Wstęp do inteligencji komputerowej – zajęcia nr 2

Jarosław Stańczak
jarek@wsisiz.edu.pl
WSISiZ

Systemy ekspertowe.

Systemy ekspertowe

System ekspertowy (SE) – w najprostszym ujęciu jest to pakiet oprogramowania, który emuluje proces podejmowania decyzji przez eksperta w danej dziedzinie. Jego wyższość nad ekspertem (który zazwyczaj bierze udział w jego tworzeniu) polega na tym, że może brać pod uwagę zdecydowanie więcej faktów, przesłanek i reguł zgromadzonych w **bazie wiedzy** systemu niż człowiek oraz może to robić znacznie szybciej. System ekspertowy może służyć również do sterowania obiektami, nad którymi człowiek nie jest w stanie zapanować z uwagi na wymagany krótki czas reakcji.

Systemy ekspertowe

W związku z funkcjami wypełnianymi przez systemy ekspertowe możemy je podzielić na:

- doradcze (wspomagające decyzje człowieka),
- autonomiczne (samodzielnie podejmujące decyzje),
- krytykujące (oceniające sytuację i reakcję ludzką na taką sytuację).

Systemy ekspertowe
wady i zalety stosowania

Zalety:

- powtarzalna, niezależna od „nastroju” eksperta
- łatwo do udokumentowania
- może zostać objaśniona na bazie posiadanej wiedzy i reguł wnioskowania
- możliwe bywa uzyskanie odpowiedzi na nietypowe pytania, które mogą być podstawą nowej wiedzy
- system można zastosować w wielu miejscach jednocześnie
- dane można wykorzystywać dowolnie długo z możliwością uzupełniania wiedzy
- można stosować go w połączeniu z innymi systemami SI i tradycyjnymi.

Wady:

- stworzenie SE wymaga ogromnego nakładu pracy
- często odwołuje się do bardzo wąskiej dziedziny wiedzy
- mechaniczne przetwarzanie wiedzy nie wywołuje skojarzeń i dodatkowych inspiracji, które występują u ekspertów i prowadzą do rozwoju dziedziny
- wymaga specyficznego systemu zapisu wiedzy i reguł.

Budowa typowego systemu ekspertowego

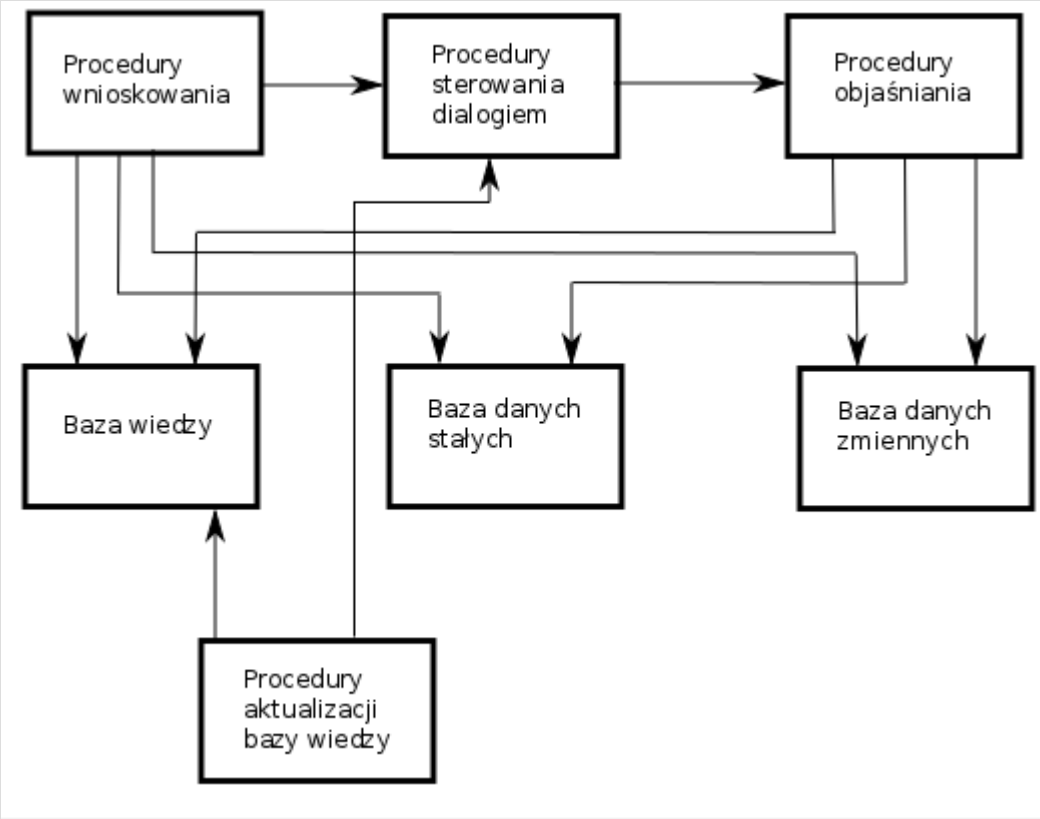
System ekspertowy składa się najczęściej z następujących elementów:

