

# Systemy wspomagania decyzji

## Opis projektu

Celem projektu jest zaprojektowanie i implementacja systemu, który na podstawie danych z czujników zamontowanych na kombinezonie strarzaka ocenia, czy strarzak kontynuować akcję, czy musi uciekać.

### 1. Opis danych wejściowych/wyjściowych:

Dane wejściowe:

- $c_1$  - Ciśnienie krwi (skurczowe)
- $c_2$  - Ciśnienie krwi (rozkurczowe)
- $p$  - Puls
- $t_1$  - Temperatura ciała
- $t_2$  - Temperatura na zewnątrz

Dane wyjściowe:

Decyzja, czy pozostać na posterunku czy uciekać.

### 2. Słowny opis modelu systemu

Starzak jest zagrożony, gdy:

- temperatura na zewnątrz jest za wysoka
- temperatura ciała jest za wysoka
- ciśnienie jest niskie i puls jest wysoki
- ciśnienie jest wysokie i puls jest wysoki
- ciśnienie jest niskie i puls jest niski

### 3. Opis modelu przy pomocy formuł zdaniowych

Na podstawie słownego opisu wyznaczamy zbiór formuł podstawowych:

$\alpha_1$  – temperatura na zewnątrz jest za wysoka

$\alpha_2$  – temperatura ciała jest za wysoka

$\alpha_3$  – ciśnienie jest niskie

$\alpha_4$  – ciśnienie jest wysokie

$\alpha_5$  – puls jest niski

$\alpha_6$  – puls jest wysoki

$\alpha_7$  – strarzak musi uciekać

$$\alpha_u(\alpha) = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$$

$$\alpha_y(\alpha) = (\alpha_7)$$

Wiążemy formuły podstawowe ze zmiennymi wejściowymi

temperatura na zewnątrz jest za wysoka, gdy  $t_2 > 70$

temperatura ciała jest za wysoka, gdy  $t_1 > 37$

ciśnienie jest za niskie, gdy  $c_1 < 100$  i  $c_2 < 65$

ciśnienie jest za wysokie, gdy  $c_1 > 150$  i  $c_2 > 100$

puls jest za niski, gdy  $p < 60$

puls jest za wysoki, gdy  $p > 120$

Definiujemy zbiór faktów

$$F_1 = \alpha_1 \rightarrow \alpha_7$$

$$F_2 = \alpha_2 \rightarrow \alpha_7$$

$$F_3 = (\alpha_3 \wedge \alpha_6) \rightarrow \alpha_7$$

$$F_4 = (\alpha_4 \wedge \alpha_6) \rightarrow \alpha_7$$

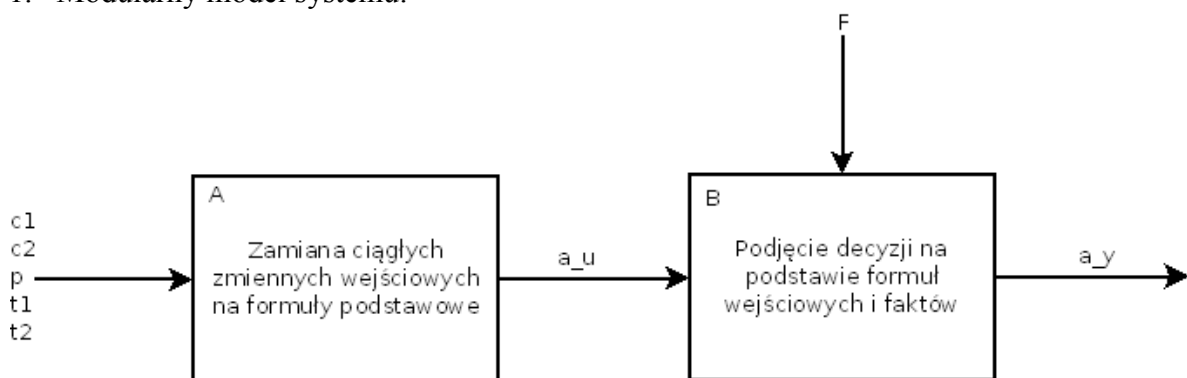
$$F_5 = (\alpha_3 \wedge \alpha_5) \rightarrow \alpha_7$$

Zapis można uprościć do następującej formuły:

$$F = \alpha_1 \vee \alpha_2 \vee (\alpha_3 \wedge \alpha_6) \vee (\alpha_4 \wedge \alpha_6) \vee (\alpha_3 \wedge \alpha_5) \rightarrow \alpha_7$$

## Implementacja projektu

### 1. Modułowy model systemu:



Moduł A – pobiera dane wejściowe i na podstawie ustalonych reguł generuje ciąg zer i jedynek, będących wartościami logicznymi kolejnych formuł podstawowych.

Moduł B – do formuł zdaniowych ze zbioru faktów (F) podstawiane są wartości logiczne formuł wejściowych, a następnie zwracane są wszystkie kombinacje formuł wyjściowych, które spełniają tak przygotowane formuły ze zbioru faktów.

