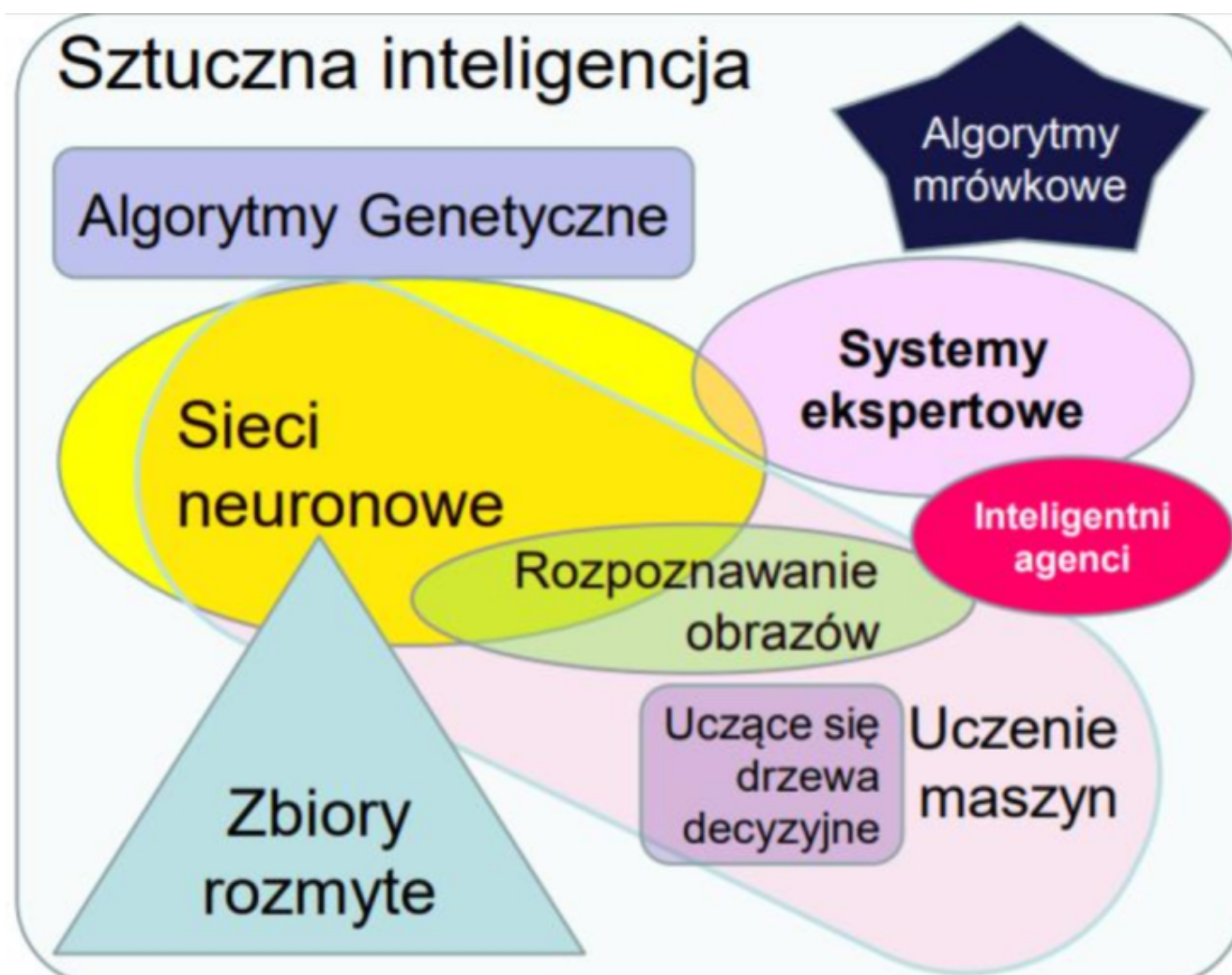


AI: Systemy Ekspertowe



Podstawową ideą symulacji kognitywnej jest konstrukcja algorytmów heurystycznych (algorytm heurystyczny może dostarczać akceptowalne rozwiązanie problemu, ale nie można przeprowadzić formalnego dowodu poprawności tego algorytmu) w celu symulacji ludzkich procesów umysłowo poznawczych (np. wnioskowania, rozwiązywania problemów, rozpoznawania obiektów, uczenia się) przez próbę odtworzenia sekwencji elementarnych kroków wykonywanych przez człowieka w trakcie tych czynności.

Model symulacji kognitywnej składa się z czterech podstawowych elementów:

- koncepcji przestrzeni stanów,
- rozwiązywania problemu przez heurystyczne przeszukiwanie (zwykle nie znamy algorytmu rozwiązania problemu),
- analizy celów i środków (jeśli dysponujemy funkcją oceny/kosztu wygenerowanych stanów, to możemy ją wykorzystać do generacji nowych stanów),
- metody redukcji problemu (zastąpieniu złożonego problemu przez sekwencję podproblemów).

Powstanie modeli konekcyjnistycznych, w których zjawiska umysłowe modeluje się jako emergentne procesy zachodzące w sieciach złożonych z prostych elementów składowych. Do modeli tych zalicza się sztuczne sieci neuronowe. Model neuronu zaproponowali W.S. McCulloch i W. Pitts w 1943 r.

Propozycja podejścia opartego na logice zamiast modeli symulacji ludzkiego myślenia (J. McCarthy). McCarthy rozwinął język Lisp. W ramach tego podejścia na początku lat siedemdziesiątych pojawił się nurt badań związanych z zastosowaniem logiki pierwszego rzędu wSI. Jego rezultatem było stworzenie Prologu. Nurt ten zaowocował następnie powstaniem modelu programowania logicznego z ograniczeniami.

To podejście jest konkurencyjne do symulacji kognitywnej i podejścia opartego na logice. System rozwiązuje konkretny problem po dostarczeniu wszelkiej możliwej wiedzy ekspertów nt. problemu. Wiedza powinna być sformalizowana w postaci danych, a system powinien być wyposażony w uniwersalny mechanizm wnioskujący oraz moduł weryfikacji poprawności działania. To podejście wykorzystuje się w modelu reprezentacji wiedzy w postaci reguł wnioskowania oraz w regułowych systemach ekspertowych.

Systemy ekspertowe

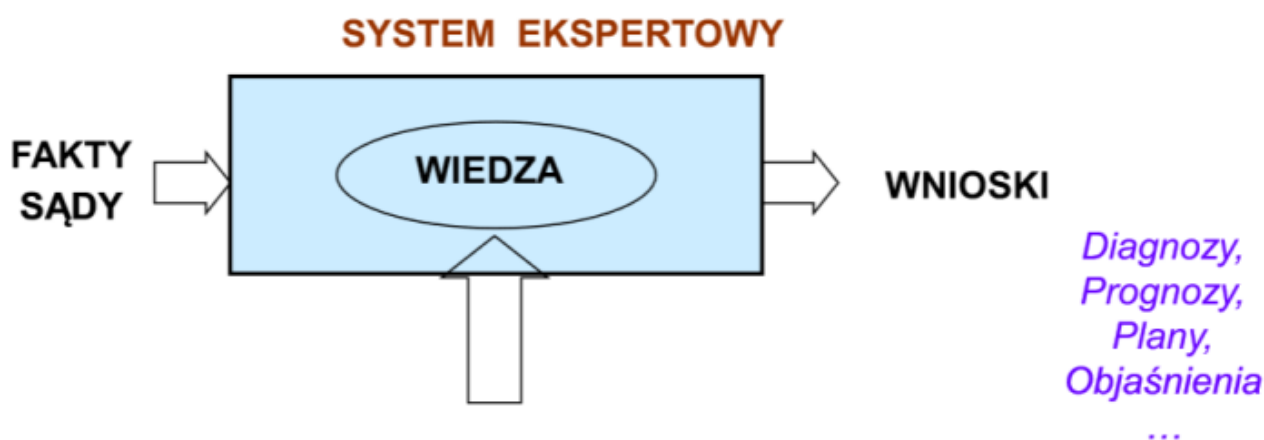
Wiedza na temat dziedziny jest bardzo cenna, gdyż pozwala określić racjonalne działania.

- Ogrom zgromadzonej wiedzy oraz subtelne nierazależności pomiędzy jej składnikami czynią jednakkorzystanie z niej procesem bardzo żmudnym.
- Efektywne wykorzystanie wiedzy wymaga więcej narzędzi wspomagających, które pozwolą ogarnąć całość oraz wyciągać skomplikowane wnioski.
- Narzędziami umożliwiającymi kompleksowe wykorzystywanie wiedzy są systemy ekspertowe.
- Podstawę systemów ekspertowych stanowią: język logiki oraz ontologia wyższa świata.

System ekspertowy (doradczy):

system informatyczny wykorzystujący procedury wnioskowania do rozwiązywania problemów, które są na tyle trudne, że wymagają znaczącej ekspertyzy specjalistów.

„wykorzystuje wiedzę eksperta do rozwiązywania takich problemów, które wymagają inteligencji człowieka -eksperta„



Ogólnie przyjmuje się, że system ekspertowy jest to system informatyczny, który ma na celu zastąpienie pracy eksperta w danej dziedzinie.

System ekspertowy może wykonywać następujące funkcje: doradzanie, analizowanie, klasyfikowanie, udzielanie informacji, diagnozowanie, uczenie się, gromadzenie doświadczeń, prognozowanie,

planowanie, nauczanie, testowanie, czyli wszystkie te zadania, które mógł realizować człowiek - ekspert.

System ekspertowy powinien:

1. „rozumieć” i analizować problem;
2. wybierać fakty niezbędne do wykonania ekspertyzy;
3. udzielać porad;
4. wyjaśniać sposób rozwiązania danego problemu.

Wiedza (niezbędna, by zapewnić odpowiedni poziom ekspertyzy), wraz z procedurami wnioskowania stanowi model ekspertyzy, posiadanej przez najlepszych specjalistów w danej dziedzinie.

Wiedza systemu eksperckiego: fakty i heurystyki.

Fakty: powszechnie akceptowane przez specjalistów.

Heurystyki: informacja subiektywna, która charakteryzuje proces oceny przez określonego specjalistę.

Heurystyki: intuicyjne domysły, przypuszczenia, zdroworozsądkowe zasady postępowania.

Poziom ekspertyzy to funkcja rozmiaru i jakości bazy wiedzy danego systemu.

Przesłanki stosowania systemów ekspertowych

- Koszty: w dłuższym okresie czasu są znacznie tańsze, pomagają w rozwiązywaniu problemów wymagających najbardziej specjalistycznej (najdroższej) wiedzy.
- Brak ekspertów w wielu dziedzinach.
- SE pracują szybciej, nie męczą się, są bardziej niezawodne niż ludzie.
- Konsekwentne, obiektywne, dokładne.
- Zawsze do dyspozycji (nie strajkują!).
- Analiza dużych ilości danych wymaga komputera.