- modułu wnioskującego
- bazy wiedzy
- modułu objaśniającego
- interfejsu z użytkownikiem/twórcą.

Budowa typowego systemu ekspertowego

W wersji minimalnej system może nie posiadać **modułu objaśniającego**, choć to dzięki niemu można prześledzić, jak powstała decyzja systemu.
Moduł wnioskujący może być uniwersalny i może być niezależny od konkretnego zastosowania systemu (choć nie musi), przetwarza on fakty i reguły z bazy wiedzy według określonych, najczęściej uniwersalnych zasad logicznych tak, aby otrzymać odpowiedź na zadane pytanie.
Dodatkowo w trakcie używania i rozwijania systemu na pewno potrzebny jest **interfejs**, prosty przy zwykłym użyciu, bardziej skomplikowany przy rozbudowie bazy faktów (np. sprawdzający, czy tworzona baza wiedzy jest spójna).
Tworzeniem **bazy wiedzy** systemu ekspertowego zajmują się zazwyczaj inżynierowie wiedzy we współpracy z ekspertami w danej dziedzinie.

Budowa systemu ekspertowego



Schemat rozbudowanego systemu ekspertowego
na podst. J. Mulawka, „Systemy ekspertowe”.

Budowa systemu ekspertowego

Moduły: **„Procedury sterowania dialogiem”**, **„Procedury objaśniania”** oraz **„Procedury aktualizacji bazy wiedzy”** odpowiadają rozbudowanemu interfejsowi użytkownika/twórcy i umożliwiają kontakt z systemem.
Moduł **„Baza danych zmiennych”** jest pamięcią roboczą przechowującą pewne fakty i parametry wprowadzone w trakcie dialogu z użytkownikiem; baza ta umożliwia odtworzenie sposobu wnioskowania systemu i przedstawienie go użytkownikowi za pomocą mechanizmu wyjaśniającego.
Moduł **„Baza danych stałych”** przechowuje dane o dziedzinie, której dedykowany jest system ekspertowy, wprowadzone przez inżyniera wiedzy; dane te raczej nie są modyfikowane (lub dość rzadko) w trakcie działania systemu.
Moduł **„Baza wiedzy**” zawiera reguły wnioskowania systemu zapisane w postaci reguł lub ram.
Moduł **„Procedury aktualizacji bazy wiedzy”** umożliwia pozyskiwanie nowej wiedzy, modyfikację zbioru reguł (edytor bazy wiedzy).
Moduł **„Procedury objaśniania”** - objaśnia strategię wnioskowania i uzyskane wyniki.

Wiedza w SE

Wiedza w systemach SI może być przechowywana symbolicznie lub niesymboliczne (parametrycznie, np. sieci neuronowe, wyuczone strategie w uczeniu maszynowym). W SE stosowane jest zazwyczaj to pierwsze podejście.

Zdarzają się jednak SE z dodatkowymi modułami obliczeniowymi, zawierającymi wiedzę w postaci parametrycznej, mogą to być np. modele pewnych zjawisk lub obiektów, które są zadane w postaci tzw. czarnej skrzynki, czyli pewnego symulatora, który nie jest modelem opisującym zjawiska w postaci równań, algorytmów, a pewnych danych liczbowych, tabel, wyuczonych sieci neuronowych.

Wiedza w SE

Na wiedzę gromadzoną w SE w postaci nieparametrycznej składają się:

- fakty: opisy, reguły, algorytmy, dane;
- relacje: zależności i skojarzenia między faktami;
- procedury: opisują raczej procesy niż dane statyczne, np. prawa fizyki opisujące jakieś zjawisko.

