

Metody zapisu wiedzy



Tablice decyzyjne
Systemy regułowe
Wnioskowanie

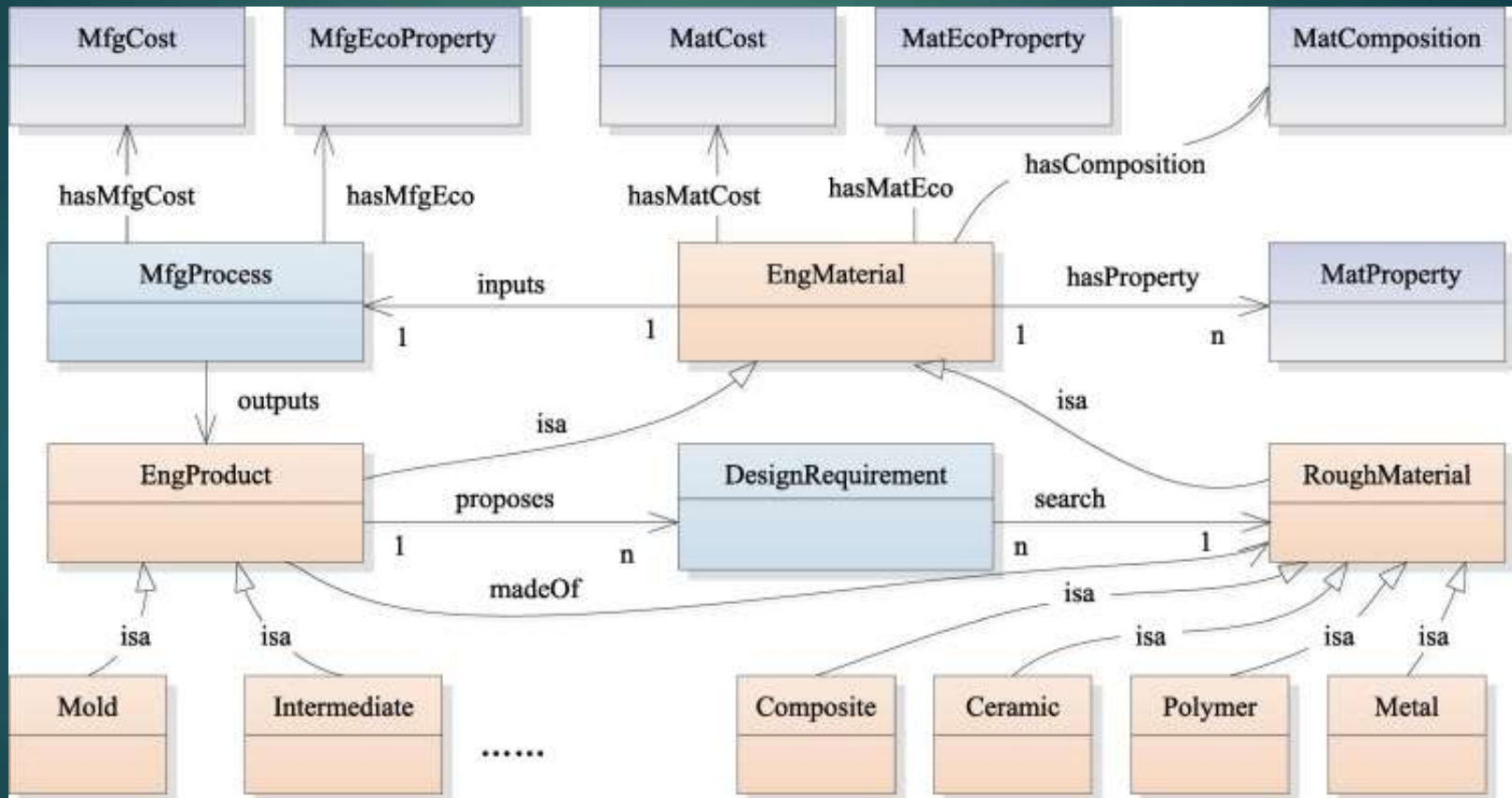
Wiedza

- ▶ Płynne połączenie doświadczenia, ocen wartości, informacji o kontekście oraz analitycznego wglądu w zagadnienia, które zapewnia ramy dla oceny i włączania nowych doświadczeń i informacji
- ▶ Wiedza organizacji wywodzi się i jest charakterystyczna dla umysłów ludzi [Davenport i Prusak 1998]

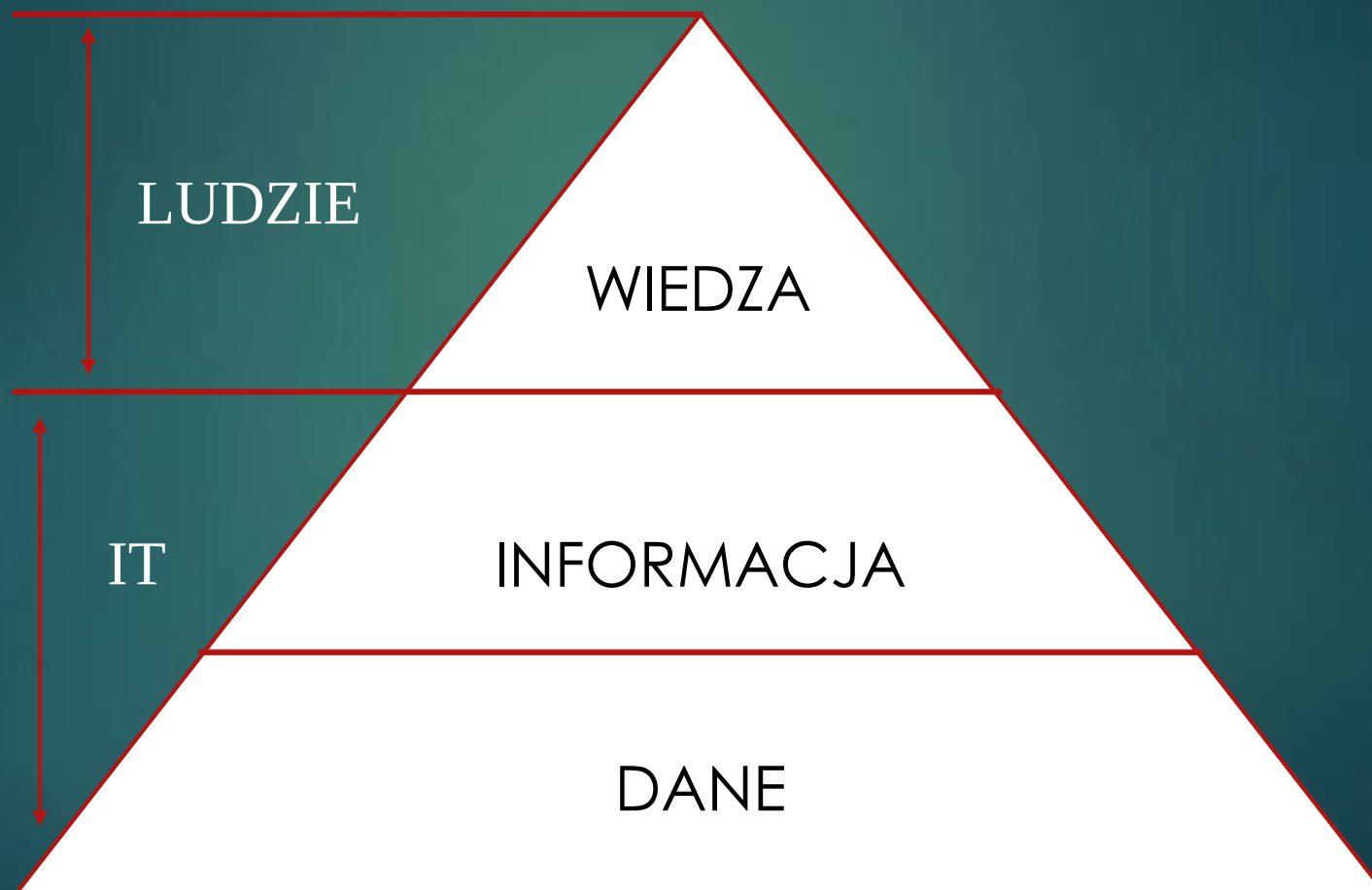
Wiedza

- ▶ Cechą wiedzy jest porządek, którego poszukiwali filozofowie zajmujący się metafizyką (czy jak wolimy to dzisiaj mówić ontologią), której zadaniem jest po prostu opisanie rzeczywistości, a więc tego co istnieje
- ▶ Tradycyjnym celem ontologii jest podział wiadomości o świecie (bycie) na kategorie definiujące rzeczy składające się na niego
- ▶ Wiedza składa się z uporządkowanych opisów, relacji i procedur