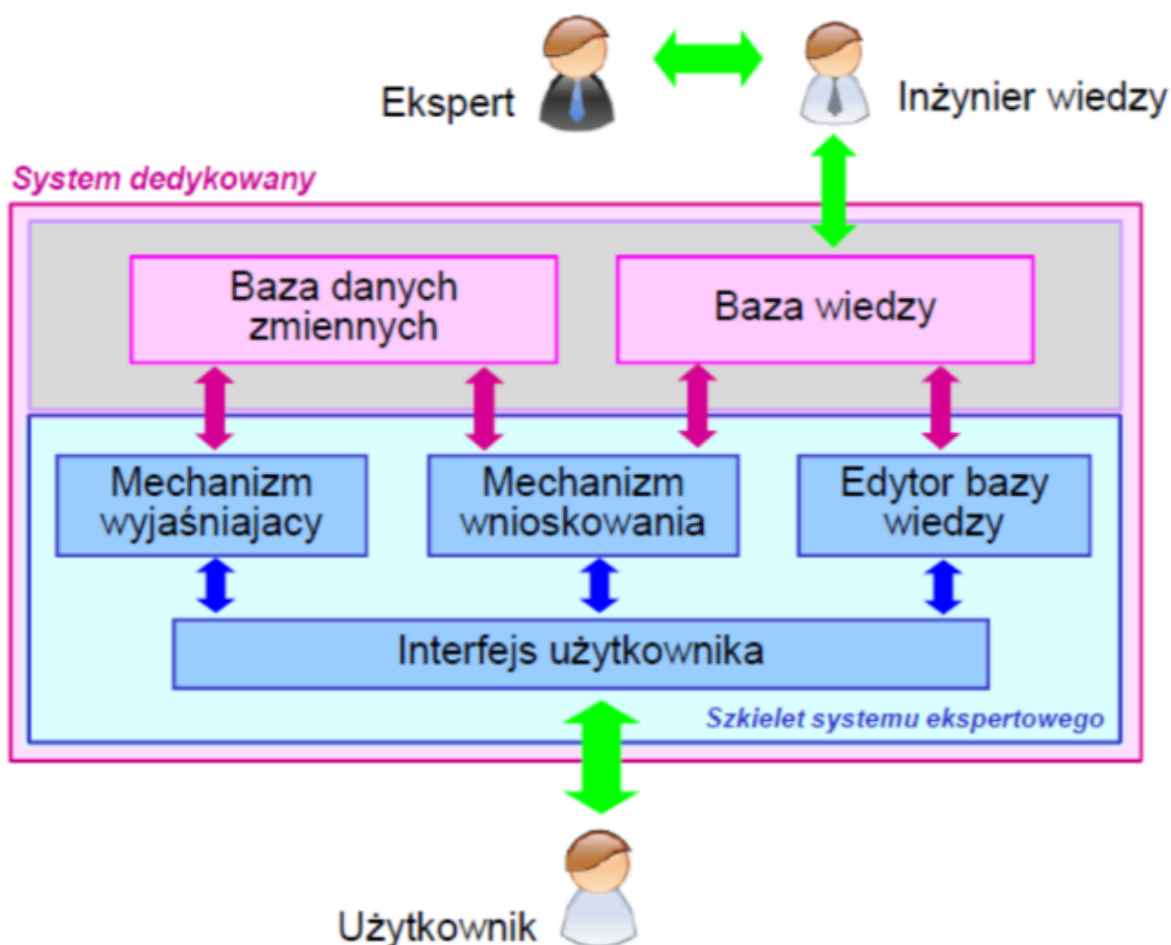
SE: systemy oparte na wszystkich sposobach reprezentacji wiedzy, najczęściej w postaci reguł produkcji.

Rodzaje systemów eksperckich

Interpretacyjne	dedukują opisy sytuacji z obserwacji lub stanu czujników np. rozpoznawanie mowy, obrazów
Predykcyjne	wnioskują o przyszłości na podstawie danej sytuacji np. prognozowanie pogody, rozwój choroby
Diagnostyczne	ocena systemu na podstawie obserwacji (wykrywanie wad) np. elektronika, mechanika
Kompletowania	konfiguruje obiekt biorąc pod uwagę istniejące ograniczenia np. konfiguracja systemu komputerowego

Planowanie	podejmuje określone działania do osiągnięcia celu
Monitorowanie	porównywanie zachodzących zjawisk z ograniczeniami nałożonymi na nie np. reakcje w elektrowniach atomowych
Sterowanie	kierowanie działaniem systemu; w skład wchodzi interpretowanie, predykcja, naprawa i monitorowanie zachowania się obiektu
Poprawianie	podaje sposób postępowania w przypadku nieprawidłowego funkcjonowania obiektu, którego system dotyczy
Naprawy	ustala kolejność wykonywania napraw uszkodzonych obiektów
Instruowania	systemy doskonalenia zawodowego dla studentów

Struktura systemu ekspertowego



Elementy systemu ekspertowego

Szkielet systemu w skład którego wchodzi:

- Mechanizm wnioskowania (maszyna wnioskująca) Główny składnik systemu ekspertowego wykonujący cały proces rozumowania w trakcie rozwiązywania problemu postawionego przez użytkownika;
- Interfejs użytkownika Umożliwia udzielanie informacji systemowi (TELL) oraz zadawanie pytań,

- odbieranie od systemu odpowiedzi i wyjaśnień (ASK);
- Edytor bazy wiedzy Pozwala modyfikować wiedzę zawartą w systemie, umożliwiając tym samym jego rozbudowę;
- Mechanizm wyjaśniający Element interfejsu użytkownika, który umożliwia uzyskanie informacji:
 - dlaczego system udzielił takiej, a nie innej odpowiedzi,
 - dlaczego system zadał użytkownikowi określone pytanie.
- Baza wiedzy

Deklaratywna postać wiedzy ekspertów z danej dziedziny zapisana za pomocą wybranego sposobu reprezentacji wiedzy, najczęściej reguł lub ram.

- Baza danych zmiennych

Pamięć robocza przechowującą pewne fakty wprowadzone w trakcie dialogu z użytkownikiem; Umożliwia odtworzenie sposobu wnioskowania systemu i przedstawienie go użytkownikowi za pomocą mechanizmu wyjaśniającego. Pozyskiwaniem wiedzy od ekspertów zajmują się inżynierowie wiedzy. Pozyskiwanie wiedzy to zwykle długi i żmudny proces, gdyż wiedza stosowana przez ekspertów ma charakter intuicyjny i praktyczny, często trudny do werbalizacji.

Wiedza w systemie eksperckim

opisy lub fakty, to podstawowe cechy i pojęcia wyrażone jako elementarne składniki zdań zapisanych w jakimś języku. Ich zadanie to identyfikacja i rozróżnianie obiektów i klas. Zawarte są w nich także wszelkiego typu reguły lub algorytmy wykorzystywane do interpretacji danych wejściowych,

relacje, to obraz zachodzących zależności i skojarzeń pomiędzy elementami wchodzącymi w skład opisów (faktami),

procedury, to mechanizmy jakie podlegają relacje i fakty.

Rodzaje baz wiedzy

baza tekstów (text base) - występuje w niej naturalna strukturalizacja informacji, wynikająca np. z ułożenia alfabetycznego. Przykładem mogą być słowniki, które zawierają wiedzę ogólną z danej dziedziny.

baza danych (data base) - zawiera uporządkowane sprecyzowane, szczegółowe informacje. Typ danych jest często numeryczny, a operacje na niej wykonywane są zdeterminowane.

baza reguł (rule base) - zawiera zbiór zależności pomiędzy obiektami występującymi w danej dziedzinie.

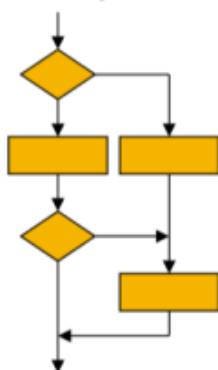
baza modeli (model base) - w niej zawarte są modele matematyczne występujące w danej dziedzinie. Możemy wyróżnić trzy typy: deterministyczne, niedeterministyczne, wartości oczekiwane.

baza wiedzy zdroworozsądkowej (common sense knowledge base) - zbiór potencjalnych, racjonalnych zachowań człowieka, reguł definiujących sposoby podejmowania decyzji.

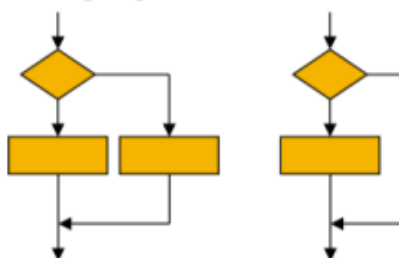
Metody reprezentacji wiedzy

Podstawowe reprezentacje wiedzy: - formuły $x = f(a, b, \dots)$ $s = v.t$

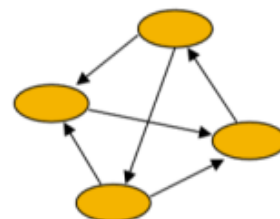
- *procedury*



- *reguły*



- *struktury*



Wiedza proceduralna mówi „jak” rozwiązać problem - określa zbiór procedur, których działanie reprezentuje wiedzę o danej dziedzinie .

Wiedza deklaratywna mówi „czym” rozwiązać problem - określa zbiór specyficznych dla danej dziedziny faktów, stwierdzeń, reguł..

Symboliczne:

- Metody bazujące na zastosowaniu logiki
 - logika konwencjonalna: rachunek zdań, rachunek predykantów, metoda rezolucji, itp..
 - logika niekonwencjonalna (rozmyta, wiele wartościowa)
 - metody wykorzystujące zapis stwierdzeń,
 - metody wykorzystujące systemy regułowe (wektory wiedzy),
- Metody z wykorzystaniem sieci semantycznych,
- Metody oparte na ramach,
- Metody używające modeli obliczeniowych.

Niesymboliczne:

- Sztuczne sieci neuronowe
- Algorytmy genetyczne

Inne:

- język naturalny
- tablice decyzyjne
- drzewka decyzyjne
- sieci Bayesa
- scenariusze
- metody stosowane w projektowaniu SI np. UML

