

# **REGUŁOWO-MODELOWE SKORUPOWE SYSTEMY EKSPERTOWE**

**Część 3: Systemy elementarne i rozwinięte z  
ocenami**

**Antoni Niederliński  
Uniwersytet Ekonomiczny  
w Katowicach**

**[antoni.niederlinski@ue.katowice.pl](mailto:antoni.niederlinski@ue.katowice.pl)**

# Koniec pewnego rozdziału...

Wszystkie systemy ekspertowe dotychczas omawiane korzystały z baz wiedzy wywodzących się z dokumentów

**logicznie i arytmetycznie deterministycznych  
(LAD)**

Dokument papierowy jest typu LAD, jeżeli wynik procedowania tego dokumentu, niezależnie od osoby procedującej, daje dla tych samych danych takie same wnioski i takie same wyniki.

# Ukryty niedeterminizm (1)

Dokument „papierowy” typu LDA może jednak zawierać **ukryty niedeterminizm**, którego źródłem są:

- modele relacyjne, np.  
model\_r(1, "bez warunku", "Dobre zabezpieczenie",  
" <=, <=" , ["A", "zabezpieczenie", "B"], 1)
- warunki dopytywalne, np.  
"dobra reputacja"



# Ukryty niedeterminizm (2)

Determinizm logiczno-arytmetyczny - właściwość dokumentów "papierowych", których procedowanie dla jednakowych danych wejściowych daje zawsze jednakowe wnioski (determinizm logiczny) i jednakowe wyniki (determinizm arytmetyczny), niezależnie od osoby procedującej.

Niedeterminizm ukryty: nieokreśloność warunków dopytywalnych i przedziały wartości dla modeli relacyjnych.

# Ukryty niedeterminizm (1)

Dla warunku dopytywalnego, np.:

"dobra reputacja"

chcemy zadeklarować ocenę tego warunku.

W tym celu posługujemy się modelem:

"Ocena\_dobra reputacja"

# Ukryty niedeterminizm (1a)

"Ocena\_Dobre reputacja"

Ocena\_ jest słowem kluczowym generującym pytanie o wartość zmiennej Dobra reputacja

Pytanie ma postać:

`Określ wartość argumentu:  
Ocena\_Dobre reputacja`



# Ukryty niedeterminizm (2)

Uwzględnienie ukrytego niedeterminizmu, np. przez deklarowanie **ocen** dla wniosków modeli relacyjnych, np. dla modelu relacyjnego:

```
model_r(100,"bez warunku","Dobre zabezpieczenie",  
        "<=,<=","A","zabezpieczenie","B",1)
```

chcemy zadeklarować ocenę zmiennej:

**"Dobre zabezpieczenie"**

W tym celu posługujemy się modelem:

**"Ocena\_Dobre zabezpieczenie"**

# Ukryty niedeterminizm (2a)

"Ocena\_ Dobre zabezpieczenie"

Ocena\_ jest słowem kluczowym generującym pytanie o wartość zmiennej **Dobre zabezpieczenie**

**Pytanie ma postać:**

**'Określ ocenę spełnienia wniosku na skali 1-10:'**



# Oceny (1)

- Oceny zmiennych logicznych są liczbami rzeczywistymi z przedziału  $[1, \dots, 10]$
- Ocena = 10 jest oceną bardzo dobrą,  
Ocena = 1 jest oceną dostateczną.
- **Oceny są ortogonalne względem logiki**

# Oceny (2) - ortogonalność

- Dedukcja logiczna dokonywana przez system wnioskujący nie zniekształca przebiegu tworzenia ocen przez tenże system wnioskujący.
- Tworzenie ocen przez system wnioskujący nie zniekształca *przebiegu dedukcji* logicznej dokonywanej przez tenże system wnioskujący.

