Reprezentacja wiedzy

Scenariusze działań z akcjami złożonymi

Projekt nr 1

Piotr Bródka
Aleksandra Bułka
Aleksandra Byczyńska
Jakub Czyżniejewski
Taras Palayda
Michał Piesio
Stanisław Wasilkowski
Kazimierz Wojciechowski

Dana jest klasa systemów dynamicznych spełniających następujące warunki:

$\boldsymbol{A_1}$ Prawo inercji.

$\boldsymbol{A_2}$ Liniowy model czasu (czas dyskretny).

$\boldsymbol{A_3}$ Niedeterminizm działań.

$\boldsymbol{A_4}$ Pełna informacja o wszystkich akcjach i wszystkich ich skutkach bezpośrednich.

$\boldsymbol{A_5}$ Akcje wykonywane są w 1 jednostce czasu.

$\boldsymbol{A_6}$ Z każdą akcją związany jest jej warunek wstępny (ew. $true$) i końcowy.

$\boldsymbol{A_7}$ W pewnych stanach akcje mogą być niewykonalne, przy czym te stany podane są albo przez podanie konkretnych chwil rozpoczęcia, albo przez określenie warunków logicznych.

$\boldsymbol{A_8}$ Nie występują warunki integralności.

Scenariuszem działań nazywamy parę $Sc = (OBS, ACS)$, gdzie OBS to zbiór obserwacji (rozumiany w sensie języka $\mathcal{AL}$), zaś ACS jest zbiorem par $\{(A_1, t_1), \ldots, (A_n, t_n)\}$ gdzie $A_i \subseteq \mathcal{A}c$ jest akcją złożoną), $t_i \in \mathbb{N}$ jest momentem jej rozpoczęcia, $i = 1, \ldots, n$. Jeśli $A_i = \emptyset$, to jest akcją pustą (nie powodującą żadnych zmian). Przyjmujemy, że:

- Wykonywane są jedynie akcje o zmiennych niezależnych.
- Efekt akcji złożonej $\{a_1, \ldots, a_k\}$ jest sumą wyników akcji składowych a_i, $i = 1, \ldots, k$.
- Jeśli w akcji złożonej $A \subseteq \mathcal{A}c$ występuje konflikt, wykonywane są (niedeterministycznie) maksymalne niekonfiktowe akcje $A' \subset A$.

Zadanie:

Opracować i zaimplementować język akcji dla specyfikacji podanej klasy systemów dynamicznych oraz odpowiadający mu język kwerend zapewniający uzyskanie odpowiedzi na następujące pytania:

- $\boldsymbol{Q_1}$ Czy podany scenariusz jest możliwy do realizacji zawsze/kiedykolwiek?
- $\boldsymbol{Q_2}$ Czy w chwili $t \geq 0$ realizacji danego scaneriusza warunek γ zachodzi zawsze/kiedykolwiek?
- $\boldsymbol{Q_3}$ Czy w trakcie realizacji scenariusza Sc akcja A jest wykonywalna w chwili $t \in \mathbb{N}$ zawsze/kiedykolwiek?

Spis treści

1 Język $\mathcal{ALC}$ - podstawowe założenia **4**

2 Syntaktyka **4**
2.1 Sygnatura 4
2.2 Literał 4
2.3 Akcja złożona 4
2.4 Formuła 4
2.5 Dziedzina akcji 5
2.6 Definicja scenariusza 5

3 Semantyka **6**
3.1 Definicje akcji 6
3.1.1 Akcje atomowe potencjalnie wykonywalne 6
3.1.2 Akcje atomowe konfliktowe 6
3.1.3 Akcje złożone całkowicie wykonywalne 7
3.1.4 Dekompozycja akcji przy użyciu funkcji MNS 7
3.2 Struktura 8
3.2.1 Rozszerzenie funkcji historii H na H^* 9
3.2.2 Struktura dla scenariusza Sc 9
3.3 Model 9
3.3.1 Definicja modelu 9

4 Kwerendy języka $\mathcal{ALC}$ i ich interpretacja **10**

5 Przykłady **12**
5.1 Warsaw standoff 12
5.2 Promocja w supermarkecie 15
5.3 Ewakuacja budynku 17
5.4 Myślał indyk o niedzieli 20

1 Język $\mathcal{ALC}$ - podstawowe założenia

- Prawo inercji.
- Liniowy model czasu (czas dyskretny).
- Niedeterminizm działań.
- Pełna informacja o wszystkich akcjach i wszystkich ich skutkach bezpośrednich.
- Akcje wykonywane są w 1 jednostce czasu.
- Z każdą akcją związany jest jej warunek wstępny (ew. *true*) i końcowy.
- W pewnych stanach akcje mogą być niewykonalne, przy czym stany takie podane są albo przez podanie konkretnych chwil rozpoczęcia, albo przez określenie warunków logicznych.
- Nie występują warunki integralności.

2 Syntaktyka

2.1 Sygnatura

Niech $\mathcal{F}$ będzie zbiorem fluentów oraz niech $\mathcal{A}c$ będzie zbiorem wszystkich akcji atomowych (atomów). Zbiory $\mathcal{F}$ oraz $\mathcal{A}c$ są rozłączne. Wówczas parę $\boxed{\Upsilon = (\mathcal{F}, \mathcal{A}c)}$ nazywamy sygnaturą.

2.2 Literał

Fluent $f \in \mathcal{F}$ lub jego zaprzeczenie $\neg f$ nazywamy literałem $\bar{f}$.

