



Systemy regułowe

wiedza heurystyczna deklaratywnie



Plan wykładu

- Wnioski z badań nad systemami automatycznego dowodzenia
- Reguły jako forma reprezentacji wiedzy
- Sterowanie w systemach regułowych
- Mocne i słabe strony systemów eksperckich
- Obszary zastosowań systemów eksperckich

Wnioski dotyczące systemów automatycznego dowodzenia

- Systemy automatycznego wnioskowania abstrahują od *semantyki* symboli i formuł
- Logika klasyczna nie umożliwia reprezentacji informacji *niepewnej*
- Zastosowanie metod automatycznego wnioskowania jest problematyczne kiedy wiedza jest *niepełna*
- Podstawowy problem to zgromadzenie i zorganizowanie *wiedzy zdroworozsądkowej* niezbędnej do analizowania praktycznych zadań



Wnioski dotyczące systemów automatycznego dowodzenia

- Jak uporać się z wiedzą niepełną i niepewną?
 - Opracować inne formy reprezentacji wiedzy
- Jak sprostać ogromnej ilości wiedzy zdroworozsądkowej niezbędnej w praktycznych problemach wnioskowania?
 - Ograniczyć funkcjonowanie systemu do mocno specjalistycznej dziedziny wiedzy

Inna reprezentacja

- Badania w dziedzinie psychologii behawiorystycznej nad ludzką pamięcią doprowadziły do obserwacji, że u człowieka utrwalone w procesie zdobywania doświadczenia zasady najczęściej przyjmują postać:

IF A THEN B



Jeśli nie startuje
to uszkodzony
RAM

Regułowa forma reprezentacji wiedzy

- Reguły mogą opisywać różnego rodzaju zależności:
 - **Proceduralne**: **IF** <sytuacja> **THEN** <akcja>
np. jeżeli pada deszcz to otwórz parasol
 - **Przyczynowo-skutkowe**: **IF** <przyczyna> **THEN** <skutek>
np. jeżeli X jest przeziębiony to X ma gorączkę
 - **Pojęciowe**: **IF** <warunek> **THEN** <wniosek>
np. jeżeli X jest zatrudniony to X jest ubezpieczony
 - **Kompozycji**: **IF** <podtyp> **THEN** <typ>
np. jeżeli X jest częścią silnika to X jest częścią auta
 - **Dziedziczenia**: **IF** <podtyp> **THEN** <typ>
np. jeżeli X jest psem to X jest ssakiem

Regułowa forma reprezentacji

- Reguły produkcji

- Proceduralne

IF <sytuacja> **THEN** <akcja>

np. **jeżeli** pada deszcz **to** zabierz parasol

- Deklaratywne (pojęciowe)

IF <przesłanka/warunek> **THEN** <konkluzja/decyzja>

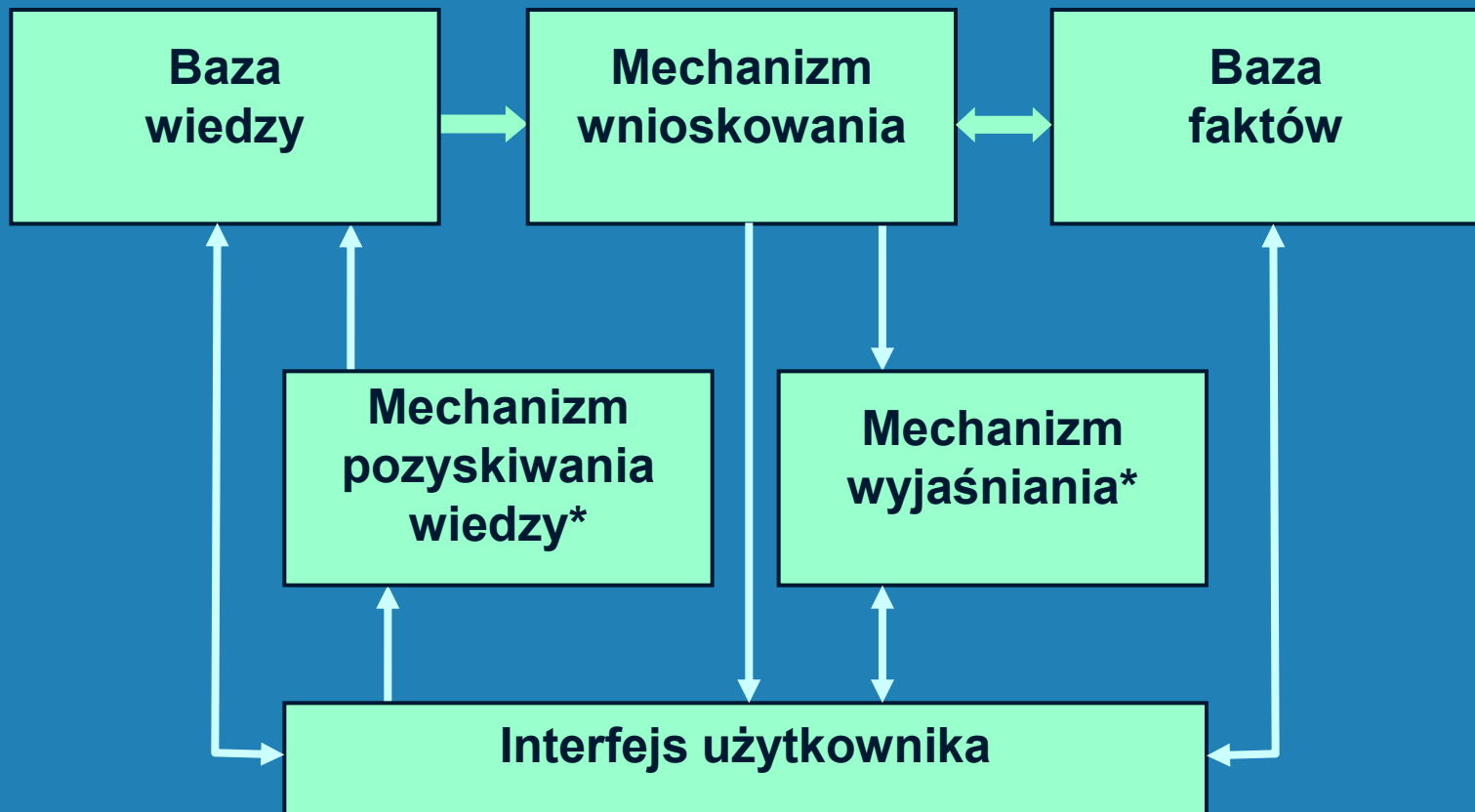
np. **jeżeli** X jest zatrudniony **to** X jest ubezpieczony

- Metareguły

IF <stan systemu> **THEN** <ustaw parametr>

np. **jeżeli** optymalne rozwiązane **to** strategia wszerz

Architektura systemu eksperckiego



** Nie zawsze występują*

Postać regułowej bazy wiedzy

- Ogólna postać :

IF $A_1 \odot \dots \odot A_n$ **THEN** $B_1 \wedge \dots \wedge B_m$

gdzie: \odot to \wedge i/lub \vee

- Przykład najprostszy (najpopularniejszy)

Reguły Horna (koniunkcja zdań atomowych oraz jedna konkluzja):

IF $A_1 \wedge \dots \wedge A_n$ **THEN** B

Zalety reguł hornowskich

- Kilka konkluzji ten sam warunek

~~IF A1 \wedge A2 \wedge A3 THEN X1 \wedge X2 .~~

IF A1 \wedge A2 \wedge A3 THEN X1

IF A1 \wedge A2 \wedge A3 THEN X2

- Alternatywne warunki ta sama konkluzja

~~IF (A1 \wedge A2 \wedge A3) \vee (B1 \wedge B2 \wedge B3) THEN W~~

IF A1 \wedge A2 \wedge A3 THEN W

IF B1 \wedge B2 \wedge B3 THEN W

- Prostsza - w rezultacie efektywniejsza - konstrukcja mechanizmu wnioskowania oraz większa czytelność (przejrzystość) bazy reguł

„Zazębianie” reguł

- Warunki odpytywalne (*ang. askeable conditions*)
 - Warunki nie będące wnioskami innych reguł – ich wartość musi być określona przez odpowiedź użytkownika na pytanie zadane podczas wnioskowania systemu
- Warunki nieodpytywalne (*ang. non-askable conditions*)
 - Warunki będące wnioskami innych reguł – ich wartość wynika pośrednio z odpowiednich reguł i wartości odpowiednich warunków odpytywalnych

„Zazębiania” można uniknąć przez „spłaszczanie” bazy wiedzy:

~~IF A \wedge B \wedge V THEN W~~

~~IF C \wedge D \wedge E THEN V~~

IF A \wedge B \wedge C \wedge D \wedge E THEN W

„Zazębianie” reguł

- Dlaczego?
 - Istnienie „zazębień” w bazie zwiększa dokładność (gradację) procesu wnioskowania – mamy jawną informację o wnioskach pośrednich, nie będących warunkami odpytywalnymi, które mogą być interesujące dla użytkownika
 - Bliższe rzeczywistości odzwierciedlenie wiedzy dziedzinowej – „spłaszczanie” reguł uniemożliwia zachowanie hierarchicznej struktury wiedzy
 - Przejrzysta i czytelna baza wiedzy – reguły mają mniej warunków

Negacje w konkluzji reguły

W regułowej bazie wiedzy nie występują równocześnie reguły z wnioskami w postaci prostej i zanegowanej

IF Urlop \wedge Pieniądze **THEN** Wycieczka

IF \neg Zdrów **THEN** \neg Wycieczka

IF \neg Zdrów(żona) **THEN** \neg Wycieczka

czasami można bazę przeformułować:

IF Urlop \wedge Pieniądze \wedge Zdrów \wedge Zdrów(żona)
THEN Wycieczka

dzięki czemu pozbędziemy się konkluzji zanegowanych.