Przykład ontologii materiałowej (logika opisowa)



Hierarchia wiedzy

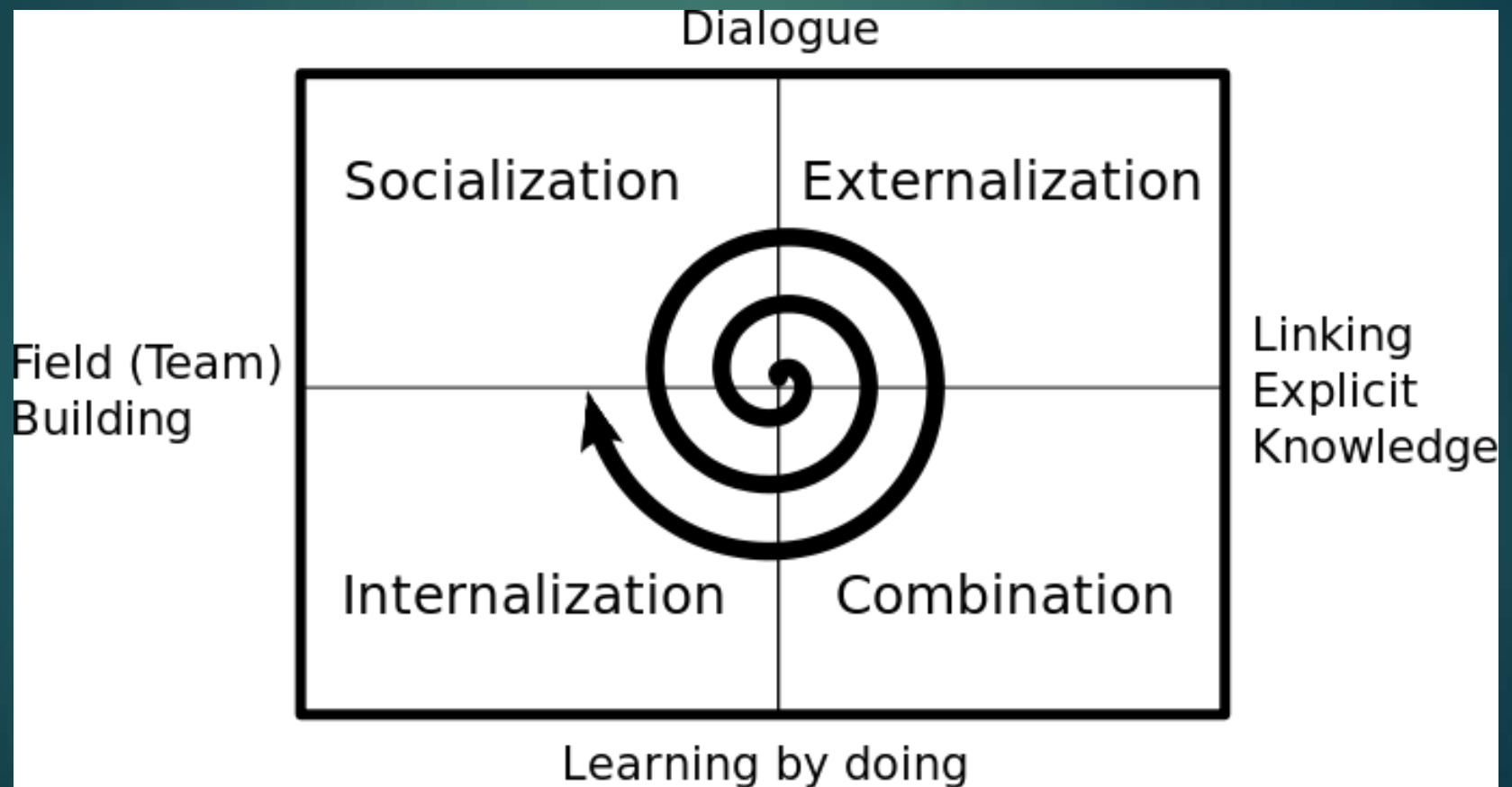


Wiedza człowieka (eksperta)

- ▶ **Wiedza milcząca** (*tacit knowledge*) – wiem jak postępować w danych okolicznościach, ale nie wiem dlaczego i nie potrafię tego wyjaśnić
- ▶ Asocjacyjna **wiedza empiryczna** – znam wiele przykładów i umiem je wykorzystać do wnioskowania
- ▶ **Wiedza głęboka** – dokładnie wiem dlaczego (oczywiście w pewnej ontologii)

Model SECI

(Nonaka i Takeuchi)



Wiedza - reprezentacja

- ▶ Reprezentacja **proceduralna** – określenie zbioru procedur, których działanie reprezentuje wiedzę o dziedzinie (np. procedura wyznaczania ekstremum funkcji)
- ▶ Reprezentacja **deklaratywna** – polegająca na określeniu zbioru specyficznych dla rozpatrywanej dziedziny faktów i reguł

Komputery wyposażone w wiedzę

- ▶ Systemy ekspertowe to „programy z wiedzą”
- ▶ Sam program realizuje procedury wnioskowania i ew. obliczania, objaśniania, uczenia, komunikowania
- ▶ Wiedza zapisana jest w osobnych zbiorach i poza tym, że może być poszerzana przez program (uczenie) jest od niego niezależna

Tabela (tablica) decyzyjna

- ▶ Składa się z
 - Listy elementarnych warunków (*fasety - przesłanki*)
 - Listy elementarnych, możliwych do podjęcia działań
(*fasety – konkluzje*)
 - Wierszy testów (reguły decyzyjne)
 - Wierszy działań – zachowania w odpowiedzi na spełnione reguły (testy)
- ▶ Reguły muszą się wzajemnie wykluczać!
- ▶ Wystąpienie sytuacji spełniającej reguły, to zdarzenie

Przykład 1: tabela decyzyjna

Światło zielone	T	N	N
Światło żółte	N	T	N
Światło czerwone	N	N	T
Jedź	X		
Przygotuj się do jazdy		X	
Zatrzymaj się			X

Przykład2: tabela decyzyjna

Klient jest wiarygodny	T	T					
Klient nie jest wiarygodny			T				
Klient płaci gotówką				T	T		
Klient dokonuje przedpłaty						T	T
Klient płaci przelewem	T	T	T				
Wartość kontraktu jest duża	T			T		T	
Wartość kontraktu jest mała		T			T		T
Decyzja ostateczna: sprzedać po cenie standardowej		X			X		X
Decyzja ostateczna: przyznać klientowi upust	X			X		X	
Decyzja ostateczna: odrzuć ofertę			X				

1. If "Klient jest wiarygodny" And " Klient płaci przelewem" And "Wartość kontraktu jest duża" Then "przyznać klientowi upust"
2. If "Klient jest wiarygodny" And " Klient płaci przelewem" And "Wartość kontraktu jest mała" Then "sprzedać po cenie standardowej,,
3. If "Klient nie jest wiarygodny" And " Klient płaci przelewem" And "Wartość kontraktu jest duża" Then konkluzja = "odrzucić ofertę,,
4. If " Klient płaci gotówką" And "Wartość kontraktu jest duża" Then "przyznać klientowi upust,,
5. If " Klient płaci gotówką" And "Wartość kontraktu jest mała" Then "sprzedać po cenie standardowej,,
6. If " Klient dokonuje przedpłaty" And "Wartość kontraktu jest duża" Then "przyznać klientowi upust,,
7. If " Klient dokonuje przedpłaty " And "Wartość kontraktu jest mała" Then "sprzedać po cenie standardowej"

Przykład: fasety

<i>Nr faktu</i>	<i>Opis</i>
1	Klient jest wiarygodny
2	Klient nie jest wiarygodny
3	Klient płaci gotówką
4	Klient dokonuje przedpłaty
5	Klient płaci przelewem
6	Wartość kontraktu jest duża
7	Wartość kontraktu jest mała
8	Decyzja wstępna: sprzedać towar
9	Decyzja ostateczna: sprzedać po cenie standardowej
10	Decyzja ostateczna: przyznać klientowi upust
11	Decyzja ostateczna: odrzucić ofertę

Przykład: baza wiedzy (przesłanki)

