frac{|P^{duże}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_{11}}|}{|P_{\mathbb{K}_{11}}|}) = \frac{2}{6}\log_2(\frac{2}{6}) +$
$\frac{4}{6}\log_2(\frac{4}{6}) = -0.918,$

$\vartheta_{\mathbb{K}_{12}}(P) = -E_{\mathbb{K}_{12}}(P) = \frac{|P^{małe}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}\log_2(\frac{|P^{małe}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}) + \frac{|P^{duże}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}\log_2(\frac{|P^{duże}_{\mathbb{K}_{12}}|}{|P_{\mathbb{K}_{12}}|}) = \frac{2}{6}\log_2(\frac{2}{6}) +$
$\frac{4}{6}\log_2(\frac{4}{6}) = -0.918$

- $\mathbb{K}_9 = <?, \text{sportowy} >$ ma największą wartość $\vartheta = 0$ w zbiorze $\mathbb{S}$ razem z $\mathbb{K}_3$, ale więcej przykładów pokrywa; $S = \{\mathbb{K}_9\}, k_* = \mathbb{K}_9,$

(c) $R = \{<?, \text{sportowy} > \rightarrow \text{duże}\},\ P = \{1, 2, 4, 6, 7, 9\},$

(d) $P \neq \phi \Rightarrow$ *znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* = <? >$ i $\vartheta_{k_*}(P) = -0.918,$
- $S' = \mathbb{S} = S \cap \mathbb{S},$

sportowy już w zbiorze P nie ma.
Dla zbioru uporządkowanego trzeba wartość funkcji oceny kompleksów atomowych obliczać przed każdym wyborem najlepszego kompleksu.
$\vartheta_{\mathbb{K}_1}(P) = -1$, $\vartheta_{\mathbb{K}_2}(P) = 0$, $\vartheta_{\mathbb{K}_3}(P) = 0$, $\vartheta_{\mathbb{K}_4}(P) = -0,721$, $\vartheta_{\mathbb{K}_5}(P) = -0,811$, $\vartheta_{\mathbb{K}_6}(P) = -0.918$, $\vartheta_{\mathbb{K}_7}(P) = -0.918$, $\vartheta_{\mathbb{K}_8}(P) = -0.918$, $\vartheta_{\mathbb{K}_{10}}(P) = -0.918$,

- $\mathbb{K}_2 =< w_2, ? >$ ma największą wartość $\vartheta = 0$ razem z $\mathbb{K}_3$, ale więcej przykładów pokrywa; $S = \{\mathbb{K}_2\}, k_* = \mathbb{K}_2$,

(e) $R = \{<?, \text{sportowy} >\rightarrow \text{duże}, < w2, ? >\rightarrow \text{małe}\}$, $P = \{1, 4, 9\}$,

(f) $P \neq \phi \Rightarrow$*znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* =<? >$ i $\vartheta_{k_*}(P) = -0.918$,
ze względu na użycie $\mathbb{K}_2$ wyklucza się wszystkie kompleksy atomowe z wartością atrybutu wiek $= w_2$ czyli $\mathbb{K}_2$,$\mathbb{K}_4$,$\mathbb{K}_5$, bo takich przykładów z wartością w_2 już w zbiorze P nie ma.
$\vartheta_{\mathbb{K}_1}(P) = -1$, $\vartheta_{\mathbb{K}_3}(P) = 0$, $\vartheta_{\mathbb{K}_6}(P) = -0.918$, $\vartheta_{\mathbb{K}_7}(P) = 0$, $\vartheta_{\mathbb{K}_8}(P) = -1$, $\vartheta_{\mathbb{K}_{10}}(P) = -0.918$,
- $\mathbb{K}_3 =< w_3, ? >$ ma największą wartość $\vartheta = 0$ razem z $\mathbb{K}_7$ i tyle samo przykładów pokrywa, ale trzeba wybrać i można zauważyć, że w zbiorze T pokrywa tylko przykłady o jednej etykiecie; $S = \{\mathbb{K}_3\}, k_* = \mathbb{K}_3$,

(g) $R = \{<?, \text{sportowy} >\rightarrow \text{duże}, < w2, ? >\rightarrow \text{małe}, < w3, ? >\rightarrow \text{duże}\}$, $P = \{1, 9\}$,

(h) $P \neq \phi \Rightarrow$*znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* =<? >$ i $\vartheta_{k_*}(P) = -1$,
- $\mathbb{K}_8 =<?, \text{minivan} >$ ma największą wartość $\vartheta = 0$ razem z $\mathbb{K}_7$ i tyle samo przykładów pokrywa, ale trzeba wybrać go wybrać, aby ostatni przykład miał etykietę duże; $S = \{\mathbb{K}_8\}, k_* = \mathbb{K}_8$,