Negacja w regułowej bazie wiedzy

Różnice między logiką a systemem regułowym:

Reguła

IF P THEN Q	P	Q
True	True	True
True	False	False

Implikacja w logice

$P \Rightarrow Q$	P	Q
True	True	True
True	False	True
True	False	False

Gdy warunki prawdziwej implikacji w logice są fałszywe, wniosek może być prawdziwy lub fałszywy. Natomiast gdy warunki prawdziwej reguły nie są prawdą, to również wniosek nie może być prawdą.

Negacja w regułowej bazie wiedzy

W systemie regułowym prawdą jest tylko to, co wynika ze zdefiniowanych reguł i faktów oraz danych wprowadzonych przez użytkownika.

Jest to przejaw *założenia o zamkniętości świata*.

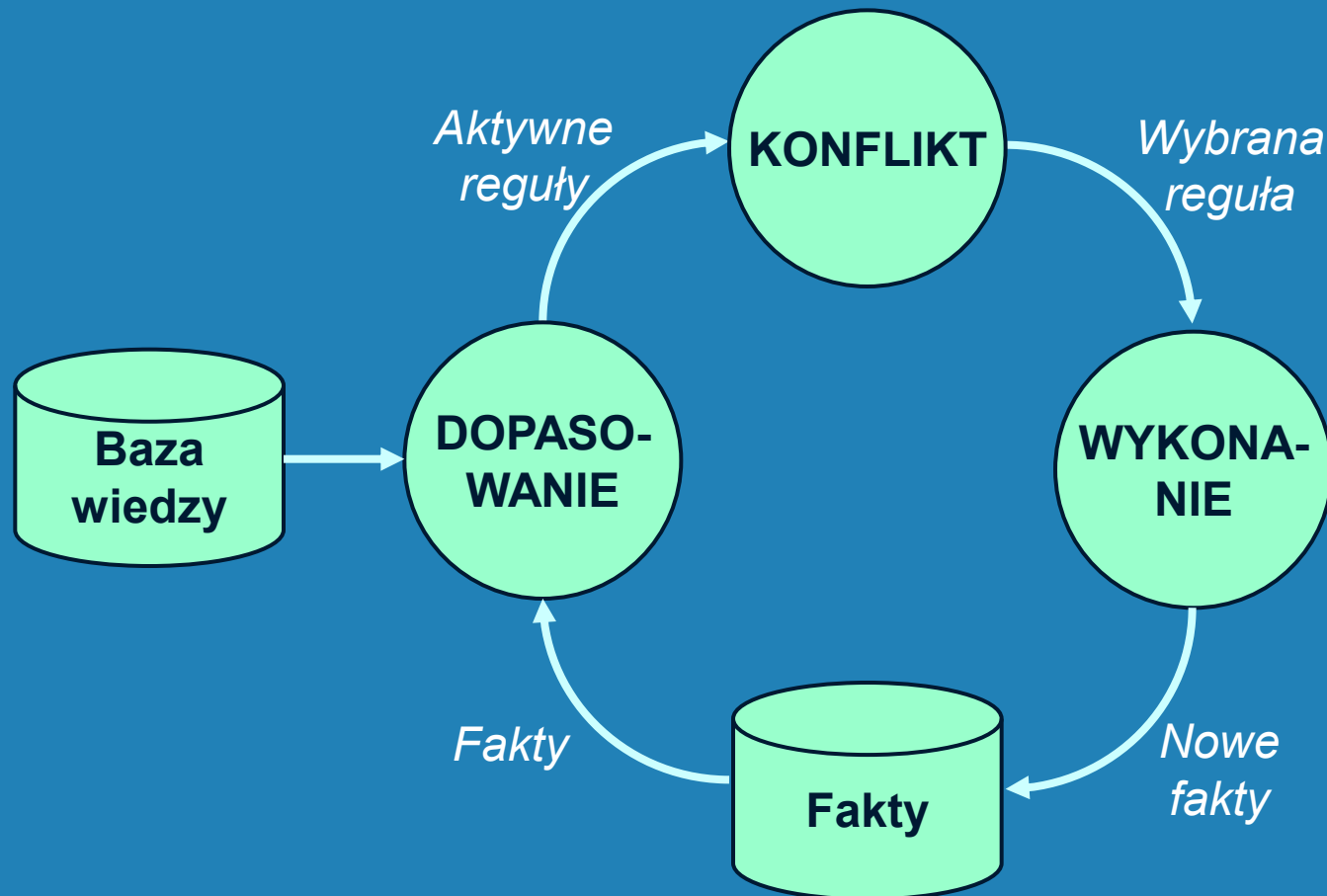
Inaczej: jeśli czegoś nie można wywnioskować na podstawie reguł i faktów, to uznaje się to za nieprawdziwe.

Rodzaje mechanizmów wnioskowania na regułach

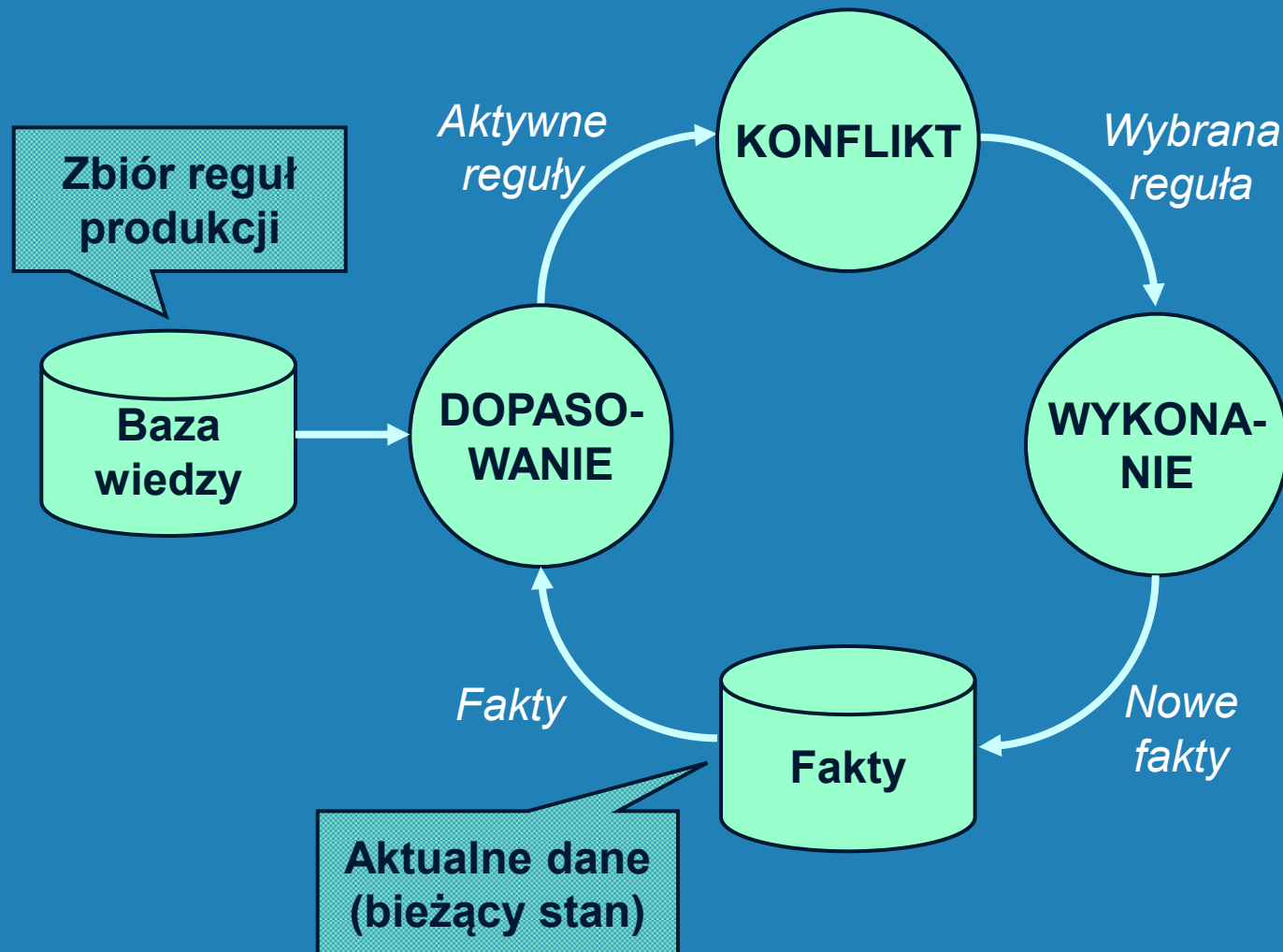
Mamy dwa podstawowe typy mechanizmów wnioskujących dla reprezentacji regułowej:

- Wnioskowanie w przód (ang. forward chaining)
 - Weryfikujemy warunki wszystkich reguł na wszystkich dostępnych faktach w celu wyprowadzenia nowych faktów za pomocą reguł
- Wnioskowanie wstecz (ang. backward chaining)
 - Szukamy reguły, która w konkluzji zawiera naszą hipotezę, sprawdzamy czy warunki użycia tej reguły są spełnione (badamy możliwość ich spełnienia przez fakty lub inne reguły)

Wnioskowanie w przód



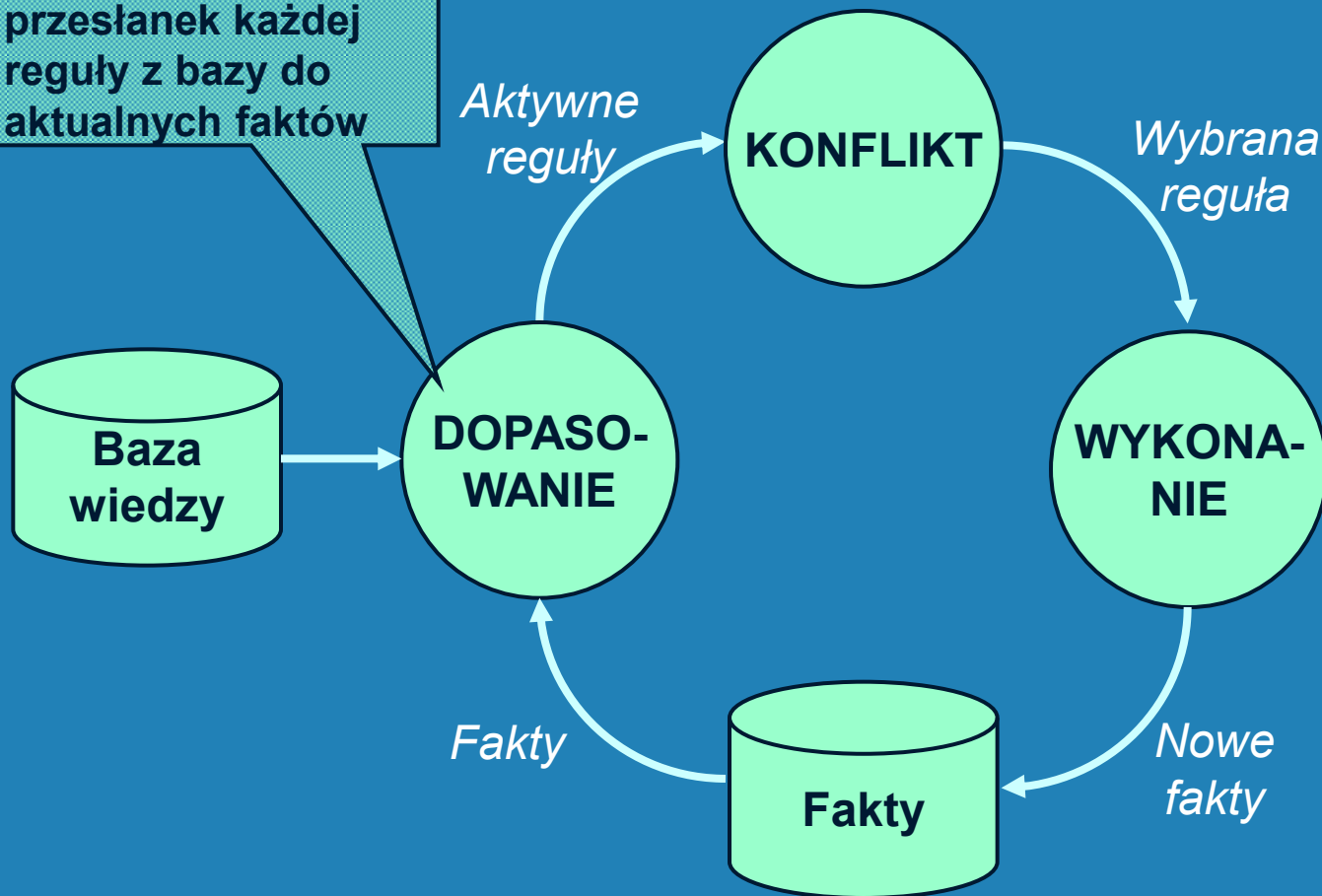
Wnioskowanie w przód



Wnioskowanie w przód

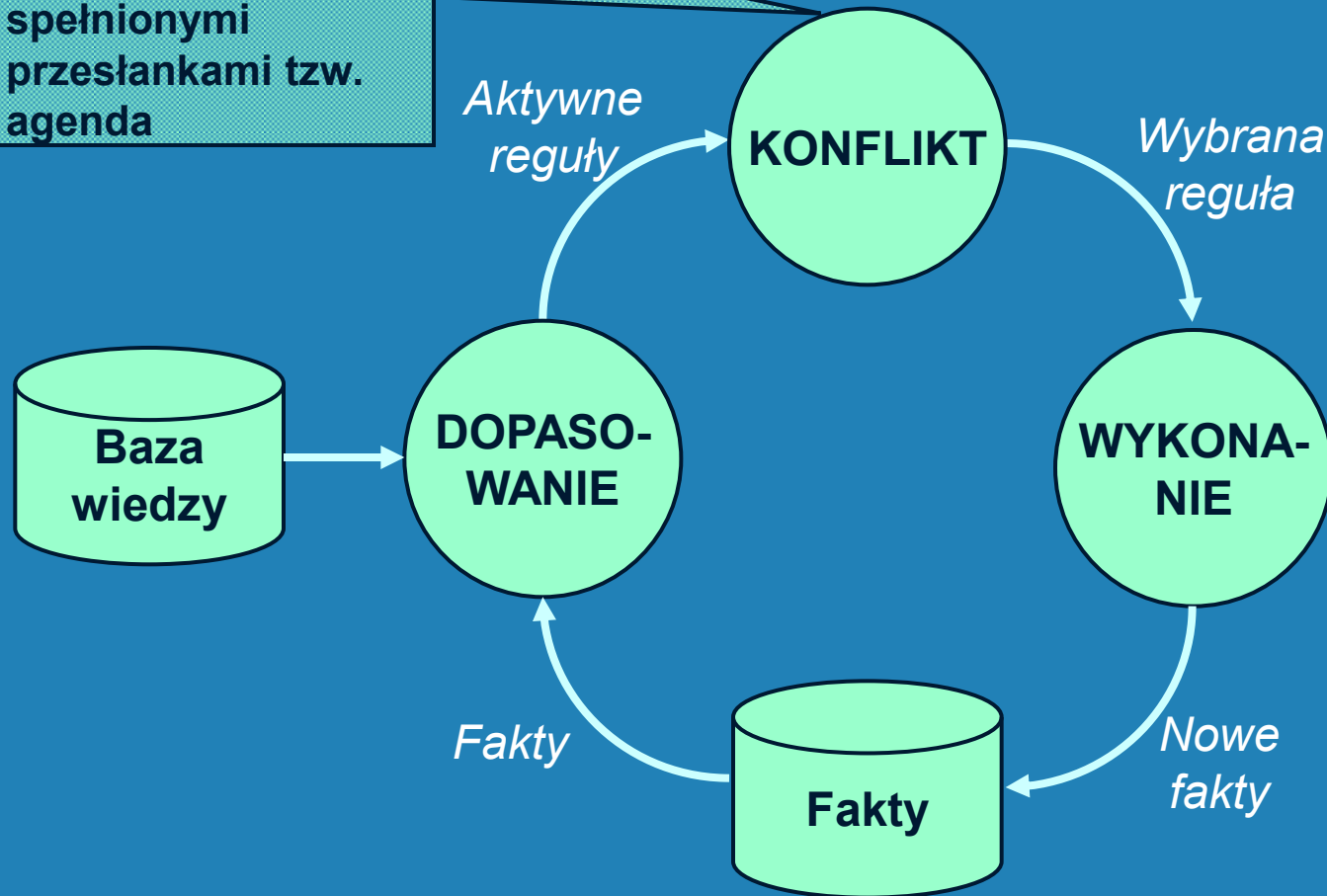
Krok 1:

dopasowanie przesłanek każdej reguły z bazy do aktualnych faktów

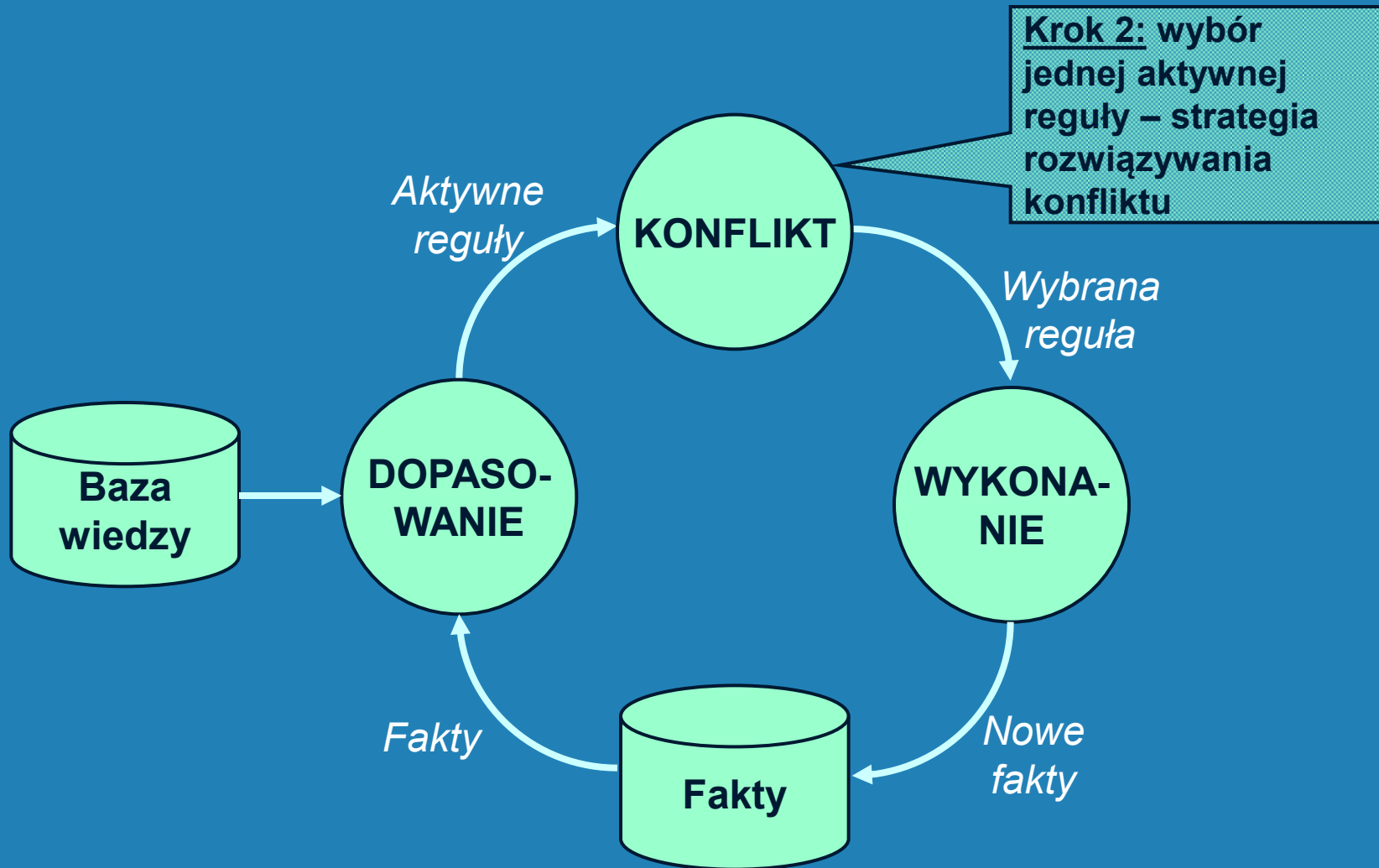


Wnioskowanie w przód

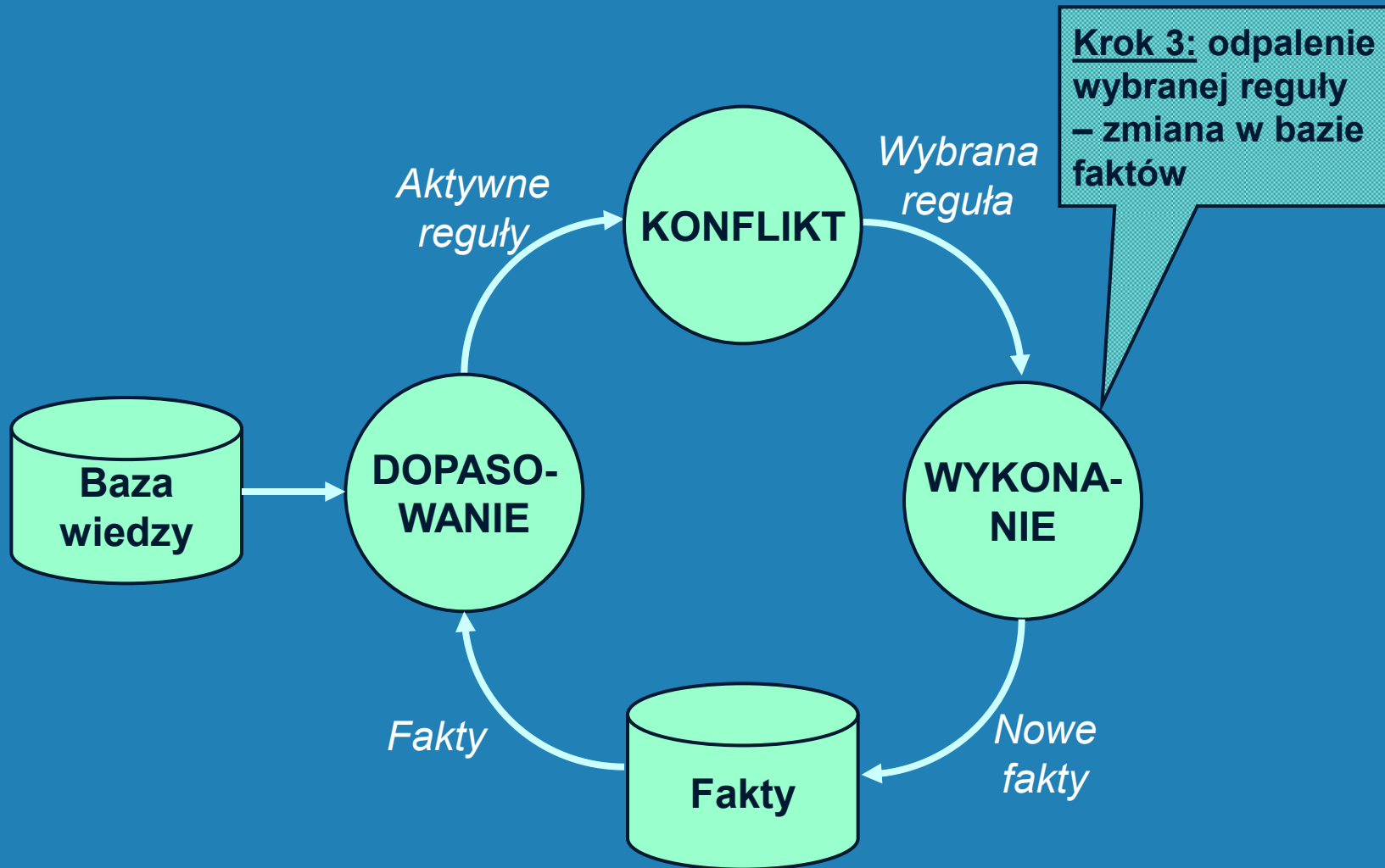
Krok 1: zbiór
aktywnych reguł ze
spełnionymi
przesłankami tzw.
agenda