	1	2	3
jeżeli	<i>Klient płaci gotówką</i>		to <i>Decyzja wstępna: sprzedać towar</i>
jeżeli	<i>Klient dokonuje przedpłaty</i>		to <i>Decyzja wstępna: sprzedać towar</i>
jeżeli	<i>Klient płaci przelewem</i>	i <i>Klient jest wiarygodny</i>	to <i>Decyzja wstępna: sprzedać towar</i>
jeżeli	<i>Klient płaci przelewem</i>	i <i>Klient nie jest wiarygodny</i>	to <i>Decyzja ostateczna: odrzucić ofertę</i>
jeżeli	<i>Decyzja wstępna: sprzedać towar</i>	i <i>Wartość kontraktu jest duża</i>	to <i>Decyzja ostateczna: przyznać klientowi upust</i>
jeżeli	<i>Decyzja wstępna: sprzedać towar</i>	i <i>Wartość kontraktu jest mała</i>	to <i>Decyzja ostateczna: odrzucić ofertę</i>

```
For k = 1 To LiczbaReguł
    prawda1 = True
    l = 1
    Do
        prawda = False
        For ll = 1 To LiczbaFaktów
            If fakty(ll) = przeslanki(k, l) Then
                prawda = True
            End If
        Next
        If Not prawda Then
            prawda1 = False
        End If
        l = l + 1
    Loop Until IsEmpty(przeslanki(k, l))

    If prawda1 Then
        fakty(LiczbaFaktów) = konkluzje(k)
        LiczbaFaktów = LiczbaFaktów + 1
    End If
Next
```

Redukcja kolumn

(Advanced Software Testing: Volume 1)

- ▶ Jeśli wartości w jednym lub więcej warunków **nie mogą** wpływać na akcje dla dwóch lub więcej kombinacji warunków, możemy zredukować tablicę decyzyjną
- ▶ Wymaga to łączenia dwóch lub więcej kolumn
- ▶ Kroki
 1. Znajdź dwie kolumny, które mają te same działania na wyjściu
 2. Zastąp te warunki, które są różne w dwóch kolumnach przez „-” oznaczające „nie ma znaczenia” lub „nie wydarzy się”

Przykład: przed redukcją

WARUNKI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Prawdziwe konto?	T	T	T	T	T	T	T	T	N	N	N	N	N	N	N	N
Aktywne konto?	T	T	T	T	N	N	N	N	T	T	T	T	N	N	N	N
W limicie?	T	T	N	N	T	T	N	N	T	T	N	N	T	T	N	N
Poprawna lokalizacja?	T	N	T	N	T	N	T	N	T	N	T	N	T	N	T	N
AKCJE																
Zatwierdź?	T	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Zadzwoń do posiadacza?	N	T	T	T	N	T	T	T	N	N	N	N	N	N	N	N
Zadzwoń do dostawcy?	N	N	N	N	T	T	T	T	T	T	T	T	T	T	T	T

Przykład: po redukcji

WARUNKI	1	2	3	5	6	7	9
Prawdziwe konto?	T	T	T	T	T	T	N
Aktywne konto?	T	T	T	N	N	N	-
W limicie?	T	T	N	T	T	N	-
Poprawna lokalizacja?	T	N	-	T	N	-	-
AKCJE							
Zatwierdź?	T	N	N	N	N	N	N
Zadzwoń do posiadacza?	N	T	T	N	T	T	N
Zadzwoń do dostawcy?	N	N	N	T	T	T	T

Metody zapisu wiedzy

- ▶ Klasyczna logika zdań
- ▶ Logika predykatów pierwszego rzędu
- ▶ Logika atrybutowa
- ▶ Logika opisowa
- ▶ Języki opisu reguł

Logika

- ▶ Logika jest formalnym językiem reprezentacji informacji takim, w którym mogą być wyciągane wnioski
- ▶ Syntaktyka definiuje zdania w języku
- ▶ Semantyka definiuje "znaczenie" zdań; tzn. definiuje prawdziwość zdań w opisywanym świecie
- ▶ Np. język arytmetyki
 - ▶ $x + 2 > y$ jest zdaniem; $x^2 + y >$ nie jest zdaniem
 - ▶ $x + 2 \geq y$ jest prawdziwe wtw $x + 2$ jest nie mniejsze niż liczba y
 - ▶ $x + 2 > y$ jest prawdziwe w świecie, gdzie $x = 7$, $y = 1$
 - ▶ $x + 2 > y$ jest nieprawdziwe w świecie, gdzie $x = 0$, $y = 6$
- ▶ Chomsky:
 - ▶ *Bezbarwne zielone idee wściekle śpią*

Logiczna konsekwencja

- ▶ Logiczna konsekwencja oznacza, że jeden fakt wynika z innego:
 - ▶ $KB \models \alpha$
- ▶ α jest logiczną konsekwencją bazy wiedzy **KB** wtedy i tylko wtedy gdy α jest prawdziwe we wszystkich światach, w których **KB** jest prawdziwe
- ▶ Np. logiczną konsekwencją bazy wiedzy **KB** zawierającej „Pies jest studentem” i „Kot jest studentem” jest zdanie „Kot jest studentem i Pies jest studentem” natomiast nie jest konsekwencją fakt „Słoń jest studentem”
- ▶ Np. $4 = x + y$ jest logiczną konsekwencją $x + y = 4$
- ▶ Logiczna konsekwencja jest relacją pomiędzy zdaniami (syntaktyka) która opiera się na semantyce
- ▶ Uwaga: umysł analizuje syntaktykę (pewnego rodzaju)

Wnioskowanie

- ▶ $KB \models_i \alpha$ - zdanie α może być wyprowadzone z **KB** procedurą ***i***
- ▶ Poprawność: ***i*** jest poprawne, jeśli
 - ▶ zawsze kiedy $KB \models_i \alpha$, to też $KB \models \alpha$
- ▶ Pełność: ***i*** jest pełne jeśli
 - ▶ zawsze kiedy $KB \models \alpha$, to też $KB \models_i \alpha$
- ▶ Cel: zdefiniować logikę, w której można wyrazić możliwie jak najwięcej i dla której istnieje poprawna i pełna procedura dowodzenia
- ▶ Tzn. ta procedura odpowie na każde pytanie, które wynika z tego, co wiadomo w bazie wiedzy **KB**

Cechy charakteryzujące system logiczny

- ▶ poprawność
- ▶ zupełność
- ▶ niesprzeczność
- ▶ rozstrzygalność

Poprawność

- ▶ Pojęcie poprawności odnosi się do relacji pomiędzy składnią a semantyką systemu logicznego
- ▶ **System jest poprawny jeśli każda konkluzja posiadająca dowód w sensie składniowym jest prawdziwa w sensie semantycznym**
- ▶ W systemie poprawnym każda formuła, którą potrafimy dowieść manipulując na symbolach i wykorzystując aksjomaty teorii jest prawdą w sensie semantycznym, tj. nie można wskazać takiej interpretacji, że formuła jest fałszywa

System niepoprawny

- ▶ Załóżmy, że w pewnym systemie logicznym przyjmujemy aksjomat:

$$\neg(p \wedge q) = (\neg p \vee q)$$

- ▶ Wówczas za poprawne musielibyśmy uznać sformułowanie:
- ▶ *Ponieważ nieprawdą jest to, że Jan jest mężczyzną i jest studentem to prawdą jest, że nie jest mężczyzną lub jest studentem*

Zupełność

- ▶ Pojęcie zupełności również odnosi się do relacji pomiędzy składnią a semantyką
- ▶ **System logiczny jest zupełny, jeśli każda konkluzja prawdziwa w sensie semantycznym może być dowiedziona na gruncie składniowym**
- ▶ W systemie zupełnym nie może istnieć formuła prawdziwa w sensie semantycznym i nie posiadająca dowodu w sensie składniowym

System niezupełny

- ▶ Załóżmy, że usuwamy z pewnego systemu prawo podwójnego przeczenia

$$p \rightarrow \neg(\neg p)$$

- ▶ Wówczas nie moglibyśmy dowieść stwierdzenia:
- ▶ *Jeżeli Jan jest studentem to nieprawdą jest, że nie jest studentem*

Niesprzeczność

- ▶ Pojęcie niesprzeczności dotyczy już wyłącznie syntaktyki i zbioru aksjomatów
- ▶ **System jest niesprzeczny jeśli nie istnieje taka formuła, że w sensie składniowym można dowieść, iż jest ona zarówno prawdziwa jak i fałszywa**

Rozstrzygalność

- ▶ Pojęcie rozstrzygalności dotyczy również tylko syntaktyki i zbioru aksjomatów
- ▶ **System jest rozstrzygalny, jeśli dla każdej formuły można dowieść w sensie składniowym, czy jest prawdziwa czy fałszywa**

