# Praca domowa Sztuczna Inteligencja

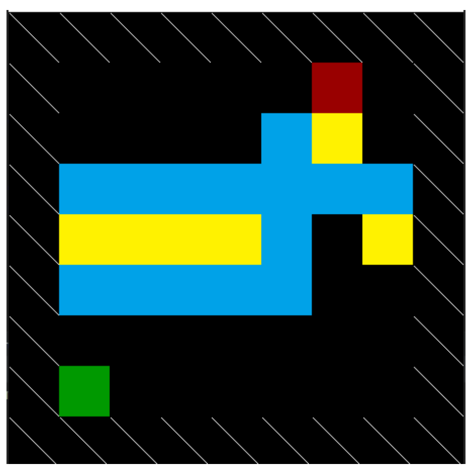
Piotr Sieński 184297

Kwiecień 2022

## 1 Szukanie najszybszej trasy jako przykład zastosowania metod przeszukiwania grafów

### 1.1 Opis problemu

Rozpatrywanym problemem jest szukanie najkrótszej ścieżki (zastosowanie np. w grach komputerowych) na przykładowej planszy (o wymiarach 7 x 7) zawierajacej 3 rodzaje pól: pola o koszcie przejścia 1 - czarne, pola o koszcie przejścia 5 - niebieskie oraz pola o koszcie przejścia 20 - żółte. Dodatkowo pole poczatkowe jest zaznaczone na zielono, a cel na czerwono.



### 1.2 Algorytm przeszukiwania

Wybranym algorytmem przeszukiwania jest algorytm A* z heurystyka określona jako sume odległości na osi x i y od celu.

#### 1.2.1 Zasada działania algorytmu

Poniższe punkty wykonywane sa w petli aż algorytm nie znajdzie pozycji końcowej.

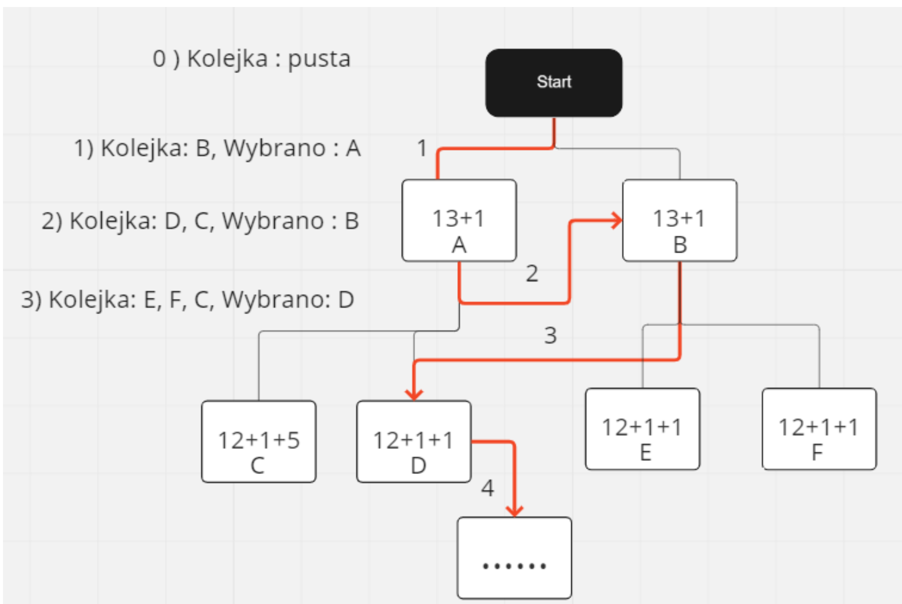
1) Algorytm w każdym kroku przeglada przylegajace do obecnie zajmowanej pozycji pola i dołacza je

do kolejki priorytetowej z priorytetem oznaczajacym sume wartości heurystyki i kosztu przemieszczenia sie do kolejnego pola.

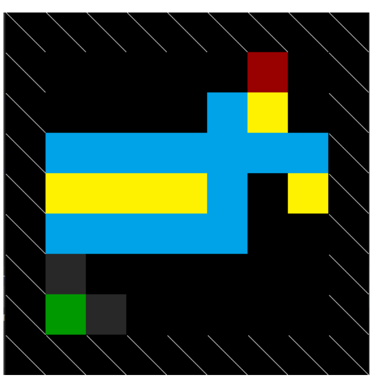
2) Algorytm pobiera z kolejki kolejnych pozycji ta o najmniejszej wartości heurystyki ustawiajac ja jako obecnie zajmowana pozycje i przechodzi z powrotem do punktu 1)

### 1.2.2 Przykładowy przebieg algorytmu

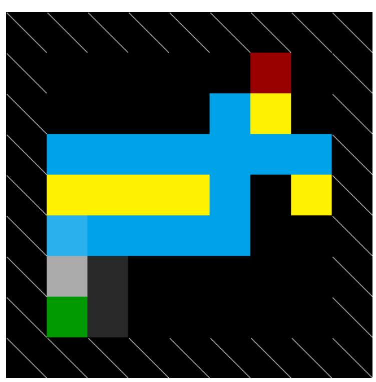
Na poniższym diagramie przedstawiono diagram przykładowego przebiegu algorytmu wraz z wygladem kolejki w każdym kroku. Na diagramie zaznaczone sa również koszty przejścia do poszczególnych pól - suma heurystyki i kosztu poruszenia sie.



