

System ekspertowy

Ekspert – osoba posiadająca wiedzę specjalistyczną w pewnej dziedzinie, najczęściej o niealgorytmicznym charakterze i umiejętność jej stosowania dla rozwiązywania problemów z tej dziedziny.

Cechy eksperta

- Wiedza dziedzinowa
- Umiejętność wnioskowania

- 44 -

System ekspertowy...

System ekspertowy (program regułowy) – program, którego zadaniem jest rozwiązywanie problemów zleczanych ekspertom.

Główne elementy systemu ekspertowego:

- Baza wiedzy – wiedza dziedzinowa, istotna dla podejmowania racjonalnych decyzji.
- System wnioskujący – system podejmowania decyzji na podstawie bazy wiedzy

- 45 -

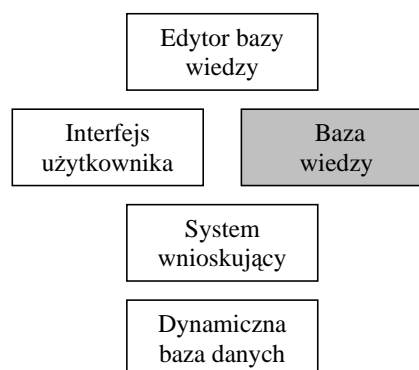
System ekspertowy...

Pozostałe elementy systemu ekspertowego:

- Edytor bazy wiedzy – umożliwia edycję, aktualizację, usuwanie bazy wiedzy
- Interfejs użytkownika – komunikacja systemu z użytkownikiem
- Dynamiczna baza danych – baza danych w postaci relacyjnej służąca przechowywaniu odpowiedzi użytkownika i wyników wnioskowania

- 46 -

Struktura systemu ekspertowego



- 47 -

Baza wiedzy

- Baza wiedzy oddzielona od systemu ekspertowego
- Najczęściej plik tekstowy, łatwość
 - czytania
 - edycji

do modyfikacji bazy wiedzy służy tzw. edytor bazy wiedzy. Umożliwia jej modyfikację bez naruszania integralności systemu wnioskującego

- 48 -

Baza reguł

- Reguły – wiedza dziedzinowa o charakterze ogólnym
- Fakty – wiedza dziedzinowa o charakterze szczegółowym

| |
|--------|
| Reguły |
| Fakty |

Przykład reguły:

„Jeżeli student x otrzymał wszystkie zaliczenia w terminie **i** student x zdał wszystkie egzaminy w terminie **to** student x zaliczył semestr”

Przykład faktu:

„Student Jan Nowak otrzymał wszystkie zaliczenia w terminie”

- 49 -

Baza reguł...

- Reguła jest zdaniem warunkowym to znaczy:
 - W języku potocznym:
Jeśli A jest prawdą i B jest prawdą to C jest prawdą
 - W języku logiki:
 $A \wedge B \Rightarrow C$
 - W języku PROLOG:
C:-A,B
 - Na potrzeby systemów ekspertowych
 $A, B \rightarrow C$
- A, B – warunki reguły
- C – wniosek reguły

- 50 -

Baza reguł... Klauzule Horna

- Fakt – może być uważany za wniosek reguły, której warunki zawsze są prawdą.
- Fakt zapisujemy w postaci A
- Klauzula Horna – reguła o jednym wniosku

Twierdzenie.

Dowolne zagadnienie dające się wyrazić w języku logiki można wyrazić za pomocą klauzul Horna.

- 51 -

Klauzule Horna...