2.3 Akcja złożona

Każdy niepusty podzbiór zbioru akcji atomowych $\mathcal{A}c$ nazywamy akcją złożoną (czyli $\boxed{A \subseteq \mathcal{A}c, A \neq \emptyset}$).

Przyjęta notacja

Akcje atomowe oznaczamy małymi literami $a, b, c, \ldots$ natomiast akcje złożone oznaczamy wielkimi literami $A, B, C, \ldots$.

2.4 Formuła

Formułę będziemy oznaczać literami greckimi $\alpha, \beta, \gamma, \ldots$. Zbiór wszystkich formuł będziemy oznaczać $\mathbb{F}$.

- Każdy fluent jest formułą: $f \in \mathbb{F}$.
- Każde zaprzeczenie formuły jest formułą: $(\neg\alpha) \in \mathbb{F}$.
- Koniunkcja dwóch formuł jest formułą: $(\alpha \wedge \beta) \in \mathbb{F}$.
- Alternatywa dwóch formuł jest formułą: $(\alpha \vee \beta) \in \mathbb{F}$.
- Implikacja, której poprzednik i następnik są formułami jest formułą: $(\alpha \implies \beta) \in \mathbb{F}$.

- Równoważność dwóch formuł jest formułą: $(\alpha \iff \beta) \in \mathbb{F}$.

Zbiorem fluentów występujących w danej formule α nazywamy $Fluents(\alpha)$, przykładowo:

$$Fluents\left((f \land \neg g) \implies (h \lor i)\right) = \{f, g, h, i\}$$

Formułą nad pewnym niepustym zbiorem fluentów $\mathcal{F}$ nazywamy każdą formułę $\alpha \in \mathbb{F}$ dla której zbiór fluentów w niej występujących jest podzbiorem $\mathcal{F}$, czyli zachodzi inkluzja $Fluents(\alpha) \subseteq \mathcal{F}$.
Zbiorem wszystkich formuł nad pewnym niepustym zbiorem fluentów $\mathcal{F}$ oznaczamy $Forms(\mathcal{F})$. Wówczas

$$\forall_{\alpha, \mathcal{F} \neq \emptyset} \quad (\alpha \in Forms(\mathcal{F})) \iff (Fluents(\alpha) \subseteq \mathcal{F})$$

2.5 Dziedzina akcji

Niech $a \in \mathcal{A}c$ będzie akcją atomową, niech $f \in \mathcal{F}$ będzie fluentem oraz niech $\pi \in \mathbb{F}$ będzie formułą. W języku $\mathcal{ALC}$ występują następujące rodzaje zdań:

- Zdania efektu fluentu:
  - $\boxed{a \textbf{ causes } \alpha \textbf{ if } \pi}$ - jeżeli zachodzi warunek π w pewnej chwili t to wykonanie akcji a w tejże chwili skutkuje prawdziwością formuły α w następnej chwili $t+1$; jeżeli warunek π nie zachodzi to akcji nie można wykonać.
  - $\boxed{a \textbf{ causes } \alpha}$ - wykonanie akcji a w pewnej chwili t skutkuje prawdziwością formuły α w następnej chwili $t+1$.
- Zdania niemożliwości wykonania akcji:
  - $\boxed{\textbf{impossible } a \textbf{ at } t}$ - w chwili t nie jest możliwe wykonanie akcji a.
- Zdania uwolnienia fluentu:
  - $\boxed{a \textbf{ releases } f \textbf{ if } \pi}$ - jeżeli zachodzi warunek π w pewnej chwili t to wykonanie akcji a w tejże chwili może (ale nie musi) zmienić wartość fluentu f.
  - $\boxed{a \textbf{ releases } f}$ - wykonanie akcji a może (ale nie musi) zmienić wartość fluentu f.

Dowolny niepusty zbiór zdań nazywamy dziedziną akcji. Z każdą akcją atomową utożsamiamy dokładnie jedno zdanie w dziedzinie akcji typu: **causes** albo **releases**.
Uwaga językowa: Termin **releases** tłumaczymy na język polski jako *uwolnić*. Odnosząc się do zdań, zawierających **releases**, będziemy mówili, że f jest *uwalniany*.

2.6 Definicja scenariusza

Scenariuszem działań Sc nazywamy parę (OBS, ACS):

- Zbiór $OBS = \{(\alpha_1, t_1), (\alpha_2, t_2), \ldots, (\alpha_n, t_n)\}$ jest zbiorem obserwacji. Poprawny zbiór OBS spełnia następujące warunki:
  - α_i jest formułą dla $i = 1, 2, \ldots, n$ (czyli $\forall_{i=1,\ldots,n}\ \alpha_i \in Forms(\mathcal{F})$).
  - t_i jest liczbą naturalną opisującą chwilę rozpoczęcia, dla $i = 1, 2, \ldots, n$ (czyli $\forall_{i=1,\ldots,n}\ t_i \in \mathbb{N}$).

- Chwile rozpoczęcia są unikalne (czyli $t_i < t_j \implies i < j$ oraz $t_i = t_j \implies i = j$).
- Intuicja: w chwili t_i zachodzi formuła α_i.