Kryteria doboru języka

- ▶ **Efektywność**, której miarą może być liczba symboli potrzebnych do reprezentacji wiedzy
- ▶ **Siła ekspresji** wyrażana w bogactwie operatorów logicznych oraz w poziomie szczegółowości
- ▶ **Adekwatność** rozumiana jako dopasowanie środków wyrazu, czyli siły ekspresji do poziomu złożoności wiedzy

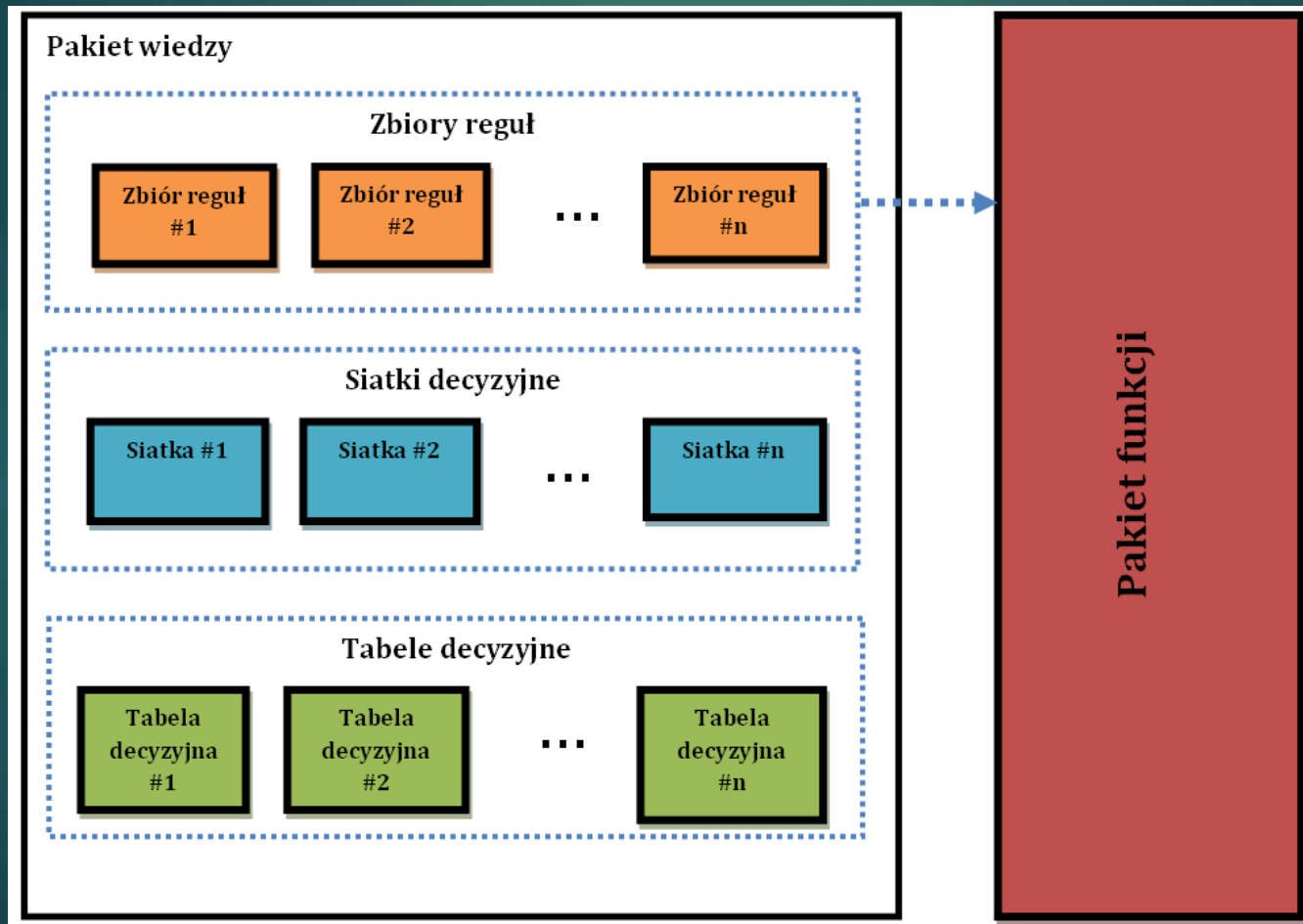
Reguły biznesowe

- ▶ Umożliwiają ekspertom przepisanie strategii przedsiębiorstwa na poziom operacyjny przy pomocy prostych stwierdzeń, które następnie mogą zautomatyzować reakcje na zdarzenia wewnątrz i w otoczeniu przedsiębiorstwa
- ▶ Reguły definiują zachowanie przedsiębiorstwa jako zbiór stwierdzeń logicznych
- ▶ Podstawową zaletą takiego podejścia do opisu procesów jest prostota i elastyczność reguł biznesowych, co pozwala na ich łatwe zastosowanie w różnych instytucjach i do różnych celów

Przykłady zastosowań

- ▶ Systemy wspomagające zarządzanie kontaktami z klientem (reguły opisują w tym przypadku ofertę, jaką w sposób zindywidualizowany można kierować do różnych klientów)
- ▶ Systemy oceny klientów (np. oceny zdolności kredytowej, oceny wartości klienta)
- ▶ Rozbudowane cenniki i regulaminy sprzedaży
- ▶ Systemy wspomagające zarządzanie łańcuchem dostaw (np. dobór dostawców)
- ▶ Kontrola kosztów i budżetowanie (reguły opisują zasady dekretowania kosztów, zasady budżetowania itp.)

System Rebit - Organizacja bazy wiedzy



Zmienne

- ▶ **Zmienne** służą do przechowywania danych, które w trakcie procesu wnioskowania mogą podlegać zmianom.
- ▶ Definicja zmiennej musi zawierać nazwę zmiennej oraz typ (dziedzinę) przechowywanych danych
- ▶ Nazwa zmiennej musi być unikalna w obrębie całej bazy wiedzy.
- ▶ Jako że zmienne są elementami, który przechowują dane, definicja zmiennej wymaga podania krotności (*multiplicity*) oraz porządku (*ordering*)

Funkcje

- ▶ **Funkcje** służą do specyfikacji operacji na danych
- ▶ Każda funkcja musi zwracać pewne dane
- ▶ Definicja funkcji wymaga podania krotności (*multiplicity*) oraz porządku (*ordering*) dla wszystkich parametrów funkcji oraz dla danych zwracanych przez funkcje
- ▶ Definicja funkcji zawiera jedynie informacje o danych wejściowych i wyjściowych, i w żaden sposób nie precyzuje wewnętrznych szczegółów algorytmu realizowanego przez funkcje

Stałe

- ▶ **Stałe** są najprostszym elementem języka, przeznaczonym do przechowywania danych, które nie podlegają zmianom
- ▶ W odróżnieniu od zmiennych i funkcji stałe nie wymagają odrębnej sekcji definiującej
- ▶ Są definiowane doraźnie w momencie pojawienia się w regule

Termy

- ▶ **Term** reprezentuje każdorazowe wystąpienie stałej, zmiennej lub funkcji w przestrzeni lub konkluzji reguły
- ▶ Termy będące wystąpieniami stałych noszą nazwę *termów stałych*, termy będące wystąpieniami zmiennych nazywa się *termami zmiennymi* i przez analogię na termy reprezentujące wystąpienie funkcji używa się określenia *termy funkcyjne*
- ▶ Argumentem termu funkcyjnego może być term stały lub zmienny

Termy - przykłady

► **Przykład 1.** Termy stałe

- 2 – term prosty reprezentujący stałą jednowartościową
- {„angielski”, „niemiecki”, „francuski”} – term prosty reprezentujący stałą wielowartościową nieuporządkowaną
- [3, 6, 1, 2] – term prosty reprezentujący stałą wielowartościową uporządkowaną

► **Przykład 2.** Termy zmienne

- *Podatek* – term prosty reprezentujący zmienną o nazwie *Podatek*

Termy - przykłady

► **Przykład 3.** Termy funkcyjne

- $add(X, 2)$ – term złożony reprezentujący funkcję dodawania; argumentami są term zmienny X i term stały 2
- $ln(Y)$ – term złożony reprezentujący funkcję do obliczania logarytmu naturalnego; argumentem jest term zmienny Y
- $max([3, 7, 1, 10])$ – term złożony reprezentujący funkcję do znajdowania maksimum; argumentem termu funkcyjnego jest term stały, wielowartościowy, uporządkowany
- $multiply(ln(Y), 3)$ – term złożony, zagnieżdżony reprezentujący funkcję mnożenia (term nadrzędny) oraz funkcję do obliczania logarytmu naturalnego (term podrzędny)