- Jeżeli z tych samych warunków mogą wynikać dwa wnioski zapisujemy w postaci dwóch klauzul Horna:
 $A, B, C \rightarrow \text{wniosek1}$
 $A, B, C \rightarrow \text{wniosek2}$
 - Jeżeli ten sam wniosek uzyskujemy w wyniku spełnienia dwóch różnych zbiorów warunków zapisujemy:
 $A, B, C \rightarrow \text{wniosek}$
 $D, E, F \rightarrow \text{wniosek}$
- Klauzule Horna upraszczają automatyczne wnioskowanie a więc budowę systemu

- 52 -

Budowa bazy wiedzy. Zagnieżdżanie reguł

- Występuje gdy wnioski jednych reguł są warunkami reguł innych.
- Występowanie zagnieżdżania oznacza, że nie o wszystkie warunki powinien system pytać
- Zbiór warunków bazy reguł dzielimy na warunki **dopytywalne** i **niedopytywalne**
- Warunki dopytywalne** – warunki nie będące wnioskami innych reguł. Ich wartość logiczna musi być określona przez użytkownika
- Warunki niedopytywalne** – warunki będące wnioskami innych reguł. Ich wartość może być wyprowadzona przez system na podstawie warunków dopytywalnych

- 53 -

Budowa bazy wiedzy. Klasyfikacja baz reguł

- Klasyfikacja ze względu na strukturę **zagnieżdżania reguł**
 - Bazy elementarne** – warunki niedopytywalne nie mogą występować w postaci zanegowanej np.:
 - $A, B, \neg C \rightarrow W$
 - $W, \neg D, E \rightarrow V$
 - $V, H, I, J \rightarrow U$(warunki niedopytywalne W, V występują w postaci niezanegowanej)

- 54 -

Budowa bazy wiedzy. Klasyfikacja baz reguł

- Bazy rozwinięte** – warunki dopytywane mogą występować w postaci zanegowanej
 - $A, B, C \rightarrow W$
 - $\neg W, \neg D, E \rightarrow V$
 - $\neg V, H, I, J \rightarrow U$(warunki niedopytywalne $\neg W, \neg V$ mogą występować w postaci zanegowanej)

- 55 -

Budowa bazy wiedzy. Spłaszczanie baz reguł

- Dowolną bazę reguł można spłaszczyć przez zastępowanie warunków niedopytywalnych listami warunków reguł, których są wnioskami – często utrata czytelności

Baza zagnieżdżona

- $A \rightarrow D$
- $F, H \rightarrow G$
- $B \rightarrow L$
- $D, J \rightarrow M$
- $C, D \rightarrow F$
- $A, E \rightarrow J$

Baza płaska

- $A \rightarrow D$
- $C, A, H \rightarrow G$
- $B \rightarrow L$
- $A, E \rightarrow M$
- $C, A \rightarrow F$
- $A, E \rightarrow J$

- 56 -

Wnioskowanie

- Dla wszystkich systemów dokładnych zakładamy, że **uważamy za nieznane to, co nie wynika z bazy reguł, faktów, faktów zadeklarowanych przez użytkownika oraz reguł wnioskowania - założenie otwartego świata**
- Założenie otwartego świata – pozostawia drzwi otwarte na przypadek opracowania innej reguły z tym samym wnioskiem lecz innymi warunkami, które być może okażą się prawdziwe

- 57 -

Wnioskowanie

- Jeżeli prawdą są wszystkie warunki implikacji regułowej prawdziwy jest również jej wniosek
- Jeżeli nieprawdą jest choć jeden warunek implikacji regułowej to wniosek jest nieukonkretniony

| Implikacja w logice | | |
|---------------------|---|-------------------|
| q | p | $q \Rightarrow p$ |
| 1 | 1 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

| Implikacja regułowa | | |
|---------------------|---|-------------------|
| q | p | $q \rightarrow p$ |
| 1 | 1 | 1 |
| 0 | ? | 1 |
| | | |

- 58 -

Ograniczenia założenia otwartego świata

- Rozpatrzmy regułę:
jeśli dostanę urlop i będę miał pieniądze wyjadę na urlop
- Reguła nie określa co się stanie, gdy któryś z jej warunków nie będzie spełniony
- Rozwiązanie – jeżeli warunki powyższej reguły są jedynymi, których spełnienie umożliwi wyjazd na urlop należałoby dopisać dwie reguły:
jeśli nie dostanę urlopu nie pojadę na urlop
jeśli nie będę miał pieniędzy nie pojadę na urlop