# Oceny dla zmiennych logicznych: warunków dopytywalnych i wniosków reguł

Zachowane wszystkie prawa algebry logiki

Zmienna logiczna bez oceny



Zmienna logiczna z oceną



**Zmienna logiczna swobodna**

**Zmienna logiczna prawdziwa**

**Zmienna logiczna nieprawdziwa**

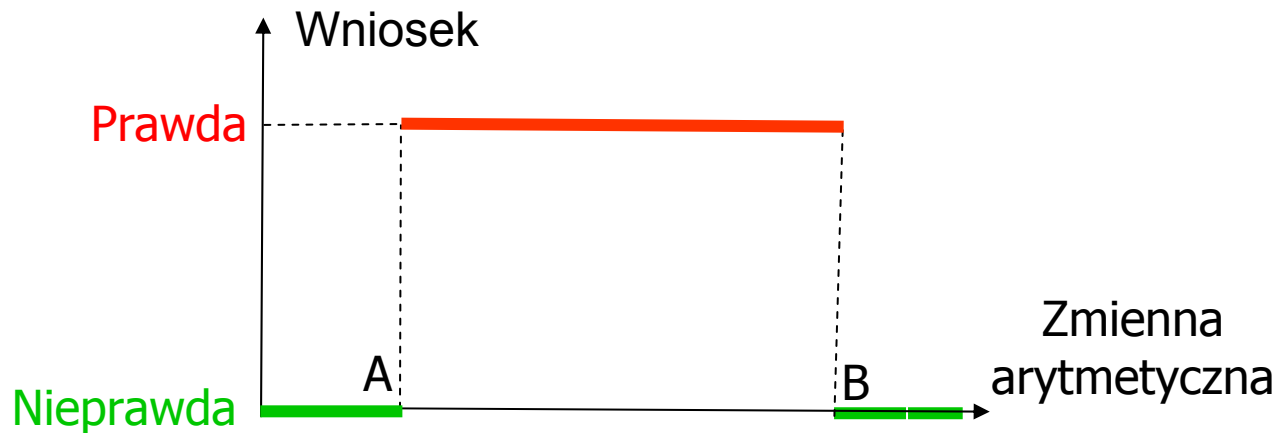


Ocena zmiennej logicznej



# Oceny dla wniosków modeli relacyjnych

'Wniosek' jest prawdą jeżeli  $A \leq \text{Zmienna arytmetyczna} \leq B$

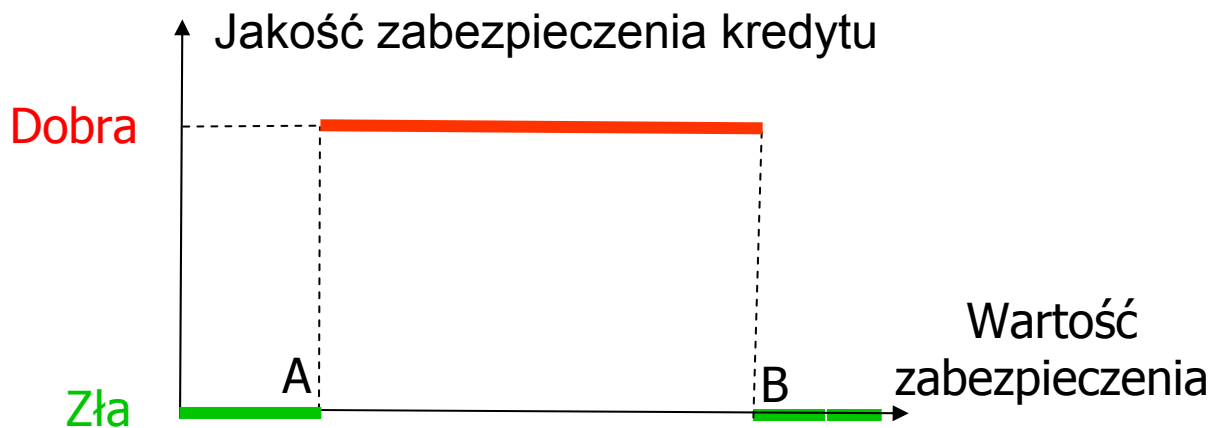


$\left. \begin{array}{l} 1 \quad 10 \\ 10 \quad 1 \end{array} \right\}$  Dwie definicje skali ocen dla zmiennej logicznej 'Wniosek'