Czasem dwa pierwsze sposoby zapisu wiedzy określa się łącznie jako reprezentację deklaratywną wiedzy, a trzeci nazywa reprezentacją proceduralną.

Na następnych slajdach zostaną pokazane sposoby zapisu wiedzy, niektóre dość skomplikowane. Generalnie problem zapisu wiedzy nie do końca jest jeszcze zbadany, gdyż nie wynaleziono uniwersalnej, jednoznacznej i jednocześnie łatwej do użycia metody jej zapisu.

Reprezentacja wiedzy w SE

Wiedzę w SE zapisuje się w postaci:

- rachunku zdań i rachunku predykatów (logika);
- metod wykorzystujących zapis stwierdzeń;
- metod wykorzystujących reguły tzw. wektory wiedzy;
- sieci semantycznych;
- metod opartych na tzw. ramach;
- metod używających tzw. modeli obliczeniowych;
- ...

Rachunek zdań

Wywodzi się wprost z **logiki matematycznej**.
Formuły składają się ze zdań mogących przyjmować wartość 0 lub 1, oznaczanych symbolami np. *P*, *Q*, ... i funktorów logicznych, najczęściej używane to: $\neg$, $\wedge$, $\vee$, $\Leftrightarrow$, $\Rightarrow$.

Rachunek zdań

przykłady

Znane tożsamości logiczne (prawa logiczne, tautologie) w rachunku zdań (wybór):

$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ – pierwsze prawo De Morgana

$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$ – drugie prawo De Morgana

$\neg(P \Rightarrow Q) \Leftrightarrow P \wedge \neg Q$ – prawo zaprzeczenia implikacji

$P \Rightarrow P$ – prawo tożsamości

$P \Leftrightarrow \neg\neg P$ – prawo podwójnego zaprzeczenia

$(P \vee Q) \Leftrightarrow (Q \vee P)$ – prawo przemienności alternatywy

$(P \wedge Q) \Leftrightarrow (Q \wedge P)$ – prawo przemienności koniunkcji

$((P \wedge Q) \wedge R) \Leftrightarrow ((P \wedge Q) \wedge R)$ – prawo łączności koniunkcji

$((P \vee Q) \vee R) \Leftrightarrow ((P \vee Q) \vee R)$ – prawo łączności alternatywy

$(P \wedge (Q \vee R)) \Leftrightarrow ((P \wedge Q) \vee (P \wedge R))$ – prawo rozdzielności koniunkcji

$(P \vee (Q \wedge R)) \Leftrightarrow ((P \vee Q) \wedge (P \vee R))$ – prawo rozdzielności alternatywy

$((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R))$ – prawo przechodniości implikacji

Rachunek zdań

sprawdzanie prawdziwości zdań logicznych

Np. $(P \wedge (Q \vee R)) \Leftrightarrow ((P \wedge Q) \vee (P \wedge R))$

P	Q	R	$Q \vee R$	$P \wedge (Q \vee R)$	$P \wedge Q$	$P \wedge R$	$((P \wedge Q) \vee (P \wedge R))$	Wynik
0	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	0	1
0	1	0	1	0	0	0	0	1
1	1	0	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0	1

Rachunek zdań

sprawdzanie prawdziwości zdań logicznych

Np. $(P \wedge (Q \vee R)) \Leftrightarrow ((P \wedge Q) \vee (P \wedge R))$

P	Q	R	$Q \vee R$	$P \wedge (Q \vee R)$	$P \wedge Q$	$P \wedge R$	$((P \wedge Q) \vee (P \wedge R))$	Wynik
0	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	0	1
0	1	0	1	0	0	0	0	1
1	1	0	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0	1

Ćwiczenie na zajęciach – proszę wykazać prawdziwość praw De Morgana i prawa rozdzielności alternatywy, używając tabelek jak wyżej.