- 59 -

Cele wnioskowania

- Wyznaczenie wszystkich wniosków prawdziwych dla początkowego zbioru faktów w oparciu o zdefiniowane reguły oraz reguły wnioskowania – wnioskowanie w przód
- Potwierdzenie (zweryfikowanie) lub brak możliwości potwierdzenia iż dana hipoteza wynika z danego początkowego zbioru faktów oraz reguł – wnioskowanie wstecz

- 60 -

Zasady wnioskowania

Wnioskowanie w systemach ekspertowych elementarnych dokładnych opiera się o dwie reguły:

- reguła odrywania** (*modus ponens*):

Dana reguła: **Jeżeli A to B**

Prawdą jest **A**

Wniosek: prawdą jest **B**

$$\frac{A \rightarrow B, A}{B}$$

- Założenie otwartego świata:**

Dana reguła: **Jeżeli A to B**

Nieprawdą jest **A**

Wniosek: **B** jest nieukonkretnione

$$\frac{A \rightarrow B, \neg A}{?B}$$

- 61 -

Wnioskowanie w przód

Przykład 1:

1. $A \rightarrow D$
2. $F, H \rightarrow G$
3. $B \rightarrow L$
4. $D, J \rightarrow M$
5. $C, D \rightarrow F$
6. $A, E \rightarrow J$

Za fakty uznajemy A, C, E, H

- 62 -

Wnioskowanie w przód

Przykład 1:

1. $P \rightarrow Q$
2. $L, M \rightarrow P$
3. $B, L \rightarrow M$
4. $A, P \rightarrow L$
5. $A, B \rightarrow L$

Za fakty uznajemy A, B. Czy można wywnioskować Q?

- 63 -

Wnioskowanie w przód - uwagi

1. Jeżeli reguła ma warunek o nieokreślonej wartości logicznej to jest pomijana
2. Jeżeli wszystkie warunki reguły są faktami to reguła jest spełniona i jej wniosek jest faktem – dodawanym do dynamicznej bazy danych
3. Jeżeli jeden z warunków nie jest faktem, to reguła jest niespełniona i jej wniosek nie jest faktem, czego nie piszemy bo to co nie wynika z bazy reguł nie jest faktem
4. Jeżeli kilka reguł ma ten sam wniosek to spełnienie co najmniej z nich czyni wniosek faktem
5. Jeżeli reguła została już testowana i jej wniosek uznano za fakt zostaje pomijana w następnym cyklu testowania

- 64 -

Wnioskowanie wstecz

- Przy wnioskowaniu w przód użytkownik deklaruje pewne fakty otrzymując na wyjściu fakty nowe, wynikające z bazy wiedzy
- Użytkownik nie musi być zainteresowany znalezieniem wszystkich faktów a jedynie stwierdzeniem prawdziwości jednego z nich – nazywamy go hipotezą
- Wynikiem wnioskowania wstecz może być:
 - Weryfikacja hipotezy – potwierdzenie
 - Brak potwierdzenia hipotezy
- Dla zweryfikowania hipotezy musi ona być wnioskiem co najmniej jednej reguły
- W przypadku gdy nie jest wnioskiem żadnej z reguł oznacza że nie wynika z bazy reguł (założenie otwartego świata)

- 65 -

Wnioskowanie wstecz

Przykład:

1. $A \rightarrow D$
2. $F, H \rightarrow G$
3. $B \rightarrow L$
4. $D, J \rightarrow M$
5. $C, D \rightarrow F$
6. $A, E \rightarrow J$

- Zweryfikować hipotezę F, przy założeniu że tylko warunki A, C są faktami.

- 66 -

Wnioskowanie wstecz

Przykład:

1. $P \rightarrow Q$
2. $L, M \rightarrow P$
3. $B, L \rightarrow M$
4. $A, P \rightarrow L$
5. $A, B \rightarrow L$

- Zweryfikować hipotezę Q, przy założeniu że tylko warunki A, B są faktami.