- Zbiór $ACS = \{(A_1, t_1), (A_2, t_2), \dots, (A_k, t_k)\}$ jest zbiorem par, gdzie pierwszy element pary to akcja złożona A_i lub zbiór pusty, a drugi element pary to chwila jej rozpoczęcia t_i. Poprawny zbiór ACS spełnia następujące warunki:
  - A_i jest akcją złożoną lub zbiorem pustym dla $i = 1, 2, \dots, k$ (czyli $\forall_{i=1,\dots,k}\ A_i \subseteq \mathcal{A}c$).
  - t_i jest liczbą naturalną opisującą chwilę rozpoczęcia, dla $i = 1, 2, \dots, k$ (czyli $\forall_{i=1,\dots,k}\ t_i \in \mathbb{N}$).
  - Chwile rozpoczęcia są unikalne (czyli $t_i < t_j \implies i < j$ oraz $t_i = t_j \implies i = j$).
  - Intuicja: jeżeli akcja złożona A_i jest zbiorem niepustym to w chwili t_i ta akcja jest *potencjalnie wykonywana* jeżeli jest wykonywalna w danej chwili, bądź wykonywalny jest jej maksymalny, w sensie inkluzji, podzbiór niekonfliktowy (szczegółowa definicja znajduje się w następnych rozdziałach). Jeśli A_i jest zbiorem pustym (akcją pustą) to taką akcję pomijamy. Uzasadnione jest to równoważnością scenariuszy Sc i $Sc' = (OBS, ACS')$ gdzie zbiór ACS' jest zbiorem ACS bez akcji pustych, (czyli $ACS' = ACS - \bigcup_{t=1}^{\infty} (\emptyset, t)$).

3 Semantyka

3.1 Definicje akcji

3.1.1 Akcje atomowe potencjalnie wykonywalne

Dla niepustej akcji złożonej $A \subseteq \mathcal{A}c$, gdzie $A = (a_1, a_2, \dots, a_n)$, rozpatrywanej w chwili t, akcje potencjalnie wykonywalne to takie akcje atomowe a_i, dla których:

- W dziedzinie akcji nie zostało zdefiniowane zdanie $\boxed{\textbf{impossible}\ a_i\ \text{at}\ t}$.
- Spełniony jest jeden z poniższych warunków:
  - W dziedzinie akcji zostało zdefiniowane zdanie $\boxed{a_i\ \textbf{causes}\ \alpha_i}$.
  - W dziedzinie akcji zostało zdefiniowane zdanie $\boxed{a_i\ \textbf{causes}\ \alpha_i\ \textbf{if}\ \pi_i}$ i warunek początkowy π_i jest spełniony w chwili t.
  - W dziedzinie akcji zostało zdefiniowane zdanie $\boxed{a_i\ \textbf{releases}\ \alpha_i}$.
  - W dziedzinie akcji zostało zdefiniowane zdanie $\boxed{a_i\ \textbf{releases}\ \alpha_i\ \textbf{if}\ \pi_i}$ i warunek początkowy π_i jest spełniony w chwili t.

$PotentiallyExecutable(A, t)$ to zbiór wszystkich akcji atomowych należących do A, które są potencjalnie wykonywalne w chwili t.

3.1.2 Akcje atomowe konfliktowe

Uwaga notacyjna: Zdania typu $\boxed{a\ \textbf{causes}\ \alpha\ \textbf{if}\ \pi}$ zdefiniowane dla akcji $a \in PotentiallyExecutable(A, t)$ możemy zapisać, dla uproszczenia, jako $\boxed{a\ \textbf{causes}\ \alpha}$. Analogicznie dla zdań zawierających **releases**.
Niech $t \in \mathbb{N}$ będzie pewną chwilą rozpoczęcia, a A pewną akcją złożoną, zawierającą co najmniej 2 akcje atomowe potencjalnie wykonywalne a_1 i a_2. Mówimy że a_1 i a_2 są konfliktowe w chwili t gdy zachodzi co najmniej jeden z warunków:

- W dziedzinie akcji są zdefiniowane dwa zdania $\boxed{a_1 \textbf{ causes } \alpha_1}$ oraz $\boxed{a_2 \textbf{ causes } \alpha_2}$ i $\neg(\alpha_1 \wedge \alpha_2)$.
- W dziedzinie akcji są zdefiniowane dwa zdania $\boxed{a_1 \textbf{ causes } \alpha}$ oraz $\boxed{a_2 \textbf{ releases } f}$ i $f \in Fluents(\alpha)$.

3.1.3 Akcje złożone całkowicie wykonywalne

Akcja złożona A jest całkowicie wykonywalna w chwili t, wtedy i tylko wtedy, gdy zachodzą następujące warunki:

- A jest niepusta.
- Zbiór $PotentiallyExecutable(A, t)$ jest równy A.
- Koniunkcja wszystkich skutków akcji atomowych wchodzących w skład A jest prawdziwa (czyli dla zdań $\boxed{a_1 \textbf{ causes } \alpha_1}$, $\boxed{a_2 \textbf{ causes } \alpha_2}$, $\ldots$, $\boxed{a_n \textbf{ causes } \alpha_n}$ zachodzi $\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n$. W zapisie pomijamy warunki początkowe, gdyż są spełnione, co wynika z punktu powyżej).

Zbiór wszystkich akcji złożonych całkowicie wykonywalnych w chwili t nazywamy $CompoundExecutables(t)$.

3.1.4 Dekompozycja akcji przy użyciu funkcji MNS

Niech A będzie niepustą akcją złożoną wykonywaną w chwili t. Szukamy rodziny M podzbiorów A, gdzie każdy element tej rodziny spełnia następujące warunki:

- Jest niepustym podzbiorem A.
- Akcja złożona będąca tymże elementem jest całkowicie wykonywalna.
- Nie istnieje niepusta akcja złożona $A'' \in M$ spełniająca powyższe warunki będąca ścisłym nadzbiorem A'.