- ontologie,
- zbiory przybliżone
- XML
- Logiczne agenty

Założenia symbolicznych metod reprezentacji wiedzy

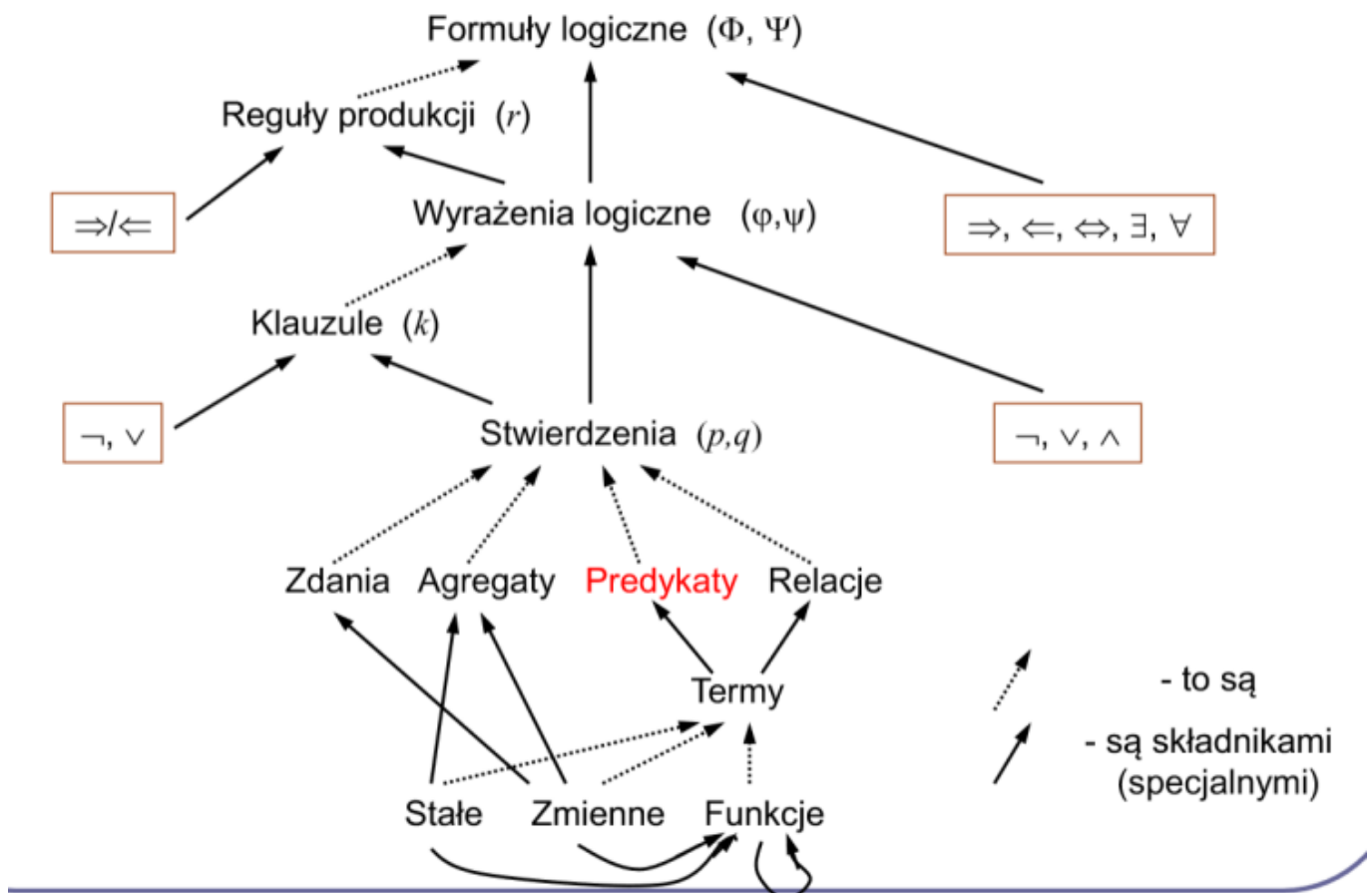
1. możliwe jest skonstruowanie modelu reprezentującego system inteligentny explicite,
2. wiedza w tych modelach powinna być reprezentowana w sposób symboliczny (np. w postaci grafów lub reguł; modele wiedzy w symbolicznej SI są konstruowane na podstawie modeli logiki, matematyki dyskretnej, teorii języków formalnych itd.),
3. (inteligentne) działania umysłowo-poznawcze można opisać przy użyciu formalnych operacji nad wyrażeniami i strukturami symbolicznymi należącymi do modelu wiedzy.

Metody symbolicznej reprezentacji można podzielić na dwie grupy.

1. Pierwsza obejmuje ogólne modele reprezentacji wiedzy i inteligentnych działań (np. podejście symulacji kognitywnej, podejście oparte na logice).
2. Druga obejmuje modele oparte na reprezentacji wiedzy, ale w ramach specyficznego aspektu czy też obszaru jej funkcjonowania (np. podejście oparte na reprezentacji wiedzy w postaci reguł wnioskowania, strukturalne modele reprezentacji wiedzy, podejście wykorzystujące gramatyki generatywne do reprezentacji wiedzy)

Modele reprezentacji wiedzy oparte o logikę

W podejściu tym przyjmuje się, że inteligentne systemy informatyczne powinny nie tyle być konstruowane jako symulatory heurystycznych reguł ludzkich procesów myślowych, ale raczej przy użyciu sformalizowanych modeli wnioskowania logicznego. System w tym podejściu nie realizuje algorytmu będącego sekwencją rozkazów/instrukcji określających jak wykonać obliczenia, aby uzyskać rozwiązanie problemu (paradygmat imperatywny). System powinien ograniczyć się jedynie do określenia, jakie pożądane własności powinno mieć rozwiązanie problemu (czyli co jest rozwiązaniem problemu, a nie jak rozwiązać problem). Samo rozwiązanie problemu powinno być przeprowadzone przez uniwersalne oprogramowanie podstawowe, które wykorzystuje programowanie logiczne lub programowanie funkcyjne (paradygmat deklaratywny). W programowaniu logicznym specyfikacja pożądanych własności rozwiązania problemu ma charakter zbioru twierdzeń/reguł wyrażonych w języku logiki. Użytkownik formułuje pytanie/hipotezę, a system SI sprawdza drogą wnioskowania logicznego przy użyciu twierdzeń/reguł zapisanych w programie, czy jest to prawda. Przykładem takiego podejścia jest Prolog.



Opis języka logiki

Wyrażenia języka składają się z term i formuł.

1. termy - encje, obiekty

- symbole stałych (zwykle z początku alfabetu):
 $a, b, \dots$
- symbole zmiennych
- n-argumentowe symbole funkcyjne $f(t_1, \dots, t_n)$,
gdzie $t_1, \dots, t_n$ 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.
 $P(t_1, \dots, t_n)$, gdzie t_n to termy dla $n \geq 1$
- 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 najslabiej wiążących - spójników i kwantyfikatorów: $\neg \forall \exists \wedge \vee \Rightarrow$

Modele reprezentacji wiedzy oparte o logikę predykatów

Logika predykatów

Podstawowe pojęcia logiki predykatów są symbol term (konstant) i predykat. Predykat P jest to „funkcja”, która ma tylko dwie wartości $P = \{\text{true}, \text{false}\}$

Term może być konstantą, np. twierdzenie ojciec (adam, rafał) lub zmienna predykat ojciec (X,Y)

Logika predykatów jest zbudowana z trzech komponentów:

- rachunek predykatów pierwszego rzędu,
- kilka twierdzeń, przedstawionych w terminach języka RP,
- reguły wnioskowania.

Rachunek predykatów pierwszego stopnia jest to zbiór logicznych formuł (LF), które mogą zawierać następujące operatory

Symbol	Znaczenie symbolu
x, y, z	Zmienne
a, b, c	Konstanty
f, g, h	Funkcje
p, q, r	Predykaty
$\exists x$	<u>Kwantor istnienia</u>
$\forall x$	<u>Kwantor uniwersalności</u>
$A \wedge B$	Operator 'I' (konjunktja)
$A \vee B$	Operator 'Lub' (dyzjunktja)
$\neg A$	Operator 'Nie' (negacja)
$A \rightarrow B$	Operator postępowania (implikacja)
$A \leftrightarrow B$	Dwustronna implikacja (warunek 'wtedy i tylko wtedy')

Przykłady zdań zapisanych w logice

Piotr nie jest wysoki

$$\neg \text{wysoki}(\text{Piotr})$$

Na stole leży owoc

$$\forall x \text{ na}(x, \text{stół}) \Rightarrow \text{owoc}(x)$$

Liczba całkowita może być parzysta lub nieparzysta

$$\forall x \text{ całkowita}(x) \Rightarrow (\text{parzysta}(x) \vee \text{nieparzysta}(x))$$

Wszyscy studenci są zdolni

$$\forall x \text{ student}(x) \Rightarrow \text{zdolny}(x)$$

Każdy na świecie student jest zdolny

$$\forall x \text{ student}(x) \wedge \text{zdolny}(x)$$

Tylko co niektórzy student jest zdolny

$$\exists x \text{ student}(x) \wedge \text{zdolny}(x)$$

Każdy delfin jest ssakiem

$$\forall x \text{ delfin}(x) \Rightarrow \text{ssak}(x)$$

Istnieje ssak, który znosi jaja

$$\exists x \text{ ssak}(x) \wedge \text{znosi_jaja}(x)$$

Każdy ogrodnik lubi słońce

$$\forall x \text{ ogrodnik}(x) \Rightarrow \text{lubi}(x, \text{słońce})$$

Wszystkie czerwone grzyby są trujące

$$\forall x (\text{grzyb}(x) \wedge \text{czerwony}(x)) \Rightarrow \text{trujący}(x)$$

Żaden czerwony grzyb nie jest trujący

$$\neg \exists x \text{ czerwony}(x) \wedge \text{grzyb}(x) \wedge \text{trujący}(x)$$

Są dokładnie dwa czerwone grzyby

$$\exists x \forall y \text{ grzyb}(x) \wedge \text{czerwony}(x) \wedge \text{grzyb}(y) \wedge \text{czerwony}(y) \wedge \neg(x = y) \wedge \forall z (\text{grzyb}(z) \wedge \text{czerwony}(z)) \Rightarrow ((x = z) \vee (y = z))$$

Każdy kogoś lubi

$$\forall x \exists y \text{ lubi}(x, y)$$

Ktoś lubi każdego

$$\exists x \forall y \text{ lubi}(x, y)$$

Możesz kochać niektórych ludzi cały czas

$$\exists x \forall t (\text{osoba}(x) \wedge \text{czas}(t)) \Rightarrow \text{można_kochać}(x, t)$$

Można kochać niek którego człowieka przez pewien okres czasu

$$\forall x \exists t (\text{osoba}(x) \wedge \text{czas}(t)) \Rightarrow \text{można_kochać}(x, t)$$

Istnieje student, który interesuje się co najmniej dwoma różnymi przedmiotami wykładanymi na jego wydziale

$$\exists x (\text{student}(x) \wedge \exists y \exists z (y \neq z \wedge \text{wykładany}(y, \text{wydział}(x)) \wedge \text{wykładany}(z, \text{wydział}(x)) \wedge \text{interesuje_się}(x, y) \wedge \text{interesuje_się}(x, z)))$$

Rozpatrzmy twierdzenie: każdy człowiek ma ojca.

a) wprowadzamy predykat *ojciec* (x, y)

$$\forall x \exists y = \text{człowiek}(x) \rightarrow \text{ojciec}(y)$$

b) zmienimy porządek dla kwantów uniwersalności $\forall$ i istnienia $\exists$:

$$\exists y \forall x = \text{Dla wszystkich ludzi } (x) \rightarrow \text{ojciec}(y)$$

To oznacza, że wszyscy ludzie mają jednego ojca – Boga

Z tego przykłady wynika, że w predykanie ma znaczenie porządek operatorów

c) w predykacji możliwe jest również użycie symboli funkcyjnych

$$\forall x (p(f(x)) \rightarrow p(x))$$

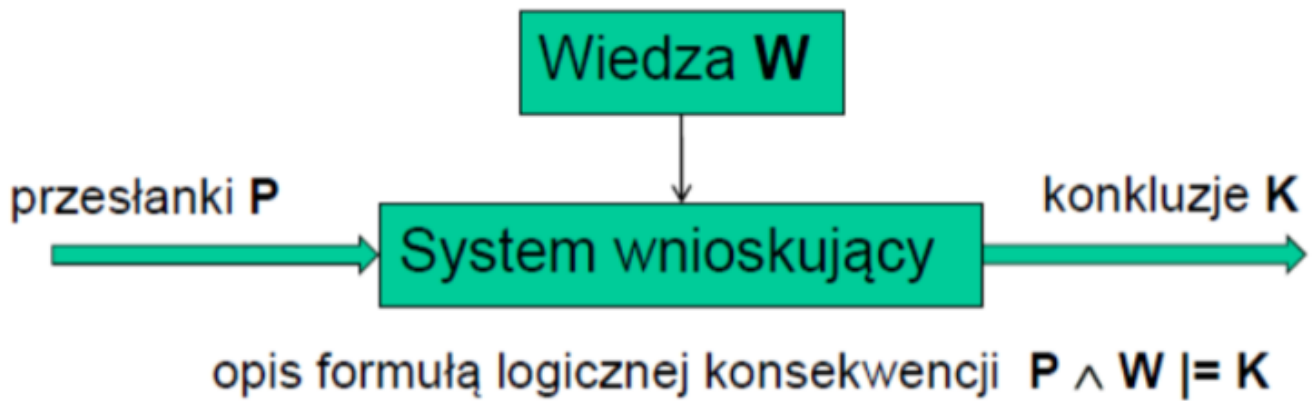
Oznaczmy $f(x) = \text{Ojciec } x$ i $p(x) = x \text{ jest człowiek}$, wtedy mamy twierdzenie:

Wszystkie stworzenia, których ojcem jest człowiek są ludźmi

Modele reprezentacji wiedzy oparte o reguły

Automatyczne wnioskowanie -podstawy dedukcji

Podstawowy schemat systemu dedukcji wykorzystującego związek logicznej konsekwencji



- Z prawdziwości P wynika prawdziwość K (logicznie)
- W systemach uczących się (dedukcyjnych) : P=dane trenujące, W=wiedza zastana, K=wiedza nabyta
- Reguła zachowania prawdy: z prawdziwych P mamy prawdziwe K
- Stanowi podstawę systemów eksperckich (ekspertowych) oraz systemów automatycznego dowodzenia twierdzeń

Rozumowanie dedukcyjne

Rozumowanie dedukcyjne jest **prostym wyciąganiem wniosków**

●w odróżnieniu od rozumowania indukcyjnego nie wymaga tworzenia nowych twierdzeń czy pojęć

■ Przykład 1:

{ P: każdy prostokąt jest czworokątem;
W: suma kątów czworokąta wynosi 360 stopni }
⇓
{ K: suma kątów prostokąta wynosi 360 stopni }

■ Przykład 2:

{ P: Jadzia jest z Polski i ma polski paszport;
W: Polak mówi po polsku }
⇓
{ K: Jadzia mówi po polsku }

Rozumowanie indukcyjne

- Rozumowanie indukcyjne przebiega w kierunku odwrotnym – ze zbioru przesłanek (**K**) wyprowadzamy wniosek ogólny, teorię czy prawo (nie zawsze prawdziwe) dotyczące **P** (nowa wiedza)
 - np. z wielokrotnych obserwacji wynika, że przed deszczem jaskółki latają nisko nad ziemią; formułujemy wniosek - jeżeli jaskółki nisko latają, to znaczy, że będzie padać deszcz (niestety, nie musi to być prawda)
- Na podstawie obserwacji faktów (prawdy) szukamy przyczyn (prawdziwych?):
 - z prawdziwości **K** nie musi wynikać prawdziwość **P** (problem PRAWDZIWOŚCI reguł)
 - z fałszywości **K** mamy fałszywe **P** (fakt niepraktyczny)
- Indukcja logiczna to sposób rozumowania polegający na wyprowadzaniu nowych pojęć, twierdzeń lub sugerowaniu możliwości zaistnienia nowych faktów na podstawie intuicyjnej analizy wejściowych przesłanek
- We wnioskowaniu indukcyjnym obok wiedzy nabytej i informacji trenującej znaczenie ma także algorytm wnioskujący

Regułowa reprezentacja wiedzy

Pierwszy system tego typu oparty był na tzw. produkcjach. Podstawowymi komponentami tego modelu są: pamięć reguł oraz pamięć robocza. Pamięć reguł odpowiada pamięci długotrwałej, a pamięć robocza – krótkotrwałej. W pamięci długotrwałej wiedza zapisana jest w postaci produkcji/reguł, które mają postać:

JEŚLI zachodzi pewien warunek, TO wykonaj pewną akcję.

Akcja ma charakter konkluzji albo jest czynnością oddziałującą na środowisko. Pamięć robocza zawiera informację dynamiczną, zmieniającą się w czasie. Jest to informacja odnośnie środowiska, która jest sprawdzana przez część warunkową reguł. Gdy warunek którejś z reguł jest spełniony, to reguła ta jest stosowana, a jej akcja wykonana (konkluzja tej reguły zapisywana jest w pamięci roboczej lub system wykonuje pewną czynność, np. włącza jakieś urządzenie lub zmienia kierunek ruchu robota).

System oparty na produkcjach znany jest przede wszystkim w swojej szczególnej wersji – modelu ekspertowego systemu regułowego.

Reguły są intuicyjnie najprostszą metodą reprezentacji wiedzy

Istnieje wiele formatów zapisu reguł. Koncepcja jest jednak zwykle ta sama:



„Działanie” reguły odbywa się według wywodzącego się z logiki schematu *modus*

$$\frac{p \rightarrow q}{p} \quad q$$

Jeżeli p implikuje logicznie q oraz p jest prawdziwe to q jest również prawdziwe.

Zapis reguł :

Z wykorzystaniem symbolu implikacji:

przesłanki $\rightarrow$ konkluzje

W postaci zbliżonej do instrukcji warunkowej:

if przesłanki then konkluzje

W postaci odwrotnie zapisanej instrukcji warunkowej: **konkluzje if przesłanki**

przesłanka

IF x_1 is F_1 and/or x_2 is F_2 and/or x_n is F_n

konkluzja

THEN y_1 is P_1 and y_2 is P_2 and y_n is P_n

$x_1, x_2, \dots, x_n, y$ - obiekty

$F_1, F_2, \dots, F_n, P_1, \dots, P_n$ - wartości

Jeśli (IF) przesłanki (warunki) To (THEN) konkluzja

Jeśli (IF) przesłanki (warunki) To (THEN) działanie (produkcja)

Jeśli (IF) przesłanki (warunki) THEN konkluzja1 ELSE konkluzja2

Przykład reguły:

IF Wynik_testu_STUDENTA is >1350 THEN Test_studenta is Zdany

IF jest ładna pogoda THEN pójść na spacer

IF pożar wykryty THEN włącz alarm

IF nie działa monitor OR płyta główna jest spalona THEN obsługa komputera nie jest możliwa Można taką regułę zamienić na zestaw dwóch równoważnych reguł nie zawierających funktora alternatywy:

IF nie działa monitor THEN obsługa komputera nie jest możliwa

IF płyta główna jest spalona THEN obsługa komputera nie jest możliwa

Rodzaje reguł

Rozróżniamy dwa rodzaje reguł ze względu na sposób uzyskiwania ostatecznych konkluzji w procesie

wnioskowania:

- 1) reguły proste - takie, które mają postać wniosków pośrednich
- 2) reguły złożone - takie, które umożliwiają bezpośrednie wyznaczenie wniosków przez system.

Przykład reguły złożonej:

IF spełnione są wszystkie warunki podpisanej umowy THEN towar można odebrać z magazynu

Reprezentacja reguł

Reguły mogą być reprezentowane:

- jako zbiór warunków JEŻELI przesłanka TO wniosek,
- w postaci drzewa decyzyjnego.

Drzewo decyzyjne jest to graficzna postać reguł.

Drzewo decyzyjne obrazuje powiązania pomiędzy faktami.

Przykład Drzewo decyzyjne diagnozy rozruchu samochodu



Przykład

IF padał deszcz THEN ulica jest mokra
 IF jechała polewaczka THEN ulica jest mokra
 IF jest ładna pogoda THEN pójde na spacer
 IF jestem zmęczony THEN nie pójde na spacer
 IF pada deszcz THEN wezmę parasol
 IF pożar wykryty THEN włącz alarm

Ulica jest mokra IF padał deszcz

ulica = „mokra” if deszcz= „padał”

Heurystyki

wszelkie prawa, kryteria, zasady i intuicje, które umożliwiają wybranie najbardziej efektywnych kierunków działania prowadzącego do osiągnięcia danego celu

Heurystyka jest rozumiana w zagadnieniach AI jako:

- praktyczna strategia poprawiająca efektywność rozwiązywania złożonych problemów
- prowadzi do rozwiązania wzdłuż najkrótszej, najbardziej prawdopodobnej drogi
- podaje proste kryterium wyboru kierunków postępowania
- powinna umożliwić uniknięcie badania tzw. ślepych uliczek i wcześniejsze wykorzystanie zdobytych w trakcie badania informacji.

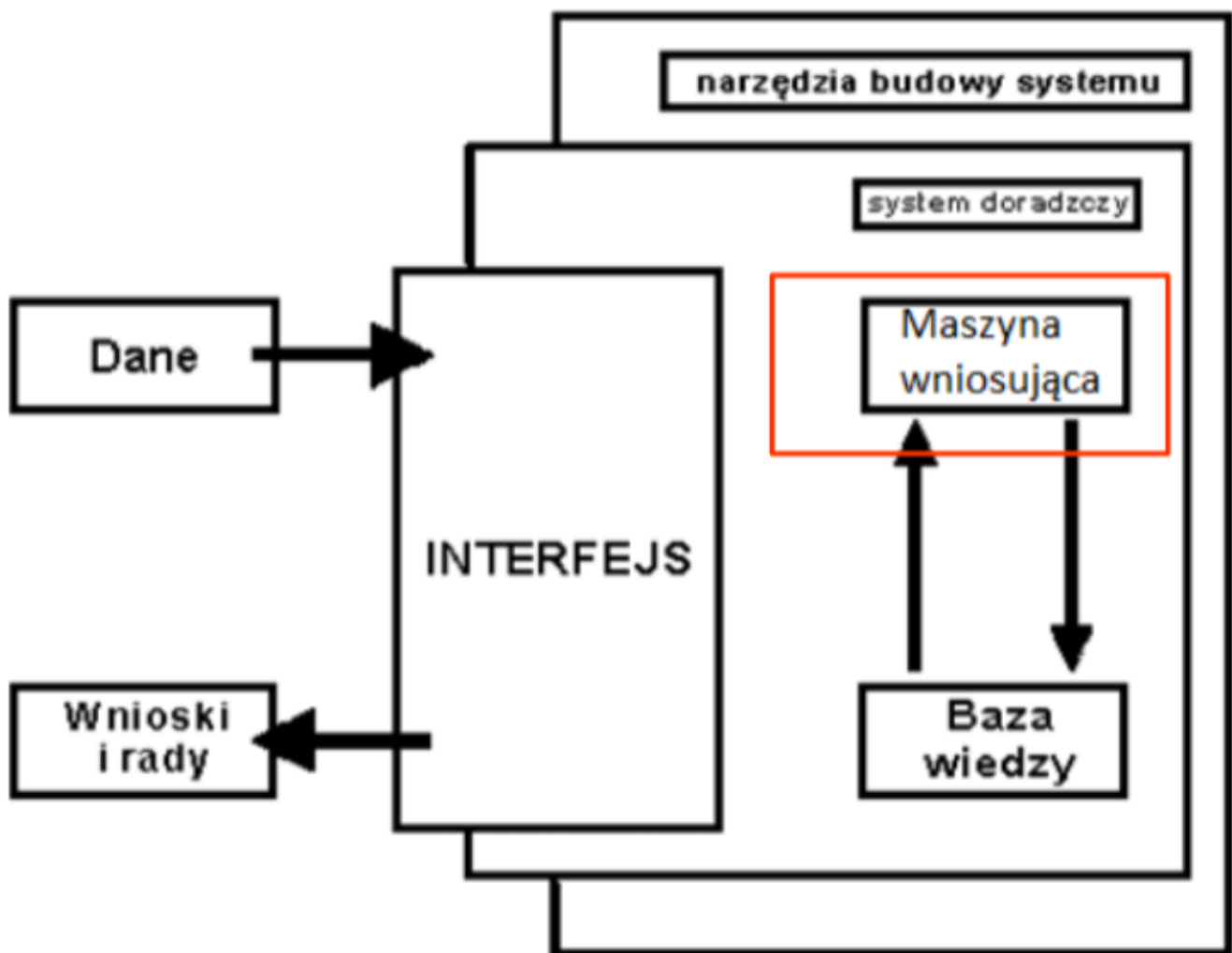
Szczególne znaczenie mają one przy rozwiązywaniu problemów o dużej złożoności obliczeniowej (gdzie dokładny algorytm zawodzi), zwłaszcza przy rozwiązywaniu problemów NP-zupełnych.