- 67 -

Regułowe systemy ekspertowe

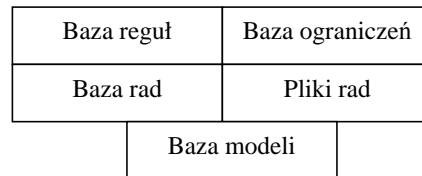
- Wiedza dziedzinowa zapisana w postaci deklaratywnej za pomocą:
 - Reguł
 - Faktów
- System wnioskujący oparty na wnioskowaniu logicznym

Systemy regułowo-modelowe (hybrydowe)

- Wiedza dziedzinowa w postaci deklaratywnej (reguły i fakty) i proceduralnej (równania, relacje)
- Wnioskowanie logiczne wspomagane obliczeniami numerycznymi i relacyjnymi

- 68 -

Struktura bazy wiedzy RMSE



- Baza reguł – wiedza dziedzinowa logiczna
- Baza ograniczeń – wiedza dziedzinowa logiczna
- Baza modeli – wiedza dziedzinowa matematyczna
- Baza rad/Pliki rad – wiedza dziedzinowa uzupełniająca i wyjaśniająca

- 69 -

System RMSE

- W systemach RMSE stosowane są wyłącznie zmienne łańcuchowe, tak zwane *inkludy*, np.: "zmienna łańcuchowa"
- Wyróżnia się zmienne łańcuchowe typu:
 - Logicznego – gdy przyjmuje tylko wartości ze zbioru {prawda, nieprawda}, np.: "wiek >= 32 lat"
 - Rzeczywistego – np.: "pi", "123.45"
 - Całkowitego – np.: "wiek", "12"
- Inkludy nie muszą być atomowe (mogą występować spacje)

- 70 -

System RMSE, baza wiedzy

- Fakty bazy dokładnej mają postać:
fakt("warunek_dopytywalny")
- Reguły bazy dokładnej mają postać:
reguła(nr_reguły, "wniosek", "lista warunków", semafor_wyświetlania)
- Lista warunków:
["warunek1", "warunek2", "warunek3", ..., "warunekN"]
- warunek1* – logiczna zmienna łańcuchowa
- semafor_wyświetlania* – wartość 0 lub 1 oznacza odpowiednio, że informacja o stosowaniu reguły nie jest bądź jest wyświetlana w trakcie wnioskowania

- 71 -

System RMSE, baza wiedzy

Baza ograniczeń

- elementarna dokładna baza ograniczeń zawiera zbiory warunków dopytywanych wzajemnie wykluczających się.
- W systemie RMSE ograniczenia zapisujemy następująco:
ograniczenie(nr_og, lista_warunków_dop_wykluczających_się)
przykład:
ograniczenie(1, ["dostanę urlop", "nie dostanę urlopu"])

- 72 -

System RMSE, baza wiedzy

Baza modeli

- Model elementarny podstawowy dokładny

*model(numer_modelu,
"warunek_startowy",
"wynik" / "wniosek",
"pierwszy_argument",
"operacja" / "relacja",
"drugi_argument",
semafor_wyświetlania)*