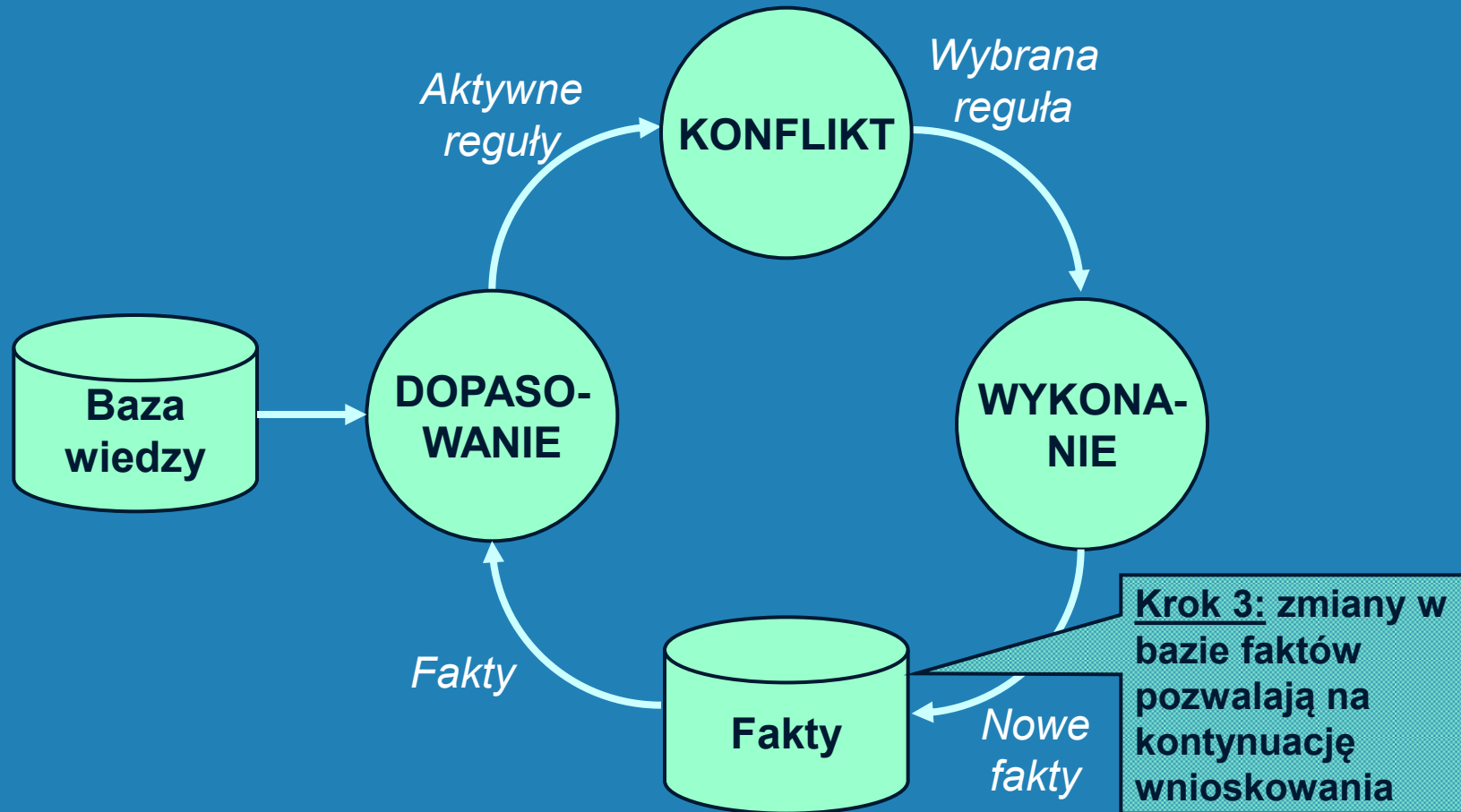
Wnioskowanie w przód



Wnioskowanie w przód



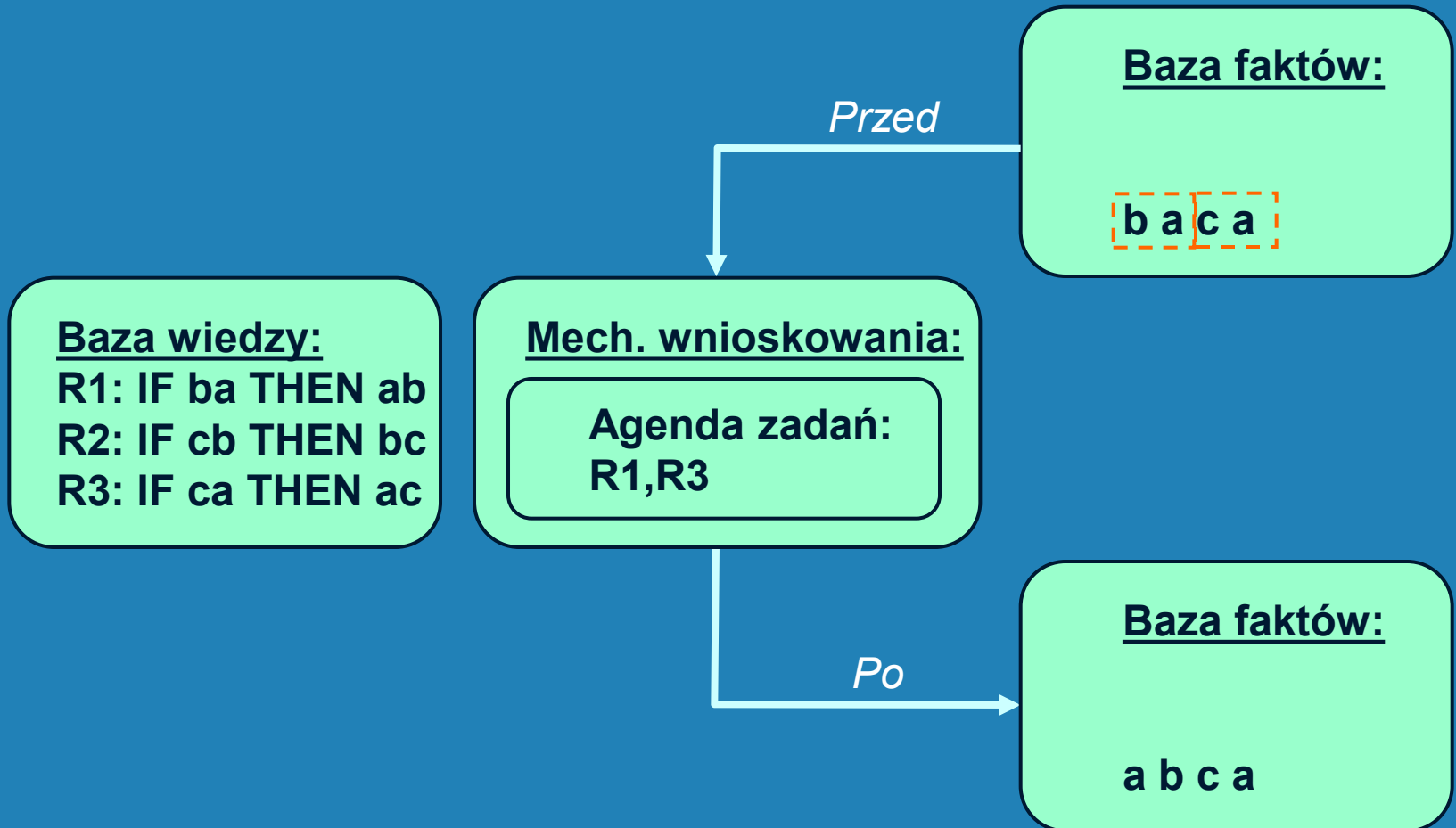
Wnioskowanie w przód



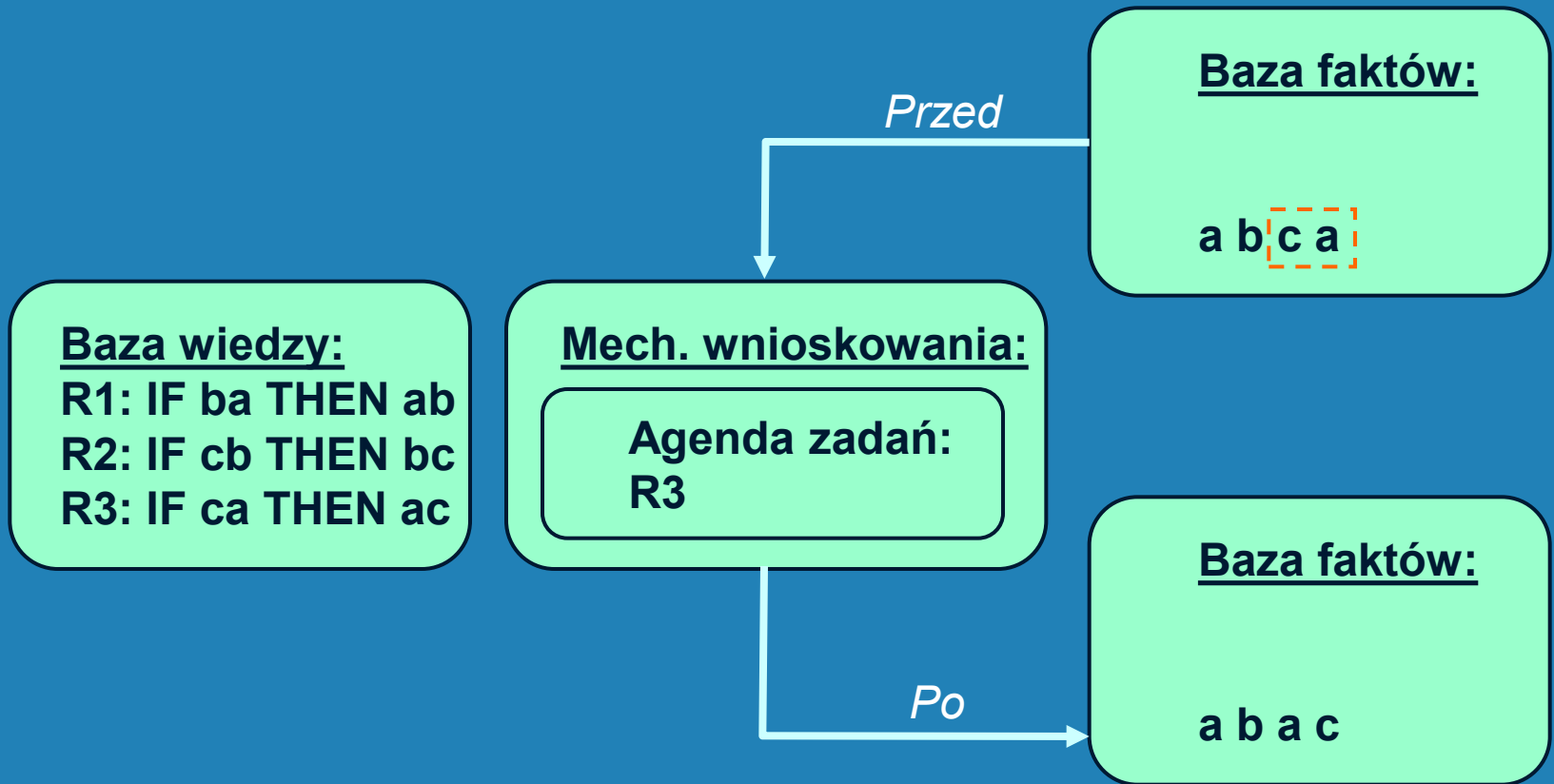
Cykl wnioskowania w przód

- Cykl rozpoznaj-wykonaj (*ang. recognize-act cycle*)
while DOPASOWANIE $\neq \emptyset$ do
 - ROZSTRZYGANIE
KONFLIKTU: wybór jednej reguły według
przyjętych kryteriów
 - WYKONANIE AKCJI: odpalenie reguły – realizacja
wszystkich konkluzji po prawej
stronie wybranej reguły
 - DOPASOWANIE: poszukiwanie w bazie wiedzy
wszystkich reguł, które mają
spełnione warunki w oparciu o
fakty
end

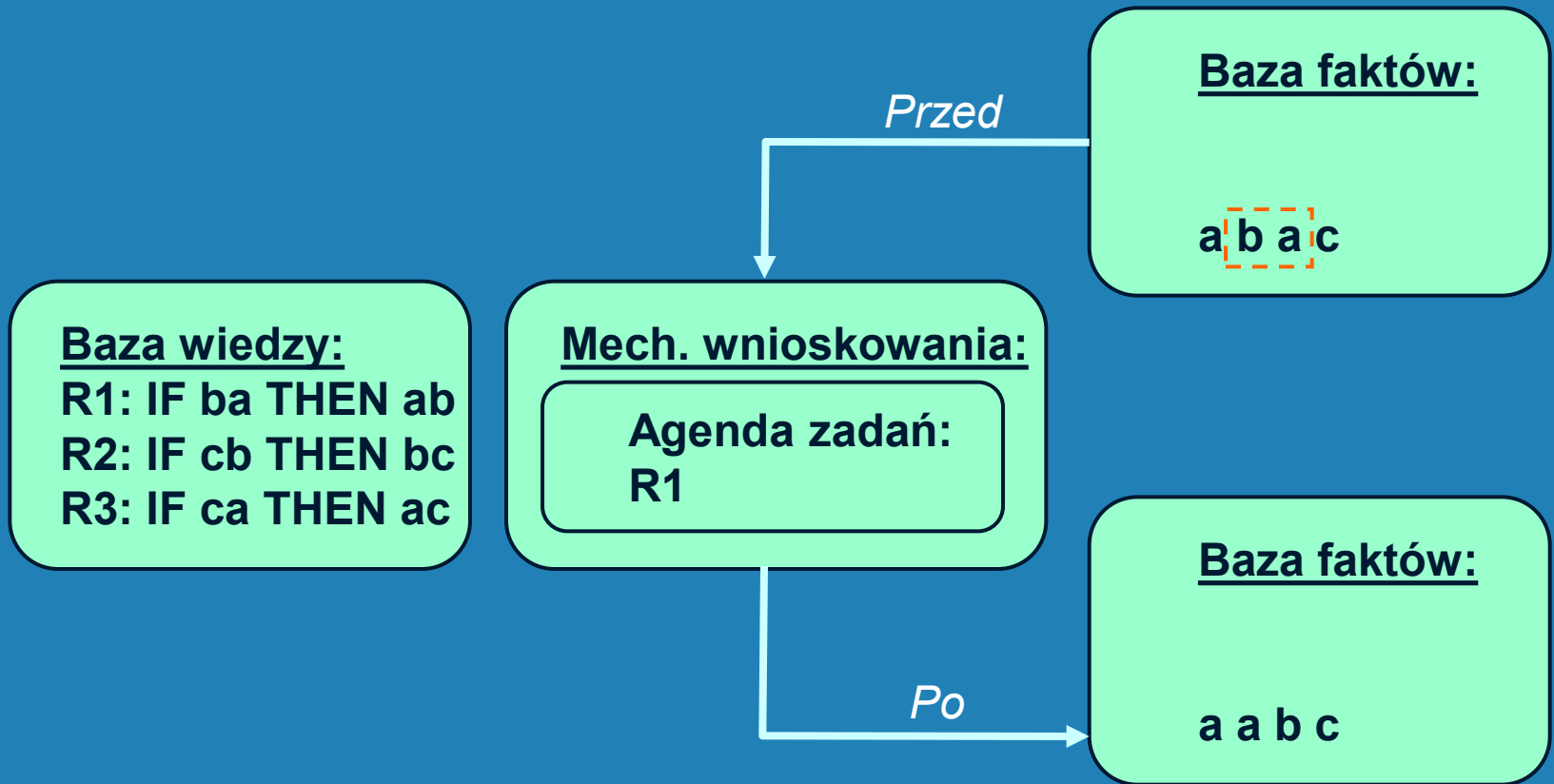
Przykład wnioskowania w przód



Przykład wnioskowania w przód



Przykład wnioskowania w przód



Kryteria rozstrzygania konfliktów

- Strategia sterowania systemu regułowego zależy od kryteriów wyboru reguł ze zbioru konfliktowego
 - **Kryterium porządkowe** – kolejność definiowania reguł w bazie wiedzy
 - **Kryterium czasowe** – moment aktywacji reguły (pośrednio poprzez czas pojawienia się danych uaktywniających)
 - **Kryterium złożoności** – stopień komplikacji warunków reguły
 - **Kryterium istotności** – regułom przypisywane są (statyczne bądź dynamiczne) priorytety numeryczne odzwierciedlające preferencje eksperta
 - **Brak jakiegokolwiek kryterium** – wybór aktywnej reguły odbywa się w sposób losowy
- Łączenie kryteriów ze sobą wymaga określenia ich względnej ważności – kryteria złożone

Kierunek wnioskowania: *w przód*

kryterium porządkowe

R1: IF start THEN $p \wedge q$

R2: IF p THEN $r \wedge s$

R3: IF q THEN $w \wedge r$

R4: IF q THEN $t \wedge u$

R5: IF s THEN v

R6: IF $v \wedge r \wedge q$ THEN goal

Pamięć robocza

start

start, p , q