(i) $R = \{<?, \text{sportowy} >\rightarrow \text{duże}, < w2, ? >\rightarrow \text{małe}, < w3, ? >\rightarrow \text{duże}, <?, \text{minivan} >\rightarrow \text{małe}\}$, $P = \{1\}$,

(j) $P \neq \phi \Rightarrow$*znajdź-kompleks*(T, P),

- $S = \{<? >\} \neq \phi, k_* =<? >$ i $\vartheta_{k_*}(P) = 0$,
Kompleks k_* tym razem ma największą wartość funkcji oceny i zostaje częścią reguły.

(k) Ostatecznie
$R = \{<?, \text{sportowy} >\rightarrow \text{duże},$
$< w2, ? >\rightarrow \text{małe},$
$< w3, ? >\rightarrow \text{duże},$
$<?, \text{minivan} >\rightarrow \text{małe},$
$<? >\rightarrow \text{duże}\}$

W uzyskanym zbiorze reguł NIE można reguł zamieniać miejscami, gdyż jest to zbiór uporządkowany. Najpierw nowe przykłady klasyfikuje reguła pierwsza, jak ona zawiedzie to druga itd.

Pyt33: Co to takiego tablice kontyngencji:

Tablice kontyngencji są znaną ze statystyki metodą prezentacji zależności występujących pomiędzy wartościami dwóch (w podstawowym przypadku) zmiennych losowych, którymi

przy dokonywaniu odkryć są atrybuty. W przypadku dwóch atrybutów a_1 i a_2 o niewielkiej liczbie wartości dyskretnych tablica kontyngencji ma wiersze odpowiadające wszystkim wartościom atrybutu a_1 (ze zbioru A_1) i kolumny odpowiadające wszystkim wartościom atrybutu a_2 (ze zbioru A_2). Element tablicy na przecięciu wiersza odpowiadającego wartości $v_1 \in A_1$ i kolumny odpowiadającej wartości $v_2 \in A_2$ jest liczbą przykładów (rekordów w rozpatrywanej tabeli), dla których a_1 ma wartość v_1 i a_2 ma wartość v_2.

W przypadku atrybutów ciągłych ich wartości są dyskretyzowane. W najprostszym przypadku, gdy dla obu atrybutów dyskretyzacja dzieli zakres wartości na dwa przedziały (małe i duże wartości), tablica kontyngencji jest czteroelementową tablicą 2×2. Dla atrybutów dyskretnych o dużej liczbie wartości wstępnie przeprowadza się na ogół agregację tych wartości (rodzaj konstruktywnej indukcji).

Znanym systemem odkrywania zależności w danych wykorzystującym tablice kontyngencji jest *FortyNiner* (49*er*) Żytkowa i Zembowicza, na którym luźno oparta jest poniższa dyskusja.

Pyt34: Wyjaśnić pojęcie odkrywania wiedzy w danych (ang. Data, knowledge mining). Czym różni się od badań statystycznych? I kiedy się stosuje algorytmy indukcyjnego odkrywania wiedzy?

Odkrywanie wiedzy czyli zależności w danych (ang. Knowledge Mining, Data Mining) łączy techniki wywodzące się z uczenia się maszyn i statystyki w celu pozyskiwania wiedzy z dużych, rzeczywistych baz danych.

Termin *data mining* ostatnio pojawia się coraz częściej nie tylko w publikacjach naukowych, lecz także w marketingowych materiałach różnych firm oferujących oprogramowanie i usługi w dziedzinie analizy danych. W praktyce dokładne znaczenie wiązane z tym terminem bywa różne, ponieważ wyraźnie opłaca się go używać, nawet jeśli jest to tylko w niewielkim stopniu uzasadnione. Funkcjonuje również inny termin, *knowledge discovery in databases*. Niektórzy rozróżniają znaczenie tych dwóch terminów, a zwłaszcza ich zakresy znaczeniowe: jeden z nich jest szerszy, chociaż trudno bez dokładnych studiów powiedzieć który. My przyjmiemy tutaj, że w obu przypadkach mowa jest o odkrywaniu zależności występujących w dużych zbiorach danych, zazwyczaj przechowywanych w bazach danych (obecnie najczęściej relacyjnych).