Rachunek predykatów
pierwszego rzędu

Jest rozszerzeniem rachunku zdań o kwantyfikatory „dla każdego” ∀ i „istnieje takie że” ∃ oraz funkcje zdaniowe (**predykaty**) np. *W(x)*, mogące przyjmować wartości logiczne 0 i 1, *x* – argument predykatu nazywany jest **termem**.
Określenie”pierwszego rzędu” pochodzi od właściwości tego rachunku w którym kwantyfikatory mogą odnosić się tylko do argumentów, będących elementami pewnych zbiorów, nie mogą być natomiast ich podzbiorami. Kwantyfikatory nie mogą się również odnosić do funkcji zdaniowych.
Istnieją rachunki predykatów wyższych rzędów, w których tego typu ograniczenia nie występują, jednak ich przydatność do użycia w systemach komputerowych jest ograniczona potencjalnie ogromną złożonością obliczeniową.
Przy użyciu rachunku predykatów pierwszego rzędu można zapisać większość zasad matematyki oraz praw otaczającego świata, stąd ich częste wykorzystanie w SE.

Rachunek predykatów

Przykład (na podst. J. Mulawka, „Systemy ekspertowe”):

$$\forall_x \{ P(x) \Rightarrow \{ \forall_y [P(y) \Rightarrow P(f(x,y))] \wedge \neg \forall_y \{ Q(x,y) \Rightarrow P(y) \} \} \}$$

W praktyce często stosuje się formuły w postaci klauzul, które można otrzymać przez przekształcenie np. zdania powyżej do postaci:

$$\neg P(x_1) \vee \neg P(y_1) \vee P(f(x_1, y_1))$$

$$\neg P(x_2) \vee Q(x_2, g(x_2))$$

$$\neg P(x_3) \vee \neg P(g(x_3))$$

Stwierdzenia

Opisują pewne obiekty, zjawiska, efekty, czynności w postaci uporządkowanej trójki lub czwórki parametrów:

(<OBIEKT>, <ATRYBUT>, <WARTOŚĆ>)

lub

(<OBIEKT>, <ATRYBUT>, <WARTOŚĆ>,<CF>)

gdzie pewne wątpliwości budzić może tylko CF – stopień pewności (certainty factor), zawarty najczęściej w przedziale [-1,1], [0,1] lub [0,10].

Reguły

Reguły uzupełniają najczęściej stwierdzenia tak, aby opis danego obiektu lub zjawiska był pełny.
Reguły mogą być zapisane w różny sposób, lecz wszystkie metody zapisu zawierają przesłanki i konkluzje (lub działania, akcje):

JEŚLI przesłanka TO konkluzja
IF przesłanka THEN akcja
IF *P* jest *u* AND *Q* jest *v* THEN *R* jest *w*

lub skrótowo:

$$(P, u) \wedge (Q, v) \Rightarrow (R, w)$$

Wektory wiedzy

Są sformalizowanym zapisem reguł, w którym tworzona jest pewnego rodzaju macierz wszystkich możliwych przesłanek i konkluzji dla wszystkich reguł (macierz wiedzy). Macierz ta ma rozmiar $(P+K) * R$.
P – wszystkie możliwe (różne) przesłanki, K – wszystkie możliwe konkluzje, R – liczba reguł.
Macierz wypełniana jest trzema symbolami oznaczającymi:
- nie dotyczy: *
- warunek/konkluzja prawdziwy: 1
- warunek/konkluzja fałszywy: 0.

Wektory wiedzy

Przykład - reguły sterujące rozwiązaniem równania kwadratowego:

R1: JEŚLI $a = 0$ TO brak równania kwadratowego
R2: JEŚLI $a \neq 0 \wedge \Delta < 0$ TO równanie ma 0 rozwiązań
R3: JEŚLI $a \neq 0 \wedge \Delta = 0$ TO równanie ma 1 rozwiązanie
R4: JEŚLI $a \neq 0 \wedge \Delta > 0$ TO równanie ma 2 rozwiązania

Można zapisać następująco jako wektory w macierzy wiedzy:

	a=0	$\Delta<0$	$\Delta=0$	$\Delta>0$	brak r.k.	0 rozw.	1 rozw.	2 rozw.
R1	1	*	*	*	1	*	*	*
R2	0	1	*	*	*	1	*	*
R3	0	*	1	*	*	*	1	*
R4	0	*	*	1	*	*	*	1