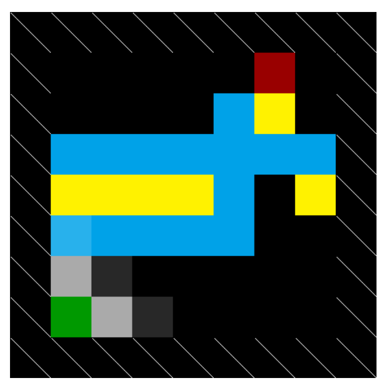

przebieg algorytmu - pierwsze 3 iteracje



krok 1

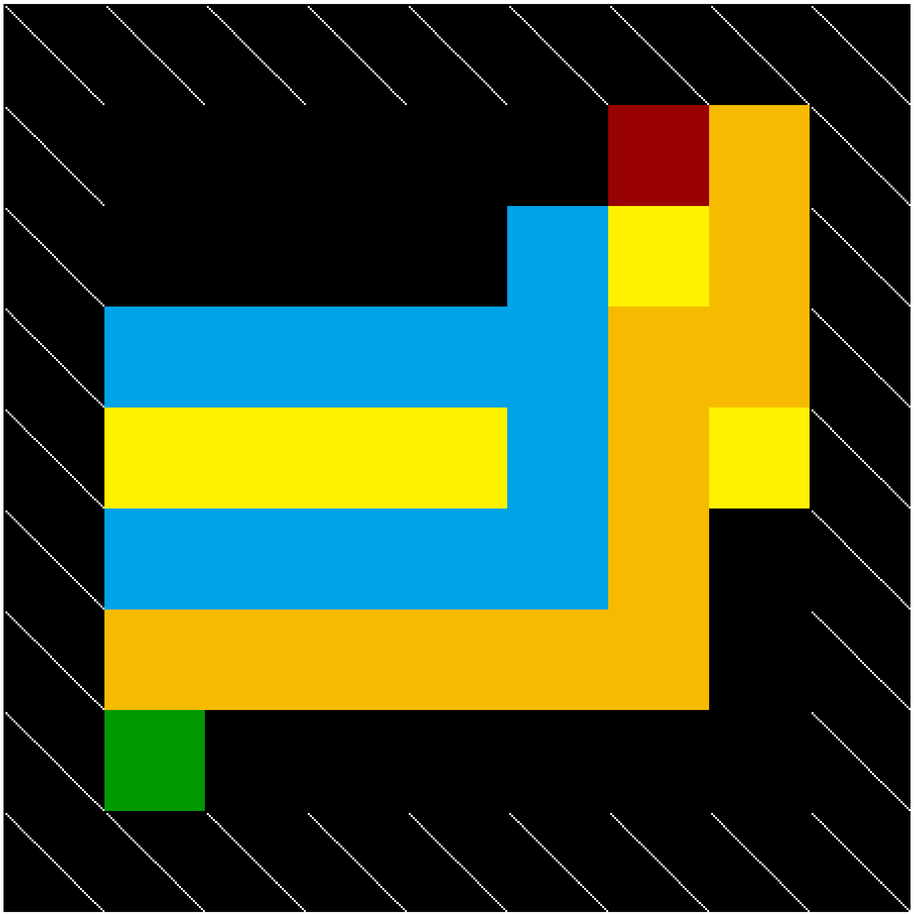


krok 2



krok 3

#### 1.2.3 Wynik działania algorytmu



Na pomarańczowo zaznaczono przebieg trasy znalezionej przez algorytm

Jak widać na powyższej ilustracji algorytm znalazł optymalna ścieżke omijajac pola o najwyższym koszcie.

# 2 Wnioskowanie w grze saper

## 2.1 Zasady gry

Gra polega na znalezieniu wszystkich bomb. Miejsca gdzie znajduja sie bomby oznaczane sa flaga. Jeśli na odkrytym polu znajduje sie liczba ( z przedziału 1 - 8) oznacza ona ile bomb znajduje sie w sasiedztwie tego pola.

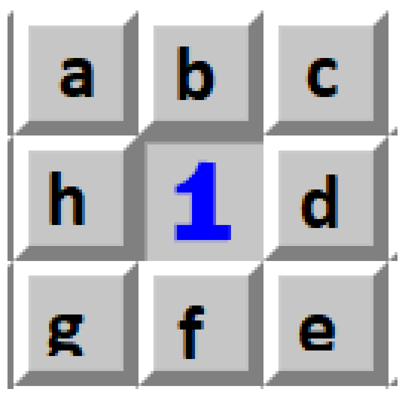
Agent grajacy w gre bedzie postepował według nastepujacych zasad:
- Pierwszy ruch w każdej grze wykonywany jest w sposób losowy.
- Agent wykonuje ruch poprzez oflagowanie lub odkrycie jednego kwadratu.
- Jeśli agent po rozważeniu wszystkich możliwych ruchów nie jest w stanie określić jednoznacznie ani jednego ruchu odkrywajacego nowy kwadrat, odkrywa losowy nie oflagowany kwadrat.

Agent w każdej "turze" (nowa "tura" rozpoczyna sie po odkryciu jakiegoś kwadratu) wnioskuje dla każdego z nieodkrytych pól przylegajacych do już odkrytych na podstawie wartości przylegajacych do niego pól odkrytych. Dodatkowo w celu uproszczenia wnioskowania dodać można reguły dla pól odkrytych określajace jednoznacznie status przylegajacych do nich pól nieodkrytych.

## 2.2 Reguły

Predykat Odkryte(x) zwraca 1 gdy pole x jest odkryte, 0 w przeciwnym wypadku, predykat Sasiaduje(x, y) zwraca 1 gdy y jest w sasiedztwie x, predykat Bomba(x) znaczy, że na danym polu znajduje sie bomba i powinno zostać ofalgowane.

Liczba n znajdujaca sie na danym odkrytym polu oznacza że w jego sasiedztwie znajduje sie dokładnie n bomb. Przykład symbolicznego zapisu powyższej reguły:




$(\mathrm{Bomba}(a) \wedge \neg Bomba(b) \wedge \neg Bomba(c) \wedge \neg Bomba(d) \wedge \neg Bomba(e) \wedge \neg Bomba(f) \wedge \neg Bomba(g) \wedge \neg Bomba(h)) \quad \vee \quad (Bomba(b) \wedge \neg Bomba(a) \wedge ...) \quad \vee \quad (Bomba(c) \wedge \neg Bomba(a) \wedge ...) \quad \vee ...$

Predykat A(x) zwraca 1, gdy liczba pól nieodkrytych wokół odkrytego pola x jest równa liczbie bomb w pobliżu tego pola

$$\forall_x A(x) \Rightarrow \forall_a (Sasiaduje(a, x) \land \neg Odkryte(a)) \Rightarrow Bomba(a) \tag{1}$$

Jeśli liczba pól nieodkrytych wokół odkrytego pola x jest równa liczbie bomb w pobliżu tego pola to wszystkie te nieodkryte pola musza być bombami.