Rodzinę zbiorów M otrzymujemy przez obliczenie funkcji $MNS(A, t)$ dla niepustej złożonej akcji A i chwili t. Funkcję MNS definiujemy w następujący sposób:

Algorithm 1 MNS algorithm

Require: niepusta akcja złożona A oraz chwila jej rozpoczęcia t

Ensure: zbiór całkowicie wykonywalnych niepustych akcji złożonych maksymalnych pod względem inkluzji

```
function MNS(A, t)
    toRemove ← ∅
    M ← ∅
    for a ∈ A do
        if PotentiallyExecutable(a, t) then
            M ← M ∪ {a}
        end if
    end for
    p ← PowerSet(M)
    p ← p − {∅}
    for A′ ∈ p − CompoundExecutables(t) do
        M ← M − {A′}
    end for
    for A′ ∈ M do
        for A″ ∈ M do
            if A′ ⊊ A″ then
                toRemove ← toRemove ∪ {A″}
            end if
        end for
    end for
    for r ∈ toRemove do
        M ← M − {r}
    end for
    return M
end function
```

3.2 Struktura

Strukturą S nazywamy trójkę (H, O, E) gdzie

1. $H(f, t) : \mathcal{F} \times \mathbb{N} \to \{0, 1\}$ jest funkcją historii. Zwraca ona wartość 1 gdy fluent f jest prawdziwy w chwili t, w przeciwnym wypadku zwraca wartość 0.

2. $O(a, t) : \mathcal{A}_c \times \mathbb{N} \to 2^{\mathcal{F}}$ to funkcja okluzji, która określa zbiór fluentów na które wpływa akcja atomowa a, która została wykonana w poprzedniej chwili jednostce $t - 1$. W przypadku zdania **causes** są to wszystkie fluenty występujące w formule α będącej konsekwencją akcji a występującej w tymże zdaniu. W przypadku zdania **releases** jest to fluent f który zostaje uwolniony (*released*).

3. $E \subseteq 2^{\mathcal{A}_c} \times \mathbb{N}$ to relacja wiążąca niepustą akcję złożoną z chwilą w której jest wykonywana. W danej chwili można wykonać co najwyżej jedną niepustą akcję złożoną (czyli $\neg(((A, t) \in E) \wedge ((B, t) \in E) \wedge (A \neq B)))$. Pierwszy element każdej pary ze zbioru E, będący niepustą akcją złożoną, jest akcją całkowicie wykonywalną.

3.2.1 Rozszerzenie funkcji historii H na H^*

Dla każdej struktury $S = (H, O, E)$ funkcja historii H naturalnie uogólnia się na zbiór formuł zgodnie z zasadami logiki zdań. Takie rozszerzenie nazywamy rozszerzoną funkcją historii H^*. Wówczas dla każdej chwili $t \in \mathbb{N}$ prawdziwe jest co następuje:

- $\forall_{f \in \mathcal{F}}\ H^*(f,t) = H(f,t)$.
- $H^*(\neg\alpha, t) = 1 - H^*(\alpha, t)$.
- $H^*(\alpha \wedge \beta, t) = min(H^*(\alpha,t), H^*(\beta,t))$.
- $H^*(\alpha \vee \beta, t) = max(H^*(\alpha,t), H^*(\beta,t))$.
- $H^*(\alpha \to \beta, t) = \begin{cases} 0 & H^*(\alpha,t) = 1 \wedge H^*(\beta,t) = 0 \\ 1 & \text{w przeciwnym przypadku} \end{cases}$
- $H^*(\alpha \leftrightarrow \beta, t) = \begin{cases} 1 & H^*(\alpha,t) = H^*(\beta,t) \\ 0 & \text{w przeciwnym przypadku} \end{cases}$

3.2.2 Struktura dla scenariusza Sc

Niech $S = (H, O, E)$ będzie strukturą dla języka $\mathcal{ALC}$, niech $Sc = (OBS, ACS)$ będzie scenariuszem, i niech D będzie dziedziną akcji. Mówimy, że $\boxed{S \text{ jest } \textbf{strukturą dla scenariusza } Sc \textbf{ względem } D}$, wtedy i tylko wtedy, gdy zachodzą następujące warunki:

- Każda obserwacja podana w scenariuszu zachodzi dla struktury (czyli $\forall_{(\alpha,t) \in OBS} \quad H^*(\alpha,t) = 1$).
- W strukturze wydarza się co najmniej jedna atomowa akcja przewidziana do wykonania w scenariuszu w chwili t (czyli $\forall_{(A,t) \in ACS, A \neq \emptyset}\ \exists_{(A',t) \in E} \quad (A' \subseteq A) \wedge (A' \neq \emptyset)$).
- Dla każdej wykonywanej akcji złożonej A zawierającej akcję atomową a (tzn. $((A,t) \in E) \wedge (a \in A)$), której skutkiem jest formuła α, wszystkie fluenty występujące w tejże formule należą do okluzji w chwili $t+1$ (czyli $Fluents(\alpha) \subseteq O(a, t+1)$). Warunek dotyczy zdań typu **causes**.
- Dla każdej wykonywanej akcji złożonej A zawierającej akcję atomową a (czyli $((A,t) \in E) \wedge (a \in A)$), która uwalnia fluent $f \in \mathcal{F}$, tenże fluent należy do okluzji w chwili $t+1$ (czyli $f \in O(a, t+1)$). Warunek dotyczy zdań typu **releases**.