Formuły atomowe

- ▶ Elementarne wyrażenia logiczne, przyjmujące wartości prawdy lub fałszu
- ▶ Na formułę atomową składa się term lewostronny, term prawostronny oraz operator relacyjny
- ▶ W notacji Backusa-Naura (BNF) formułę atomową zapisuje się następująco:
 - ▶ **<LHTerm> <operator> <RHTerm>**
- ▶ gdzie:
 - ▶ **<LHTerm>** – term lewostronny
 - ▶ **<RHTerm>** – term prawostronny
 - ▶ **<operator>** – operator relacji

Formuły atomowe ze zmiennymi, stałymi i funkcjami

- ▶ $X = 2$
- ▶ $\text{Przychód} > \text{Koszty}$
- ▶ $\text{Stawka} \in [20, 30, 40]$
- ▶ $\text{add}(X, 2) \neq \text{ln}(Y)$

Reguły

► **RULE** *r*:

IF

<premise>

THEN

<conclusion>

► gdzie:

- *r* – identyfikator reguły
- *<premise>* – przesłanka reguły
- *<conclusion>* – konkluzja reguły

Przesłanki

- ▶ Przesłanki reguły mają postać koniunkcji formuł atomowych:
 - $\langle atom_#1 \rangle$ **and** $\langle atom_#2 \rangle$ **and** ... **and** $\langle atom_#n \rangle$
- ▶ gdzie :
 - $\langle atom_#i \rangle$ – i -ta formuła atomowa
 - **and** – operator koniunkcji

Konkluzje

- ▶ Konkluzja reguły zawiera jedną formułę atomową z dodatkowymi ograniczeniami
- ▶ Ograniczenia odnoszą się do termu występującego po lewej stronie formuły oraz dopuszczalnych operatorów
- ▶ Term lewostronny musi być termem zmiennym, a jedynym dopuszczalnym operatorem jest operator „=”, interpretowany w tym przypadku jako podstawienie wartości termu z prawej strony pod zmienną występującą po lewej stronie
- ▶ **<Variable> = <RHTerm>**
- ▶ gdzie:
 - <Variable> – zmienna
 - <RHTerm> – term

Reguły - przykłady

► RULE 1:

IF

$(X > 2)$ **and** $(Y = 2)$

THEN

$Z = \text{„dobry”}$

► RULE 2:

IF

$GatunekStali = \text{„St32”}$

THEN

$ZamiennikGatunku = \{\text{„St32”, „S235”, „DC01”}\}$

Tabele decyzyjne

Economy	=Good	=Good	=Good	=Good	=Good	=Good	=Good	=Good
Logistic	=Good	=Good	=Good	=Good	=Good	=Good	=Medium	=Medium
Quality	=No	=No	=No	=Yes	=Yes	=Yes	=No	=No
Confidence	=Good	=Medium	=Bad	=Good	=Medium	=Bad	=Good	=Medium
Provider	=Medium	=Medium	=Medium	=Good	=Good	=Medium	=Medium	=Medium

RULE 1:

IF

(Economy="Good") and (Logistic="Good") and (Quality="No") and (Confidence="Good")

THEN

Provider="Medium"

Siatki decyzyjne

<i>Dostępność</i>	Opel	Audi	Fiat
White	Tak	Nie	Tak
Blue	Nie	Tak	Tak
Red	Nie	Tak	Nie

RULE 3:

IF

(Marka="Audi")

THEN

Kolor={"Blue", "Red"}

Przykład – cennik blach stalowych

- ▶ Producent blach stalowych w kręgach uzależnia cenę produktów od następujących parametrów:
 - ▶ wymiary blachy (grubość, szerokość),
 - ▶ gatunek stali
 - ▶ masa kręgów,
 - ▶ wartość zamówienia,
 - ▶ dodatkowe wymagania,
 - ▶ ocena klienta itp.

Kreği stali



Cena bazowa (zł/tonę)

grubość/ szerokość	700-999	1000-1149	1150-1299	1300-1449	1450-1550
0,40- 0,49	3000	0	0	0	0
0,50-0,59	2900	2850	0	0	0
0,60-0,69	2800	2750	2700	0	0
0,70-0,74	2780	2720	2680	2600	0
0,75-2,50	2700	2670	2650	2600	2580

Dopłaty za gatunek stali (zł/tonę)

Gatunek stali	Dopłata
DC01, DC02	0
DC03, DC04	100
St37-2G, St37-3G	50
H260LA, H300LA	200
H340LA	250
DC01 EK	220

Dopłata za masę kręgów i dodatkowe wymagania (zł/tonę)

- ▶ Dopłata za masę kręgu poniżej 15 ton – 50 zł
- ▶ Dopłata za masę kręgu poniżej 7,5 tony – 100 zł
- ▶ Dopłata za dodatkowe zabezpieczenie powierzchni – 100 zł
- ▶ Dopłata z specjalne opakowanie kręgów – 150 zł

Ocena handlowa

Wartość zamówienia	Ocena klienta	Upust
< 50 000	nieznany	0
< 50 000	trudny	0
< 50 000	premium	5%
> 50 000	nieznany	2%
> 50 000	trudny	3%
> 50 000	premium	8%

Przykład zamówienia

- ▶ Wymiar blachy gr. 0,65 mm szerokość 1050 mm
- ▶ Gatunek DC03
- ▶ Kręgi 12 ton
- ▶ Wymagane specjalne zabezpieczenie
- ▶ Masa zamówienia 24 tony
- ▶ Klient premium

Kalkulacja

- ▶ Cena bazowa: 2 750 zł/tonę
- ▶ Dopłata za gatunek : 100 zł/tonę, cena: 2 850 zł/tonę
- ▶ Dopłata za masę kręgu 50 zł, cena: 2 900 zł/tonę
- ▶ Dopłata za specjalne zabezpieczenie 100 zł, cena: 3 000 zł/tonę
- ▶ Wartość zamówienia: 72 000 zł
- ▶ Upust 8%
- ▶ Koszt zamówienia: 66 240 zł

Definiujemy siatkę

0.9828 PL

Edycja Widok Narzędzia Okno Pomoc



Aktywny zbiór: CenyBazowe

Przepływ reguł

Składniki

Obiekty

Typy enumerowane

Typy zakresowe

Zmienne

Źródła wiedzy

Siatki decyzyjne

Tabele decyzyjne

Reguły

Siatki decyzyjne

Dodaj siatkę

Wybierz zbiór dla siatki CenyBazowe

☒ Ręcznie określ wiersze i kolumny

☐ Na podstawie gotowej enumeracji

Liczba kolumn

5

Liczba wierszy

5

Typ danych dla wierszy i kolumn

Zakres

Określ typ i krotność wartości w komórkach

Typ prosty



Zakres



Enumeracja



Real

Utwórz siatkę

Tworzymy siatkę

0.9828 PL

Widok Narzędzia Okno Pomoc



Aktywny zbiór: ☐

w reguł

Siatki decyzyjne

Edytuj siatkę

	<700;999>		<1000;1149>		<1150;1299>		<1300;1449>		<1450;1550>
<0.40;0.49>	3000	✓	0	✓	0	✓	0	✓	0
<0.50;0.59>	2900	✓	2850	✓	0	✓	0	✓	0
<0.60;0.69>	2800	✓	2750	✓	2700	✓	0	✓	0
<0.70;0.74>	2780	✓	2720	✓	2680	✓	2600	✓	0
<0.75;2.50>	2700	✓	2670	✓	2650	✓	2600	✓	2580

Nazwa siatki:

CenyBazowe

Opis:

Save Grid

Tworzymy reguły

0.9828 PL

dok Narzędzia Okno Pomoc

Aktywny zbiór: CenyBazowe_rules

Reguły

Nazwa	Opis	Podgląd
CenyBazowe_0		IF Col_Real between [700;999] AND Row_Real between [0.40;0.49] THEN CenyBazow
CenyBazowe_1		IF Col_Real between [1000;1149] AND Row_Real between [0.40;0.49] THEN CenyBaz
CenyBazowe_2		IF Col_Real between [1150;1299] AND Row_Real between [0.40;0.49] THEN CenyBaz
CenyBazowe_3		IF Col_Real between [1300;1449] AND Row_Real between [0.40;0.49] THEN CenyBaz
CenyBazowe_4		IF Col_Real between [1450;1550] AND Row_Real between [0.40;0.49] THEN CenyBaz
CenyBazowe_5		IF Col_Real between [700;999] AND Row_Real between [0.50;0.59] THEN CenyBazow
CenyBazowe_6		IF Col_Real between [1000;1149] AND Row_Real between [0.50;0.59] THEN CenyBaz
CenyBazowe_7		IF Col_Real between [1150;1299] AND Row_Real between [0.50;0.59] THEN CenyBaz
CenyBazowe_8		IF Col_Real between [1300;1449] AND Row_Real between [0.50;0.59] THEN CenyBaz
CenyBazowe_9		IF Col_Real between [1450;1550] AND Row_Real between [0.50;0.59] THEN CenyBaz
CenyBazowe_10		IF Col_Real between [700;999] AND Row_Real between [0.60;0.69] THEN CenyBazow
CenyBazowe_11		IF Col_Real between [1000;1149] AND Row_Real between [0.60;0.69] THEN CenyBaz
CenyBazowe_12		IF Col_Real between [1150;1299] AND Row_Real between [0.60;0.69] THEN CenyBaz
CenyBazowe_13		IF Col_Real between [1300;1449] AND Row_Real between [0.60;0.69] THEN CenyBaz
CenyBazowe_14		IF Col_Real between [1450;1550] AND Row_Real between [0.60;0.69] THEN CenyBaz
CenyBazowe_15		IF Col_Real between [700;999] AND Row_Real between [0.70;0.74] THEN CenyBazow
CenyBazowe_16		IF Col_Real between [1000;1149] AND Row_Real between [0.70;0.74] THEN CenyBaz

Wykorzystaj edytor



Dodaj regułę

Usuń regułę

Edytuj regułę

Importuj

Zamknij

Uruchamiamy wnioskowanie wprzód



Witaj, amaciol | [O programie](#) | [Pomoc](#) | [Wyloguj](#)

Menu



Wnioskowanie



Repozytorium



Sesje

Sesja wnioskownia

Sesja: 20121118132159641
Data rozpoczęcia: 2012-11-18 13:21:59
Data ostatniej operacji: 2012-11-18 13:21:59

Wnioskowanie: W przód
Baza wiedzy: Wykład1
Zbiór reguł: CenyBazowe_rules

Fakty początkowe

Wnioskowanie w przód w systemie REBIT polega na tworzeniu wszystkich możliwych faktów na podstawie pewnej wiedzy początkowej. W bieżącym kroku należy wybrać jedną lub więcej zmiennych. W kolejnym kroku użytkownik będzie proszony o utworzenie wiedzy początkowej poprzez nadanie wartości wybranym zmiennym.

Dostępne zmienne

=>

<=

Wybrane zmienne

Row_Real
Col_Real

Wprowadzamy znane fakty



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#)

Menu



Wnioskowanie



Repozytorium



Sesje

Sesja wnioskownia

Sesja: 20121118132159641
Data rozpoczęcia: 2012-11-18 13:21:59
Data ostatniej operacji: 2012-11-18 13:21:59

Wnioskowanie: W przód
Baza wiedzy: Wykład1
Zbiór reguł: CenyBazowe_rules

Dane

Opis:
Row_Real:
Typ: rzeczywisty

Opis:
Col_Real:
Typ: rzeczywisty

Dalej

Przerwij

Uzyskujemy wynik



Witaj, amaciol | [O programie](#) | [Pomoc](#) | [Wyl](#)

Menu



skowanie



ozytorium



Sesje

Sesja wnioskownia

Sesja: 20121118132159641
Data rozpoczęcia: 2012-11-18 13:21:59
Data ostatniej operacji: 2012-11-18 13:28:34

Wnioskowanie: W przód
Baza wiedzy: Wyklad1
Zbiór reguł: CenyBazowe_rules

Wynik

Zmienna	Wartość
CenyBazowe	2750

Fakty

Zmienna	Wartość	Kategoria
Row_Real	0,65	Zmienna wejściowa

Szczegóły

Info: zmienna: Row_Real (source), multiplicity: Single otrzymała wartość 0,65, multiplicity: Single
Info: Result: code = variable_value_set, Ustalono wartość: [0,65], dla zmiennej: Row_Real
Info: Row_Real

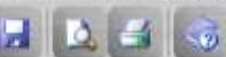
☐ Czy zapisać sesję?

Zakończ

Tworzymy zbiór reguł dla dopłat

EST 0.9828 PL

Edycja Widok Narzędzia Okno Pomoc



Aktywny zbiór: ☒

Opisy reguł

Pakiet

Reguły

Typy enumerowane

Typy zakresowe

Zmienne

dla wiedzy

Siatki decyzyjne

Tabele decyzyjne

Reguły

Pakiet

Zbiór Reguł	Nazwa zbioru reguł	Dialekt	Źródło
Siatki	CenyBazowe_rules	Basic	GeneratedFromGrid
Tabele			

Nazwa: DopłataZaGatunek

Źródło: UserDefined

Dialekt: Basic

Edytuj Usuń Aktywuj Dodaj Zbiór Reguł

Tworzymy zmienne

0.9828 PL

Widok Narzędzia Okno Pomoc

Aktywny zbiór: DopłataZaGatunek

Reguły Zmienne Dodaj zmienną

Nazwa: dopłata

Opis:

Typ: ☒ Typ prosty ☐ Typ enumerowany ☐ Typ zakresowy

Real

Krotność: Single

Dodaj Anuluj

W tym zmienne wielokrotne

0.9828 PL

Widok Narzędzia Okno Pomoc

Aktywny zbiór: DopłataZaGatunek

Reguły

Reguły

Zmienne

Dodaj zmienną

Nazwa GatunekStali

Opis

Typ

☒ Typ prosty

☐ Typ enumerowany

☐ Typ zakresowy

Krotność MultipleUnordered

Wszystkie niezbędne zmienne

0.9828 PL

Narzędzia Okno Pomoc

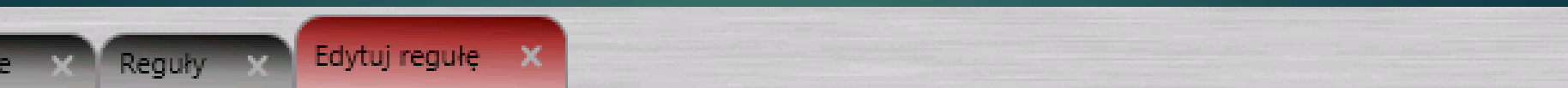
Aktywny zbiór: DopłataZaGatunek

Reguły Zmienne

Nazwa	Opis	Wielokrotność	Typ
dopłata		Single	Real
GatunekStali		MultipleUnordered	String
gatunek		Single	String

Dodaj zmienna Usuń zmienna Edytuj zmienna Import Zamknij

Reguły



```
1  RULES
2  RULE R1
3  IF gatunek in ["DC01";"DC02"]
4  THEN doplata = 0
5  RULE R2
6  IF gatunek in ["DC03";"DC04"]
7  THEN doplata = 100
8  RULE R3
9  IF gatunek in ["St372G";"St373G"]
10 THEN doplata = 50
11 RULE R4
12 IF gatunek in ["H260LA";"H300LA"]
13 THEN doplata = 200
14 RULE R5
15 IF gatunek = "H340LA"
16 THEN doplata = 250
17 RULE R6
18 IF gatunek = "DC01 EK"
19 THEN doplata = 220
20
21
22
23
```

Testujemy reguły – wnioskując wprzód



Witaj, amaciol | O programie | P

Wnioskowanie - zbiory reguł

W przód

W tył

Pełne

Baza wiedzy ▲	Baza wiedzy - opis	Zbiór reguł	Zbiór reguł - opis	Kategoria	Akcja	
baza		CenyBazowe_rules		Siatka	W przód	W
bazaTest5		Test1		Reguły	W przód	W
Dopłaty		Ceny_rules		Siatka	W przód	W
Dopłaty		DopłatyZaGatunek		Reguły	W przód	W
Siatka		Ceny_rules		Siatka	W przód	W
Wykład1		CenyBazowe_rules		Siatka	W przód	W
Wykład1		DoplataZaGatunek		Reguły	W przód	W

Wybieramy zmienne, których wartość znamy



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#) | [Wylo](#)

Sesja wnioskownia

Sesja: 20121119114002982
Data rozpoczęcia: 2012-11-19 11:40:02
Data ostatniej operacji: 2012-11-19 11:40:02

Wnioskowanie: W przód
Baza wiedzy: Wyklad1
Zbiór reguł: DopłataZaGatunek

Fakty początkowe

Wnioskowanie w przód w systemie REBIT polega na tworzeniu wszystkich możliwych faktów na podstawie pewnej wiedzy początkowej. W bieżącym kroku należy wybrać zmienne, które będą stanowić wiedzę początkową.