# Oceny (5) – wybór skali ocen

'Wniosek' określa jakość zabezpieczenia kredytu

Zabezpieczenie jest dobre jeżeli  $A \leq \text{Wartość zabezpieczenia} \leq B$



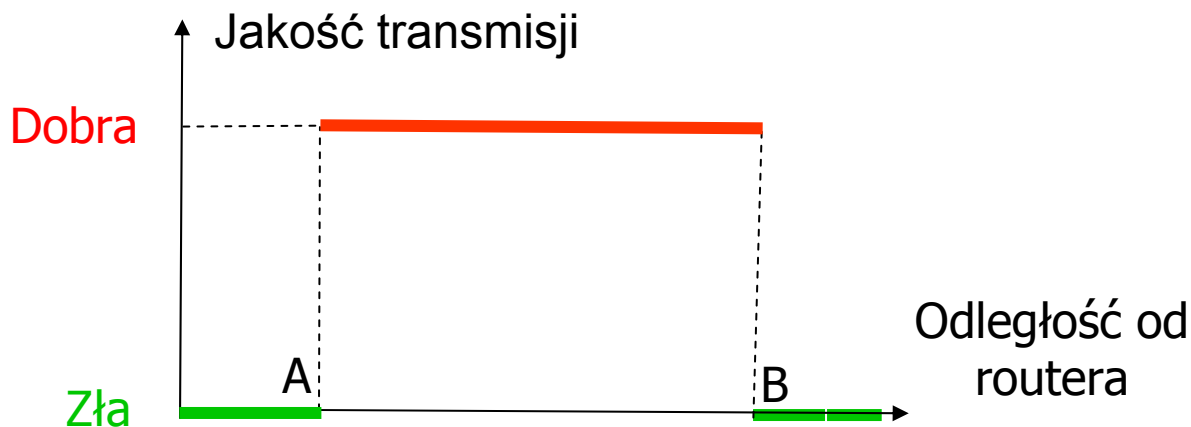
1 10

Im bliżej górnego kresu tym lepsze zabezpieczenie

# Oceny (6) – wybór skali ocen

‘Wniosek’ określa jakość transmisji bezprzewodowej

Transmisja jest dobra jeżeli  $A \leq \text{Odległość od routera} \leq B$



10 1

Im bliżej dolnego kresu tym lepsza transmisja



Co zrobić jeżeli :

'Wniosek' jest prawdą jeżeli  $A \leq$  Zmienna arytmetyczna

'Wniosek' jest prawdą jeżeli Zmienna arytmetyczna  $\leq B$

Żadna zmienna arytmetyczna opisująca rzeczywistość nie jest nigdy nieograniczenie duża lub nieograniczenie mała.

Właściwość ta dała o sobie znać przy okazji wprowadzania ocen.

A więc relacje:

'Wniosek' jest prawdą jeżeli  $A \leq$  Zmienna arytmetyczna

'Wniosek' jest prawdą jeżeli Zmienna arytmetyczna  $\leq B$

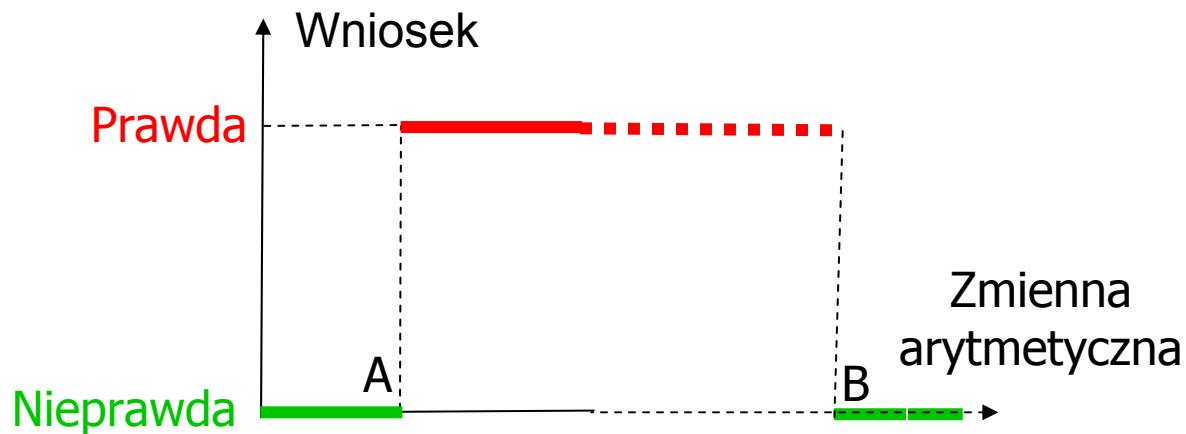
należy zastąpić relacją:

'Wniosek' jest prawdą jeżeli  $A \leq$  Zmienna arytmetyczna  $\leq B$

# Zmienna arytmetyczna jest ograniczona od dołu

Jej ocena wymaga uzupełnienia zmiennej arytmetycznej domyślnym kresem górnym B.