- 73 -

| | |
|--|--|
| <p>System RMSE, baza wiedzy</p> <p>Baza modeli</p> <ul style="list-style-type: none"> • ”wynik” – rzeczywista zmienna łańcuchowa dla modelu arytmetycznego a logiczna dla modelu relacyjnego • argumenty – rzeczywiste lub całkowite zmienne łańcuchowe • ”operacja” – ”+”, ”-”, ”*”, ”/”, ”div”, ”mod”, ”min”, ”max”, ”%”, ”A^N”, ”zaokraglenie_do_N”, ”sqrt”, ”sin”, ”cos”, ”tan”, ”arctan”, ”log”, ”ln”, ”exp”, ”round”, ”abs”, ”=” • W przypadku modeli jednoargumentowych, drugi argument jest równy ”0”. <p>- 74 -</p> | <p>System RMSE, baza wiedzy</p> <p>Baza modeli</p> <ul style="list-style-type: none"> • ”relacja” – w przypadku modeli relacyjnych testuje relacje ”>”, ”<”, ”>=”, ”<=”, ”==”, ”<>”, ”><” <p>Interpretacja modelu arytmetycznego:</p> <p><i>Model(Nr, ”start”, ”wynik”, ”x1”, ”+”, ”x2”, semafor_wyświetlania)</i></p> <p>Jeżeli <i>start</i> jest prawdą to <i>wynik</i>=$x1+x2$</p> <p>Jeżeli <i>start</i> nie jest prawdą to <i>wynik</i> jest nieokreślony</p> <p>- 75 -</p> |
| <p>System RMSE, baza wiedzy</p> <p>Baza modeli</p> <p>Interpretacja modelu relacyjnego:</p> <p><i>Model(Nr, ”start”, ”wniosek”, ”x1”, ”<=”, ”x2”, semafor_wyświetlania)</i></p> <p>Jeżeli <i>start</i> jest prawdą oraz $x1 \leq x2$ to <i>wniosek</i> jest prawdą</p> <p>Jeżeli <i>start</i> nie jest prawdą lub $x1 > x2$ to <i>wniosek</i> jest nieokreślony</p> <p>- 76 -</p> | <p>System RMSE, baza wiedzy</p> <p>Baza modeli</p> <ul style="list-style-type: none"> • Model elementarny rozszerzony dokładny <p><i>model(numer_modelu, ”warunek_startowy”, ”wynik” / ”wniosek”, ”operacja” / ”relacja”, Lista_argumentów, semafor_wyświetlania)</i></p> <p>- 77 -</p> |
| <p>System RMSE, baza wiedzy</p> <p>Baza modeli</p> <ul style="list-style-type: none"> • ”wynik” – rzeczywista zmienna łańcuchowa dla modelu arytmetycznego a logiczna dla modelu relacyjnego • argumenty – rzeczywiste lub całkowite zmienne łańcuchowe • ”operacja” – w przypadku modeli arytmetycznych wykonują operacje: ”+”, ”*”, ”min_list”, ”max_list” • ”relacja” – w przypadku modeli relacyjnych testuje relacje ”<”, ”<=”, ”<=<”, ”<=<= <p>- 78 -</p> | <p>System RMSE, baza wiedzy</p> <p>Baza modeli</p> <p>Interpretacja modelu arytmetycznego:</p> <p><i>Model(Nr, ”start”, ”wynik”, ”+”, [”x1”, ”x2”, ..., ”xn”] semafor_wyświetlania)</i></p> <p>Jeżeli <i>start</i> jest prawdą to <i>wynik</i>=$x1+x2+...+xn$</p> <p>Jeżeli <i>start</i> nie jest prawdą to <i>wynik</i> jest nieokreślony</p> <p>- 79 -</p> |

System RMSE, baza wiedzy

Baza modeli

- Dla modeli rozszerzonych relacyjnych dokładnych *Lista_argumntów* jest postaci:

["ograniczenie dolne", "wielkość testowana", "ograniczenie górne"]

Interpretacja modelu relacyjnego:

Model(Nr, "start", "wniosek", ["ogr1", "x", "ogr2"], "<,<=", semafor_wyświetlania)

Jeżeli *start* jest prawdą oraz $ogr1 < x <= ogr2$ to *wniosek* jest prawdą

System RMSE, baza wiedzy

Baza modeli

- Model elementarny liniowy dokładny

model liniowy(numer_modelu, "warunek_startowy", "wynik", Lista_współczynników, Lista_argumentów, semafor_wyświetlania)

System RMSE, baza wiedzy

Baza modeli

- Model elementarny wielomianowy dokładny

model_wielomianowy(numer_modelu, "warunek_startowy", "wynik", Lista_współczynników, Lista_potęg, Lista_argumentów, semafor_wyświetlania)

System RMSE, baza wiedzy

Baza modeli

- Podobnie jak w przypadku reguł modele mogą się zagnieżdżać
 - wynik jednego modelu może być argumentem innego,
 - wniosek jednego modelu może być warunkiem stosowania innego
 - wniosek modelu relacyjnego może być warunkiem reguły
 - wniosek reguły może być warunkiem startowym modelu