Dostępne zmienne

GatunekStali

=>

<=

Wybrane zmienne

gatunek

Zmienna

Nazwa: GatunekStali

Opis:

Wprowadzamy wartość



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#)

Sesja wnioskownia

Sesja: 20121119114002982
Data rozpoczęcia: 2012-11-19 11:40:02
Data ostatniej operacji: 2012-11-19 11:40:02

Wnioskowanie: W przód
Baza wiedzy: Wykład1
Zbiór reguł: DopłataZaGatunek

Dane

Opis:

gatunek: DC03

Typ: tekstowy

Dalej

Przerwij

Uzyskujemy odpowiedź



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#)

Sesja wnioskownia

Sesja: 20121119114002982
Data rozpoczęcia: 2012-11-19 11:40:02
Data ostatniej operacji: 2012-11-19 11:42:03

Wnioskowanie: W przód
Baza wiedzy: Wyklad1
Zbiór reguł: DopłataZaGatunek

Wynik Fakty Szczegóły

Zmienna	Wartość
dopłata	100

☐ Czy zapisać sesję?

Reguły dotyczące dopłat za masę kręgu

Reguły

Edytuj regułę

```
1  RULES
2  RULE R11
3  IF MasaKregu > 15
4  THEN DoplataZaMase = 0
5  RULE R12
6  IF MasaKregu between [7.5;15]
7  THEN DoplataZaMase = 50
8  RULE R13
9  IF MasaKregu < 7.5
10 THEN DoplataZaMase = 100
11
12
13
14
15
16
17
18
19
20
21
22
```

Wnioskowanie wprzód



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#) |

Sesja wnioskownia

Sesja: 20121119122422721

Data rozpoczęcia: 2012-11-19 12:24:22

Data ostatniej operacji: 2012-11-19 12:24:22

Wnioskowanie: W przód

Baza wiedzy: Wyklad11

Zbiór reguł: DopłatyZaMaseKregu

Fakty początkowe

Wnioskowanie w przód w systemie REBIT polega na tworzeniu wszystkich możliwych faktów na podstawie pewnej wiedzy początkowej. W bieżącym kroku należy wybrać zmienne, które będą stanowić wiedzę początkową.

Dostępne zmienne

=>

<=

Wybrane zmienne

MasaKregu

Dalej

Przerwij

Anuluj

Wprowadzamy dane



Witaj, amaciol | [O programie](#) |

Sesja wnioskownia

Sesja: 20121119122422721
Data rozpoczęcia: 2012-11-19 12:24:22
Data ostatniej operacji: 2012-11-19 12:24:22

Wnioskowanie: W przód
Baza wiedzy: Wyklad11
Zbiór reguł: DopłatyZaMaseKregu

Dane

Opis: Masa kręgu w tonach

MasaKregu:

Pole musi mieć wartość

Typ: rzeczywisty

Dalej

Przerwij



Wynik



Witaj, amaciol | [O programie](#) | [P](#)

Sesja wnioskownia

Sesja: 20121119122422721
Data rozpoczęcia: 2012-11-19 12:24:22
Data ostatniej operacji: 2012-11-19 12:27:30

Wnioskowanie: W przód
Baza wiedzy: Wyklad11
Zbiór reguł: DopłatyZaMaseKregu

Wynik

Fakty

Szczegóły

Zmienna	Wartość
DoplataZaMase	50

☐ Czy zapisać sesję?



Reguły dotyczące specjalnych zabezpieczeń

Reguły X

Edytuj regułę X

RULES

RULE R21

IF SpecjalneZabezpieczenie = true

THEN DopłataZaZabezpieczenie = 100

RULE R22

IF SpecjalneZabezpieczenie = false

THEN DopłataZaZabezpieczenie = 0

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Testowe wnioskowanie wprzód



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#) |

Sesja wnioskownia

Sesja: 20121119131143576
Data rozpoczęcia: 2012-11-19 13:11:43
Data ostatniej operacji: 2012-11-19 13:11:43

Wnioskowanie: W przód
Baza wiedzy: Wyklad11
Zbiór reguł: DopłatySpecjalne

Fakty początkowe

Wnioskowanie w przód w systemie REBIT polega na tworzeniu wszystkich możliwych faktów na podstawie pewnej wiedzy początkowej. bieżącym kroku należy wybrać zmienne, które będą stanowić wiedzę początkową.

Dostępne zmienne

=>

<=

Wybrane zmienne

SpecjalneZabezpieczenie

Wprowadzanie danych



Witaj, amaciol | O programie | Pomoc

Sesja wnioskownia

Sesja: 20121119131427581

Data rozpoczęcia: 2012-11-19 13:14:27

Data ostatniej operacji: 2012-11-19 13:14:27

Wnioskowanie: W przód

Baza wiedzy: Wykład11

Zbiór reguł: DopłatySpecjalne

Dane

Opis:

SpecjalneZabezpieczenie:

prawda
fałsz

Pole musi mieć wartość

Dalej

Przerwij

Wynik



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#)

Sesja wnioskownia

Sesja: 20121119131427581

Data rozpoczęcia: 2012-11-19 13:14:27

Data ostatniej operacji: 2012-11-19 13:15:42

Wnioskowanie: W przód

Baza wiedzy: Wyklad11

Zbiór reguł: DopłatySpecjalne

Wynik

Fakty

Szczegóły

Zmienna	Wartość
DoplataZaZabezpieczenie	100

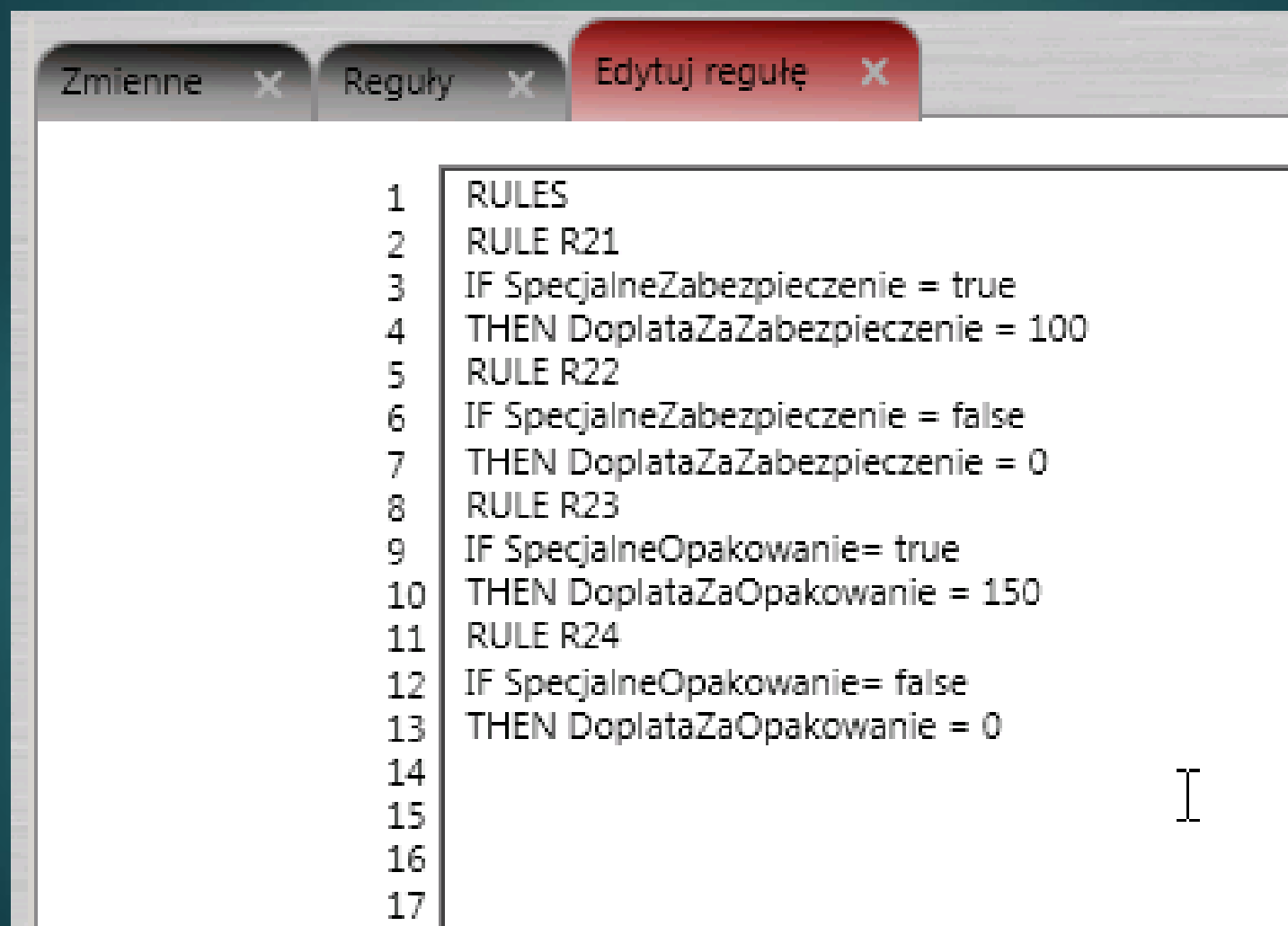
☐ Czy zapisać sesję?