3.3 Model

3.3.1 Definicja modelu

Mówimy, że struktura $S = (H, O, E)$ jest modelem scenariusza $Sc = (OBS, ACS)$ względem dziedziny akcji D wtedy i tylko wtedy gdy zachodzą następujące warunki:

M₁ Każdy fluent którego wartość ulega zmianie z chwili t na chwilę $t+1$, należy do funkcji okluzji (czyli $\forall_{f \in \mathcal{F}}\ H(f,t) \neq H(f,t+1) \implies \exists_{(A,t) \in E}\ \exists_{a \in A}\ f \in O(a, t+1)$).

M_2 Każdy fluent występujący w koniunkcji skutków akcji atomowych (dalej nazywany *Consequence*) wchodzących w skład wykonywanej akcji złożonej może się zmienić tylko wtedy, gdy zachodzi następujący warunek: Koniunkcja skutków akcji dla której rozpatrywany fluent (powiedzmy f_k) przybiera wartość *true* nie jest równoważna koniunkcji skutków akcji dla której rozpatrywany fluent przybiera wartość *false* (czyli $\forall_{f_k \in \mathcal{F}}\ H(f_k, t) \neq H(f_k, t{+}1) \implies Consequence(f_1, f_2, \ldots, f_{k-1}, true, f_{k+1}, \ldots, f_n) \not\Leftrightarrow$

$Consequence(f_1, f_2, \ldots, f_{k-1}, false, f_{k+1}, \ldots, f_n)$).

M_3 Zachodzą wszystkie obserwacje podane w scenariuszu (czyli $\forall_{(\alpha,t) \in OBS}\ H^*(\alpha, t) = 1$).

M_4 Wykonywana niepusta akcja złożona A jest maksymalna pod względem inkluzji (czyli $\forall_{(A',t) \in E, A' \neq \emptyset}\ \exists_{(A,t) \in ACS, A \neq \emptyset}\ A' \in MNS(A, t)$).

Mówimy, że scenariusz Sc jest spójny (*consistent*) względem dziedziny akcji D wtedy i tylko wtedy gdy istnieje dla niego model względem tejże dziedziny. W przeciwnym przypadku scenariusz nazywamy niespójnym (*inconsistent*).

4 Kwerendy języka $\mathcal{ALC}$ i ich interpretacja

Q_{1_a} Odpowiedź na kwerendę $\boxed{\textbf{necessarily executable scenario } Sc}$ (scenariusz jest zawsze wykonywalny) jest pozytywna, wtedy i tylko wtedy, gdy istnieje co najmniej jeden model oraz gdy nie istnieje jakikolwiek model sprzeczny, czyli dla wszystkich modeli i dla każdej pary zachodzi warunek $\forall_{(A,t) \in ACS, A \neq \emptyset}\ MNS(A, t) \neq \emptyset$ oraz zachodzą absolutnie wszystkie obserwacje ze zbioru OBS, również te z późniejszych chwil.

Q_{1_b} Odpowiedź na kwerendę $\boxed{\textbf{possibly executable scenario } Sc}$ (scenariusz jest wykonywalny kiedykolwiek) jest pozytywna, wtedy i tylko wtedy, gdy istnieje taki model, że dla każdej pary zachodzi warunek $\forall_{(A,t) \in ACS, A \neq \emptyset}\ MNS(A, t) \neq \emptyset$ oraz zachodzą absolutnie wszystkie obserwacje ze zbioru OBS, również te z późniejszych chwil.

Q_{2_a} Odpowiedź na kwerendę $\boxed{\textbf{necessarily accessible } \gamma \textbf{ at } t \textbf{ when } Sc}$ (warunek zawsze zachodzi) jest pozytywna, wtedy i tylko wtedy, gdy istnieje co najmniej jeden model oraz gdy dla każdego modelu $S = (H, O, E)$ scenariusza Sc względem dziedziny akcji D warunek γ jest prawdziwy w chwili t (czyli $H^*(\gamma, t) = 1$) oraz nie istnieje model który byłby sprzeczny w chwili t lub wcześniejszej.

Q_{2_b} Odpowiedź na kwerendę $\boxed{\textbf{possibly accessible } \gamma \textbf{ at } t \textbf{ when } Sc}$ (warunek zachodzi kiedykolwiek) jest pozytywna, wtedy i tylko wtedy, gdy istnieje model $S = (H, O, E)$ scenariusza Sc względem dziedziny akcji D, w którym warunek γ jest prawdziwy w chwili t (czyli $H^*(\gamma, t) = 1$).

Q_{3_a} Odpowiedź na kwerendę $\boxed{\textbf{necessarily executable } A \textbf{ at } t \textbf{ when } Sc}$ (akcja złożona jest wykonywalna zawsze) jest pozytywna, wtedy i tylko wtedy, gdy istnieje co najmniej jeden model oraz gdy dla każdego modelu $S = (H, O, E)$ scenariusza Sc względem dziedziny akcji D niepusta akcja złożona A w chwili t jest całkowicie wykonywalna (definicję można znaleźć w rozdziale 3.1.3, czyli $A \in CompoundExecutables(t)$) oraz nie istnieje model który byłby sprzeczny w chwili t lub wcześniejszej.

Q_{3_b} Odpowiedź na kwerendę $\boxed{\textbf{possibly executable } A \textbf{ at } t \textbf{ when } Sc}$ (akcja złożona jest wykonywalna kiedykolwiek) jest pozytywna, wtedy i tylko wtedy, gdy istnieje model $S = (H, O, E)$ dla scenariusza Sc

względem dziedziny akcji D, taki, że w chwili t niepusta akcja złożona A jest całkowicie wykonywalna (definicję można znaleźć w rozdziale 3.1.3, czyli $A \in CompoundExecutables(t)$).

5 Przykłady

5.1 Warsaw standoff

Rysunek 1: Przykład Warsaw standoff w prawdziwym życiu

Podczas spotkania trzech gangsterów w trakcie rozmowy dochodzi do sprzeczki, która kończy się każdym z nich celującym w jednego z kolegów. Gangsterzy 1 i 2 są bardzo szybcy, natomiast gangster 3 ma słabszy refleks od kolegów. Każdy z gangsterów ma karabin, lecz może on pochodzić z gorszej partii, z której nie wszystkie karabiny działają po załadowaniu. Są jednak także karabiny w pełni sprawne. Jeśli karabin zadziała, nie trzeba go przeładowywać, mają one duże magazynki. Gangsterzy celują do siebie w następujący sposób i wiedzą, że jeśli ktorykolwiek strzeli, zrobią to także pozostali (gangster 3 z opóźnieniem). Muszą jednak być skierowani w stronę przeciwnika, którego chcą trafić, aby go zabić. Celują do siebie zgodnie z poniższym schematem:

Rysunek 2: Gangster 1 celuje w gangstera 2 swoim "niepewnym" "karabinem", gangster 2 w gangstera 3 "niepewnym" "karabinem", a gangster 3 w gangstera 1 swoim sprawnym "karabinem". Mimo, że na ilustracji gangsterzy władają szablą zamiast karabinami, konfiguracja jest w pełni poprawnie zilustrowania.

Domena akcji:
$SHOOT12 \; causes \; \neg ALIVE2 \; if \; ALIVE1 \wedge \neg JAMMED1 \wedge FACING12$

$SHOOT13\ causes\ \neg ALIVE3\ if\ ALIVE1 \wedge \neg JAMMED1 \wedge FACING13$
$SHOOT21\ causes\ \neg ALIVE1\ if\ ALIVE2 \wedge \neg JAMMED2 \wedge FACING21$
$SHOOT23\ causes\ \neg ALIVE3\ if\ ALIVE2 \wedge \neg JAMMED2 \wedge FACING23$
$SHOOT31\ causes\ \neg ALIVE1\ if\ ALIVE3 \wedge \neg JAMMED3 \wedge FACING31$
$SHOOT32\ causes\ \neg ALIVE2\ if\ ALIVE3 \wedge \neg JAMMED3 \wedge FACING32$
$impossible\ SHOOT31\ at\ 0$
$impossible\ SHOOT32\ at\ 0$
$impossible\ SHOOT31\ at\ 1$
$impossible\ SHOOT32\ at\ 1$

$ROTATE12\ causes\ FACING12\ if\ ALIVE1$
$ROTATE13\ causes\ FACING13\ if\ ALIVE1$
$ROTATE21\ causes\ FACING21\ if\ ALIVE2$
$ROTATE23\ causes\ FACING23\ if\ ALIVE2$
$ROTATE31\ causes\ FACING31\ if\ ALIVE3$
$ROTATE32\ causes\ FACING32\ if\ ALIVE3$

$LOAD1\ releases\ JAMMED1\ if\ ALIVE1$
$LOAD2\ releases\ JAMMED2\ if\ ALIVE2$
$LOAD3\ causes\ \neg JAMMED3\ if\ ALIVE3$

Rozważmy następujący scenariusz:
$Sc = (OBS, ACS)$
$OBS = \{$
$(ALIVE1 \wedge ALIVE2 \wedge ALIVE3 \wedge$
$\neg JAMMED1 \wedge \neg JAMMED2 \wedge \neg JAMMED3 \wedge$
$FACING12 \wedge FACING23 \wedge FACING31, 0),$
$(\neg JAMMED1 \vee \neg JAMMED2, 1)$
$\}$
$ACS = \{$
$(\{LOAD1, LOAD2, LOAD3\}, 0),$
$(\{SHOOT12, SHOOT23\}, 1),$
$(\{ROTATE13, ROTATE21, SHOOT31\}, 2),$
$(\{SHOOT13, SHOOT21, ROTATE32\}, 3),$
$(\{SHOOT32\}, 4)$
$\}$

	0→1	1→2	2→3	3→4	4→5
	LOAD1		ROTATE13	SHOOT13	
	LOAD2	SHOOT12	ROTATE21	SHOOT21	
	LOAD3	SHOOT23	ROTATE31	ROTATE32	SHOOT32

0	1	2	3	4	5
$a1$	$a1$	$a1$	$\neg a1*$	$\neg a1*$	$\neg a1?$
$a2$	$a2$	$\neg a2*$	$\neg a2?$	$\neg a2?$	$\neg a2*$
$a3$	$a3$	$\neg a3*$	$\neg a3?$	$\neg a3*$	$\neg a3?$
$\neg j1$	$j1*$	$j1?$	$j1?$	$j1?$	$j1?$
$\neg j2$	$j2*$	$j2?$	$j2?$	$j2?$	$j2?$
$\neg j3$	$\neg j3$	$\neg j3$	$\neg j3$	$\neg j3$	$\neg j3$
$f12$	$f12$	$f12$	$\neg f12*$	$\neg f12?$	$\neg f12?$
$\neg f13$	$\neg f13$	$\neg f13$	$f13*$	$f13?$	$f13?$
$\neg f21$	$\neg f21$	$\neg f21$	$f21*$	$f21?$	$f21?$
$f23$	$f23$	$f23$	$\neg f23*$	$\neg f23?$	$\neg f23?$
$f31$	$f31$	$f31$	$f31$	$\neg f31*$	$\neg f31?$
$\neg f32$	$\neg f32$	$\neg f32$	$\neg f32$	$f32*$	$f32?$

Rysunek 3: Timeline scenariusza Warsaw standoff

Jeżeli oba karabiny pierwszego i drugiego gangstera są działające to potyczkę przeżywa pierwszy gangster, ponieważ jednocześnie drugi wyeliminuje trzeciego kiedy pierwszy wyeliminuje drugiego. Trzeci gangster ma opóźniony refleks, więc nie zdąży zabić pierwszego.

Jeżeli tylko karabin drugiego gangstera jest niedziałający to potyczkę przeżywa trzeci gangster, ponieważ drugi gangster zostanie zabity przez pierwszego, a w następnej chwili podczas gdy pierwszy gangster obraca karabin w stronę trzeciego zostaje przez niego postrzelony gdyż trzeci gangster zdążył w porę naładować broń.

Jeżeli tylko karabin pierwszego gangstera jest niedziałający to potyczkę przeżywa drugi gangster, ponieważ trzeci gangster ma opóźniony refleks i drugi gangster zdąży go zastrzelić, następnie obróci się w stronę pierwszego, bezbronnego gangstera aby go następnie zabić.

Z powyższych rozważań odpowiedź na kwerendę **possibly accessible** $ALIVE1 \wedge \neg ALIVE2 \wedge \neg ALIVE3$ **at** 5 **when** Sc będzie pozytywna.

Odpowiedź na kwerendę **possibly accessible** $ALIVE3 \wedge JAMMED1$ **at** 5 **when** Sc będzie negatywna, gdyż trzeci gangster przetrwa jedynie gdy drugi gangster będzie miał uszkodzoną broń, ale nie istnieje konfiguracja w której obie bronie jednocześnie pierwsza i druga są uszkodzone.

0 → 1	1 → 2	2 → 3	3 → 4	4 → 5
LOAD1 LOAD2 LOAD3	SHOOT12 SHOOT23	ROTATE13 ROTATE21 ROTATE31	SHOOT13 SHOOT21 ROTATE32	SHOOT32

0	1	2	3	4	5
$a1$	$a1$	$a1$	$a1*$	$\neg a1*$	$\neg a1$
$a2$	$a2$	$a2*$	$a2$	$a2$	$a2*$
$a3$	$a3$	$\neg a3*$	$\neg a3$	$\neg a3*$	$\neg a3$
$\neg j1$	$j1*$	$j1$	$j1$	$j1$	$j1$
$\neg j2$	$\neg j2*$	$\neg j2$	$\neg j2$	$\neg j2$	$\neg j2$
$\neg j3$	$\neg j3$	$\neg j3$	$\neg j3$	$\neg j3$	$\neg j3$
$f12$	$f12$	$f12$	$\neg f12*$	$\neg f12$	$\neg f12$
$\neg f13$	$\neg f13$	$\neg f13$	$f13*$	$f13$	$f13$
$\neg f21$	$\neg f21$	$\neg f21$	$f21*$	$f21$	$f21$
$f23$	$f23$	$f23$	$\neg f23*$	$\neg f23$	$\neg f23$
$f31$	$f31$	$f31$	$f31$	$f31*$	$f31$
$\neg f32$	$\neg f32$	$\neg f32$	$\neg f32$	$\neg f32*$	$\neg f32$

Rysunek 4: Timeline scenariusza Warsaw standoff kiedy zepsuty jest tylko karabin pierwszego gangstera - przeżywa drugi gangster

5.2 Promocja w supermarkecie

W supermarkecie Świat Telewizorów rozpoczęła się promocja, w której można zakupić elektronikę z bardzo wysoką zniżką. W sklepie znajdują się trzy alejki: z telewizorami, komputerami i konsolami do gier. Po sklepie krąży dwóch klientów. Każdy z nich może przeszukać najbliższe stoisko w celu zakupu przecenionego produktu. Klienci mogą też przejść do innego stoiska. Promocja z racji wyczerpania zapasów już się kończy i półki są puste, jednak co pewien czas obsługa sklepu uzupełnia niektóre stoiska ostatnimi produktami z magazynu. Klient pierwszy jest z innego miasta, więc gubi się w sklepie, oraz łatwo się rozprasza - np. chcąc przejść ze stoiska z komputerami do stoiska z konsolami może się zatrzymać przy stoisku z telewizorami. Każdy zakup przecenionego produktu czyni kupującego klienta usatysfakcjonowanym.

Dziedzina dla zagadnienia:
$BUY\_TV\_1\ causes\ HAPPY\_1\ \wedge\ \neg\ TV\ if\ NEAR\_TV\_1\ \wedge\ TV$
$BUY\_TV\_2\ causes\ HAPPY\_2\ \wedge\ \neg\ TV\ if\ NEAR\_TV\_2\ \wedge\ TV$
$BUY\_COMPUTER\_1\ causes\ HAPPY\_1 \wedge \neg\ COMPUTER\ if\ NEAR\_COMPUTER\_1 \wedge COMPUTER$
$BUY\_COMPUTER\_2\ causes\ HAPPY\_2 \wedge \neg\ COMPUTER\ if\ NEAR\_COMPUTER\_2 \wedge COMPUTER$
$BUY\_CONSOLE\_1\ causes\ HAPPY\_1\ \wedge\ \neg\ CONSOLE\ if\ NEAR\_CONSOLE\_1\ \wedge\ CONSOLE$
$BUY\_CONSOLE\_2\ causes\ HAPPY\_2\ \wedge\ \neg\ CONSOLE\ if\ NEAR\_CONSOLE\_2\ \wedge\ CONSOLE$

