Reprezentacja wiedzy - projekt

Anna Bekas

Anna Kosiorek

Mateusz Bąkała

Bartosz Woźniak

Marcin Chudy

Paweł Wiśniewski

Jakub Suchan

Jakub Fijałkowski

1 października 2017

1. Definicja

1.1. 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, który ma zostać opracowany w ramach tego projektu, jest opisany w rozdziale 1.2, Język akcji. Język kwerend został opisany w rozdziale 1.5 Język kwerend.

W rozdziale 2 Przykłady znajdują się przykłady wykorzystania zaprojektowanego języka akcji/kwerend. Obrazują one konkretne przypadki zastosowania języka, sposób działania kwerend oraz zawierają wyniki działania programu (odpowiedzi na zadane kwerendy).

1.2. Język akcji

Język akcji, będący przedmiotem tego zadania projektowego, posiada następujące założenia:

1. Spełnia prawo inercji,
2. Zachowuje sekwencyjność działań,
3. Dopuszcza akcje niedeterministyczne,
4. Posiada liniowy model czasu z czasem dyskretnym,
5. Posiada pełną informację o wszystkich:
   (a) akcjach,
   (b) skutkach bezpośrednich.
6. Akcja języka posiada:

   (a) warunek początkowy (lub `true`, jeśli można ją wykonać w każdych warunkach),
   (b) czas trwania $d \geqslant 1$, $d \in \mathbb{N}$,
   (c) efekt akcji.

7. Podczas trwania akcji, wartości zmiennych, na które ona wpływa, nie są znane,
8. Występują efekty:
   (a) środowiskowe: zmieniają wartość zmiennych systemu,
   (b) dynamiczne: wystąpienie akcji może wywołać wystąpienie innych akcji po $d \geqslant 0$ 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, w których akcja może być wykonana.
10. Pewne stany mogą rozpocząć wykonywanie pewnych akcji.

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

1.2.1. Sygnatura języka

Definicja 1.1. Sygnaturą języka nazywamy parę $\psi = (\mathcal{F}, \mathcal{A})$, gdzie:

$\mathcal{F}$ – zbiór zmiennych (fluentów)

$\mathcal{A}$ – zbiór akcji

1.2.2. Oznaczenia

W dziedzinie definiowanego języka będą używane następujące oznaczenia:

$f, g \in \mathcal{F}$ – fluenty

$A, B \in \mathcal{A}$ – akcje

d_A, d_B – czasy trwania akcji odpowiednio A oraz B

$\alpha, \pi \in Forms(\mathcal{F})$ – warunki

Dziedzinę języka będziemy oznaczali symbolem D.

Formuły (warunki) złożone z fluentów definiujemy następująco:

$Forms(\mathcal{F}) \ni \alpha := f|\neg f|\alpha \wedge \pi|\alpha \vee \pi$

przy czym $\neg$ oznacza logiczną negację, $\wedge$ logiczną koniunkcje a $\vee$ logiczną alternatywę. Dodatkowo wprowadza się dwa dodatkowe symbole specjalne: $true$ oznaczający prawdę, oraz $false$ oznaczający fałsz.

1.3. Syntaktyka

1.3.1. Rodzaje zdań

A **lasts** d_A **causes** α **if** π

Akcja A po d_A jednostkach czasu powoduje spełnienie warunku α, pod warunkiem, że zaszedł warunek π.

A **lasts** d_A **releases** f **if** π

Akcja A po d_A jednostkach czasu, powoduje uwolnienie fluentu f, jeśli zachodzi warunek π.

π **triggers** A

Spełnienie warunku π wyzwala wykonanie akcji A (tj. jeśli zajdzie stan, w którym π jest prawdziwe, to akcja A może się rozpocząć).

A **invokes** B **after** d **if** π

Akcja A powoduje wykonanie akcji B po d jednostkach czasu od zakończenia akcji A, jeśli zaszedł warunek π w momencie rozpoczęcia akcji A.