Działanie maszyny wnioskującej (kontrolera wyvodu)

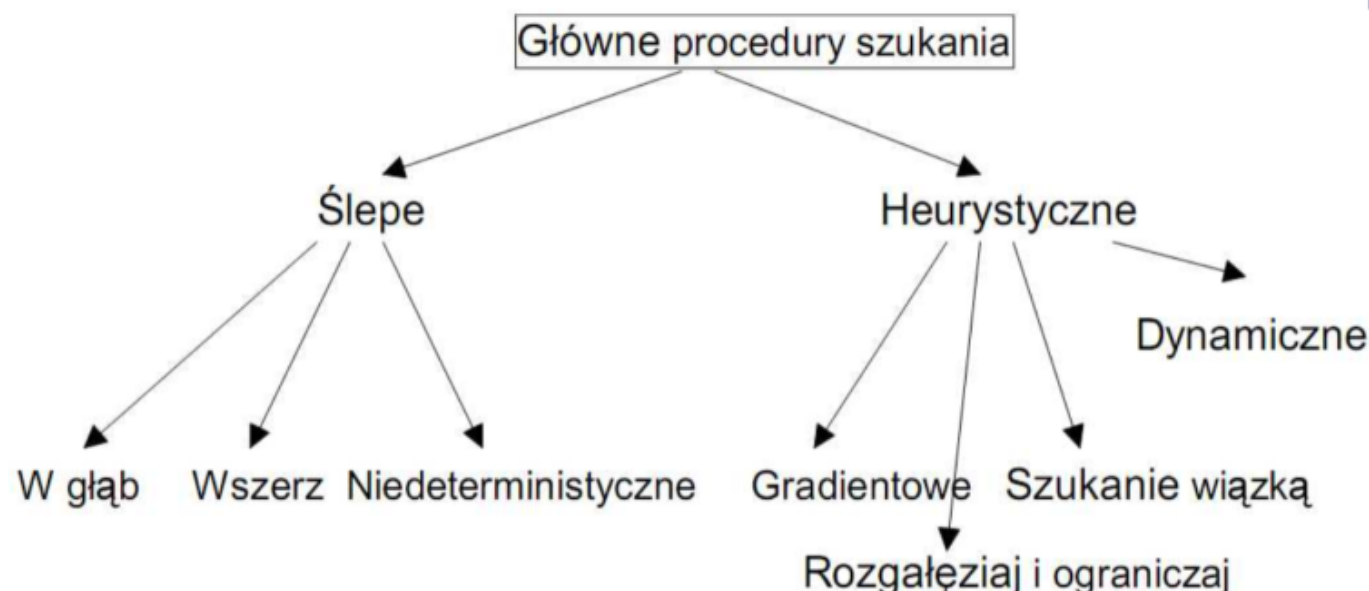
W oparciu o heurystyki działają maszyny wnioskujące. które dają odpowiedź na następujące pytania:

jak zacząć proces wnioskowania?

- którą regułę zastosować, gdy jest kilka reguł aktywnych?
- jak znaleźć następne reguły?



Metody przeszukiwania grafów (drzewka)



Metody poszukiwań rozwiązań

W procesie poszukiwania rozwiązań wymagane jest najczęściej określenie

- zbioru stanów przestrzeni rozwiązywanego problemu (w tym stanu początkowego i zbioru stanów końcowych)
- zbioru operatorów przekształcających te stany (operatory stosowane do stanów generują nowe stany –powstają)

Rozwiązanie polega na określeniu ciągu operatorów przekształcających stan początkowy w stan końcowy

Korzysta się z pewnych algorytmów, czyli strategii realizujących poszczególne metody przeszukiwań (od metod ślepych po heurystyczne)

Wyszukiwanie wiedzy w bazie wiedzy

Głównym zadaniem strategii przeszukiwania jest wybór operatorów, określających gałęzie grafu przestrzeni stanów możliwe do wyboru podczas rozwiązywania danego problemu. Większość wyników teoretycznych dotyczących strategii przeszukiwania uzyskano dla drzew, a więc szczególnego typu grafów. Parametry charakteryzujące przeszukiwane grafy to:

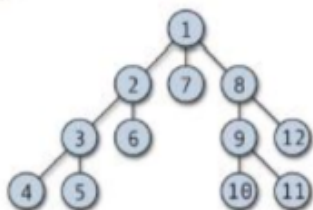
- głębokość węzła,
- stopień rozgałęzienia węzła (liczba potomków),
- stopień rozgałęzienia gałęzi (ograniczenie liczby gałęzi wychodzących z jednego węzła).

Strategie przeszukiwania grafów:

- w głąb
- z powracaniem

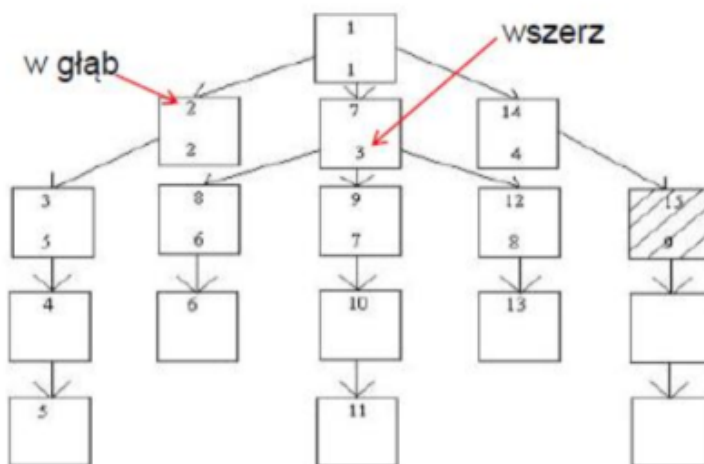
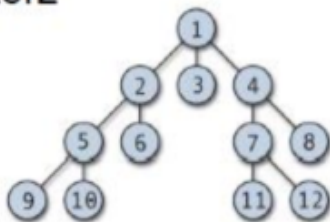
- wszerz
- zachłanna
- "najpierw najlepszy"

- w głąb



- z powracaniem

- wszerz



zachłanna -badane są nowe węzły i najbardziej obiecujący z nich jest wybierany do dalszej ekspansji; wykorzystuje lokalną optymalizację i nie są w niej możliwe powroty do żadnego przodka aktualnie badanego węzła (strategia nieodwracalna)

Wnioskowanie (Maszyna wnioskująca)

Maszyna wnioskująca (kontroler wyводу, *inference engine*) jest to program prowadzący wnioskowanie.

Na podstawie faktów i reguł maszyna wnioskująca generuje nowe fakty będące odpowiedziami na pytania użytkownika.

Podstawową zasadą wnioskowania jest reguła *modus ponens*,

$$[(p \rightarrow q) \wedge p] \rightarrow q$$

Jeżeli z P wynika Q i P jest prawdziwe, to prawdziwe jest Q

Modus ponendo ponens (sposób potwierdzający przez potwierdzenie) – tautologia rachunku zdań i analogiczny schemat wnioskowania dedukcyjnego.

Tautologia rachunku zdań mówi, że jeśli uznajemy prawdziwość poprzednika prawdziwej implikacji, to musimy uznać też prawdziwość jej następnika:

Zgodnie z tą regułą:

Wnioskowanie może być prowadzone:

- **bez hipotezy** - generowanie nowych faktów tak długo, aż zostaną odpalone wszystkie możliwe reguły;
Pytanie „*Co wynika z XXX ?*”
- **z hipotezą** - weryfikacja (udowodnienie) konkretnej hipotezy na podstawie faktów zawartych w bazie wiedzy;
Pytanie „*Czy jest prawdą, że XXX ?*”

Modus tollens - Modus tollens (modus tollendo tollens, łac. sposób zaprzeczający przy pomocy zaprzeczenia) – wnioskowanie logiczne, reguła logiki mówiąca,

$$[(p \Rightarrow q) \wedge \neg q] \Rightarrow \neg p$$

że jeśli z P wynika Q oraz że Q jest fałszywe, to musimy zaakceptować też fałszywość P.

Przykład:

1) Jeżeli prawdą jest że nie jest wilgotno, i jednocześnie gdyby padał deszcz, to byłoby wilgotno, więc nie pada deszcz

2) Jeżeli prawdą jest że pada deszcz i wiedząc, że nie może jednocześnie padać deszcz i być susza więc nie ma suszy

Algorytmu wnioskowania (Maszyna wnioskująca)

Podstawowe algorytmy wnioskowania:

- w przód;
 - wnioskowania z hipotezą lub bez.
- wstecz;
 - tylko wnioskowania z hipotezą.
- mieszane;
 - rodzaj wnioskowania przełączany w zależności od reguły.

Algorytm wnioskowania

- Znalezienie kolejnej reguły możliwej do odpalenia,
 - kolejność wyboru reguł zależy od przyjętej strategii
- Odpalenie reguły,
- Dodanie uzyskanych wniosków do bazy wiedzy,
- Koniec gdy:
 - znaleziono odpowiedź (wnioskowanie z hipotezą),
 - gdy wszystkie reguły zostały odpalone (bez hipotezy).

Baza reguł:

$R_1: P \wedge Q \Rightarrow S$
 $R_2: S \wedge T \Rightarrow U$
 $R_3: R \Rightarrow T$
 $R_4: S \wedge R \Rightarrow V$

Baza faktów

$BF = \{ P, Q, R \}$

Wnioskowanie w przód

Można odpalić reguły R_1 i R_3 .

Odpalenie reguły R_1 ;

Wynik $BF = \{ P, Q, R, S \}$

Można odpalić reguły R_3 i R_4

Odpalenie reguły R_3 ;

Wynik $BF = \{ P, Q, R, S, T \}$

Można odpalić reguły R_2 i R_4

Odpalenie reguły R_2 ;

Wynik $BF = \{ P, Q, R, S, T, U \}$

Można odpalić regułę R_4

Odpalenie reguły R_4 ;

Wynik $BF = \{ P, Q, R, S, T, U, V \}$

Wnioskowanie wstecz

Hipoteza: V

Czy $V \in BF$? → **NIE**

Hipoteza jest wnioskiem R_4

Należy sprawdzić jej przesłanki:

Czy $S \in BF$? → **NIE**

Hipoteza robocza: S

Hipoteza robocza jest wnioskiem R_1

Należy sprawdzić jej przesłanki:

Czy $P \in BF$? → **TAK**

Czy $Q \in BF$? → **TAK**

Odpalić regułę R_1

Hipoteza robocza S została

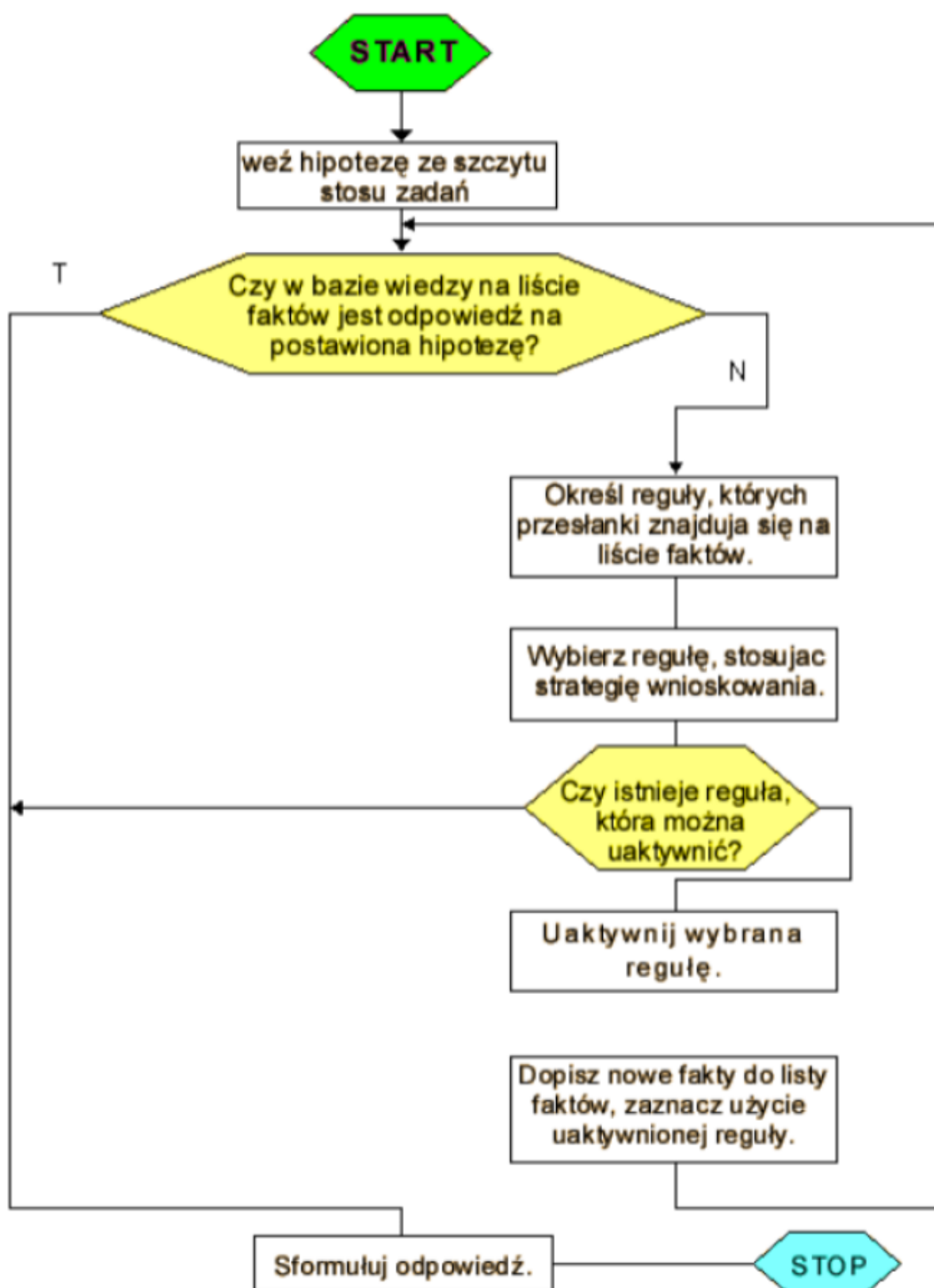
udowodniona, $BF = \{ P, Q, R, S \}$

Czy $R \in BF$? → **TAK**

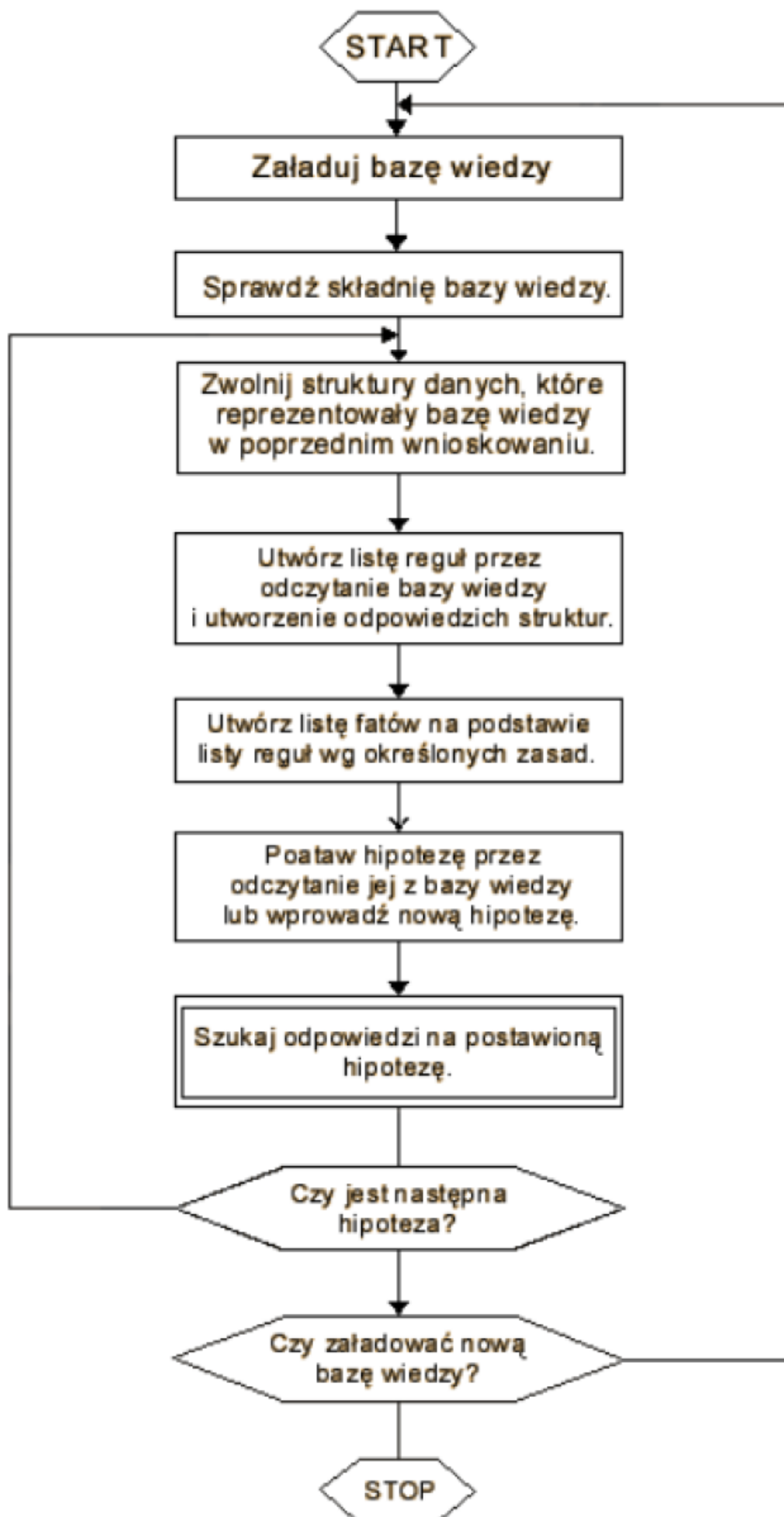
Odpalić regułę R_4

Hipoteza V została udowodniona, $BF = \{ P, Q, R, S, V \}$

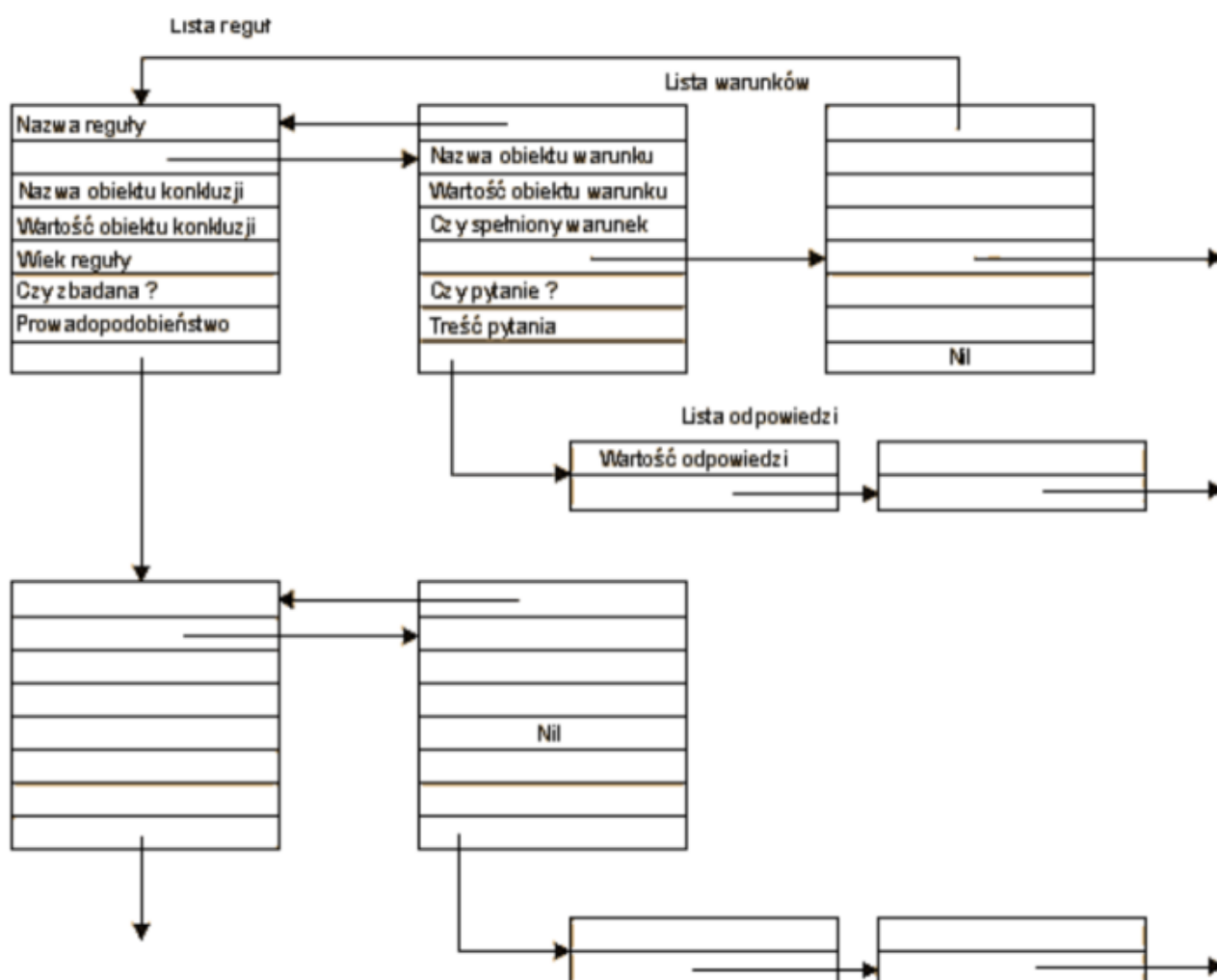
Algorytm maszyny wnioskującej w przód



Algorytm maszyny wnioskującej wstecz



Lista jako - schemat struktury wiedzy

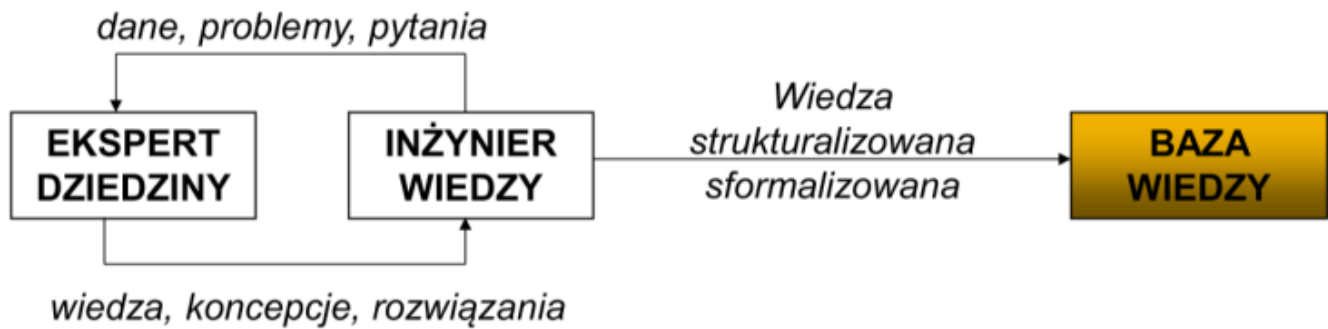


Pozyskiwanie wiedzy

Pozyskiwanie wiedzy w ścisły sposób jest związane z zagadnieniami uczenia się maszyn. Do zadań systemów wyposażonych w umiejętność uczenia się należy zaliczyć:

- formułowanie nowych pojęć,
- wykrywanie nie znanych dotychczas prawidłowości w danych,
- tworzenie reguł decyzyjnych,
- przyswajanie nowych pojęć i struktur drogą uogólniania przykładów i analogii,
- modyfikowanie,
- uogólnianie i precyzowanie danych,
- zdobywanie wiedzy drogą konwersacji z ludźmi,
- uogólnianie obserwacji dokonanych sztucznymi zmysłami (tj. czujnikami),
- generowanie wiedzy zrozumiałej dla człowieka.

Jedną z metod pozyskiwania wiedzy jest prowadzenie dialogu z ekspertem.



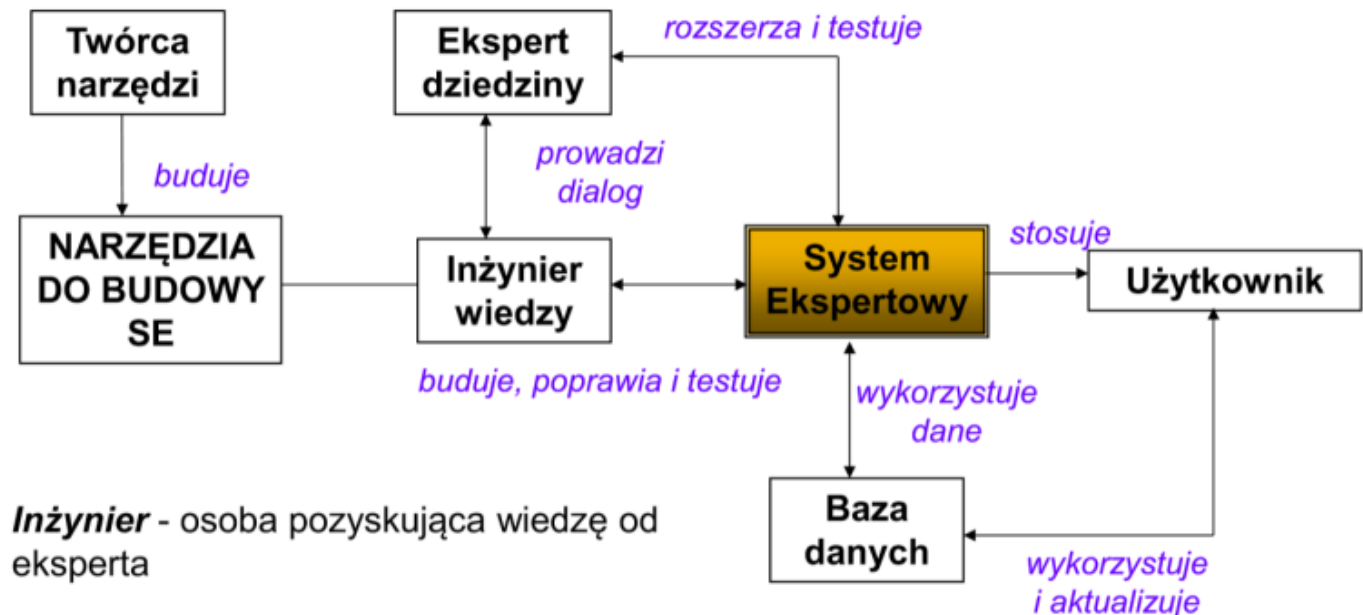
Oto przykładowy zestaw pytań zadawanych ekspertowi przez inżyniera wiedzy:

- jak różni się omawiany problem od prototypowych problemów danej dziedziny ?
- jakich rodzajów danych wymaga problem ?
- jakie typy rozwiązań są adekwatne dla problemu ?
- czy problem może być zdekomponowany na nie oddziaływujące na siebie podproblemy ?
- jakie dziedziny wiedzy są potrzebne do rozwiązania problemu ?
- w jaki sposób odbywa się wyjaśnianie rozwiązania problemu ?

Metody Pozyskiwania Wiedzy

- manualne
 - wywiad
 - analiza protokołów
 - analizowanie problemu
 - kwestionariusze
 - obserwacja eksperta w miejscu pracy
 - „burza mózgów”
 - wykorzystanie symulacji komputerowej
- Metody półautomatyczne
 - bezpośrednie zapisywanie wiedzy
 - pozyskiwanie wiedzy na podstawie instrukcji
 - pozyskiwanie wiedzy na podstawie analogii
 - pozyskiwanie wiedzy na podstawie przykładów
- Metody automatyczne
 - uczenie indukcyjne
 - uczenie dedukcyjne

Role twórców systemów eksperckich



Trudności pozyskania wiedzy od eksperta

- Czasami nie ma ekspertów danego problemu
- Nie satysfakcjonujący poziom wiedzy eksperta
- Ekspert nie wyraża zgody lub nie okazuje chęci do współpracy
- Niektórzy eksperci nie potrafią wyartykułować swojej wiedzy, doświadczenia.

Zasady pozyskania wiedzy od eksperta

1. Wybrać eksperta posiadającego satysfakcjonujący poziom wiedzy.
2. Umówić się na pierwsze spotkanie. Spotkanie powinno być nieformalne i odbywać się w zrelaksowanej atmosferze.
3. Na pierwszym spotkaniu: prosto i jasno wyjaśnić co chcemy osiągnąć; nakreślić plan, formę i cel następnych spotkań.
4. Inżynier wiedzy powinien zrozumieć problem i jego otoczkę oraz poznać terminologię dziedziny problemu.
5. Pytać eksperta o wyjaśnienia jak również obserwować jego pracę podczas rozwiązywania konkretnego problemu.
6. Być cierpliwym, unikać krytyki, pytać o niezrozumiałe kwestię.
7. Skrupulatnie notować przebieg każdej konsultacji z ekspertem.
8. Inżynier wiedzy buduje model bazy wiedzy eksperta, nie swojej.

Zalety i wady ekspertyzy sztucznej i ekspertyzy naturalnej

Ekspertyza naturalna	Ekspertyza sztuczna
Wady:	Zalety:
- tracąca na wartości z upływem czasu - trudna do przeniesienia - trudna w dokumentacji - nie dająca się przewidzieć - kosztowna	- łatwa do przeniesienia - łatwa w dokumentacji - zgodna z bazą wiedzy - dostępna
Zalety:	Wady:
- twórcza - adaptacyjna - wykorzystanie zmysłów - szeroki zakres - wiedza zdrowego rozsądku	- nie inspirująca - wymaga wprowadzenia wiedzy - wejścia symboliczne - wąski zakres - wiedza przetwarzana w sposób mechaniczny

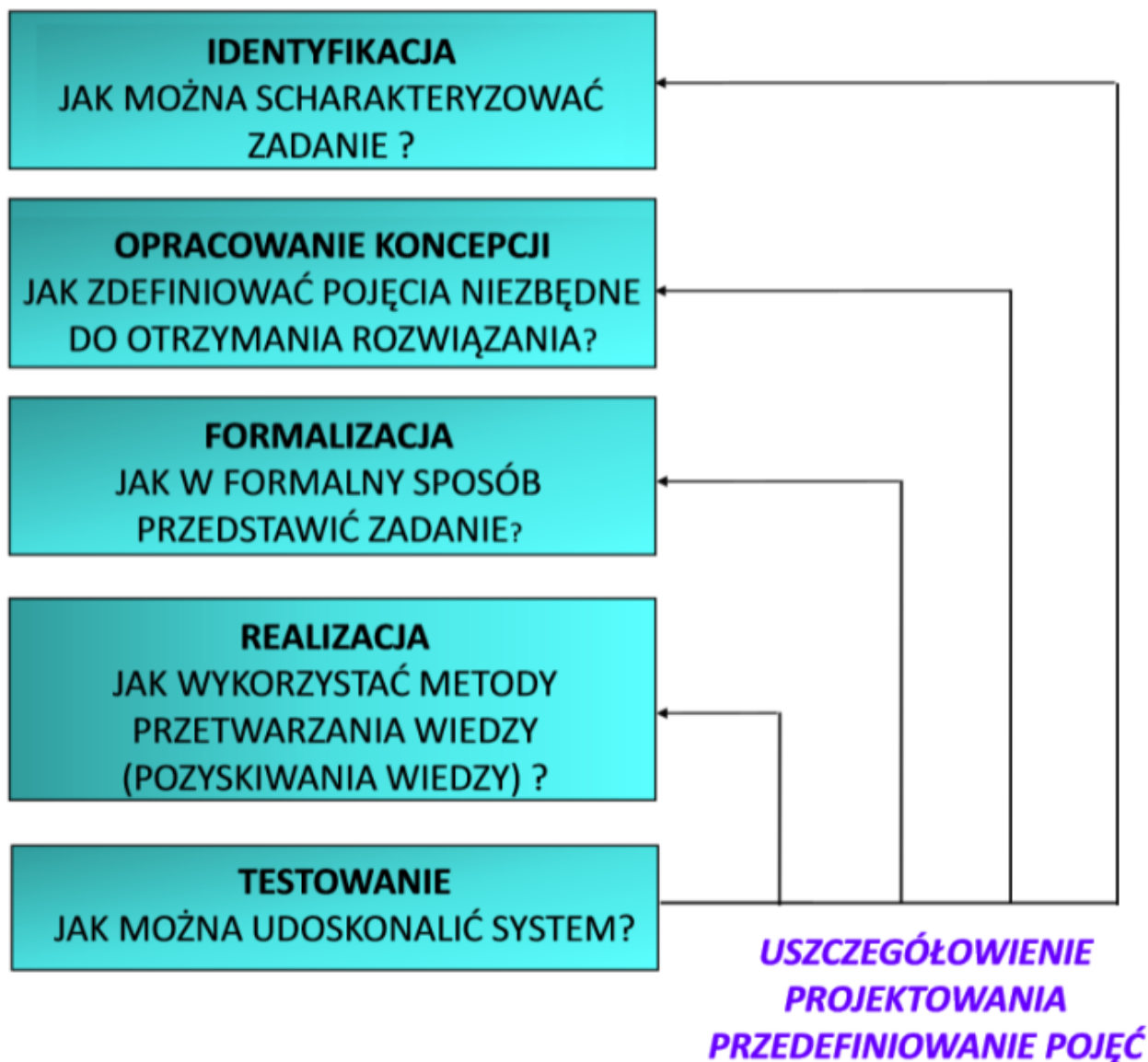
Porównanie terminologii klasycznego przetwarzania danych i inżynierii wiedzy