Sieci semantyczne

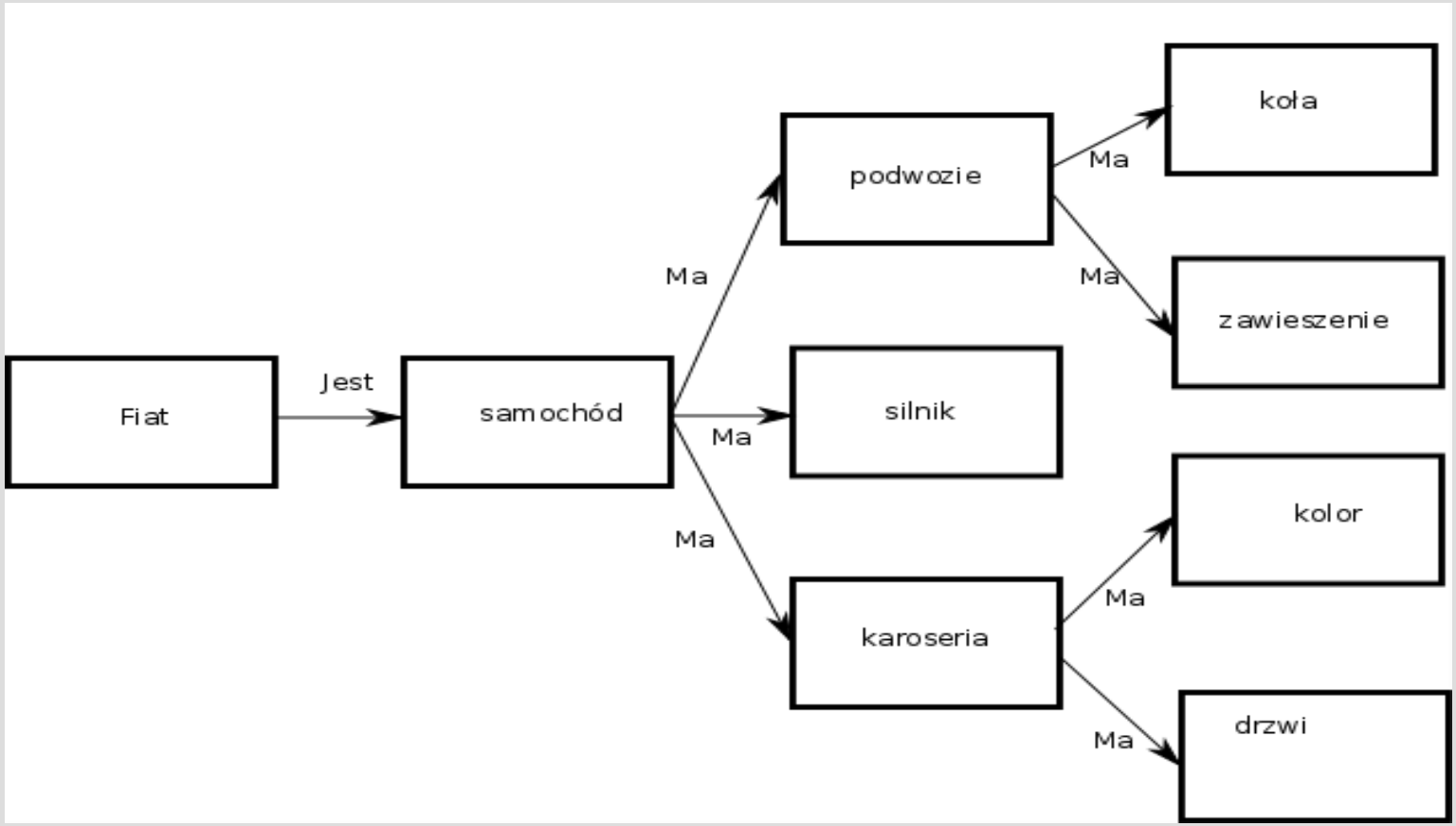
Są pewnym rodzajem zapisu **asocjacji**, skojarzeń pomiędzy obiektami, który zbliżony jest do modelu działania ludzkiej pamięci. W tej metodzie zapisu pewne terminy wyjaśniane są przez inne terminy, przy czym powstaje skierowany graf powiązań, relacji między węzłami-obiektami, często z pętlami, w którym można znaleźć potrzebne wyjaśnienia. Tworzy się specyficzna logika opisująca pewną dziedzinę wiedzy.

Łukom w tym grafie można przypisać znaczenie relacji, można też przypisać pewne wagi, określające stopień pewności danego powiązania.

Węzły w grafie powiązań mogą reprezentować obiekty fizyczne, obiekty wirtualne (czynności, wydarzenia), mogą posiadać też tzw. deskryptory, czyli dodatkowy opis węzła, pokazujący pewne cechy charakterystyczne obiektu.

Sieci semantyczne

Przykład sieci semantycznej z relacjami „Jest” i „Ma”:



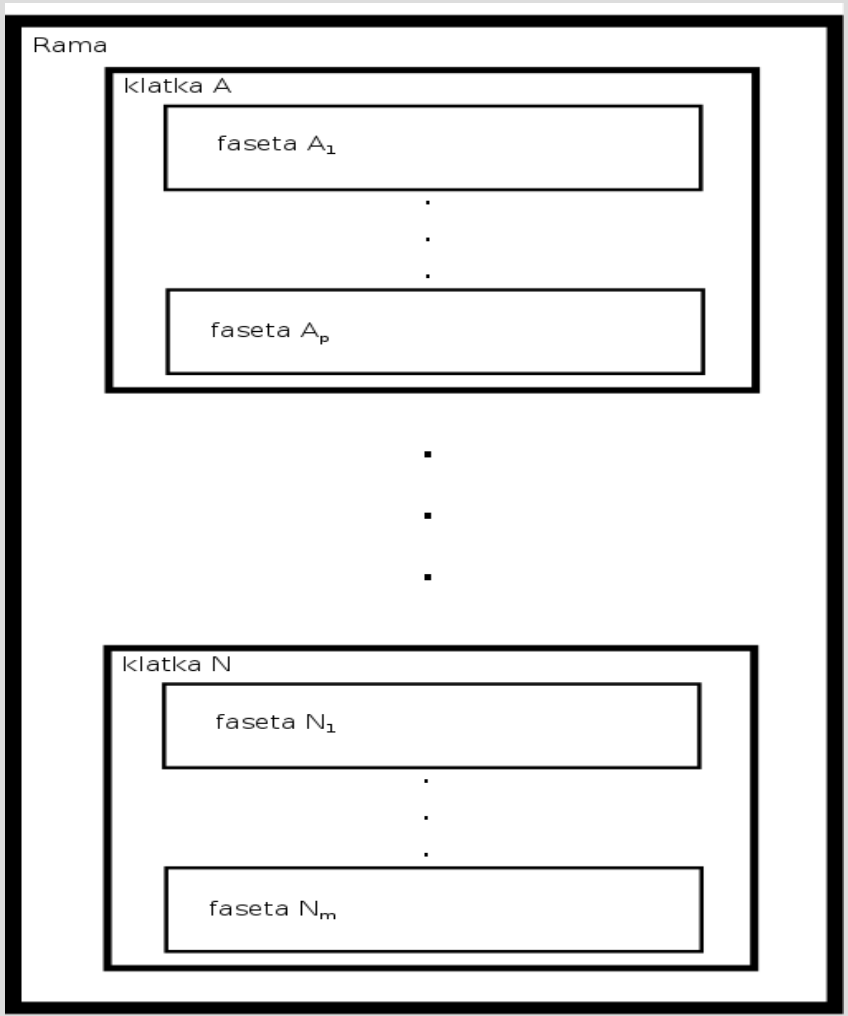
Ramy

Są z kolei pewnym rozszerzeniem stwierdzeń, umożliwiającym usystematyzowane opisywanie obiektów za pomocą faktów, reguł i procedur umieszczonych w odpowiednich strukturach danych.

Ramy są najwyższą w hierarchii obiektów strukturą i zawierają **klatki**, które z kolei mogą grupować **fasety**.

Rama jest opisem pewnego obiektu lub klasy obiektów, klatki opisują jego cechy, które z kolei mogą potrzebować cech jeszcze niższego rzędu, czyli faset. Klatki mogą mieć odwołania do innych ram. Ramy, klatki i fasety mają swoje unikalne nazwy, tak aby można było jej jednoznacznie rozróżnić. Istnieje bardzo wiele predefiniowanych typów klatek i faset. Ramy mogą dziedziczyć właściwości po innych ramach, stąd możliwa jest też hierarcha ram.
Odpowiednio przygotowane ramy umożliwiają wnioskowanie na podstawie tylko ram, bez dodatkowych reguł (reguły mogą być zapisane w postaci ram).

Ramy - schemat



Modele obliczeniowe

Modele obliczeniowe to swoisty sposób zapisu danych służących do konkretnych obliczeń. Jest to również pewnego rodzaju rozszerzenie stwierdzeń, w którym buduje się pewne modele z odpowiednimi formułami oraz zmiennymi wejściowymi i wyjściowymi. Istotnym jest tu pojęcie relacji, które wiąże zmienne. Celem modelu jest policzenie wszystkich niewiadomych na podstawie formuł i danych. Działanie systemu polega na przeszukiwaniu całej „bazy danych” w kierunku znalezienia nowych możliwych do policzenia zmiennych. Dodatkowo następuje porównywanie obliczonych wyników na podstawie różnych możliwych formuł, co umożliwia weryfikację wyników.

Modele obliczeniowe stanowią raczej pewne dodatkowe moduły, które przeprowadzają obliczenia dla właściwego oprogramowania, np. systemu ekspertowego.

Wnioskowanie

Ze względu na metodę wnioskowania SE dzieli się na trzy kategorie:

- systemy działające w oparciu o logikę dwuwartościową (Boole'a)
- systemy wykorzystujące logiki wielowartościowe
- systemy z logiką rozmytą.

Wnioskowanie

Wnioskowanie w logice dwuwartościowej (a taką zajmiemy się na bieżących zajęciach) oparte jest na regule **modus ponendo ponens** (zwanej także regułą odrywania):

$$\frac{(A \Rightarrow B), A}{B}$$

lub w innym zapisie: $((A \Rightarrow B) \wedge A) \Rightarrow B$
co oznacza, że jeśli z przesłanki A wynika B i A jest prawdziwe, to B też jest prawdziwe.

Wnioskowanie

Wnioskowanie w logice dwuwartościowej można oprzeć także na regule **modus tollendo tollens**:

$$\frac{(A \Rightarrow B), \neg B}{\neg A}$$

lub w innym zapisie: $((A \Rightarrow B) \wedge \neg B) \Rightarrow \neg A$
co oznacza, że jeśli z przesłanki A wynika B i B jest nieprawdziwe, to A też jest nieprawdziwe.

Wnioskowanie

Podstawowe typy wnioskowania to:

- wnioskowanie w przód
- wnioskowanie wstecz
- wnioskowanie mieszane

Wnioskowanie w przód

Wnioskowanie w przód polega na generowaniu nowych faktów na podstawie istniejących w bazie dopóty, dopóki nie pojawi się fakt będący poszukiwaną hipotezą lub nie da się uaktywnić żadnych nowych reguł.

Niestety często nowe fakty pojawiają się lawinowo, co może doprowadzić do zablokowania wnioskowania zbyt wielką liczbą faktów i reguł do „odpalenia”. Dlatego stosuje się techniki sterowania wnioskowaniem w przód. Techniki te są oparte na **heurystykach**, czyli pewnych spostrzeżeniach, które nie gwarantują sukcesu, ale często go zapewniają.

Wnioskowanie w przód
strategie sterowania wnioskowaniem

- strategia świeżości – uaktywnianie reguł dodanych w procesie wnioskowania najpóźniej
- strategia blokowania – blokowanie już użytych reguł
- strategia specyficzności – skłonność do użycia reguł o większej liczbie przesłanek (bardziej szczegółowych/specyficznych)
- strategia przypadkowości – losowy wybór reguł do „odpalenia”.

Wnioskowanie w przód przykład

Są cztery reguły i dwa fakty:

- R1: Jeśli A i C to B
- R2: Jeśli A i E to C
- R3: Jeśli D i C to B
- R4: Jeśli C i B to D

dane są A i E, udowodnić D.

Rozwiązanie:

1. Z R2 otrzymujemy C
2. Z R1 otrzymujemy B
3. Z R4 otrzymujemy D – koniec!

Wnioskowanie wstecz

W metodzie tej, jak można się domyślić, działanie zachodzi w odwrotnym kierunku:

- najpierw poszukuje się, czy dowodzona hipoteza jest w bazie faktów, jeśli tak, to wnioskowanie jest skończone
- jeśli nie jest w bazie, to czy jest konkluzją jakiejś reguły
- jeśli nie jest konkluzją, to nie można jej dowieść i wnioskowanie skończone
- jeśli jest konkluzją pewnej reguły, to sprawdzane są jej przesłanki
- jeśli są w bazie, to stosując wnioskowanie w przód, dowiedziemy hipotezy,
- jeśli nie, szukamy ich w konkluzjach innych reguł i tak powtarzamy działanie aż skończą się reguły lub udowodnimy hipotezę.