A **impossible at** t

Akcja A nie może rozpocząć się w chwili czasu t.

Przy zdaniach, w których występuje warunek `if` π, dopuszcza się pominięcie warunku. W takim przypadku zakłada się, że $\pi = true$.

1.4. Semantyka

1.4.1. Semantyczna struktura interpretacyjna języka

Definicja 1.2. Semantyczną strukturą interpretacyjną języka AL nazywamy czwórkę $S = (H, O, E, T)$, taką że:

- $H : \mathcal{F} \times \mathbb{N} \rightarrow \{0, 1\}$ jest funkcją historii. Pozwala ona sprawdzić wartość danego fluentu w danej chwili czasu. Funkcję historii można naturalnie rozszerzyć definiując funkcję $H^* : Forms\,(\mathcal{F}) \times \mathbb{N} \rightarrow \{0, 1\}$ jako funkcję, która danej formule logicznej przyporządkowuje jej wartość (prawdę lub fałsz) w chwili czasu t na podstawie wartości jakie w chwili t przyjmują fluenty wchodzące w skład tej formuły.

- $O : \mathcal{A} \times \mathbb{N} \rightarrow 2^{\mathcal{F}}$ jest funkcją okluzji. Dla pewnej ustalonej akcji $A \in \mathcal{A}$ oraz chwili czasu $t \in \mathbb{N}$, funkcja $O(A, t)$ zwraca zbiór fluentów, na które akcja A, trwająca w chwili t, ma wpływ i dla których wartość w chwili t jest wciąż nieokreślona.

- $E \subseteq \mathcal{A} \times \mathbb{N}$ jest relacją wystąpień akcji. Para (A, t) należy do relacji E jeśli akcja A została rozpoczęta w czasie t.

- $T \in \mathbb{N}$ jest czasem zakończenia wszystkich akcji. Może to być dowolnie duża, skończona liczba naturalna. Mówi ona do kiedy powinny być zakończone wszystkie akcje.

1.4.2. Czas trwania akcji

Czas trwania akcji A, która rozpoczyna się w chwili t jest wyznaczany za pomocą następującego algorytmu:

```
dmax_A ← 0
for zdanie (a lasts d_A causes α if π) ∈ D do
    if H* (π, t) = 1 and d_A > dmax_A then
        dmax_A ← d_A
    end if
end for
for zdanie (a lasts d_A releases f if π) ∈ D do
    if H* (π, t) = 1 and d_A > dmax_A then
        dmax_A ← d_A
    end if
end for
return dmax_A
```

Intuicyjnie czas trwania akcji A rozpoczynającej się w danej chwili t definiujemy jako maksymalny z czasów d_A występujących we wszystkich zdaniach typu *causes* i *releases* z dziedziny D, których warunki rozpoczęcia akcji A są spełnione w chwili t.

Od tego momentu, stwierdzenia „czas trwania akcji" odnoszą sę do czasu wyznaczonego przez ten algorytm, chyba że z kontekstu wynika inaczej.

1.4.3. Pozostałe założenia semantyczne języka

Niech A, B będą akcjami, f – fluentem, α, π – formułami, d_A, d_B – liczbami naturalnymi (oznaczającymi czas trwania odpowiadających im akcji) oraz $fl(\alpha)$ będzie zbiorem fluentów występujących w α. Wtedy dla zdań języka muszą być spełnione następujące warunki:

- Dla każdego zdania (A lasts d_A causes α if π) $\in D$ i dla każdej chwili $t \in \mathbb{N}$, jeśli $H^*(\pi, t) = 1$ oraz $(A, t) \in E$, wtedy $H^*(\alpha, t + d_A) = 1$ oraz dla każdego $p \in \mathbb{N}$, takiego że $1 \leqslant p \leqslant d_A$ zachodzi $fl(\alpha) \subseteq O(A, t + p)$.