W przypadku relacyjnych baz danych można przyjąć, że zależności poszukuje się w tabeli zawierającej wiele rekordów, z których każdy stanowi zestaw wartości pewnej liczby atrybutów o różnych typach. Zależności można uznać za interesujące, jeśli dotyczą atrybutów ważnych dla posiadacza danych. Są one natomiast użyteczne, jeśli charakteryzują się:

- dużym zakresem (czyli zachodzą dla wielu rekordów),

- dużą dokładnością (czyli występują od nich co najwyżej niewielkie odchylenia),
- dużym znaczeniem statystycznym (czyli nie są przypadkowe).

Statystyczne metody analizy danych są w większości znane od wielu lat i stosowane z powodzeniem do rozwiązywania wielu praktycznych problemów w różnych obszarach zastosowań, lecz używane w tradycyjny sposób napotykają na pewne ograniczenia. W uproszczeniu, pozwalają one na wykrywanie korelacji między różnymi zjawiskami (np. wartościami różnych atrybutów w bazie danych), wykrywanie występujących trendów, dopasowywanie równania do zbioru punktów pomiarowych, wykrywanie skupień itd., ale nie generują wyjaśnień tych zależności i ich opisów w abstrakcyjnej, symbolicznej postaci, użytecznej do wyciągania wniosków. Aby odkrywane za pomocą analizy statystycznej zależności były w pełni użyteczne, konieczny jest najczęściej daleko idący udział doświadczonego użytkownika przy ich stosowaniu i interpretacji. Można natomiast powiedzieć, że większość metod uczenia się ma na celu odkrycie (nauczenie się) zależności w sposób maksymalnie zautomatyzowany i utworzenie ich opisu, który jest łatwy do interpretacji i pozwala na wnioskowanie.

O omawianych przez nas dotychczas algorytmach indukcyjnego uczenia się na podstawie przykładów (np. ID3, AQ, CN2, COBWEB itd.) można powiedzieć nie popełniając żadnego nadużycia, że są metodami odkrywania zależności w bazach danych. Jednak stosując którykolwiek z tych algorytmów do odkrywania wiedzy w bazach danych zakładamy natychmiast, że zbiór trenujący jest *bardzo duży*: liczba rekordów liczona jest na ogół w tysiącach albo milionach. O ile algorytmy te nie mogą uczyć się użytecznej wiedzy przy zbyt małej ilości danych, duża ilość danych stwarza inne problemy.

Koszt obliczeniowy wszystkich metod zależy oczywiście od ilości danych, na których operują, i w niektórych przypadkach może okazać się trudny do akceptacji. Wówczas można wziąć pod uwagę przeprowadzenie redukcji rozmiaru dostępnego zbioru danych. Należy to oczywiście uczynić w sposób, który z dużym prawdopodobieństwem pozostawi w zredukowanych danych interesujące i użyteczne zależności występujące w oryginalnym zbiorze danych. Redukcja taka może mieć postać

- pozostawienia tylko interesujących i istotnych atrybutów (rodzaj konstruktywnej indukcji),
- pozostawienia tylko najbardziej reprezentatywnych rekordów,
- przeprowadzenia grupowania jako fazy przetwarzania wstępnego i odkrywania dalszych zależności dla grup zamiast pojedynczych rekordów.

Inny problem wiąże się z tym, że w przypadku bardzo dużej ilości przykładów może być niemożliwe znalezienie hipotezy, która przy dostatecznie dużej dokładności byłaby na tyle prosta, aby jej interpretacja przez człowieka prowadziła do interesujących wniosków. Lepszym pomysłem może być poszukiwanie hipotez o ograniczonym zakresie -- zachodzących dla wyznaczonych w systematyczny sposób fragmentów bazy danych.

Z całą pewnością algorytmy stosowane do odkrywania wiedzy w bazach danych muszą sobie radzić z ich zaszumieniem i niekompletością. W pierwszym przypadku oznacza to stosowanie różnych technik zapobiegających nadmiernemu dopasowaniu do przypadkowych

(nieistotnych) danych, a w drugim np. wypełnianie w odpowiedni sposób nieznanych wartości atrybutów.

Przy uwzględnieniu wszystkich tych praktycznie istotnych problemów do odkrywania zależności w bazach danych mogą być i są rzeczywiście stosowane metody uczenia się już przez nas poznane. W ramach tego wykładu krótko omówimy dwie inne metody dokonywania odkryć:

- odkrywanie zależności za pomocą tablic kontyngencji,
- odkrywanie zależności dla atrybutów numerycznych w postaci równań.

Pyt35: Opisać algorytm Apriori odzyskiwania asocjacji (ang. Knowledge mining) z danych uporządkowanych zbiorów umieszczonych w tabeli uzyskanej na przykład z bazy relacyjnej: