

REPREZENTACJA WIEDZY

## REALIZACJE SCENARIUSZY DZIAŁAŃ

PROJEKT NR 5

SZEF:

**ROBERT JAKUBOWSKI**

AUTORZY:

MARIUSZ AMBROZIAK

PAWEŁ BIELICKI

KAROL BOCIAN

HANNA DZIEGCIAR

KAROL DZITKOWSKI

MATEUSZ JANKOWSKI

WIKTOR RYCIUK

WARSZAWA, 19 KWIETNIA 2014

## Spis treści

<b>1. Opis zadania</b>	<b>3</b>
<b>2. Opis języka akcji</b>	<b>3</b>
2.1. Sygnatura języka . . . . .	4
2.2. Opis domeny . . . . .	4
2.3. Scenariusze działań . . . . .	5
2.4. Semantyka . . . . .	5
<b>3. Opis języka kwerend</b>	<b>6</b>
<b>4. Przykłady</b>	<b>8</b>
4.1. Pytanie czy dany scenariusz może wystąpić . . . . .	8
4.2. Pytanie czy dany warunek zachodzi w danym czasie . . . . .	9
4.3. Pytanie czy dana akcja jest wykonywana w pewnym czasie . . . . .	11
4.4. Pytanie czy podany cel jest osiągalny . . . . .	12

# 1 Opis zadania

Zadaniem projektu jest opracowanie i zaimplementowanie:

- języka akcji pewnej klasy systemów dynamicznych,
- języka kwerend, zapewniającego uzyskanie odpowiedzi na określone pytania.

Szczegółowy opis klasy systemów dynamicznych oraz języka akcji jest opisany w rozdziale 'Opis języka akcji', natomiast język kwerend oraz zadawane pytania znajdują się w rozdziale 'Opis języka kwerend'. W tym dokumencie znajdują się również przykłady. Pokazują one konkretne przypadki użycia oraz oczekiwane wyniki działania programu.

# 2 Opis języka akcji

Język akcji, zaprojektowany na potrzeby zadania, spełnia następujące warunki:

1. Prawo inercji.
2. Sekwencyjność działań.
3. Możliwe akcje niedeterministyczne.
4. Liniowy model czasu - czas dyskretny.
5. Pełna informacja o wszystkich:
  - (a) akcjach,
  - (b) skutkach bezpośrednich.
6. Akcja posiada:
  - (a) warunek początkowy,
  - (b) czas trwania  $t \geq 1$ ,  $t \in \mathbb{N}$ ,
  - (c) efekt akcji.
7. Podczas trwania akcji, wartości zmiennych, na które ona wpływa, nie są znane.
8. Występujące rodzaje efektów:
  - (a) środowiskowe: zmiany wartości zmiennych systemu,
  - (b) dynamiczne: wystąpienie akcji może wywołać wystąpienie innych akcji po  $d \geq 0$  jednostkach czasu od jej zakończenia.
9. W pewnych stanach akcje mogą być niewykonalne. Takie stany są określone przez podanie konkretnych punktów czasowych, albo przez określenie warunków logicznych.
10. Stany opisywane częściowo przez obserwacje.
11. Pewne stany mogą rozpocząć wykonywanie pewnych akcji.

Językiem odpowiadającym powyższym założeniom jest język *AL* opisujący domeny akcji z czasem liniowym.

## 2.1 Sygnatura języka

Sygnaturą języka jest następująca trójka zbiorów:  $\psi = (F, Ac, \mathbb{N})$ , gdzie:

$F$  – zbiór zmiennych inercji (fluentów)

$Ac$  – zbiór akcji

$\mathbb{N}$  – zbiór liczb naturalnych (czas trwania akcji)

## 2.2 Opis domeny

### Oznaczenia:

$f \in F$  – fluent

$Ac_i, Ac_j \in Ac$  – akcje

$\alpha, \pi \in Forms(F)$  – warunki

$d_i, d \in \mathbb{N}$

### Rodzaje zdań występujących w projektowanym języku (domena języka):

- $(Ac_i, d_i)$  causes  $\alpha$  if  $\pi$   
Akcja  $Ac_i$  trwająca  $d_i$  jednostek czasu powoduje stan  $\alpha$ , jeśli zachodzi warunek  $\pi$ .
- $(Ac_i, d_i)$  invokes  $(Ac_j, d_j)$  after  $d$  if  $\pi$   
Akcja  $Ac_i$  trwająca  $d_i$  jednostek czasu powoduje wykonanie akcji  $Ac_j$  trwającej  $d_j$  jednostek czasu po  $d$  jednostkach czasu od zakończenia akcji  $Ac_i$ , jeśli przy jej rozpoczęciu zachodzi warunek  $\pi$ .
- $(Ac_i, d_i)$  releases  $f$  if  $\pi$   
Akcja  $Ac_i$  trwająca  $d_i$  jednostek czasu powoduje uwolnienie  $f$  po zakończeniu akcji  $Ac_i$ , jeśli zachodzi warunek  $\pi$ .
- $\pi$  triggers  $(Ac_i, d_i)$   
Akcja  $Ac_i$  trwająca  $d_i$  jednostek czasu jest wykonywana, jeśli zajdzie warunek  $\pi$ .
- impossible  $(Ac_i, d_i)$  at  $d$   
Akcja  $Ac_i$  trwająca  $d_i$  jednostek czasu jest niewykonalna w chwili  $d$ .
- impossible  $(Ac_i, d_i)$  if  $\pi$   
Akcja  $Ac_i$  trwająca  $d_i$  jednostek czasu jest niewykonalna, jeśli zachodzi warunek  $\pi$ .

## 2.3 Scenariusze działań

Scenariusze działań opisane są w następujący sposób:

- $Sc = (OBS, ACS)$
- $OBS = \{(\gamma_1, t_1), \dots, (\gamma_m, t_m)\}$ , gdzie:  
 $m \geq 0$  – liczba obserwacji,  
każda obserwacja jest stanem częściowym (stanem spełniającym warunek  $\gamma_i$  w pewnym punkcie czasu  $t_i$ ).  
 $\gamma_i \in Forms(F)$
- $ACS = \{((Ac_1, d_1), t_1), \dots, ((Ac_n, d_n), t_n)\}$ , gdzie:  
 $n \geq 0$ ,  
 $Ac_i$  – akcja,  
 $d_i$  – czas trwania akcji,  
 $t_i$  – punkt w czasie (rozpoczęcie akcji).

## 2.4 Semantyka

**Definicja 2.1.** *Semantyczną strukturą języka AL nazywamy system  $S = (H, O, E, T_\infty)$  taki, że:*

- $H : Forms(F) \times \mathbb{N} \rightarrow \{0, 1\}$  jest funkcją historii, pozwala ona stwierdzić, jaki stan ma pewny fluent lub czy dana formuła jest spełniona, dla określonej chwili czasu  $t$ .
- $O : Ac \times \mathbb{N} \rightarrow 2^F$  jest funkcją okluzji. Dla pewnej ustalonej akcji  $A \in Ac$ , chwili czasu  $t \in \mathbb{N}$ , funkcja  $O(A, t)$  zwraca zbiór fluentów, na który akcja  $A$  ma wpływ, jeśli będzie ona trwała w chwili  $t$ . Wartość funkcji okluzji będziemy nazywać regionem okluzji.
- $E \subseteq Ac \times \mathbb{N} \times \mathbb{N}$  jest relacją wykonania akcji. Trójka  $(A, t, d)$  należy do relacji  $E$  jeśli akcja  $A$  trwająca  $d$  jednostek czasu jest rozpoczęta w czasie  $t$ . W naszym modelu zakładamy warunek sekwencyjności działań. Oznacza on, że tylko jedną akcję możemy wykonać w danym czasie tak, więc jeśli  $(t_1, t_1 + d_1) \subseteq (t_2, t_2 + d_2)$  lub  $(t_2, t_2 + d_2) \subseteq (t_1, t_1 + d_1)$  oraz  $(A, t_1, d_1) \in E$  i  $(B, t_2, d_2) \in E$ , to  $A = B$ ,  $t_1 = t_2$ ,  $d_1 = d_2$ . Natomiast jeżeli  $(t_1, t_1 + d_1) \not\subseteq (t_2, t_2 + d_2)$ ,  $(t_2, t_2 + d_2) \not\subseteq (t_1, t_1 + d_1)$  oraz  $(A, t_1, d_1) \in E$  i  $(B, t_2, d_2) \in E$ , to  $t_1 + d_1 < t_2$  lub  $t_2 + d_2 < t_1$ .
- $T_\infty \in \mathbb{N}$  jest czasem zakończenia scenariusza, może to być dowolnie duża ustalona liczba naturalna. Mówi ona do kiedy powinny być zakończone wszystkie akcje w danym scenariuszu. Oznacza to, że nie będziemy rozpatrywać nieskończonych scenariuszy.

Niech:  $A, B$  będą akcjami,  $f$  - fluentem,  $\alpha, \pi$  - będą formułami,  $d, d_2, d_3$  - liczbami naturalnymi (oznaczającymi czas trwania akcji) oraz  $fl(\alpha)$  będzie zbiorem fluentów występujących w  $\alpha$ . Wtedy dla zdań języka AL muszą być spełnione następujące warunki:

- Dla każdego wyrażenia  $((A, d) \text{ causes } \alpha \text{ if } \pi) \in D$  i dla każdego momentu w czasie  $t \in \mathbb{N}$ , jeżeli  $H(\pi, t) = 1$  oraz  $(A, t, d) \in E$ , wtedy  $H(\alpha, t + d) = 1$  oraz dla każdego momentu w czasie  $d' \in \mathbb{N}$  takiego, że  $1 \leq d' \leq d$  mamy  $fl(\alpha) \subseteq O(A, t + d')$ .
- Dla każdego wyrażenia  $((A, d) \text{ release } f \text{ if } \pi) \in D$  i dla każdego momentu czasu  $t \in \mathbb{N}$ , jeżeli  $H(\pi, t) = 1$  oraz  $(A, t, d) \in E$ , wtedy dla każdego momentu w czasie  $d' \in \mathbb{N}$  takiego, że  $1 \leq d' \leq d$  mamy  $f \in O(A, t + d')$ .

- Dla każdego wyrażenia  $(\pi \text{ triggers } (A, d)) \in D$  i dla każdego momentu czasu  $t \in \mathbb{N}$ , jeżeli  $H(\pi, t) = 1$  oraz  $t + d \leq T_\infty$ , wtedy  $(A, t, d) \in E$ .
- Dla każdego wyrażenia  $((A, d_1) \text{ invokes } (B, d_2) \text{ after } d \text{ if } \pi) \in D$  i dla każdego momentu czasu  $t \in \mathbb{N}$ , jeżeli  $H(\pi, t) = 1$ ,  $(A, t, d_1) \in E$  oraz  $t + d + d_1 + d_2 \leq T_\infty$ , wtedy  $(B, t + d + d_1, d_2) \in E$ .
- Dla każdego wyrażenia  $(\text{impossible } (A, d) \text{ at } t)$  mamy  $(A, t, d) \notin E$
- Dla każdego wyrażenia  $(\text{impossible } (A, d) \text{ if } \pi)$  i dla każdego momentu czasu  $t \in \mathbb{N}$ , jeżeli  $H(\pi, t) = 1$ , wtedy  $(A, t, d) \notin E$ .

**Definicja 2.2.** Niech  $S = (H, O, E, T_\infty)$  będzie strukturą języka  $AL$ ,  $Sc = (OBS, ACS)$  będzie scenariuszem, oraz  $D$  dziedziną. Powiemy, że  $S$  jest strukturą dla  $Sc$  zgodną z opisem domeny  $D$ , jeśli:

- Dla każdej obserwacji  $(\alpha, t) \in OBS$  mamy  $H(\alpha, t) = 1$
- $ACS \subseteq E$
- Dla każdej akcji  $A \in Ac$  oraz dla każdego czasu rozpoczęcia akcji i jej długości  $t, d \in \mathbb{N}$ , jeżeli  $(A, t, d) \in E$ , to  $t + d \leq T_\infty$ .

**Definicja 2.3.** Niech  $O_1, O_2: X \rightarrow 2^Y$ . Mówimy, że  $O_1 \prec O_2$  jeżeli  $\forall x \in X \ O_1(x) \subseteq O_2(x)$  oraz  $O_1 \neq O_2$ .

**Definicja 2.4.** Niech  $S = (H, O, E, T_\infty)$  będzie strukturą dla scenariusza  $Sc = (OBS, ACS)$  zgodną z opisem dziedziny  $D$ . Mówimy, że  $S$  jest  $O$ -minimalną strukturą, jeżeli nie istnieje struktura  $S' = (H', O', E', T'_\infty)$  dla tego samego scenariusza i domeny taka, że  $O' \prec O$ .

**Definicja 2.5.** Niech  $S = (H, O, E, T_\infty)$  będzie strukturą dla scenariusza  $Sc = (OBS, ACS)$  zgodną z opisem domeny  $D$ .  $S$  będziemy nazywać modelem  $Sc$  zgodnym z opisem  $D$  jeżeli:

- $S$  jest  $O$ -minimalny
- Dla każdego momentu w czasie  $t, d \in \mathbb{N}$ ,  $\{f \in F: H(f, t) \neq H(f, t + d)\} \subseteq O(A, t + d)$  dla pewnej akcji  $A$ .
- Nie istnieje, żadna struktura  $S' = (H', O', E')$  dla  $Sc$  zgodna z opisem  $D$ , która spełnia poprzednie warunki oraz taka, że  $E' \subset E$ .

**Uwaga 2.1.** Nie dla każdego scenariusza można ułożyć model. Mówimy, że scenariusz  $Sc$  jest zgodny jeśli istnieje do niego model zgodny z dziedziną  $D$ .

### 3 Opis języka kwerend

Zdefiniowany język akcji może być odpytywany przez poniżej zaprezentowany język kwerend, który zapewnia uzyskanie odpowiedzi  $TRUE/FALSE$  na następujące pytania:

**Q1.** Czy podany scenariusz jest możliwy do realizacji zawsze/kiedykolwiek?

- *always/ever executable Sc*  
Oznacza, że scenariusz  $Sc$  zawsze/kiedykolwiek jest możliwy do realizacji.

**Q2.** Czy w chwili  $t \geq 0$  realizacji podanego scenariusza warunek  $\gamma$  zachodzi zawsze/kiedykolwiek?

- *always/ever  $\gamma$  at  $t$  when  $Sc$*

Oznacza, że zawsze/kiedykolwiek w chwili  $t$  realizacji scenariusza  $Sc$  zachodzi warunek  $\gamma$ .

- *always/ever  $\gamma$  when  $Sc$*

Oznacza, że zawsze/kiedykolwiek w pewnej chwili  $t$  realizacji scenariusza  $Sc$  zachodzi warunek  $\gamma$ .

**Q3.** Czy w chwili  $t$  realizacji scenariusza wykonywana jest akcja  $A$ ?

- *always/ever performing  $A$  at  $t$  when  $Sc$*

Oznacza, że w chwili  $t$  realizacji scenariusza  $Sc$  zachodzi akcja  $A$ .

- *always/ever performing  $A$  when  $Sc$*

Oznacza, że istnieje pewna chwila w realizacji scenariusza  $Sc$ , w której zachodzi akcja  $A$ .

- *always/ever performing at  $t$  when  $Sc$*

Oznacza, że w chwili  $t$  realizacji scenariusza  $Sc$  zachodzi pewna akcja.

**Q4.** Czy podany cel  $\gamma$  jest osiągalny zawsze/kiedykolwiek przy zadanym zbiorze obserwacji  $OBS$ ?

- *always/ever accessible  $\gamma$  when  $Sc$*

Oznacza, że cel  $\gamma$  jest osiągalny zawsze/kiedykolwiek przy zadanym zbiorze obserwacji  $OBS$  przy realizacji scenariusza  $Sc$ .

**Uwaga 3.1.** Warunek *always* zachodzi jeśli odpowiedź na kwerendę we wszystkich ścieżkach wykonania jest *TRUE*, natomiast warunek *ever* zachodzi jeśli istnieje co najmniej jedna taka ścieżka.

## Semantyka kwerend w języku

Niech  $Sc$  będzie scenariuszem, a  $D$  opisem domeny języka. Powiemy, że kwerenda  $Q$  jest konsekwencją  $Sc$  zgodnie z  $D$  (ozn.  $Sc, D \models Q$ )

- zapytanie kwerendą  $Q$  postaci *executable  $Sc$*   
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  zgodnego z  $D$  scenariusz  $Sc$  zakończy się w sposób prawidłowy tj. wszystkie akcje zostaną zakończone, czyli aktualnie nie występuje akcja trwająca i żadna inna akcja nie zostanie wywołana.
- zapytanie kwerendą  $Q$  postaci  *$\gamma$  at  $t$  when  $Sc$*   
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  scenariusza  $Sc$  zgodnego z  $D$  znajdzie  $H(\gamma, t) = 1$
- zapytanie kwerendą  $Q$  postaci  *$\gamma$  when  $Sc$*   
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  scenariusza  $Sc$  zgodnego z  $D$  znajdzie  $\exists t \in N \ H(\gamma, t) = 1$ .
- zapytanie kwerendą  $Q$  postaci *performing  $A$  at  $t$  when  $Sc$*   
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  scenariusza  $Sc$  zgodnego z  $D$  znajdzie  $\exists d \in N \ (A, t, d) \in E$ .
- zapytanie kwerendą  $Q$  postaci *performing  $A$  when  $Sc$*   
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  scenariusza  $Sc$  zgodnego z  $D$  znajdzie  $\exists t \in N \ \exists d \in N \ (A, t, d) \in E$ .

- zapytanie kwerendą  $Q$  postaci *performing at t when Sc*  
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  scenariusza  $Sc$  zgodnego z  $D$  zajdzie  $\exists_{A \in Ac} \exists_{d \in N} (A, t, d) \in E$ .
- zapytanie kwerendą  $Q$  postaci *accessible  $\gamma$  when Sc*  
zwróci wynik *TRUE* jeśli dla każdego modelu  $S = (H, O, E)$  scenariusza  $Sc$  zgodnego z  $D$  zajdzie  $\exists(A_0, \dots, A_n \in ACS, n \geq 0) \exists_{t \in N} H(\gamma, t) = 1$ .

**Uwaga 3.2.** Jeśli warunek nie zajdzie program zwróci wartość *FALSE*.

## 4 Przykłady

### 4.1 Pytanie czy dany scenariusz może wystąpić

#### 4.1.1 Historia

Michał jest pracującym studentem. W środę o godzinie 8.00 powinien pojawić się w pracy zupełnie trzeźwy. Mimo to we wtorek postanowił pójść do baru. Jeśli Michał się napije, stanie się pijany. Jeśli pójdzie spać przestanie być pijany, ale stanie się skacowany, co również będzie niedopuszczalne w jego pracy.

#### 4.1.2 Opis akcji

$(drink, 2)$  **causes** *drunk*  
 $(sleep, 8)$  **causes**  $\neg drunk$   
 $(sleep, 8)$  **causes** *hangover* **if** *drunk*

#### 4.1.3 Scenariusze

$Sc = (OBS, ACS)$   
 $OBS = \{(\neg drunk, 0), (\neg hangover, 0), (\neg drunk, 10), (\neg hangover, 10)\}$   
 $ACS = \{((drink, 2), 0), ((sleep, 8), 2)\}$

$Sc_2 = (OBS_2, ACS_2)$   
 $OBS_2 = \{(\neg drunk, 0), (\neg hangover, 0), (\neg drunk, 10), (\neg hangover, 10)\}$   
 $ACS_2 = \{((sleep, 8), 1)\}$

#### 4.1.4 Kwerendy

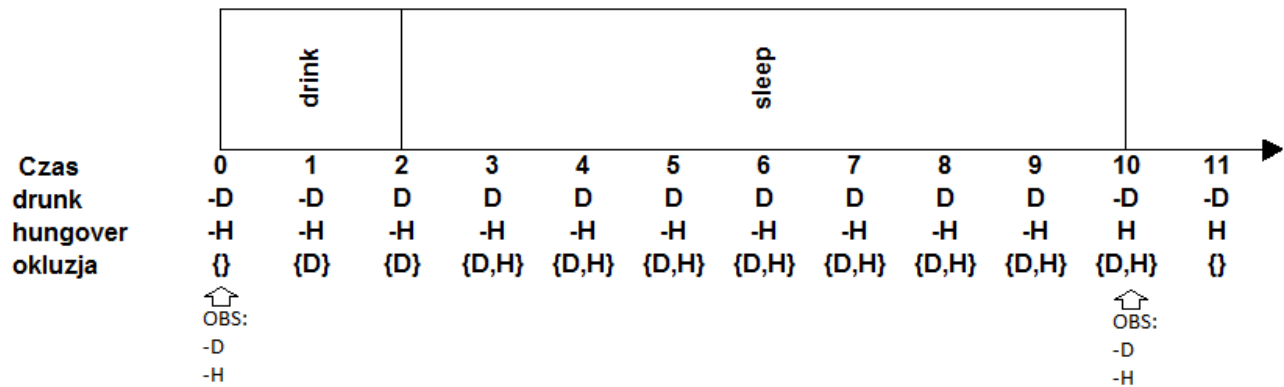
1. **ever executable**  $Sc$
2. **ever executable**  $Sc_2$



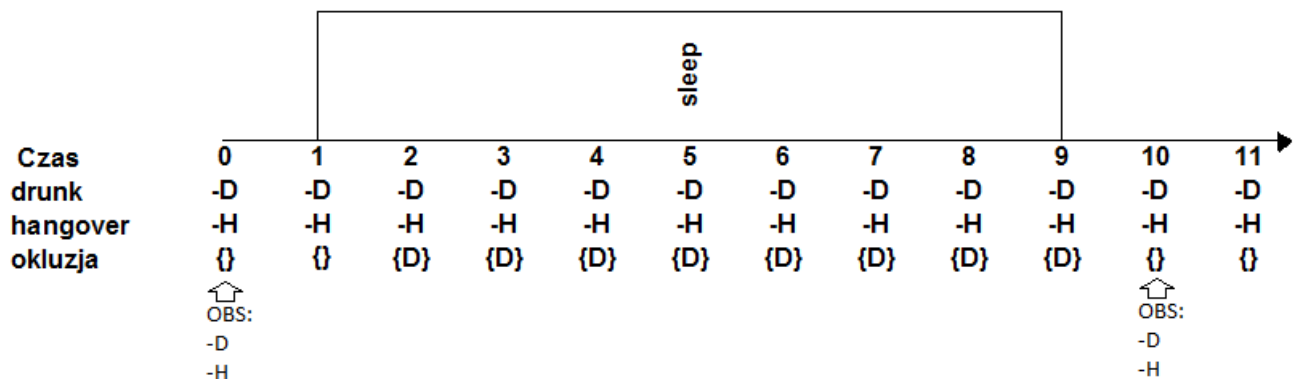
#### 4.1.5 Analiza

Odpowiedzi na kwerendy to odpowiednio:

1. FALSE zgodnie z diagramem 1.
2. TRUE zgodnie z diagramem 2.



Rysunek 1: Diagram dla scenariusza  $Sc$



Rysunek 2: Diagram dla scenariusza  $Sc_2$

Scenariusz  $Sc_2$  jest w pełni poprawny i wykonywalny. Scenariusza  $Sc$  nie można wykonać, ponieważ wymaga on by w chwili 10 fluenty *drunk* i *hangover* miały wartość FALSE, jednak w tej chwili zmienna *hangover* ma wartość TRUE.

## 4.2 Pytanie czy dany warunek zachodzi w danym czasie

### 4.2.1 Historia

Mick i Sarah są parą, więc mają wspólne produkty spożywcze, ale posiłki zwykle jadają oddzielnie. Pewnego dnia Sarah chce zrobić ciasto, a Mick naleśniki. Nie mogą być one robione w tym samym czasie ze względu na konieczność użycia miksera do przygotowania obu. Ponadto, zrobienie jednego lub drugiego dania zużywa cały zapas jajek dostępnych w mieszkaniu, więc trzeba je potem dokupić.

#### 4.2.2 Opis akcji

$(making\_panc, 1) \text{ causes } \neg eggs \text{ if } eggs$   
 $(making\_cake, 1) \text{ causes } \neg eggs \text{ if } eggs$   
 $(buy\_eggs, 2) \text{ causes } eggs$

#### 4.2.3 Scenariusz

$Sc = (OBS, ACS)$   
 $OBS = \{(eggs, 0)\}$   
 $ACS = \{((making\_panc, 1), 0), ((making\_cake, 1), 2)\}$

#### 4.2.4 Kwerendy

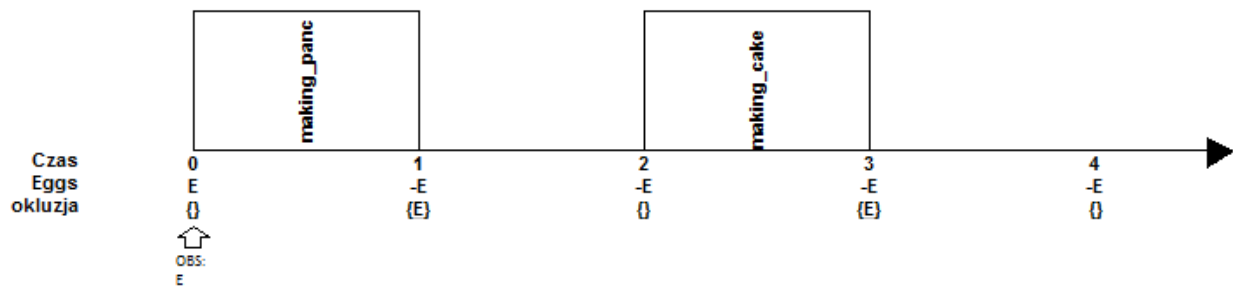
1. *eggs* at 0 **when**  $Sc$
2. *eggs* at 2 **when**  $Sc$

#### 4.2.5 Analiza

Odpowiedzi na kwerendy to odpowiednio:

1. TRUE,
2. FALSE.

Zgodnie z diagramem dla scenariusza  $Sc$ :



Rysunek 3: Diagram dla scenariusza  $Sc$

Oczywiście warunek akcji *making\_panc* nie jest spełniony w momencie 2.

### 4.3 Pytanie czy dana akcja jest wykonywana w pewnym czasie

#### 4.3.1 Historia

Mamy Billa i psa Maxa. Jeśli Bill idzie, to Max biegnie przez jakiś czas. Jeśli Bill gwizdże, Max szczeka przez jakiś czas. Jeśli Bill zatrzymuje się, Max również. Jeśli Bill przestaje gwizdać, to Max przestaje szczekać.

#### 4.3.2 Opis akcji

$(goes\_Bill, 2)$  **causes**  $run\_Max$   
 $(goes\_Bill, 2)$  **invokes**  $(runs\_Max, 2)$  **after** 0  
 $(runs\_Max, 2)$  **causes**  $\neg run\_Max$   
 $(whistles\_Bill, 1)$  **causes**  $bark\_Max$   
 $(whistles\_Bill, 1)$  **invokes**  $(barks\_Max, 1)$  **after** 0  
 $(barks\_Max, 1)$  **causes**  $\neg bark\_Max$

#### 4.3.3 Scenariusz

$Sc = (OBS, ACS)$   
 $OBS = \{(\neg run\_Max, 0), (\neg bark\_Max, 0), (run\_Max, 3), (\neg run\_Max, 5), (bark\_Max, 6), (\neg bark\_Max, 7), (run\_Max, 9), (\neg run\_Max, 11)\}$   
 $ACS = \{((goes\_Bill, 2), 1), ((whistles\_Bill, 1), 5), ((goes\_Bill, 2), 7)\}$

#### 4.3.4 Kwerendy

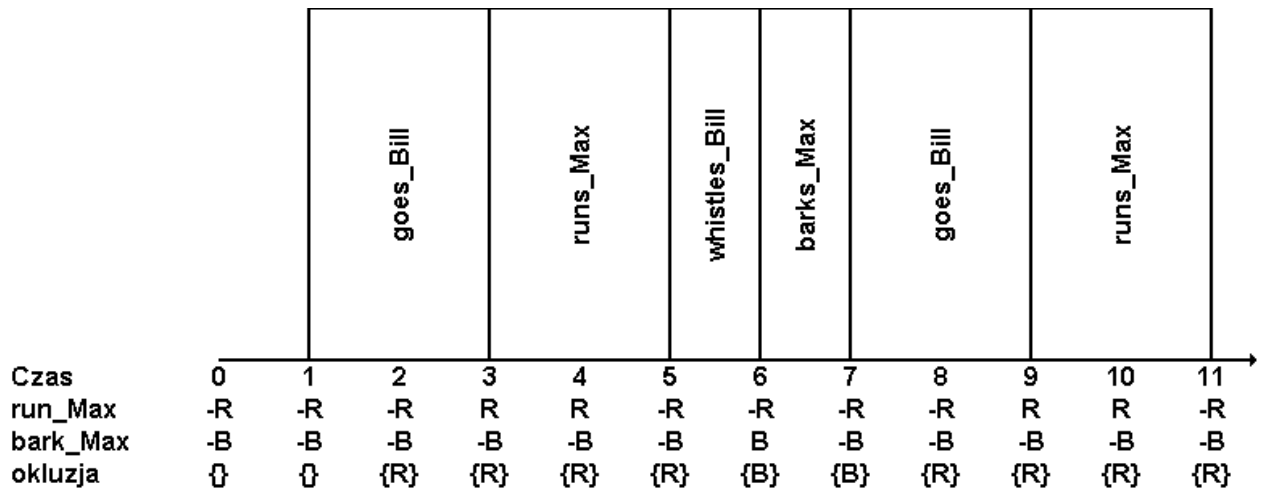
1. **performing**  $runs\_Max$  **at** 8 **when**  $Sc$
2. **performing**  $runs\_Max$  **when**  $Sc$
3. **performing** **at** 8 **when**  $Sc$

#### 4.3.5 Analiza

Odpowiedzi na powyższe kwerendy są następujące:

1. FALSE,
2. TRUE,
3. TRUE.

Ilustruje to poniższy diagram:



Rysunek 4: Diagram dla scenariusza  $Sc$

#### 4.4 Pytanie czy podany cel jest osiągalny

##### 4.4.1 Historia

Mamy Billa oraz komputer. Bill może nacisnąć przycisk *Włącz* lub odłączyć komputer od zasilania. Początkowo komputer jest wyłączony i podłączony do zasilania. Jeżeli zostanie naciśnięty jego przycisk *Włącz* oraz komputer jest podłączony do zasilania, to komputer włączy się. Odłączenie komputera od prądu powoduje, że komputer będzie odłączony od zasilania oraz wyłączony.

##### 4.4.2 Opis akcji

$(clicks\_button\_on, 1)$  **invokes**  $(switches\_on\_computer, 2)$  **after** 0 **if**  $connect\_power\_computer$   
 $(switches\_on\_computer, 2)$  **causes**  $on\_computer$   
 $(disconnects\_power, 1)$  **causes**  $\neg connect\_power\_computer$   
 $(disconnects\_power, 1)$  **causes**  $\neg on\_computer$

##### 4.4.3 Scenariusz

$Sc = (OBS, ACS)$   
 $OBS = \{(\neg on\_computer, 0), (connect\_power\_computer, 0), (on\_computer, 4), (\neg on\_computer, 5), (\neg connect\_power\_computer, 5)\}$   
 $ACS = \{(clicks\_button\_on, 1), 1), ((disconnects\_power, 1), 4), ((clicks\_button\_on, 1), 5)\}$

##### 4.4.4 Kwerendy

1.  $on\_computer$  **at** 7 **when**  $Sc$
2. **accessible**  $on\_computer$  **when**  $Sc$

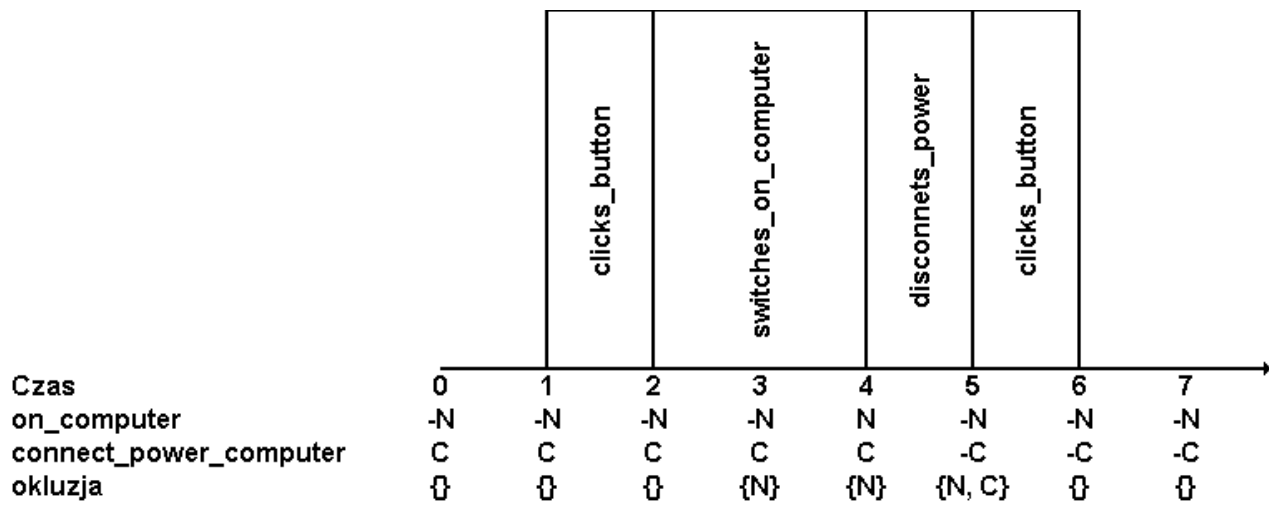
#### 4.4.5 Analiza

Odpowiedzi na powyższe kwerendy są następujące:

1. FALSE,
2. TRUE.

Należy dodać, że cel osiągnięto już w chwili czasu równej 4, mimo iż komputer jest wyłączony od chwili czasu równej 5 i stan ten nie ulega już zmianie.

Ilustruje to poniższy diagram:



Rysunek 5: Diagram dla scenariusza *Sc*