Moduł A oraz zbiory  $\alpha_u$ ,  $\alpha_y$  i F zależą od wybranego systemu decyzyjnego, natomiast moduł B jest uniwersalny i może być połączony z dowolnie zaimplementowanym modulem A.

Program zostanie napisany w języku Java. Do stworzenia GUI programu zostanie użyta technologia SWING.

### 2. Implementacja modułu A

Moduł ten jest dość prosty, więc jego implementacja nie wymaga komentarza.

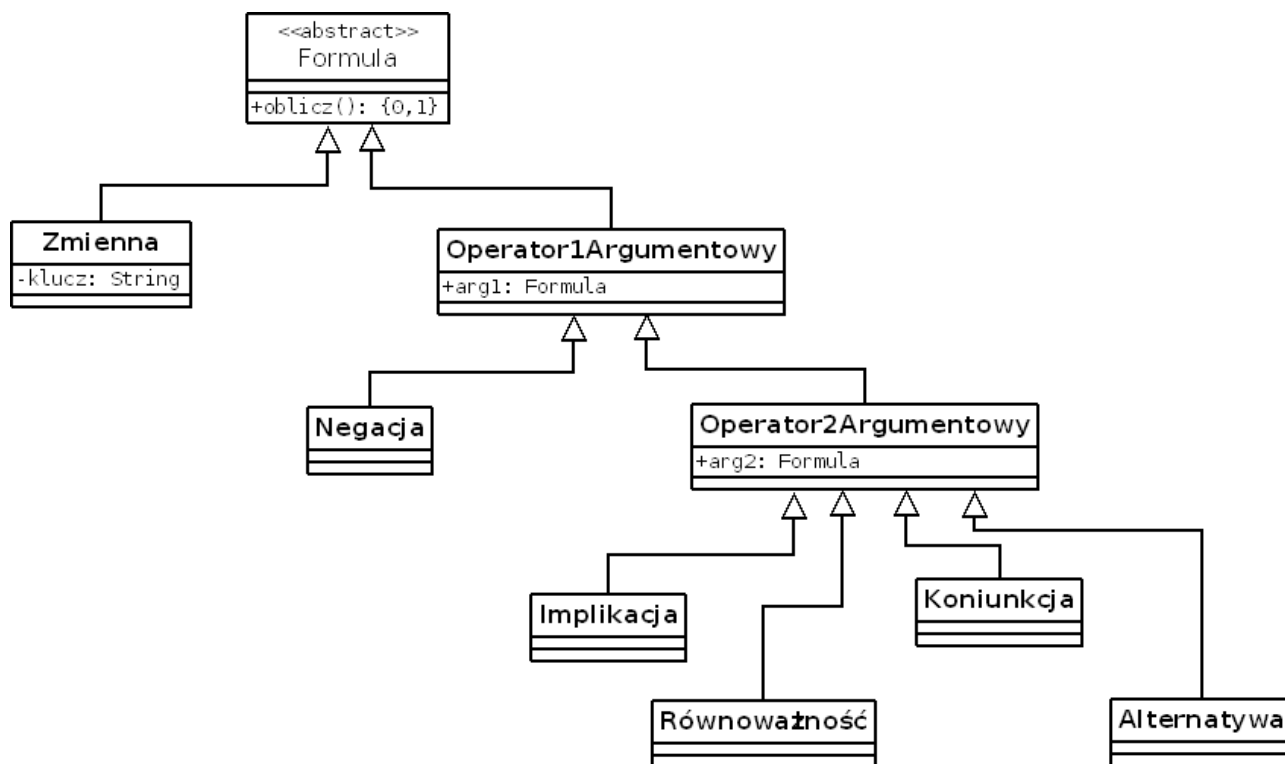
### 3. Implementacja modułu B

1. F jest zbiorem faktów (formuł zdaniowych), zapisanych jako ciągi znaków. Zbiór ten jest parsowany, następnie na jego podstawie zostaje stworzone drzewo rozbioru formuły (DRF). Liśćmi tego drzewa są formuły podstawowe (wejściowe i wyjściowe) traktowane jako zmienne.
2. Gdy zostanie podany wektor wartości logicznych formuł z  $\alpha_u$ , odpowiadające im zmienne na drzewie są ustalane.
3. Wartość logiczna DRF zostaje obliczona dla wszystkich możliwych kombinacji wartości logicznych nie ustalonych zmiennych.

4. Dla wszystkich kombinacji wartości zmiennych, dla których wartość logiczna drzewa będzie 1, zwracane są wartości zmiennych wyjściowych.

W podstawowej wersji, punkt pierwszy zostanie pominięty i DRF zostanie zapisane na sztywno w programie.

Uproszczony diagram klas:



Klasa **Zmienna** posiada klucz do słownika, w którym przechowywane są wartości zmiennych.