‘Wniosek’ jest prawdą jeżeli  $A \leq$  Zmienna arytmetyczna

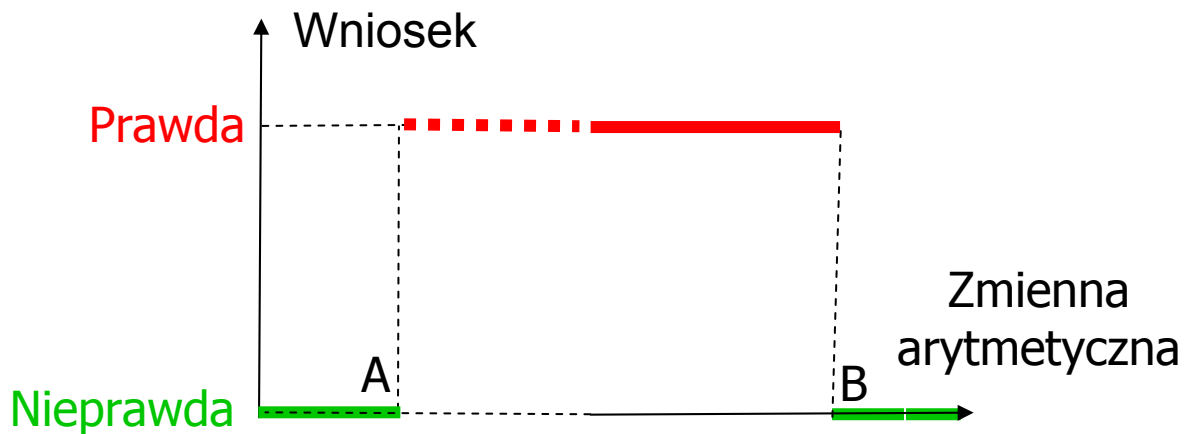


$\left. \begin{array}{l} \overbrace{1 \quad 10} \\ \underbrace{10 \quad 1} \end{array} \right\} \text{Dwie definicje skali ocen dla}$   
 $\text{zmiennej logicznej 'Wniosek'}$

# Zmienna arytmetyczna jest ograniczona od góry

Jej ocena wymaga uzupełnienia zmiennej arytmetycznej domyślnym kresem dolnym A.

'Wniosek' jest prawdą jeżeli Zmienna arytmetyczna  $\leq B$



$\left. \begin{array}{l} 1 \quad 10 \\ 10 \quad 1 \end{array} \right\}$  Dwie definicje skali ocen dla zmiennej logicznej 'Wniosek'



# Cechy ocen

Stosowanie ocen nie wymaga rezygnacji z *zasady wyłączonego środka* ("*tertium non datur*"), zgodnie z którą **każde zdanie jest albo prawdą, albo nieprawdą.**

Stosowanie *ocen* nie narusza tej zasady: wnioski i warunki dopytywalne pozostają albo prawdziwe, albo nieprawdziwe, lecz zyskują dodatkowy (*nie-arystotelesowy*) atrybut, którym jest ich *ocena*.

# Cechy ocen (4)

Oceny można deklarować dla:

- 1) warunków dopytywalnych
- 2) wniosków modeli relacyjnych

Oceny te mogą być propagowane na oceny wniosków reguł.

# Oceny (4a)

Ocena zmiennej logicznej o nazwie "zmienna"  
jest zmienną arytmetyczną o nazwie

"Wartość\_zmienna"

i wartości z przedziału  $[1, \dots, 10]$ ,

Np. dla zmiennej logicznej:

"Dobre zabezpieczenie"

może być:

"Wartość\_Dobre zabezpieczenie" = "7"



# Oceny (4b)

Dla wprowadzenia ocen użyteczny jest model arytmetyczny

model(Nr, "Warunek Startowy", "Wartość\_zmienna", "Ocena\_zmienna",

"=", "0", 1)

dla operacji:

"Ocena\_zmienna"

Generuje zapytanie o wartość argumentu "zmienna" i przyrównujący odpowiedź do wartości zmiennej "Wartość\_zmienna".