Predykat B(x) zwraca 1, gdy liczba pól nieodkrytych wokół odkrytego pola x jest równa liczbie pól oflagowanych w pobliżu tego pola

$$\forall_x B(x) \Rightarrow \forall_a (Sasiaduje(a, x) \land \neg Odkryte(a)) \Rightarrow \neg Bomba(a) \tag{2}$$

Jeśli liczba pól nieodkrytych wokół odkrytego pola x jest równa liczbie pól oflagowanych w pobliżu tego pola to wszystkie te nieodkryte pola musza nie być bombami.

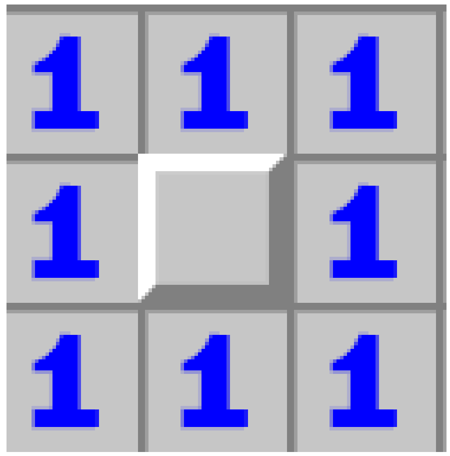
Przebieg wnioskowania:

Najpierw podejmowana jest próba dedukcji stanów pól nieodkrytych na podstawie reguł (1) i (2), jeśli to zawiedzie i nie zostanie odkryte żadne nowe pole podejmowana jest próba dedukcji stanu każdego pola przylegajacego do pól odkrytych na podstawie ogólnej reguły gry.

## 2.3 Przykłady wnioskowania

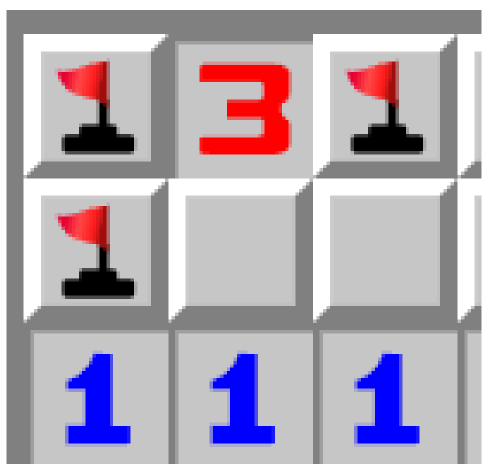
### 2.3.1 Stan w którym możliwa jest dedukcja przy pomocy reguły (1)

Jesteśmy w stanie wywnioskować, że zakryte pole jest bomba

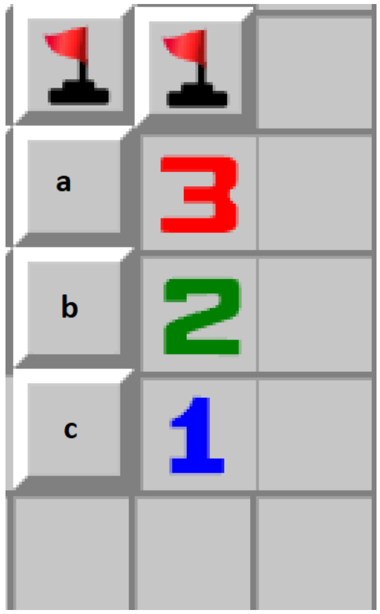



### 2.3.2 Stan w którym możliwa jest dedukcja przy pomocy reguły (2)

Jesteśmy w stanie wywnioskować że zakryte pola nie sa bombami



#### 2.3.3 Stan w którym możliwe jest wnioskowanie przy pomocy reguł ogółnych




Ze wzgledu na trudnośc w przedstawieniu bazy wiedzy w postaci klauzul wnioskowanie dla danego pola bedzie odbywać sie poprzez założenie że na danym polu jest bomba i generowanie nowych twierdzeń aż pojawi sie sprzeczność (lub nie).

W powyższym przypadku bedziemy próbować określić stan kwadratu oznaczonego jako **b**. Zapisujemy wiec reguły pochodzace ze wszystkich odkrytych pól sasiadujacych z **b**.

$$(Bomba(b) \wedge \neg Bomba(c)) \quad \vee \quad (Bomba(c) \wedge \neg Bomba(b)) \tag{3}$$

$$\begin{aligned} &(Bomba(a) \wedge Bomba(b) \wedge \neg Bomba(c)) \quad \vee \\ \vee \quad &(Bomba(a) \wedge Bomba(c) \wedge \neg Bomba(b)) \quad \vee \\ \vee \quad &(Bomba(b) \wedge Bomba(c) \wedge \neg Bomba(a)) \end{aligned} \tag{4}$$

$$(Bomba(a) \wedge \neg Bomba(b)) \quad \vee \quad (Bomba(b) \wedge \neg Bomba(a)) \tag{5}$$

Reguła (3) pochodzi z pola z cyfra 1, (4) od pola z cyfra 2, (5) od pola z cyfra 3.

**Wnioskowanie :**

1) Do bazy wiedzy składajacej sie z (3), (4) i (5) dodajemy

$$Bomba(b) \tag{6}$$

2) Na podstawie (3) i (6) do bazy wiedzy dodajemy

$$\neg Bomba(c) \tag{7}$$

3) Na podstawie (4) i (6) do bazy wiedzy dodajemy

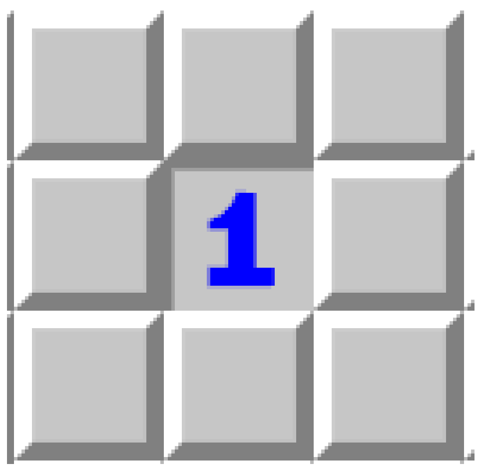
$$(Bomba(a) \wedge \neg Bomba(c)) \quad \vee \quad (Bomba(c) \wedge \neg Bomba(a)) \tag{8}$$

4) Na podstawie (5) i (6) do bazy wiedzy dodajemy

$$\neg Bomba(a) \tag{9}$$

5) Na podstawie (7), (8) i (9) otrzymujemy sprzeczność, wiec na polu **b** nie ma bomby.
6) Wiedzac że na **b** nie ma bomby z (3), (4) i (5) można wywnioskować że bomby znajduja sie na **a** i **c**. Odkrywamy wiec pole **b**, a **a** i **c** flagujemy.

#### 2.3.4 Stan w którym nie możliwe jest jednoznaczne wnioskowanie




Jak widać wnioskowanie z żadnej z reguł nie doprowadzi do jednoznacznych wyników. W takim przypadku agent powinien wykonać losowy ruch.

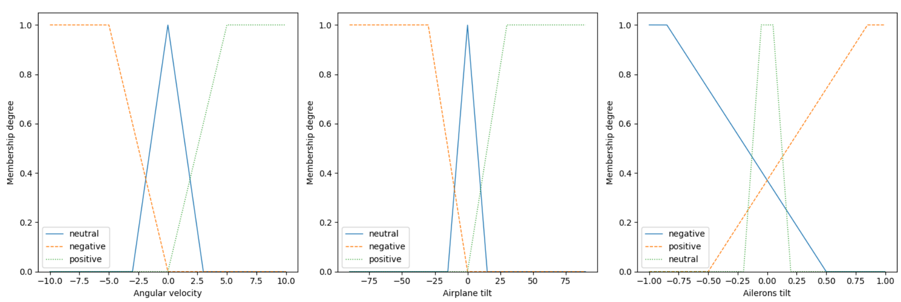
# 3 Symulacja komponentu autopilota samolotu jako przykład zastosowania logiki rozmytej

## 3.1 Opis problemu

Rozpatrywanym problemem jest symulacja komponentu autopilota samolotu odpowiedzialnego za utrzymanie stałego przechyłu samolotu. Wejściami dla systemu logiki rozmytej beda: przechył samolotu w stopniach od -90 do 90 oraz predkość przechylania sie samolotu w stopniach na sekunde (od -10 do 10). Wyjściem bedzie poziom wychylenia lotek samolotu od -1 do 1, gdzie -1 oznacza wychylenie powodujace wychylenie samolotu w lewa, a 1 w prawa.

## 3.2 Zmienne lingwistyczne i funkcje przynależności

Za równo dla predkości katowej jak i przechyłu zdefiniowane sa po trzy zmienne lingwistyczne określajace negatywna, średnia (bliska zeru) oraz pozytywna wartość wejścia. Zmienne pozytywne i negatywne maja przypisane funkcje przynależności trapezoidalne, a zmienna średnia funkcje trójkatna. Zmienne przypisane do parametru wyjściowego różnia sie tym, że zmienna średnia ma przypisana funkcje trapezoidalna.



## 3.3 Reguły

Zostały zdefiniowane nastepujace reguły:

$$\begin{aligned} &\textit{Jeśli przechyl jest średni i predkość jest średnia} \\ &\textit{to wychylenie lotek jest neutralne} \end{aligned} \tag{10}$$

$$\begin{aligned} &\textit{Jeśli przechyl jest ujemny ipredkość jest } (\textit{ujemna lub średnia lub neutralna}) \\ &\textit{to wychylenie lotek jest dodatnie} \end{aligned}$$

(11)

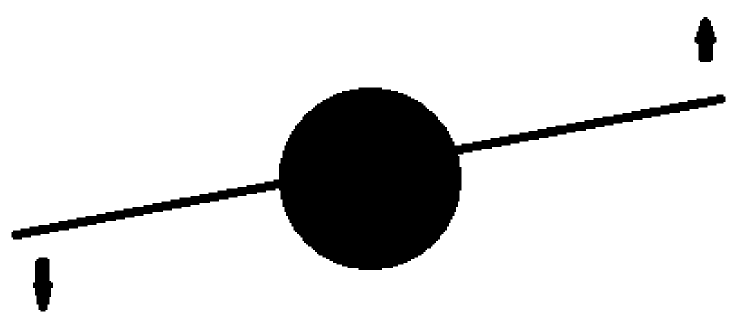
$$\begin{aligned} &\textit{Jeśli przechyl jest dodatni i predkość jest } (\textit{ujemna lub średnia lub neutralna}) \\ &\textit{to wychylenie lotek jest ujemne} \end{aligned}$$

(12)

## 3.4 Wnioskowanie

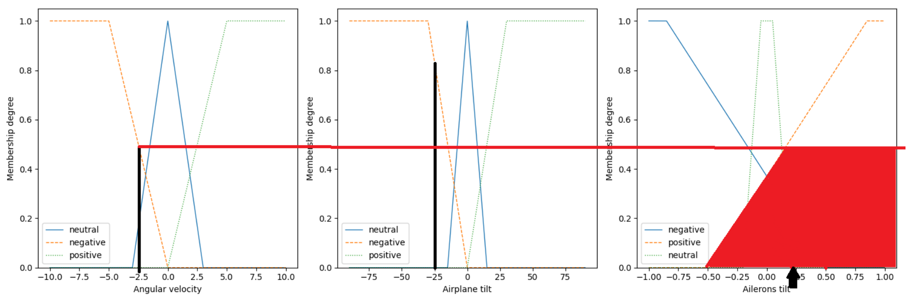
### 3.4.1 Przykład

Dane: przechył = -25 stopni, predkość przechyłu = -2 stopnie / sekunde



Ilustracja wychylenia samolotu i kierunku przechyłu

**Rozmywanie:**
przechył negatywny (-25) = 0.85
przechył pozytywny (-25) = 0
przechył średni (-25) = 0
predkość negatywna (-2) = 0.5
predkość średnia (-2) = 0.2
predkość pozytywna (-2) = 0
**Operacje rozmyte:**
Reguła 1 : min[0, 0.2]=0
Reguła 2: min[0.85, max(0.5, 0.2, 0)]=0.5
Reguła 3: min[0, max(0.5, 0.2, 0)] = 0
Funkcja wychylenie lotek dodatnie odcieta na 0.5
**Kompozycja:** Przedstawiona na rysunku poniżej




Graficzna reprezentacja wnioskowania z zaznaczona decyzja

**Precyzowanie:**
Środek ciężkości trapezu c można wyznaczyć ze wzoru

$$c = (h/3) \cdot (((2 \cdot a) + b)/(a + b))$$

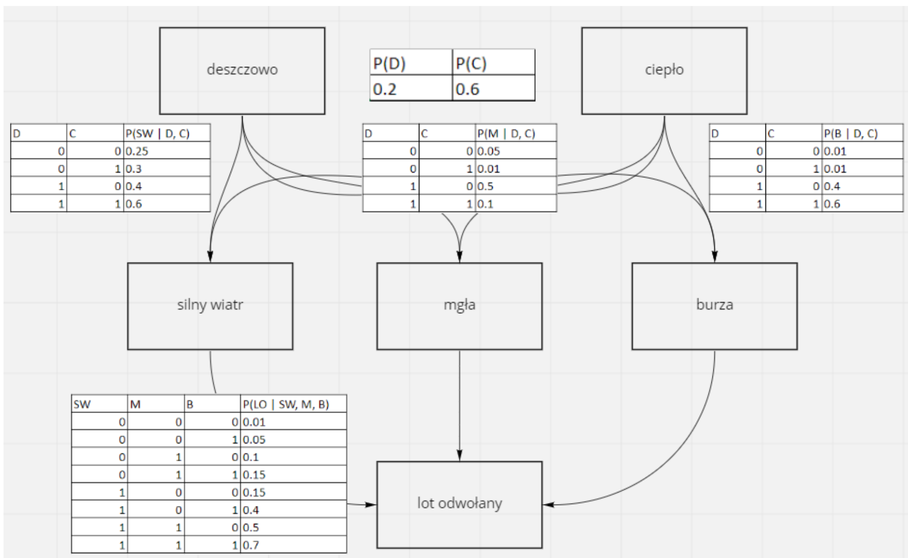
a = 0.75, b = 1.5, h = 0.5, wiec
**Odpowiedź : 0.22**

# 4 Wnioskowanie o prawdopodobieństwie odwołania lotu samolotem na podstawie Sieci Bayesowskiej

## 4.1 Opis problemu

Rozpartrywanym problemem jest obliczanie prawdopodobieństwa odwołania lotu samolotem na podstawie Sieci Bayesowskiej. Po obliczeniu rozkładu łacznego prawdopodobieństwa możemy zadawać pytania na bazie sieci.

## 4.2 Sieć Bayesowska




Możemy wyznaczyć łaczny rozkład prawdopodobieństwa jako:

$$
\begin{gathered}
P(LO, SW, M, B, D, C) = P(LO|S, M, B) \cdot P(SW, M, B, D, C) = \\
P(LO|SW, M, B) \cdot P(SW|D, C) \cdot P(M|D, C) \cdot P(B|D, C) \cdot P(D) \cdot P(C)
\end{gathered} \tag{13}
$$

## 4.3 Wnioskowanie

### 4.3.1 Obliczanie prawdopodobieństwa odwołania lotu jeśli jest ciepło, deszczowo i wieje silny wiatr

Korzystajac z (13) obliczamy sume poszczególnych prawdopodobieństw

$$Szukane\ prawdopodobie\acute{n}stwo\ :\ P(LO|SW,D,C) =$$
$$P(LO,SW,M,B,D,C) + P(LO,SW,\neg M,B,D,C)+$$
$$P(LO,SW,M,\neg B,D,C) + P(LO,SW,\neg M,\neg B,D,C)$$

$$\begin{aligned} &P(LO,SW,M,B,D,C) = \\ =&P(LO|SW,M,B) \cdot P(SW|D,C) \cdot P(M|D,C) \cdot P(B|D,C) \cdot P(D) \cdot P(C) = \\ =&0.7 \cdot 1 \cdot 0.1 \cdot 0.6 \cdot 1 \cdot 1 = 0.042 \end{aligned} \tag{14}$$

$$\begin{aligned} &P(LO,SW,\neg M,B,D,C) = \\ =&P(LO|SW,\neg M,B) \cdot P(SW|D,C) \cdot P(\neg M|D,C) \cdot P(B|D,C) \cdot P(D) \cdot P(C) = \\ =&0.4 \cdot 1 \cdot 0.9 \cdot 0.6 \cdot 1 \cdot 1 = 0.216 \end{aligned} \tag{15}$$

$$\begin{aligned} &P(LO,SW,M,\neg B,D,C) = \\ =&P(LO|SW,M,\neg B) \cdot P(SW|D,C) \cdot P(M|D,C) \cdot P(\neg B|D,C) \cdot P(D) \cdot P(C) = \\ =&0.5 \cdot 1 \cdot 0.1 \cdot 0.4 \cdot 1 \cdot 1 = 0.02 \end{aligned} \tag{16}$$

$$\begin{aligned} &P(LO,SW,\neg M,\neg B,D,C) = \\ =&P(LO|SW,\neg M,\neg B) \cdot P(SW|D,C) \cdot P(\neg M|D,C) \cdot P(\neg B|D,C) \cdot P(D) \cdot P(C) \\ =&0.15 \cdot 1 \cdot 0.9 \cdot 0.4 \cdot 1 \cdot 1 = 0.054 \end{aligned} \tag{17}$$

$$P(LO|SW,D,C) = 0.042 + 0.216 + 0.02 + 0.054 = 0.332 \tag{18}$$

**Wiec szukane prawdopodobieństwo wynosi 0.332**

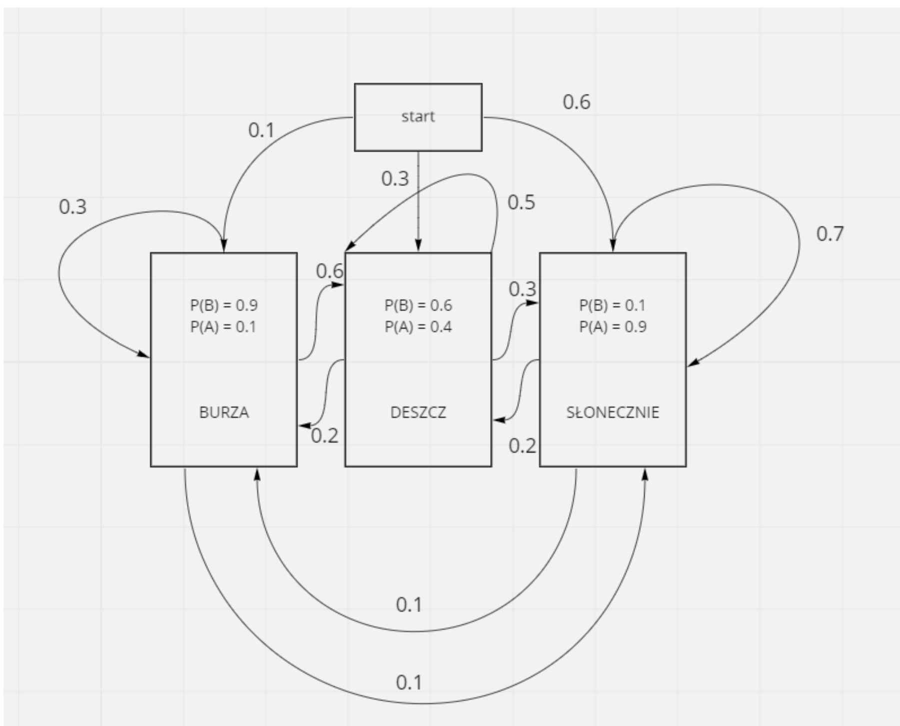
# 5 Wnioskowanie o pogodzie na danym lotnisku na podstawie przekierowań lotów przy użyciu Ukrytych Modeli Markova

## 5.1 Opis problemu

Rozpatrywanym problemem jest przewidzenie pogody w ciagu pewnego okresu czasu na danym lotnisku na podstawie danych o tym, czy loty na to lotnisko zostały przekierowane czy nie.

## 5.2 Model

Na poniższym modelu strzałkami oznaczone sa prawdopodobieństwa przejśc miedzy kolejnymi stanami, bloki oznaczaja ukryte stany, a w środku każdego bloku znajduje sie informacja o prawdopodobieństwie wygenerowania danego stanu obserwowanego.
P oznacza prawdopodobieństwo przekierowania lotu podczas danego stanu i NP analogicznie prawdopodobieństwo że lot nie został przekierowany.

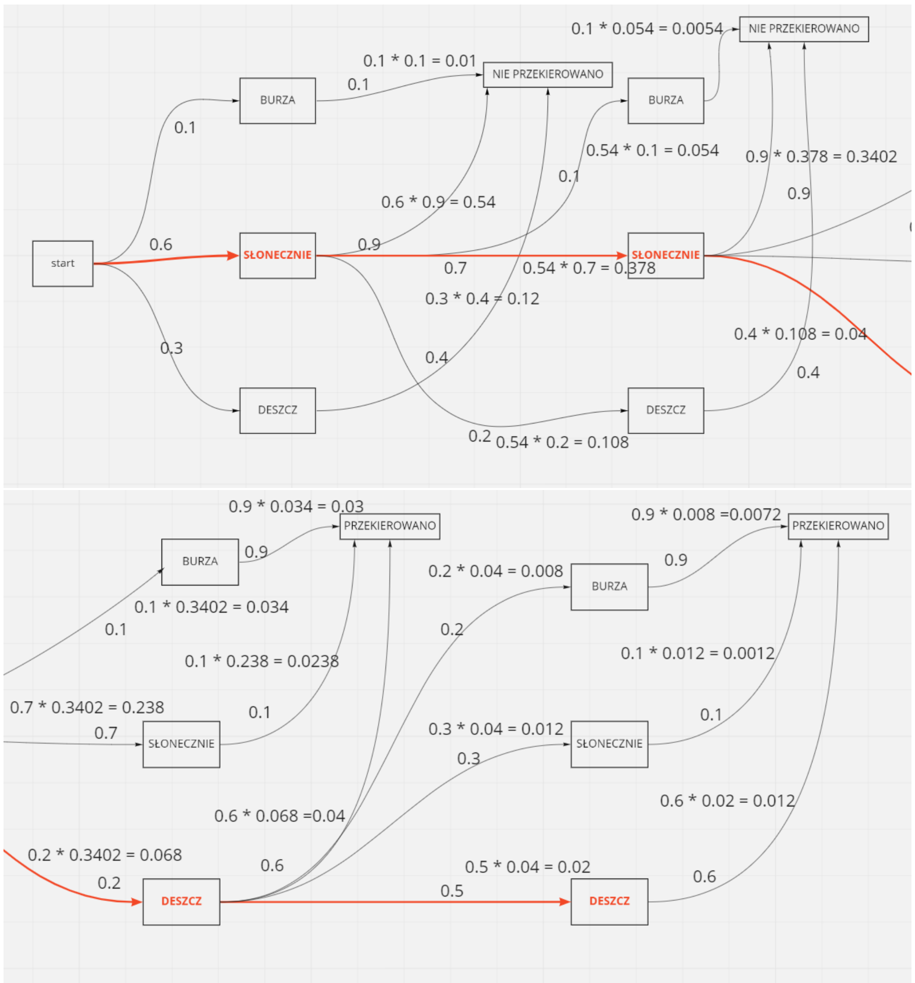



## 5.3 Wnioskowanie

Wnioskiwanie polega na znalezieniu najbardziej prawdopodobnej sekwencji stanów ukrytych dla danej sekwencji stanów zaobserwowanych i obliczeniu tego prawdopodobieństwa