start, p , q , r , s

start, p , q , r , s , w

start, p , q , r , s , w , t , u

start, p , q , r , s , w , t , u , v

start, p , q , r , s , w , t , u , goal

Zbiór konfliktowy

R1

R2, R3, R4

R3, R4, R5

R4, R5

R5

R6

\emptyset

Odpalona reguła

R1

R2

R3

↪ *nawrót*

R4

↪ *nawrót*

R5

R6

Wnioskowanie w tył

procedure backward(*Cel*)

1. if *Cel* matches *Baza Faktów* then return(true)
2. for each *Reguła*: conclusions(*Reguła*) match *Cel* do
 for each *P* in premises(*Reguła*) do:
 if backward(*P*)= false then // rekurencja
 continue // następna reguła
 if $\forall P$ backward(*P*)=true then // warunki spełnione
 add(*Cel*, *Baza Faktów*) and return(true) //cel spełniony
3. if \forall *Reguła*: premises(*Reguła*) = false then return(false)
4. if $\neg \exists$ *Reguła*: conclusions(*Reguła*) match *Cel* then
 if ask_user(*Cel*)=true then
 add(*Cel*, *Baza Faktów*) and return(true)
 if ask_user(*Cel*)=false then return(false)

Kierunek wnioskowania: w tył

strategia: w głąb kryterium: porządkowe

R1: IF start THEN $p \wedge q$
 R3: IF q THEN $w \wedge r$
 R5: IF s THEN v

R2: IF p THEN $r \wedge s$
 R4: IF q THEN $t \wedge u$
 R6: IF $v \wedge r \wedge q$ THEN goal

<u>Pamięć celów</u>	<u>Zbiór konfliktowy</u>	<u>Wybrana reguła</u>
goal	R6	R6
v, r, q, goal	R5, R6	R5
s, v, r, q, goal	R2, R5, R6	R2
p, s, v, r, q, goal	R1, R2, R5, R6	R1
start, p, s, v, r, q, goal	R1, R2, R5, R6	←dodajemy fakt start
start ←	R1, R2, R5, R6	⇐odpalamy R1 (mamy p i q)
start, p, s, v, r, q, goal	R2, R5, R6	⇐odpalamy R2 (mamy r i s)
start, p, s, v, r, q, goal	R5, R6	⇐odpalamy R5 (mamy v)
start, p, s, v, r, q, goal	R6	⇐odpalamy R6 (mamy goal)
start, p, s, v, r, q, goal	∅	⇐ brak niespełnionych
		⇐koniec!