Wnioskowanie wstecz przykład

Przykładowe zadanie jest takie samo jak poprzednio:

- R1: Jeśli A i C to B
- R2: Jeśli A i E to C
- R3: Jeśli D i C to B
- R4: Jeśli C i B to D

dane są A i E, udowodnić D.

Rozwiązanie:

1. D nie jest w bazie faktów, jest konkluzją R4
2. C i B nie są w bazie faktów, są konkluzjami R1, R2, R3
3. R2 można „odpalić” (R1 i R3 - nie można) - C dodaje się do bazy faktów
4. Teraz R1 można „odpalić” - B dodaje się do bazy faktów
5. W takim razie można „odpalić” R4 i D jest udowodnione!

Wnioskowanie mieszane

W metodzie tej część bazy reguł przetwarza się metodą wnioskowania w przód, a część metodą wnioskowania wstecz. Najczęściej stosuje się tu też pewne heurystyki zarządzające wnioskowaniem, np. dopóki jest to możliwe (są reguły do „odpalenia”), stosuje się wnioskowanie wstecz (nowe fakty generowane bardziej oszczędnie), a gdy ich brakuje, system przełącza się na wnioskowanie w przód. Daje to oszczędności w wykorzystanej pamięci i szybkości działania wnioskowania.

Wnioskowanie mieszane przykład

Przykładowe zadanie jest takie samo jak poprzednio:
wnioskowanie w przód:

- R1: Jeśli A i C to B
- R2: Jeśli A i E to C

wnioskowanie wstecz:

- R3: Jeśli D i C to B
- R4: Jeśli C i B to D

dane są A i E, udowodnić D.
Rozwiązanie:
1. D nie jest w bazie faktów, jest konkluzją R4 (wstecz)
2. C i B nie są w bazie faktów, B jest konkluzją R3, ale nie można jej „odpalić” metodą wnioskowania wstecz
3. R2 można „odpalić” metodą wnioskowania wprzód (przełączenie metody) - C dodaje się do bazy faktów
4. W dalszym ciągu nie można nic „odpalić” metodą wnioskowania wstecz
5. Za to R1 można „odpalić” metodą wnioskowania w przód (przełączenie) - B dodaje się do bazy faktów
6. W takim razie można „odpalić” (metodą wstecz) R4 i D jest udowodnione!

Wnioskowanie

– przykład do samodzielnego przećwiczenia opisanymi metodami

Mamy pięć reguł i dwa fakty:

- R1: Jeśli A i C to B
- R2: Jeśli A i ¬E to C
- R3: Jeśli D i C to B
- R4: Jeśli D i B to F
- R5: Jeśli C i B to D.

Dane są A i ¬E.

Udowodnić F metodami wnioskowania wprzód, wstecz i mieszanego.

Praktyczne zastosowania SE

- diagnozowanie chorób (np. system Mycin, Emycin, Casnet, Caduceus) i wywołujących je drobnoustrojów;
- identyfikacja struktur molekularnych (np. Dendral) – ustalanie struktury nieznanych związków chemicznych na podstawie analizy widm spektroskopowych, generowanie możliwych struktur połączeń cząsteczek;
- systemy doradcze przy usuwaniu usterek w różnego rodzaju sprzęcie (np. Delta/Cats, Mistral);
- analiza danych geologicznych, poszukiwanie złóż (np. Prospector);
- sterowanie robotami, pojazdami;
- prognozowanie pogody;
- analiza notowań giełdowych;
- system porad prawnych.

Praktyczne zastosowania SE

Wymienione wyżej systemy są przykładami systemów dedykowanych. Istnieją też systemy szkieletowe, które można zastosować w dowolnych zagadnieniach po wypełnieniu baz reguł i baz faktów, są to np. komercyjne systemy Exsys i Sphinx.

Istnieją także SE open source: PyKnow, Benchflow, Mandrax, Drools.

Do tworzenia SE powstały też specjalne języki programowania, takie jak LISP i CLIPS.