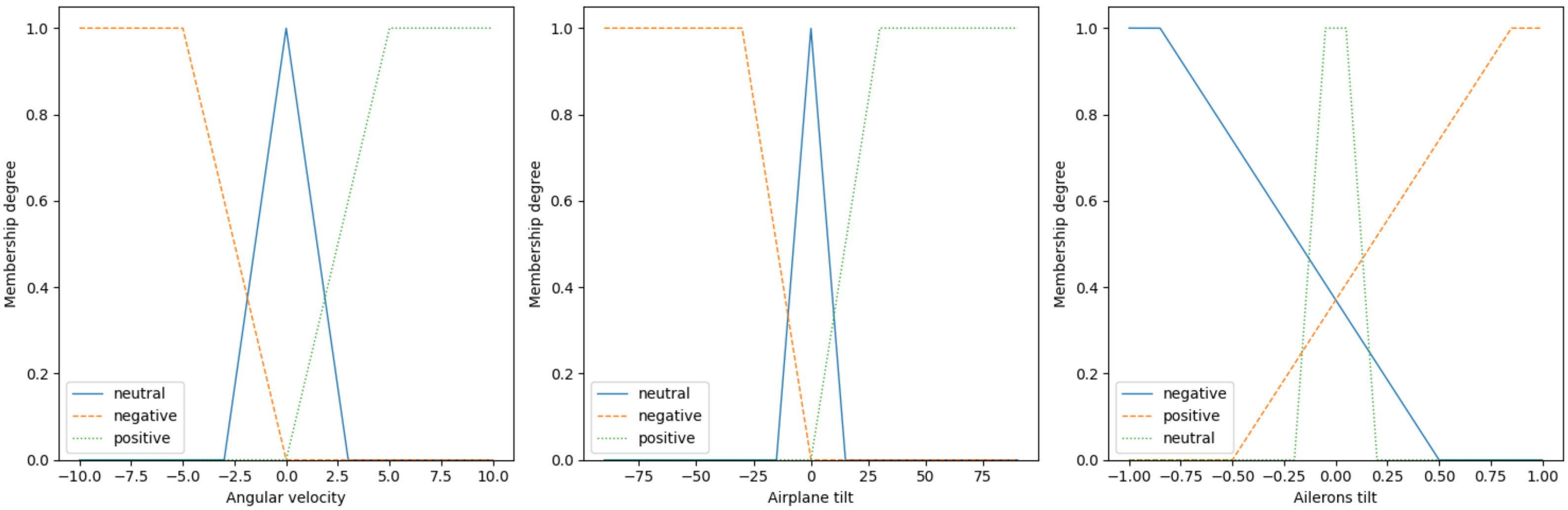
### 5.3.1 Dany stan: Nie przekierowano, Nie przekierowano, Przekierowano, Przekierowano

Wnioskowanie polega na analizie prawdopodobieństw przejść miedzy kolejnymi stanami i wybieranie tych przejść, które gwarantuja najwyższe prawdopodobieństwo. Poniżej przedstawiony jest diagram wraz z obliczeniami.