*Pokreślone cele ∈ Baza-Faktów

Kierunek wnioskowania: w tył

strategia: w głąb kryterium: porządkowe

R1: IF start THEN $p \wedge q$
 R3: IF q THEN $w \wedge r$
 R5: IF s THEN v

R2: IF p THEN $r \wedge s$
 R4: IF q THEN $t \wedge u$
 R6: IF $r \wedge v \wedge q$ THEN goal

Pamięć celów

goal

r, v, q, goal

p, r, v, q, goal

start, p, r, v, q, goal

start ←

start, p, r, v, q, goal

start, p, v, r, s, q, goal

start, p, v, r, s, q, goal

start, p, v, r, s, q, goal

start, p, v, r, s, q, goal

start, p, v, r, s, q, goal

Zbiór konfliktowy

R6

(R2; R3), R6

R1, (R2; R3), R6

R1, (R2; R3), R6

R1, (R2; R3), R6

(R2; R3), R6

R6

R5, R6

R5, R6

R6

∅

Wybrana reguła

R6

R2 (ewent. nawrót na R3)

R1

← dodajemy fakt start

← odpalamy R1 (mamy p i q)

← odpalamy R2 (mamy r i s)

← brakuje v

R5

← odpalamy R5 (mamy v)

← odpalamy R6 (mamy goal)

← brak niespełnionych

← koniec!

*Pokreślone cele ∈ Baza-Faktów

Kierunek wnioskowania: w tył

strategia: w głąb kryterium: porządkowe

R1: IF start THEN $p \wedge q$
R3: IF q THEN $w \wedge r$
R5: IF s THEN v

R2: IF p THEN $r \wedge s$
R4: IF q THEN $t \wedge u$
R6: IF $q \wedge r \wedge v$ THEN goal

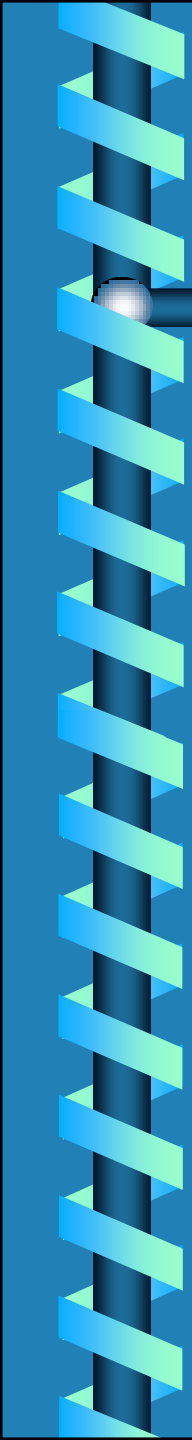
Jako ćwiczenie
zrobić sobie dla
kolejności: q, r, v



Mechanizmy wyjaśniania w systemach eksperckich

Rodzaje wyjaśnień wykorzystywanych w systemach eksperckich:

- WHAT – podaje wszystkie fakty, które zostały wywiedzione do momentu wydania tego polecenia
- HOW – informuje jak została wywiedziona konkluzja wskazana przez użytkownika
- WHY – wskazuje regułę, której aktywacja spowodowała zadanie określonego pytania w trakcie wnioskowania
- WHATIF – pozwala zbadać konsekwencji różnych odpowiedzi na zadane pytanie



Problemy projektowania regułowej bazy wiedzy



Problemy przy projektowaniu regułowej bazy wiedzy

Negatywne zjawiska występujące w trakcie prac nad bazą reguł:

- Niedeterminizm
- Niezgodność
- Sprzeczność
- Nadmiarowość

Niedeterminizm

Dla dwóch reguł o treści:

IF C1 THEN W1

IF C2 THEN W2

zachodzi niedeterminizm (*ang. ambiguity*) bazy wiedzy, jeżeli istnieje stan opisany przez formułę F taki, że równocześnie:

$F \models C1$ i $F \models C2$ oraz $W1 \neq W2$

\models - log. konsekw.

Są to zatem reguły, których akcje można wykonać równocześnie ale ich konkluzje (skutki) są różne.

Niezgodność

Dla dwóch reguł o treści:

IF C1 THEN W1

IF C2 THEN W2

zachodzi niezgodność (*ang. ambivalence*) bazy wiedzy, jeżeli istnieje stan opisany przez formułę F taki, że równocześnie:

$F \models C1$ i $F \models C2$ oraz $\not\models_I W1 \wedge W2$ w przyjętej interpretacji I

Są to zatem reguły, które można odpalić równocześnie ale ich konkluzje są konfliktowe (nie mogą być równocześnie prawdziwe).

Sprzeczność

Dla dwóch reguł o treści:

IF C1 THEN W1

IF C2 THEN W2

zachodzi sprzeczność (*ang. logical inconsistency*) bazy wiedzy, jeżeli istnieje stan opisany przez formułę F , taki że równocześnie:

$F \models C1$ i $F \models C2$ oraz $\not\models W1 \wedge W2$ w każdej interpretacji

Są to zatem reguły, które można odpalić równocześnie ale ich konkluzje są logicznie sprzeczne (nie mogą być równocześnie prawdziwe w żadnej interpretacji).

Sprzeczność reguł powiązanych

Założmy, że mamy bazę reguł:

R1: **IF** $K \wedge B$ **THEN** H

R2: **IF** $C \wedge D \wedge X$ **THEN** K

R3: **IF** $\neg H \wedge A$ **THEN** X

spłaszczając R3 do R2 otrzymamy:

R2': **IF** $C \wedge D \wedge \neg H \wedge A$ **THEN** K

zaś spłaszczenie R2' do R1 daje sprzeczną regułę:

IF $C \wedge D \wedge \neg H \wedge A \wedge B$ **THEN** H

Jest to przykład sprzeczności zewnętrznej reguł.

Sprzeczność reguł powiązanych

Założmy, że mamy bazę reguł:

R1: **IF** $C \wedge B \wedge Y$ **THEN** X

R2: **IF** $A \wedge \neg B$ **THEN** Y

splaszczając R2 do R1 otrzymamy sprzeczną regułę:

IF $C \wedge B \wedge A \wedge \neg B$ **THEN** X

Jest to przykład sprzeczności wewnętrznej reguł.

Wykrywanie sprzeczności występujących na skutek zagnieżdżeń reguł może być kosztowne i w ogólnym przypadku wymagać pełnego przeglądu reguł.

Nadmiarowość

Baza wiedzy R reprezentowana przez zbiór reguł $R = R_1 \wedge R_2 \wedge \dots \wedge R_m$ jest nadmiarowa (redundantna), jeśli istnieje baza R' otrzymana z R poprzez usunięcie z niej składowej (reguły) R_i , taka że: $R \models R'$ i $R' \models R$.

Zatem, baza jest nadmiarowa jeżeli istnieje możliwość usunięcia z niej co najmniej jednej z reguł i otrzymana w ten sposób baza jest logicznie równoważna pierwotnej bazie wiedzy.

Pochłanianie reguł

Reguła postaci **IF** A **THEN** B pochłania regułę (jest ogólniejsza od reguły) postaci **IF** C **THEN** D wtedy i tylko wtedy, gdy $C \models A$ i $B \models D$.

Reguła pochłaniająca posiada zatem mniej wymagające warunki ale mocniejsze konkluzje.

Przykład

R1: **IF** $P \wedge Q$ **THEN** $H \wedge G$

R2: **IF** $P \wedge Q \wedge S$ **THEN** H

Reguła R1 pochłania regułę R2.

Nadmiarowość: warianty

IF $A \wedge B$ **THEN** W

IF $C \wedge D$ **THEN** A

IF $E \wedge F$ **THEN** B

~~**IF** $C \wedge D \wedge E \wedge F$ **THEN** W~~

Występowanie reguł wielokrotnych

~~**IF** $A \wedge B \wedge C$ **THEN** W~~ - - -

IF $A \wedge B$ **THEN** W

Występowanie reguł pochłoniętych

~~**IF** $A \wedge B \wedge C$ **THEN** W~~ - - -

~~**IF** $A \wedge B \wedge \neg C$ **THEN** W~~ - - -

IF $A \wedge B$ **THEN** W

Występowanie reguł o zbędnych warunkach



Mocne strony systemów eksperckich

- Mniejszy koszt ekspertyzy przy większej prędkości jej uzyskiwania
- Stała i większa dostępność ekspertyzy
- Dostępność wyjaśnień
- Ekspertyza pozbawiona wpływu czynników emocjonalnych
- Możliwość instruowania użytkowników – inteligentny nauczyciel początkujących ekspertów
- Możliwa modyfikacja bazy wiedzy bez naruszania integralności systemu



Słabe strony systemów eksperckich

- Duży rozmiar bazy wiedzy w poważnych zastosowaniach
- Brak mechanizmów autonomicznego pozyskiwania wiedzy przez system – nie generują nowej wiedzy
- Brak wiedzy przyczynowo-skutkowej – wiedza zawarta w regułach nie musi odzwierciedlać faktycznego procesu podejmowania decyzji