Ocena\_ - jest słowem kluczowym

# Oceny (5)

```
model (110,"bez warunku","Wniosek",  
      "Argument",">=","Kres_dolny",1)
```

Dla tego modelu można wprowadzić ocenę prawdziwego wniosku "Wniosek" za pomocą modelu arytmetycznego podstawowego o postaci:

```
model(111,"Wniosek","Wartość Wniosek",  
      "Ocena_Wniosek","=","0",1)
```

Model ten generuje zapytanie o ocenę zmiennej logicznej "Wniosek" i przyrównujący odpowiedź do wartości zmiennej arytmetycznej "Wartość Wniosek".

# Oceny (6)

Inny użyteczny model:

```
model(Nr,"Warunek Startowy","Liczba",  
      "Nr_reguły","liczba_warunków_reguły","0",1)
```

"liczba\_warunków\_reguły" - operacja arytmetyczna wyznaczająca wartość zmiennej "Liczba" jako liczbę warunków reguły "Nr\_reguły"

Obliczanie liczby warunków reguły 'na piechotę' może czasem być kłopotliwym źródłem pomyłek.



# Propagacja oceny warunków na ocenę wniosku

Można ją dokonać na szereg różnych sposobów:

1. Wartość wniosku można wyznaczyć jako średnią wartości warunków.
2. Wartość wniosku można wyznaczyć jako minimum wartości warunków.
3. Wartość wniosku można wyznaczyć jako maksimum wartości warunków.

# Wartość wniosku jako średnia wartości warunków (1)

Dana są reguły:

`reguła(1, "Wniosek", ["Warunek_11", "Warunek_21",  
"Warunek_31"], 1)`

`reguła(2, "Wniosek", ["Warunek_12", "Warunek_22",  
"Warunek_32"], 1)`

z których tylko jedna może być spełniona.

# Wartość wniosku jako średnia wartości warunków (2)

Jeżeli chcemy obliczyć wartość wniosku "Wniosek", musimy zastąpić je regułami:

**reguła(1, "Wniosek\_1", ["Warunek\_11", "Warunek\_21",  
"Warunek\_31"], 1)**

**reguła(2, "Wniosek\_2", ["Warunek\_12", "Warunek\_22",  
"Warunek\_32"], 1)**

**reguła(3, "Wniosek", ["Wniosek\_1",  
"Wartość Wniosek\_1 > 0"], 1)**

**reguła(4, "Wniosek", ["Wniosek\_2",  
"Wartość Wniosek\_2 > 0"], 1)**



# Wartość wniosku jako średnia wartości warunków (3)

1) Wyznaczamy sumę ocen warunków dla "Wniosek\_1":

```
model_r(101,"Wniosek_1","Wartość Wniosek_1","+",  
        ["Ocena_Warunek_11",  
         "Ocena_Warunek_21",  
         "Ocena_Warunek_31"],1)
```

2) Wyznaczamy liczbę warunków reguły o numerze 1:

```
model(102,"Wniosek_1","Liczba warunków Wniosek_1","1",  
       "liczba_warunków_reguły", "0",1)
```

# Wartość wniosku jako średnia wartości warunków (3)

3) Wyznaczamy średnią ocenę dla "Wniosek\_1":

```
model_r(103,"Wniosek_1", "Średnia ocena Wniosek_1",  
        "Wartość Wniosek_1","/",  
        " Liczba warunków Wniosek_1",1)
```

4) Zaokrąglamy średnią ocenę do dwóch pozycji po przecinku:

```
model(104,"Wniosek_1","Wartość finalna Wniosek_1 ",  
      "Średnia ocena Wniosek_1", „zaokrąglenie do N”,  
      "2", 1)
```

# Wartość wniosku jako średnia wartości warunków (4)

5) Dodatkowo tworzymy model:

```
model(103,"Wniosek_1",  
      "Wartość finalna Wniosek_1 > 0",  
      "Wartość finalna Wniosek_1", ">","0",1)
```

dzięki któremu przy wnioskowaniu wstecz weryfikującym hipotezę **"Wniosek\_1"** zostanie również wyznaczona **"Wartość finalna Wniosek\_1"**



# Wartość wniosku jako średnia wartości warunków (5)

1) Wyznaczamy sumę ocen warunków dla "Wniosek\_2":

```
model_r(1010,"Wniosek_2","Wartość Wniosek_2","+",  
        ["Ocena_Warunek_12",  
         "Ocena_Warunek_22",  
         "Ocena_Warunek_32"],1)
```

2) Wyznaczamy liczbę warunków reguły o numerze 2:

```
model(1020,"Wniosek_2","Liczba warunków Wniosek_2", "2",  
       "liczba_warunków_reguły", "0",1)
```