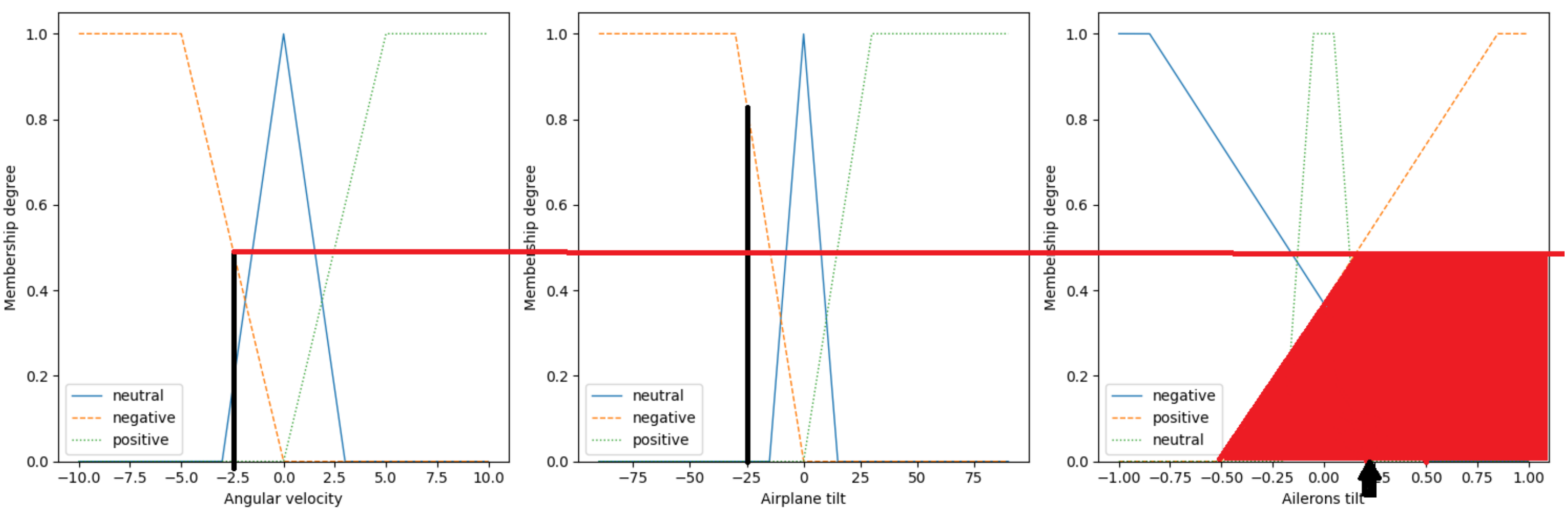



**Odpowiedź: Słonecznie, Słonecznie, Deszcz, Deszcz, z prawdopodobieństwem 0.012**

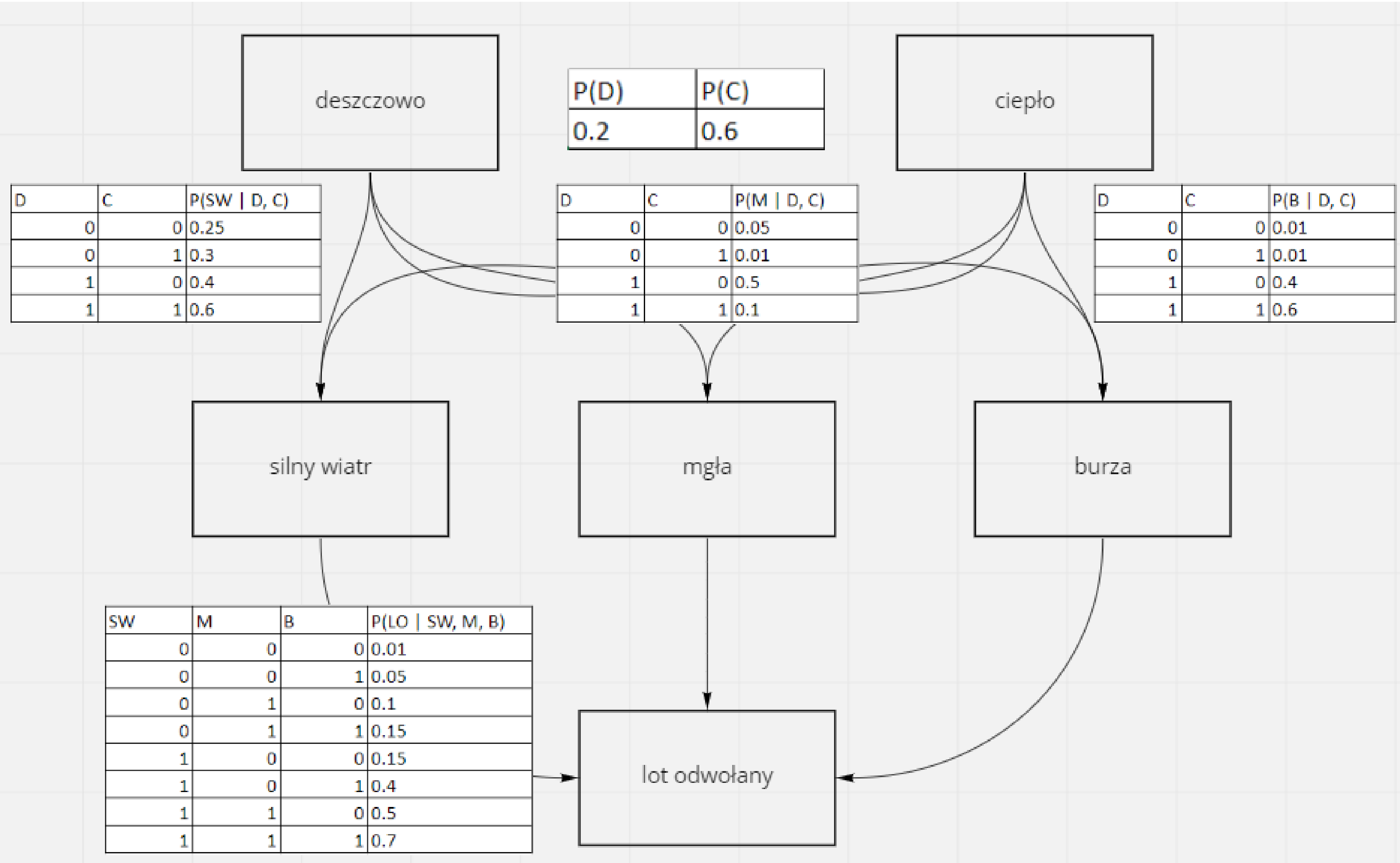
## 6 Ilustracje w wiekszym formacie




Pełen diagram z zadania 3.2




Pełen diagram z zadania 3.3




| P(D) | P(C) |
|---|---|
| 0.2 | 0.6 |

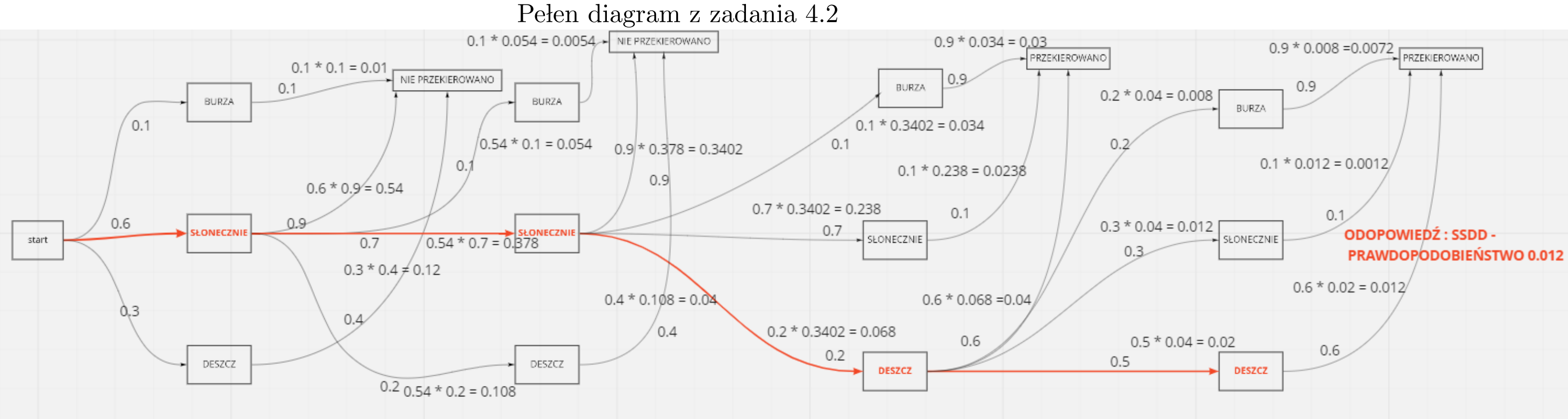
| D | C | P(SW \| D, C) |
|---|---|---|
| 0 | 0 | 0.25 |
| 0 | 1 | 0.3 |
| 1 | 0 | 0.4 |
| 1 | 1 | 0.6 |

| D | C | P(M \| D, C) |
|---|---|---|
| 0 | 0 | 0.05 |
| 0 | 1 | 0.01 |
| 1 | 0 | 0.5 |
| 1 | 1 | 0.1 |

| D | C | P(B \| D, C) |
|---|---|---|
| 0 | 0 | 0.01 |
| 0 | 1 | 0.01 |
| 1 | 0 | 0.4 |
| 1 | 1 | 0.6 |

| SW | M | B | P(LO \| SW, M, B) |
|---|---|---|---|
| 0 | 0 | 0 | 0.01 |
| 0 | 0 | 1 | 0.05 |
| 0 | 1 | 0 | 0.1 |
| 0 | 1 | 1 | 0.15 |
| 1 | 0 | 0 | 0.15 |
| 1 | 0 | 1 | 0.4 |
| 1 | 1 | 0 | 0.5 |
| 1 | 1 | 1 | 0.7 |

Pełen diagram z zadania 4.2




Pełen diagram z zadania 5