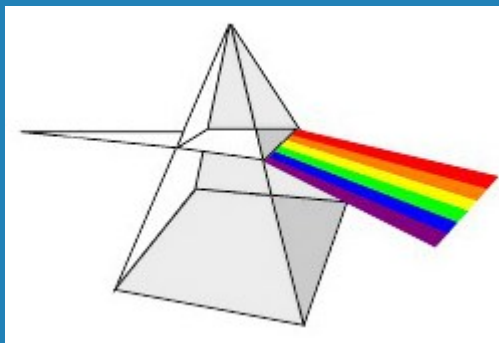
Słynne systemy eksperckie

- DENDRAL (1960) – Stanford – Rozpoznawanie struktur molekularnych związków organicznych na podstawie danych spektroskopowych
- MYCIN (1970) – Stanford – Diagnozowanie i terapia infekcji bakteryjnych krwi
- PROSPECTOR (1979) – SRI International – Lokalizacja i ocena złóż geologicznych
- XCON (1982) – VAX – Konfiguracja komputerów DEC Vax

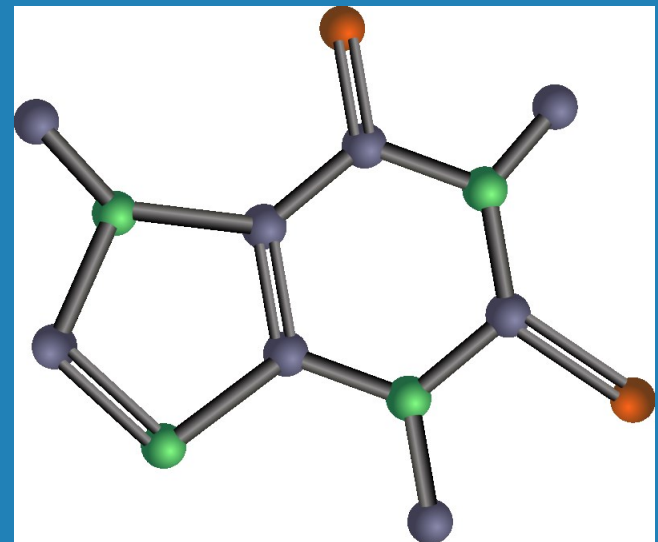
Zakres zastosowań systemów eksperckich

- Interpretacja danych

Formowanie wniosków na podstawie danych



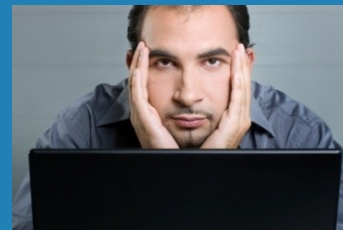
DENDRAL



Zakres zastosowań systemów eksperckich

- **Prognozowanie**

Przewidywanie potencjalnych skutków zaistniałych sytuacji



Zakres zastosowań systemów eksperckich

- Diagnostyka

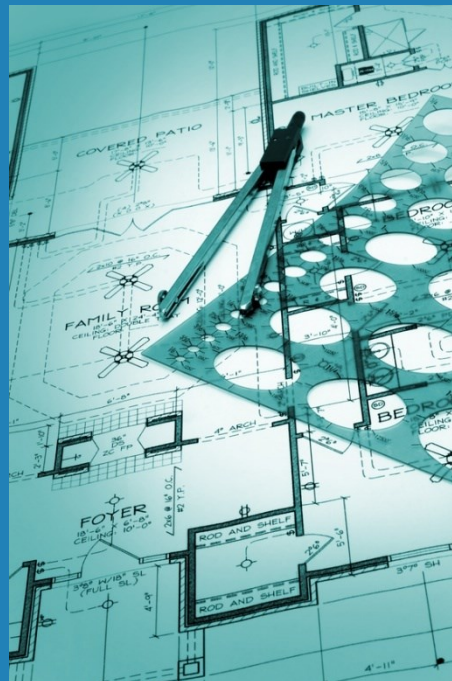
Badanie objawów i wnioskowanie na temat faktycznych przyczyn niesprawności



Zakres zastosowań systemów eksperckich

- **Projektowanie**

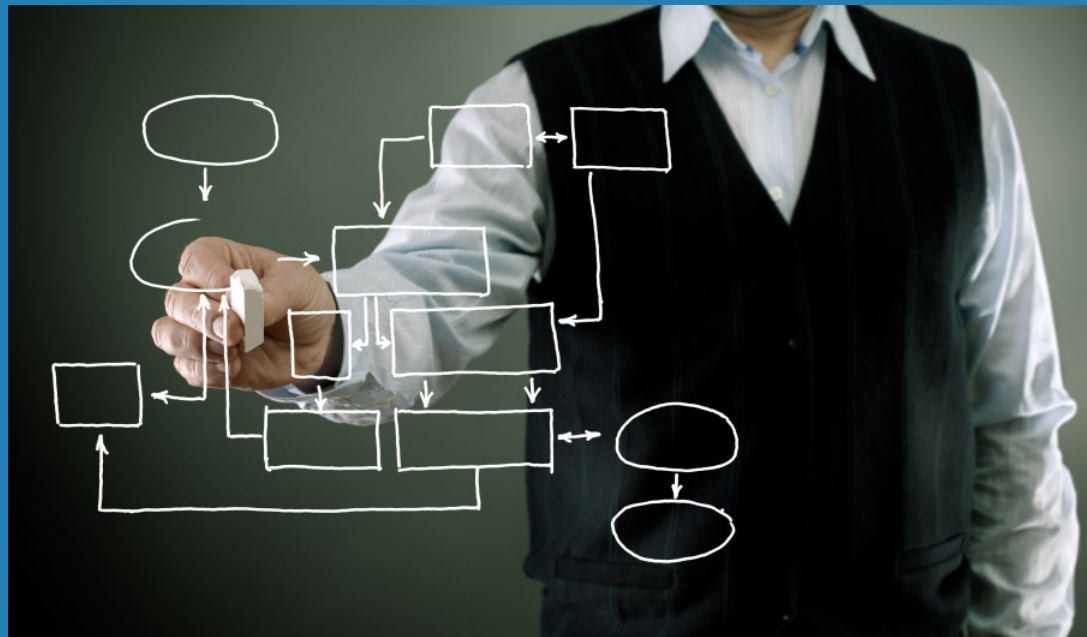
Dobór konfiguracji układu oparty na określonych kryteriach przy uwzględnieniu przyjętych ograniczeń



Zakres zastosowań systemów eksperckich

- **Planowanie**

Konstrukcja sekwencji działań prowadzących od przyjętego stanu początkowego do stanu docelowego



Zakres zastosowań systemów eksperckich

- **Szkolenie i instruktaż**

Wykrywanie i korygowania błędów w rozumieniu przedmiotu danej dziedziny



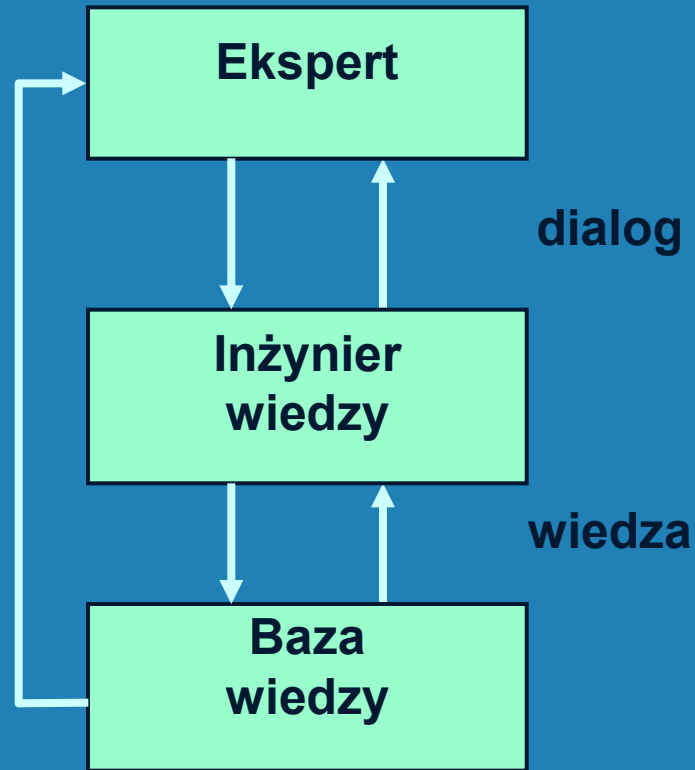
Zakres zastosowań systemów eksperckich

- **Monitorowanie i sterowanie**

Porównywanie obserwowanego działania z oczekiwanym funkcjonowaniem oraz nadzór nad aktywnością złożonych systemów



Inżynieria wiedzy



Narzędzia projektowania systemów eksperckich

- Szkieletowe systemy eksperckie

Specjalistyczne narzędzia informatyczne, w których użytkownik musi dostarczyć tylko bazę wiedzy, wyposażone w:

- Język opisu reguł
- Edytor reguł (bazy wiedzy)
- Interpreter reguł i mechanizm wnioskowania
- Debugger procesu wnioskowania
- Narzędzia weryfikacji poprawności bazy wiedzy
- Mechanizmy obsługi we-wy
- Interfejs użytkownika