Dodajemy zmienne

Zmienne ×					
Reguły ×					
Edytuj regułę ×					
	Nazwa	Opis	Wielokrotność	Typ	
	SpecjalneZabezpieczenie		Single	Boolean	
	DoplataZaZabezpieczenie		Single	Real	
	SpecjalneOpakowanie		Single	Boolean	
	DoplataZaOpakowanie		Single	Real	

Dodajemy reguły



The screenshot shows a software interface for editing rules. At the top, there are three tabs: 'Zmienne' (Variables), 'Reguły' (Rules), and 'Edytuj regułę' (Edit rule), which is currently selected and highlighted in red. Below the tabs is a large text area containing a list of rules, numbered 1 through 17. The rules are as follows:

```
1 RULES
2 RULE R21
3 IF SpecjalneZabezpieczenie = true
4 THEN DopłataZaZabezpieczenie = 100
5 RULE R22
6 IF SpecjalneZabezpieczenie = false
7 THEN DopłataZaZabezpieczenie = 0
8 RULE R23
9 IF SpecjalneOpakowanie = true
10 THEN DopłataZaOpakowanie = 150
11 RULE R24
12 IF SpecjalneOpakowanie = false
13 THEN DopłataZaOpakowanie = 0
14
15
16
17
```

A vertical cursor is positioned at the end of line 17.

Wybieramy dane do testu



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#) |

Sesja wnioskownia

Sesja: 20121119173108269
Data rozpoczęcia: 2012-11-19 17:31:08
Data ostatniej operacji: 2012-11-19 17:31:08

Wnioskowanie: W przód
Baza wiedzy: Wykład11
Zbiór reguł: DopłatySpecjalne

Fakty początkowe

Wnioskowanie w przód w systemie REBIT polega na tworzeniu wszystkich możliwych faktów na podstawie pewnej wiedzy początkowej. W bieżącym kroku należy wybrać zmienne, które będą stanowić wiedzę początkową.

Dostępne zmienne

=>

<=

Wybrane zmienne

SpecjalneZabezpieczenie
SpecjalneOpakowanie

Dalej

Przerwij

A

Wprowadzamy dane



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#) | [Wyl](#)

Sesja wnioskownia

Sesja: 20121119173108269
Data rozpoczęcia: 2012-11-19 17:31:08
Data ostatniej operacji: 2012-11-19 17:31:08

Wnioskowanie: W przód
Baza wiedzy: Wyklad11
Zbiór reguł: DopłatySpecjalne

Dane

Opis:
SpecjalneZabezpieczenie:
Typ: *logiczny*

Opis:
SpecjalneOpakowanie:
Typ: *logiczny*

Dalej

Przerwij

Anuluj

Uzyskujemy wynik



[Witaj, amaciol](#) | [O programie](#) | [Pomoc](#)

Sesja wnioskownia

Sesja: 20121119173108269
Data rozpoczęcia: 2012-11-19 17:31:08
Data ostatniej operacji: 2012-11-19 17:33:18

Wnioskowanie: W przód
Baza wiedzy: Wyklad11
Zbiór reguł: DopłatySpecjalne

Wynik

Fakty

Szczegóły

Zmienna	Wartość
DoplataZaZabezpieczenie	100
DoplataZaOpakowanie	0

☐ Czy zapisać sesję?



I historię



Witaj, amaciol | [O programie](#) | [P](#)

Sesja wnioskownia

Sesja: 20121119173108269

Data rozpoczęcia: 2012-11-19 17:31:08

Data ostatniej operacji: 2012-11-19 17:33:18

Wnioskowanie: W przód

Baza wiedzy: Wyklad11

Zbiór reguł: DopłatySpecjalne

Wynik

Fakty

Szczegóły

Zmienna	Wartość	Kategoria
SpecjalneZabezpieczenie	true	Zmienna wej
SpecjalneOpakowanie	false	Zmienna wej
DopłataZaZabezpieczenie	100	Zmienna wyn
DopłataZaOpakowanie	0	Zmienna wyn

☐ Czy zapisać sesję?



Projektujemy tabelę decyzyjną

Typy enumerowane

Tabele decyzyjne

Dodaj tabelę

Wybierz zbiór dla tabeli: OcenaHandlowa

Określ zmienne dla tabeli decyzyjnej:

Przesłanki

Nazwa	Określenie typu	Typ	Wielokrotność
Wartosc	BasicType	Real	Single
Klient	EnumType	Klient	Single

Dodaj nową pozycję

Konkluzja

Nazwa	Określenie typu	Typ	Wielokrotność
Upust	BasicType	Real	Single

Wypełnij iloczynem kartezjańskim ☐

Utwórz tabelę

Wpisujemy dane

enummerowane		Tabele decyzyjne		Nowa tabela		
koszt	< ☒ 50000	< ☒ 50000	< ☒ 50000	>= ☒ 50000	>= ☒ 50000	>= ☒ 50000
trudność	= ☒ nieznan ☒	= ☒ trudny ☒	= ☒ premium ☒	= ☒ nieznan ☒	= ☒ trudny ☒	= ☒ premium ☒
ryzyko	= ☒ 0	= ☒ 0	= ☒ 0.05	= ☒ 0.02	= ☒ 0.03	= ☒ 0.08

Nazwa tabeli:

OcenaHandlowa

Opis:

Zapisz tabelę

Reguły

Reguły X

Nazwa	Opis	Podgląd
Upust_0	Generated from decision table"OcenaHandlowa" using simple converter	IF Wartosc < 50000 AND Klient
Upust_1	Generated from decision table"OcenaHandlowa" using simple converter	IF Wartosc < 50000 AND Klient
Upust_2	Generated from decision table"OcenaHandlowa" using simple converter	IF Wartosc < 50000 AND Klient
Upust_3	Generated from decision table"OcenaHandlowa" using simple converter	IF Wartosc >= 50000 AND Klie
Upust_4	Generated from decision table"OcenaHandlowa" using simple converter	IF Wartosc >= 50000 AND Klie
Upust_5	Generated from decision table"OcenaHandlowa" using simple converter	IF Wartosc >= 50000 AND Klie

Wykorzystaj edytor ☒

Dodaj regułę

Usuń regułę

Edytuj regułę

Importuj

Zamknij

Wprowadzamy dane



Witaj, amaciol | [O programie](#) | [Pomoć](#)

Sesja wnioskownia

Sesja: 20121119185754389
Data rozpoczęcia: 2012-11-19 18:57:54
Data ostatniej operacji: 2012-11-19 18:57:54

Wnioskowanie: W przód
Baza wiedzy: Wyklad12
Zbiór reguł: OcenaHandlowa_rules

Dane

Opis:
Wartosc:

Typ: rzeczywisty

Opis:
Klient:

Typ: wyliczeniowy

Dalej

Przerwij

I uzyskujemy wynik



Witaj, amaciol | O programie | P

Menu

Wnoskowanie

Wzrostorium

Sesje

Sesja wnioskownia

Sesja: 20121119185754389
Data rozpoczęcia: 2012-11-19 18:57:54
Data ostatniej operacji: 2012-11-19 19:00:03

Wnoskowanie: W przód
Baza wiedzy: Wyklad12
Zbiór reguł: OcenaHandlowa_rules

Wynik

Fakty

Szczegóły

Zmienna	Wartość
Upust	0,08

☐ Czy zapisać sesję?