- Dla każdego zdania (A lasts d_A releases f if π) $\in D$ i dla każdej chwili $t \in \mathbb{N}$, jeżeli $H^*(\pi, t) = 1$ oraz $(A, t) \in E$, to dla każdego $p \in \mathbb{N}$, takiego że $1 \leqslant p \leqslant d_A$ mamy $f \in O(A, t + p)$.

- Dla każdego zdania (π triggers A) $\in D$ i dla każdej chwili $t \in \mathbb{N}$, jeśli $t + d_A \leqslant T$ oraz $H^*(\pi, t) = 1$, to $(A, t) \in E$, gdzie d_A jest czasem wykonywania akcji A przy danym stanie systemu.

- Dla każdego zdania (A invokes B after d if π) $\in D$ i dla każdej chwili $t \in \mathbb{N}$, jeśli $t + d_A + d + d_B \leqslant T$ oraz $H^*(\pi, t) = 1$, $(A, t) \in E$, to $(B, t + d + d_A) \in E$, gdzie d_A oraz d_B oznacza odpowiednio czas wykonywania akcji A i B przy danym stanie systemu.

- Dla każdego zdania (A impossible at t) para $(A, t) \notin E$.

1.4.4. Scenariusze działań

Definicja 1.3. Scenariuszem nazywamy trójkę $Sc = (OBS, ACS, T')$, taką że:

$OBS = \{(\gamma_1, t_1), (\gamma_2, t_2), \ldots, (\gamma_n, t_n)\}, n \geqslant 0, \gamma_i \in Forms(\mathcal{F}), t_i \in \mathbb{N}$

Jest to zbiór par zawierających **obserwacje**, czyli warunek γ_i, który jest spełniony w chwili t_i.

$ACS = \{(A_1, t_1), (A_2, t_2), \ldots, (A_m, t_m)\}, m \geqslant 0, A_i \in \mathcal{A}, t_i \in \mathbb{N}$

Jest to zbiór par zawierający **zaobserwowane akcje**, czyli obserwacje wystąpienia akcji A_i w chwili t_i.

T' Jest czasem trwania scenariusza. Wystąpienia akcji, które nie zakończyły się przed czasem T', nie będą rozważane.

Definicja 1.4. Niech $S = (H, O, E, T)$ będzie semantyczną strukturą interpretacyjną definiowanego języka, $Sc = (OBS, ACS, T')$ będzie scenariuszem, a D dziedziną. Powiemy, że S **jest strukturą interpretacyjną dla** Sc **zgodną z opisem dziedziny** D, jeśli:

- $T = T'$

- Dla każdej obserwacji $(\alpha, t) \in OBS$ zachodzi $H^*(\alpha, t) = 1$

- $ACS \subseteq E$

- Dla każdej akcji $A \in \mathcal{A}$ trwającej d_A oraz dla każdej chwili $t \in \mathbb{N}$, jeśli $(A, t) \in E$, to $t + d_A \leqslant T$

- Dla każdych różnych par (A, t_1), $(B, t_2) \in E$ spełniony jest warunek:
  $t_2 \geqslant t_1 + d_A \vee t_1 \geqslant t_2 + d_B$, gdzie d_A, d_B są odpowiednio czasami trwania akcji A i B.

Definicja 1.5. Niech $O_1,\ O_2 : X \to 2^Y$. Mówimy, że $O_1 \prec O_2$ jeżeli $\forall x \in X\ O_1(x) \subsetneq O_2(x)$.

Definicja 1.6. Niech $S = (H, O, E, T)$ będzie strukturą interpretacyjną dla scenariusza $Sc = (OBS, ACS, T')$ zgodną z opisem dziedziny D. Mówimy, że S jest O-minimalną strukturą interpretacyjną, jeżeli nie istnieje struktura $S' = (H, O', E', T)$ dla scenariusza Sc i dziedziny D taka, że $O' \prec O$.

Definicja 1.7. Niech $S = (H, O, E, T)$ będzie strukturą interpretacyjną dla scenariusza $Sc = (OBS, ACS, T')$ zgodną z opisem dziedziny D. S będziemy nazywać modelem Sc zgodnym z opisem D jeżeli:

- S jest O-minimalny.
- Dla każdej chwili $t \in \mathbb{N}$ zbiór $\{f \in \mathcal{F}\colon H(f,t) \neq H(f,t+1)\} \subseteq O(A, t+1)$ dla pewnej akcji A.
- Nie istnieje żadna struktura interpretacyjna $S' = (H, O', E', T)$ dla Sc zgodna z opisem dziedziny D, która spełnia poprzednie warunki oraz taka, że $E' \subset E$.

Definicja 1.8. Mówimy, że scenariusz Sc jest **spójny** z opisem dziedziny D, jeśli istnieje do niego model S zgodny z opisem D.

Uwaga 1.9. Dla pewnych scenariuszy model może nie istnieć.

1.5. Język kwerend

Zdefiniowany język akcji może być odpytywany przez poniżej zaprezentowany język kwerend, który zapewnia uzyskanie odpowiedzi prawda($true$)/fałsz($false$) na następujące pytania:

always/ever executable Sc

Czy podany scenariusz Sc jest możliwy do realizacji zawsze/kiedykolwiek?
Odpowiedź $true$ na to pytanie oznacza istnienie zawsze/kiedykolwiek modelu S dla scenariusza Sc do chwili T włącznie, gdzie T jest czasem trwania scenariusza.

always/ever holds γ **at** t **when** Sc

Czy w chwili $t \in \mathbb{N}$ realizacji scenariusza Sc warunek γ zachodzi zawsze/kiedykolwiek?

A **always/ever occurs at** t **when** Sc

Czy w chwili $t \in \mathbb{N}$ realizacji scenariusza Sc jest wykonywana akcja A?

Uwaga 1.10. Warunki **always** zachodzą, jeśli odpowiedzią na kwerendę we wszystkich ścieżkach wykonania (modelach) jest $true$, natomiast warunek **ever** zachodzi, jeśli istnieje co najmniej jedna taka ścieżka.

Uwaga 1.11. Scenariusze są analizowane tylko do końca czasu trwania scenariusza. Jeśli do tego czasu warunek jest spełniony, to odpowiedzią jest $true$, w przeciwnym razie odpowiedzią jest $false$.

1.5.1. Semantyka kwerend

Definicja 1.12. Niech Sc będzie scenariuszem, a D opisem dziedziny języka. Mówimy, że kwerenda Q jest **konsekwencją** Sc zgodnie z D (ozn. $Sc, D \mathrel{|}\!\approx Q$), jeśli Q jest postaci

- **always executable** Sc, to kwerenda zwróci $true$, jeśli **każda struktura interpretacyjna** $S = (H, O, E, T)$ dla Sc zgodna z D jest modelem Sc zgodnym z D.
- **ever executable** Sc, to kwerenda zwróci $true$, jeśli **pewna struktura interpretacyjna** $S = (H, O, E, T)$ dla Sc zgodna z D jest modelem Sc zgodnym z D.
- **always holds** γ **at** t **when** Sc, to kwerenda zwróci $true$, jeśli dla **każdego modelu** $S = (H, O, E, T)$ scenariusza Sc zgodnego z D zachodzi $H(\gamma, t) = 1$.
- **ever holds** γ **at** t **when** Sc, to kwerenda zwróci $true$, jeśli dla **pewnego modelu** $S = (H, O, E, T)$ scenariusza Sc zgodnego z D zachodzi $H(\gamma, t) = 1$.
- A **always occurs at** t **when** Sc, to kwerenda zwróci $true$, jeśli dla **każdego modelu** $S = (H, O, E, T)$ scenariusza Sc zgodnego z D zachodzi $(A, t) \in E$.
- A **ever occurs at** t **when** Sc, to kwerenda zwróci $true$, jeśli dla **pewnego modelu** $S = (H, O, E, T)$ scenariusza Sc zgodnego z D zachodzi $(A, t) \in E$.

Uwaga 1.13. Jeśli warunek nie zajdzie, kwerenda zwraca wartość $false$.

2. Przykłady

2.1. Pytanie czy scenariusz jest możliwy do realizacji

2.1.1. Historia

Mateusz jest studentem MSI. Musi wykonać na dzisiejsze zajęcia z RW projekt. Jednak dostał zaproszenie od koleżanki na ślizgawkę politechniki, która zaczyna się o północy. Wie również, że na zajęcia musi przyjść z gotowym projektem i być wypoczęty. Jeżeli pójdzie na łyżwy, to zmęczy się szybką jazdą. Ale jeżeli będzie spał 8 godzin, to zmęczenie ustąpi. Ponadto koledzy ze starszych lat powiedzieli mu, że jeżeli jest się wyspanym, to projekt można zrobić w 5 godzin. Jeśli jest się zmęczonym, napisanie projektu zajmuje 10 godzin.

2.1.2. Opis akcji

work **lasts** *5* **causes** *project* **if** $\neg$*tired*

work **lasts** *10* **causes** *project* **if** *tired*

sleep **lasts** *8* **causes** $\neg$*tired*

skates **lasts** *1* **causes** *tired*

2.1.3. Scenariusze

$Sc_1 = (OBS_1, ACS_1, 16)$

$OBS_1 = \{(\neg project,\ 0),\ \ (\neg tired \wedge project,\ 16)\}$

$ACS_1 = \{(skates,\ 0),\ \ (work,\ 2),\ (sleep,\ 7)\}$

$Sc_2 = (OBS_2,\ ACS_2, 16)$

$OBS_2 = \{(\neg project,\ 0),\ (\neg tired \wedge project,\ 16)\}$

$ACS_2 = \{(work,\ 0),\ \ (sleep,\ 6))\}$

2.1.4. Kwerendy

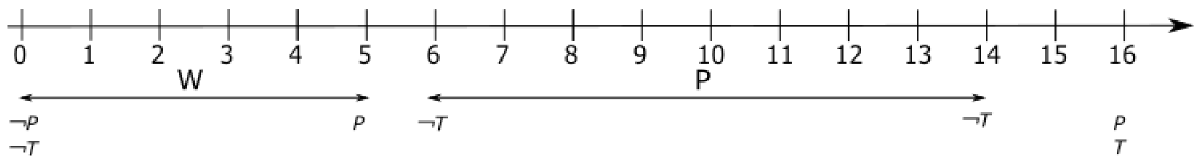
1. **ever executable** *Sc1*

2. **ever executable** *Sc2*

2.1.5. Analiza

Scenariusz Sc_1 zakłada, że Mateusz poszedł na łyżwy, a zatem był zmęczony rozpoczynając pracę. Zmęczenie powoduje, że akcja *work* trwa aż 10 godzin. W chwili 7 miała się zacząć akcja *sleep*, jednak w tym czasie nadal trwa akcja *work*, zatem akcja *sleep* nie może się zacząć. To zaś oznacza, że dla powyższego scenariusza nie istnieje model (co więcej – nie istnieje struktura interpretacyjna zgodna z opisem scenariusza). Zatem kwerenda *ever executable* Sc_1 zwróci fałsz.

Scenariusz Sc_2 zakłada, że Mateusz rozpoczyna pracę nad projektem w chwili 0. Jeśli na początku był zmęczony, to praca zajmuje mu 10 godzin, co jest sprzeczne z tym, że w chwili 6 miała rozpocząć się akcja *sleep*. Załóżmy więc, że Mateusz na początku nie był zmęczony. Pracę skończył w ciągu pięciu godzin. Nic nie stoi w tym przypadku na przeszkodzie, aby Mateusz poszedł spać w chwili 6 i spał przez 8 godzin. W chwili 16 Mateusz jest wypoczęty i ma zrobiony projekt, więc obserwacja końcowa zachodzi. Oczywiste jest, że w opisanej sytuacji dla scenariusza



Rysunek 2.1: Model M_1

Sc_2 istnieje model zgodny z opisem dziedziny D. Zatem kwerenda *ever executable* Sc_2 zwróci prawdę.

2.2. Pytanie czy dany warunek zachodzi w danym czasie

2.2.1. Historia

Ania ma zamiar ugotować niedzielny obiad. Ugotowanie pysznego obiadu zajmuje 2 godziny. Oczywiście obiad gotowany za długo przypala się.

W trakcie dnia do Ani może zadzwonić jej czarujący absztyfikant, co spowoduje, że Ania się w nim zakocha i będzie myślała o nim cały dzień. Rozmowa telefoniczna trwa godzinę. Jeżeli Ania jest zakochana to zapomni o gotującym się obiedzie i przypomni sobie o nim godzinę za późno. Dodatkowo, wiadomo, że Ania rozpoczęła gotowanie po pierwszej godzinie.

2.2.2. Opis akcji

rozmowa lasts 1 causes ZAKOCHANA if ¬ ZAKOCHANA

gotowanie lasts 2 causes OBIAD

gotowanie lasts 3 causes OBIAD ∧ SPALONY if ZAKOCHANA

2.2.3. Scenariusz

$Sc_1 = (OBS_1, ACS_1, 4)$

$OBS_1 = \{(\neg OBIAD \land \neg SPALONY, 0), (OBIAD, 4)\}$

$ACS_1 = \{(gotowanie, 1)\}$

2.2.4. Kwerendy

Czy zawsze po 4 godzinach Ania ugotowała nieprzypalony obiad i jest zakochana?

always holds $\neg SPALONY \land ZAKOCHANA$ *at 4 when* Sc_1

Czy kiedykolwiek po 4 godzinach Ania ugotowała nieprzypalony obiad i jest zakochana?

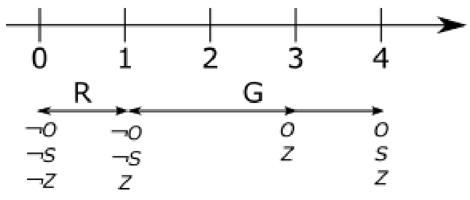
ever holds $\neg SPALONY \land ZAKOCHANA$ *at 4 when* Sc_1

2.2.5. Analiza

Na poniższych rysunkach będziemy stosować skrócone oznaczenia fluentów ($O - OBIAD$, $S - SPALONY$, $Z - ZAKOCHANA$) oraz akcji ($R - rozmowa$, $G - gotowanie$).

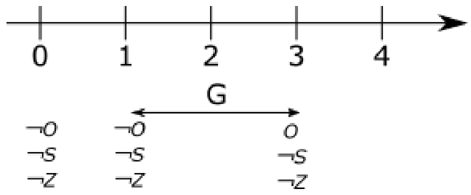
Poszukujemy takich modeli w których Ania po 4 godzinach ugotowała obiad, a na początku obiad nie był ugotowany. Wobec tego musiała się wykonać akcja *gotowanie* i wiemy, że rozpoczęła się w chwili 1. Możliwe są następujące sytuacje:

1. Ania nie była zakochana na początku. W chwili 0 zadzwonił telefon, a potem Ania ugotowała obiad, który się przypalił.



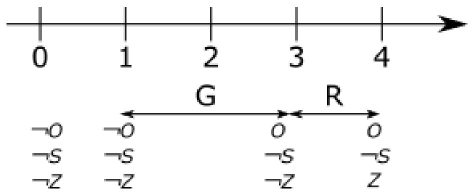
Rysunek 2.2: Sytuacja 1

2. Ania nie była zakochana na początku. W chwili 1 ugotowała nieprzypalony obiad, co trwało 2 godziny.



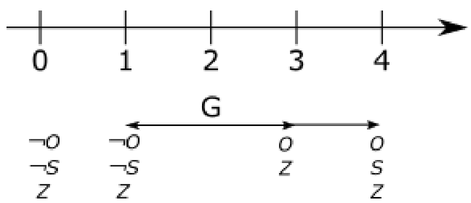
Rysunek 2.3: Sytuacja 2

3. Ania nie była zakochana na początku. W chwili 1 ugotowała nieprzypalony obiad, co trwało 2 godziny. W chwili 3 zadzwonił telefon i Ania się zakochała.



Rysunek 2.4: Sytuacja 3

4. Ania była zakochana na początku. Wówczas rozpoczynając gotowanie w chwili 1, ugotowała przypalony obiad.



Rysunek 2.5: Sytuacja 4

Każda z powyższych czterech sytuacji jest zgodna ze scenariuszem. Wobec tego można dla nich wyznaczyć modele odpowiednio M_1, M_2 M_3 M_4.

1. Dla sytuacji pierwszej model możemy zdefiniować następująco:
   $M_1 = (H_1, O_1, E_1, T_1)$, gdzie kolejno:

Funkcja historii H_1:

H	0	1	2	3	4
O	0	0	?	1	1
S	0	0	?	?	1
Z	0	1	1	1	1

Funkcja okluzji O_1:

O	0	1	2	3	4
R	$\emptyset$	$\{Z\}$	$\emptyset$	$\emptyset$	$\emptyset$
G	$\emptyset$	$\emptyset$	$\{O, S\}$	$\{O, S\}$	$\{O, S\}$

$E_1 = \{(R, 0), (G, 1)\}$,

$T_1 = 4$

2. Dla sytuacji drugiej model możemy zdefiniować następująco:

   $M_2 = (H_2, O_2, E_2, T_2)$, gdzie kolejno:

Funkcja historii H_2:

H	0	1	2	3	4
O	0	0	?	1	1
S	0	0	0	0	0
Z	0	0	0	0	0

Funkcja okluzji O_2:

O	0	1	2	3	4
R	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
G	$\emptyset$	$\emptyset$	$\{O\}$	$\{O\}$	$\emptyset$

$E_2 = \{(G, 1)\}$,

$T_2 = 4$

3. Dla sytuacji trzeciej model możemy zdefiniować następująco:

   $M_3 = (H_3, O_3, E_3, T_3)$, gdzie kolejno:

Funkcja historii H_3:

H	0	1	2	3	4
O	0	0	?	1	1
S	0	0	0	0	0
Z	0	0	0	0	1

Funkcja okluzji O_3:

O	0	1	2	3	4
R	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{Z\}$
G	$\emptyset$	$\emptyset$	$\{O\}$	$\{O\}$	$\emptyset$

$E_3 = \{(R, 3), (G, 1)\}$,

$T_3 = 4$

4. Dla sytuacji czwartej model możemy zdefiniować następująco:

   $M_4 = (H_4, O_4, E_4, T_4)$, gdzie kolejno:

Funkcja historii H_4:

H	0	1	2	3	4
O	0	0	?	1	1
S	0	0	?	?	1
Z	1	1	1	1	1

Funkcja okluzji O_4:

O	0	1	2	3	4
R	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
G	$\emptyset$	$\emptyset$	$\{O, S\}$	$\{O, S\}$	$\{O, S\}$

$E_4 = \{(G, 1)\}$,

$T_4 = 4$

Uwaga: Znaki *?* w funkcji historii oznaczają, że w tym miejscu funkcja historii może przyjąć dowolną wartość 0 lub 1. Wypełniając te wartości w różny sposób dostaniemy różne modele.

Kwerenda *always holds* $\neg SPALONY \wedge ZAKOCHANA$ *at 4 when* Sc_1 zwróci fałsz, ponieważ nie jest prawdą że w każdym modelu postulowany warunek jest spełniony (nie jest on spełniony w modelach M_1, M_2, M_4).

Kwerenda *ever holds* $\neg SPALONY \wedge ZAKOCHANA$ *at 4 when* Sc_1 zwróci prawdę, bo istnieje model dla scenariusza Sc_1, w którym warunek zawarty w kwerendzie jest prawdziwy (jest to model M_3).