# Wartość wniosku jako średnia wartości warunków (6)

5) Dodatkowo tworzymy model:

```
model(1030,"Wniosek_2",  
      "Wartość finalna Wniosek_2 > 0",  
      "Wartość finalna Wniosek_2", ">","0",1)
```

dzięki któremu przy wnioskowaniu wstecz weryfikującym hipotezę "**Wniosek\_2**" zostanie również wyznaczona "**Wartość finalna Wniosek\_2**"

# Wartość wniosku jako minimum wartości warunków (1)

Dana są reguły:

```
reguła(1, "Wniosek", ["Warunek_11", "Warunek_21",  
"Warunek_31"], 1)
```

```
reguła(2, "Wniosek", ["Warunek_12", "Warunek_22",  
"Warunek_32"], 1)
```



# Wartość wniosku jako minimum wartości warunków (2)

Jeżeli chcemy obliczyć wartość wniosku "Wniosek", musimy zastąpić je regułami:

reguła(1, "Wniosek\_1", ["Warunek\_11", "Warunek\_21", "Warunek\_31"], 1)

reguła(2, "Wniosek\_2", ["Warunek\_12", "Warunek\_22", "Warunek\_32"], 1)

reguła(3, "Wniosek", ["Wniosek\_1", "Wartość Wniosek\_1 > 0"], 1)

reguła(4, "Wniosek", ["Wniosek\_2", "Wartość Wniosek\_2 > 0"], 1)

# Wartość wniosku jako minimum wartości warunków (3)

1) Wyznaczamy minimum ocen warunków:

```
model_r(101,"Wniosek_1","Wartość Wniosek_1", "min_list",  
        ["Ocena_Warunek_11",  
         "Ocena_Warunek_21",  
         "Ocena_Warunek_31"],1)
```

2) Zaokrąglamy wartość wniosku **Wniosek\_1** do 2 pozycji po przecinku:

```
model(102,"Wniosek_1","Wartość finalna Wniosek_1",  
      "Wartość Wniosek_1","zaokrąglenie_do_N","2", 1)
```

# Wartość wniosku jako minimum wartości warunków (4)

5) Dodatkowo tworzymy model:

```
model(103,"Wniosek_1",  
      "Wartość finalna Wniosek_1 > 0",  
      "Wartość finalna Wniosek_1", ">","0",1)
```

dzięki któremu przy wnioskowaniu wstecz weryfikującym hipotezę **"Wniosek\_1"** zostanie również wyznaczona **"Wartość finalna Wniosek\_1"**



# Wartość wniosku jako minimum wartości warunków (5)

1) Wyznaczamy minimum ocen warunków:

```
model_r(1010,"Wniosek_2","Wartość Wniosek_2", "min_list",  
        ["Ocena_Warunek_12",  
         "Ocena_Warunek_22",  
         "Ocena_Warunek_32"],1)
```

2) Zaokrąglamy wartość wniosku **Wniosek\_2** do 2 pozycji po przecinku:

```
model(1020,"Wniosek_2","Wartość finalna Wniosek_2",  
      "Wartość Wniosek_2","zaokrąglenie_do_N","2", 1)
```

# Wartość wniosku jako minimum wartości warunków (6)

5) Dodatkowo tworzymy model:

```
model(1030,"Wniosek_2",  
      "Wartość finalna Wniosek_2 > 0",  
      "Wartość finalna Wniosek_2", ">","0",1)
```

dzięki któremu przy wnioskowaniu wstecz weryfikującym hipotezę "**Wniosek\_2**" zostanie również wyznaczona "**Wartość finalna Wniosek\_2**"

# Analiza SWOT (1)

Jedna z podstawowych i popularnych heurystycznych technik analitycznych, umożliwiających porządkowanie informacji o szeroko rozumianym stanie obiektu i jego otoczeniu w celu określenia równie szeroko rozumianej strategii zarządzania tym obiektem.



# Analiza SWOT (2)

- Obiektem tym może być np. firma lub projekt.
- Możemy *SWOT* również zastosować do siebie (*SWOT YOURSELF*).
- Przyjmujemy w dalszym ciągu, że obiektem tym jest firma.