Konwencjonalne przetwarzanie danych	Inżynieria wiedzy
Programista, analityk systemów	Inżynier wiedzy
Program, System	System Ekspertowy
Baza danych	Baza wiedzy
Reprezentacja i użycie danych	Reprezentacja i użycie wiedzy
Algorytmy	Heurystyki
Efektywna manipulacja dużymi bazami danych	Efektywna manipulacja dużymi bazami wiedzy

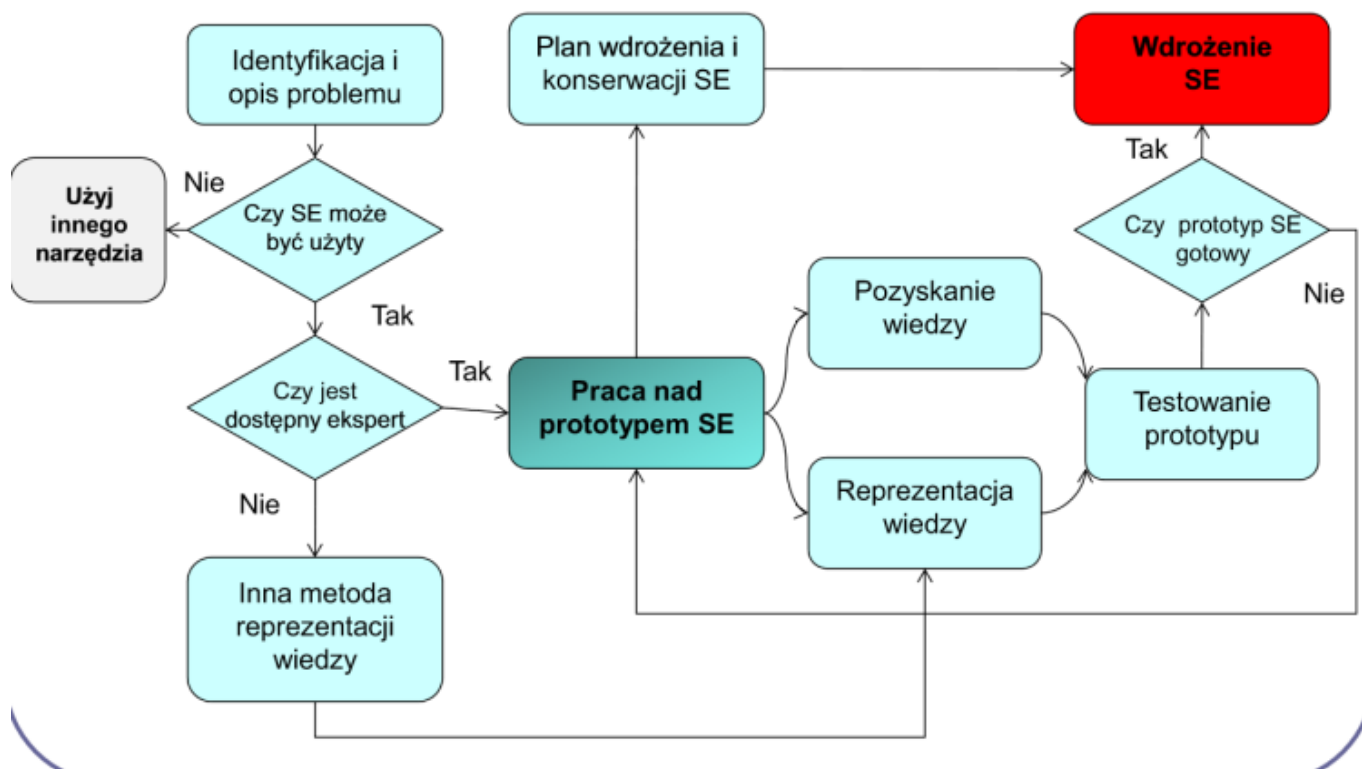
Etapy tworzenia SE

- Analiza problemu - oceny, czy budowa SE dla danego problem ma sens.

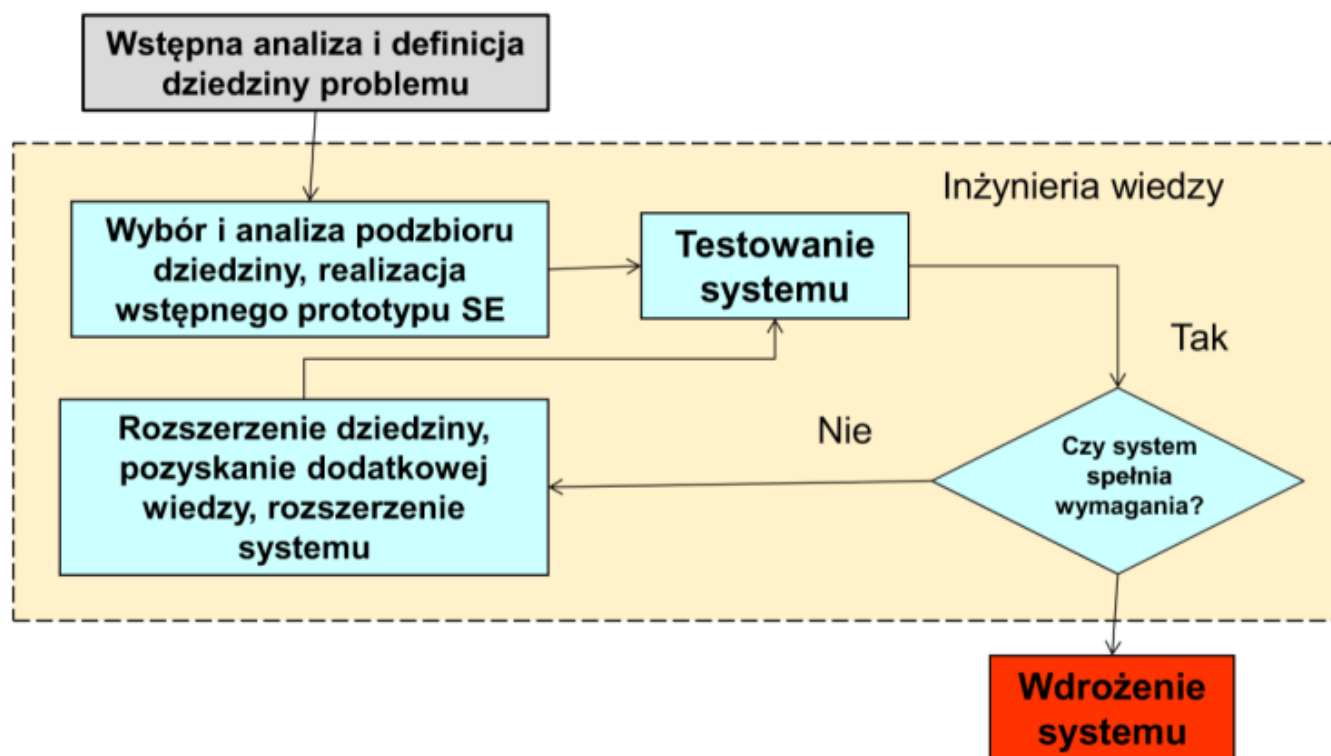
- Specyfikacja systemu - szczegółowego określenia funkcji i oczekiwań.
- Akwizycja wiedzy - zgromadzenia, wydobywania z ekspertów i organizacji wiedzy.
- Wybór metody reprezentacji wiedzy i narzędzi do budowy systemu.
- Realizacja systemu - utworzenia bazy wiedzy, reguł wnioskowania, systemu wyjaśniającego rozumowanie i prowadzenia dialogu z użytkownikiem.
- Weryfikacja i testowanie systemu.



Etapy budowy SE



Przyrostowy model cyklu rozwojowego systemu ekspertowego



Wybrane narzędzia do tworzenia SE

1. CLIPS, monk.monk.ca/clips/clipsdown.htm
2. Acquire, www.aiinc.ca/acquire/acquire.shtml
3. DecisionPro, www.vanguardsw.com/decisionpro
4. EZ-Xpert, www.ez-xpert.com
5. XpertRule www.attar.com/pages/info_kb.htm
6. ExSys, www.exsys.com/productmain.html
7. PS-Shell, www.aitech.com.pl/index_sphinx.htm

Zalety systemów eksperckich

- **Dostępność wiedzy eksperckiej** – umożliwiają dostęp do wiedzy specjalistycznej nawet tam, gdzie nie ma ekspertów.
- **Powtarzalność** – system zawsze udziela odpowiedzi według tych samych reguł, eliminując ludzką zmienność.
- **Szybkość podejmowania decyzji** – mogą analizować dane i podejmować decyzje szybciej niż człowiek.
- **Przechowywanie wiedzy** – wiedza ekspertów jest zapisana w systemie, dzięki czemu nie ginie z ich odejściem.
- **Wspomaganie pracy ekspertów** – mogą służyć jako narzędzie wspomagające decyzje, nie zastępując ich całkowicie.
- **Dostępność 24/7** – systemy działają bez przerwy, niezależnie od pory dnia i stanu zdrowia.

Wady systemów eksperckich

- **Ograniczona elastyczność** – działają tylko w ramach zaprogramowanych reguł; nie radzą sobie dobrze z sytuacjami niestandardowymi.
- **Koszt stworzenia** – stworzenie i utrzymanie dobrego systemu eksperckiego może być czasochłonne i kosztowne.
- **Zależność od jakości wiedzy** – jakość decyzji zależy od tego, jak dobrze została zebrana i zapisana wiedza.
- **Brak intuicji i zdrowego rozsądku** – systemy nie mają „czucia” kontekstu, którym często kierują się ludzie.
- **Trudności z aktualizacją** – modyfikacja reguł może być skomplikowana, szczególnie w dużych systemach.
- **Brak zdolności uczenia się** – klasyczne systemy eksperckie nie uczą się same; wymagają ręcznego wprowadzania zmian.

Zastosowania systemów eksperckich

- **Medycyna**

- Diagnostyka chorób (np. system MYCIN, DXplain)
- Wspomaganie decyzji klinicznych
- Dobór leczenia i leków

- **Finanse i bankowość**

- Ocena zdolności kredytowej
- Wykrywanie oszustw finansowych
- Doradztwo inwestycyjne

- **Inżynieria i przemysł**

- Diagnostyka usterek w maszynach
- Planowanie produkcji
- Monitorowanie jakości procesów

- **Prawo**

- Wspomaganie analizy przypadków prawnych
- Doradztwo prawne w oparciu o bazę przepisów i orzeczeń

- **Rolnictwo**

- Doradztwo w zakresie nawożenia, ochrony roślin i upraw
- Wczesne wykrywanie chorób roślin i zwierząt

- **Wojsko i obronność**

- Analiza sytuacji taktycznych
- Wspomaganie dowodzenia i decyzji w czasie rzeczywistym

- **Edukacja**

- Systemy inteligentnego nauczania (Intelligent Tutoring Systems)
- Diagnoza błędów ucznia i personalizacja materiałów

- **Zarządzanie i logistyka**

- Wspomaganie decyzji w zarządzaniu projektami
- Optymalizacja tras dostaw i zarządzania łańcuchem dostaw

- **Informatyka i bezpieczeństwo IT**

- Wykrywanie zagrożeń i intruzów
- Automatyczne diagnozowanie awarii systemów

- **Geologia i geofizyka**

- Poszukiwanie złóż surowców naturalnych
- Ocena ryzyka sejsmicznego i geologicznego

From:

<https://wiki.ostrowski.net.pl/> - **Kacper's Wiki**

Permanent link:

https://wiki.ostrowski.net.pl/doku.php?id=notatki:ai_sys_ekspertowe&rev=1748002144

Last update: **2025/05/23 14:09**