# Analiza SWOT (3)

Istota *SWOT* : przyporządkowaniu informacji o stanie firmy i otoczenia firmy czterem grupom czynników strategicznych:

- **S (*Strengths*)** – mocne strony firmy (MSF): wszystko to co stanowi atut, przewagę, zaletę analizowanej firmy.
- **W (*Weaknesses*)** – słabe strony firmy (SSF): wszystko to co stanowi słabość, barierę, wadę analizowanej firmy.
- **O (*Opportunities*)** – możliwości w otoczeniu firmy (MOF): wszystko to co w otoczeniu firmy stwarza dla niej możliwość korzystnej zmiany.
- **T (*Threats*)** – zagrożenia w otoczeniu firmy (ZOF): wszystko to co w otoczeniu firmy stwarza dla niej niebezpieczeństwo niekorzystnej zmiany.

# Analiza SWOT (4)

	<b>MSF</b> Mocne strony firmy	<b>SSF</b> Słabe strony firmy
<b>MOF</b> Możliwości w otoczeniu firmy	<b>MSF i MOF</b>  Strategia agresywna	<b>SSF i MOF</b>  Strategia konkurencyjna
<b>ZOF</b> Zagrożenia w otoczeniu firmy	<b>MSF i ZOF</b>  Strategia konserwatywna	<b>SSF i ZOF</b>  Strategia defensywna



# Przykład analizy SWOT (1)

Firma elektroniczna produkująca systemy alarmowe

# Przykład analizy SWOT (2)

Grupy czynników strategicznych	Stan firmy/otoczenia
MSF	<p>innowacyjny produkt:</p> <ul style="list-style-type: none"> <li>- modułowy system alarmowy dla budynków pod specjalnym nadzorem</li> <li>- przechowywanie nagrań na bieżąco w chmurze</li> <li>- zasilanie z ogniw paliwowych</li> <li>- skrajnie zminiaturyzowane wielofunkcyjne czujniki</li> </ul> <p>wysoka rentowność:</p> <ul style="list-style-type: none"> <li>- wysoka rentowność sprzedaży</li> <li>- wysoka rentowność kapitału</li> </ul> <p>duży kapitał:</p> <ul style="list-style-type: none"> <li>- wysoka wartość środków trwałych</li> <li>- posiadanie akcji i obligacji</li> <li>- duża ilość środków pieniężnych na rachunkach bankowych</li> </ul> <p>dobrze wykwalifikowana załoga:</p> <ul style="list-style-type: none"> <li>- duża wiedza teoretyczna oraz praktyczna pracowników</li> <li>- duże doświadczenie pracowników</li> </ul> <p>dobra obsługa klienta:</p> <ul style="list-style-type: none"> <li>- dobre szkolenia</li> <li>- szybki reset systemu po jego zadziałaniu</li> <li>- dogodne warunki gwarancji</li> <li>- infolinia z szyfrem</li> </ul>
SSF	brak własnych budynków, konflikty personalne w zarządzie
MOF	korzystne umowy kooperacyjne, brak produktów konkurencyjnych, dostęp do kredytów
ZOF	zagrożona dostawa komponentów z powodu trzęsienia ziemi w Tajlandii, podkupywanie pracowników przez inne firmy

# Przykład analizy SWOT (3)

Baza wiedzy SwotOceny.BED:

RESwotOceny.BED

MOSwotOceny.BED

Brak bazy ograniczeń wynika stąd, że analiza SWOT nie robi użytku z negacji warunków dopytywalnych (np. 'złe szkolenia', 'brak dostępu do kredytów') i negacji warunków niedopytywalnych (np. 'nieinnowacyjny produkt', 'słabo wykwalifikowana załoga'), gdyż negacje te nie są prawdą.

Przy analizie SWOT interesują nas tylko warunki, które są prawdą dla każdej z czterech grup czynników strategicznych



## Przykład analizy SWOT (4)

Celem bazy wiedzy 'SwotOceny.BED' jest wyznaczenie

- dla rozpatrywanej firmy elektronicznej – strategii o najwyższej ocenie.

# Przykład analizy SWOT (5)

Dla oceny wartości wniosków MSF, SSF, MOF, ZOF zastosowano *ocenianie absolutne*.

Obecnie bowiem mamy do czynienia z przypadkami reguł o bardzo różnych liczbach warunków i o różnych atrybutach dla których uśrednianie może fałszować relacje pomiędzy wartościami wniosków:

więcej warunków w niektórych regułach to więcej czynników przemawiających za wnioskiem, co przy analizie SWOT ma znaczenie