$MOVE\_1\ causes$
$(NEAR\_TV\_1\ \wedge\ \neg\ NEAR\_COMPUTER\_1\ \wedge\ \neg\ NEAR\_CONSOLE\_1)\ \vee$
$(\neg\ NEAR\_TV\_1\ \wedge\ NEAR\_COMPUTER\_1\ \wedge\ \neg\ NEAR\_CONSOLE\_1)\ \vee$
$(\neg\ NEAR\_TV\_1\ \wedge\ \neg\ NEAR\_COMPUTER\_1\ \wedge\ NEAR\_CONSOLE\_1)$

$GO\_TO\_TV\_2\ causes\ NEAR\_TV\_2\ \wedge\ \neg\ NEAR\_COMPUTER\_2\ \wedge\ \neg\ NEAR\_CONSOLE\_2$
$GO\_TO\_CONSOLE\_2\ causes\ NEAR\_CONSOLE\_2 \wedge \neg\ NEAR\_COMPUTER\_2 \wedge \neg\ NEAR\_TV\_2$
$GO\_TO\_COMPUTER\_2\ causes\ NEAR\_COMPUTER\_2 \wedge \neg\ NEAR\_TV\_2 \wedge \neg\ NEAR\_CONSOLE\_2$

$ADD\_TV\ causes\ TV$
$ADD\_COMPUTER\ causes\ COMPUTER$
$ADD\_CONSOLE\ causes\ CONSOLE$

Przykładowy scenariusz:
$OBS = ($
$\{\neg\ HAPPY\_1,\ \neg\ HAPPY\_2,\ \neg\ TV,\ \neg\ COMPUTER,\ \neg\ CONSOLE,$
$NEAR\_TV\_1,\ \neg\ NEAR\_COMPUTER\_1,\ \neg\ NEAR\_CONSOLE\_1,$
$\neg\ NEAR\_TV\_2,\ \neg\ NEAR\_COMPUTER\_2,\ NEAR\_CONSOLE\_2\}$
$, 0)$
$ACS = \{$
({MOVE_1, GO_TO_COMPUTER_2, ADD_TV}, 0)
({BUY_TV_1, BUY_COMPUTER_2, MOVE_1, GO_TO_CONSOLE_2, ADD_CONSOLE}, 1)
({BUY_TV_1, BUY_CONSOLE_2}, 2)
}

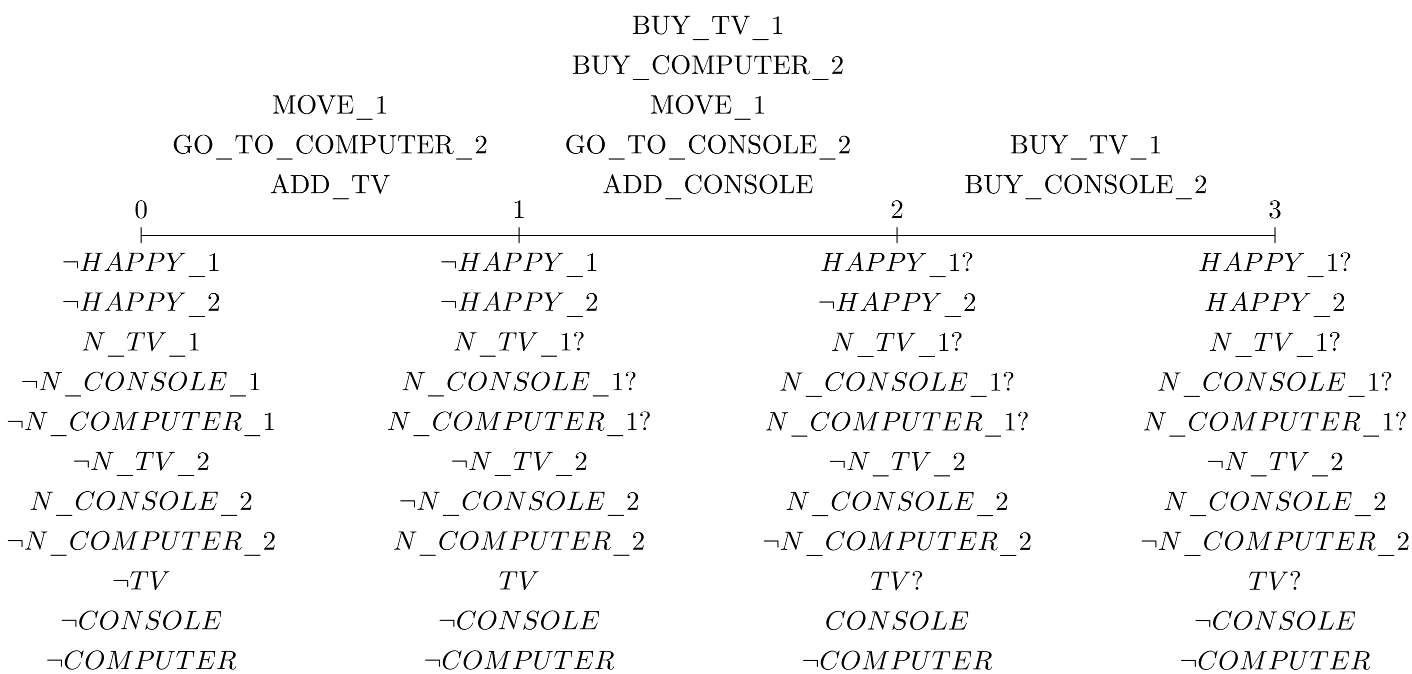
W tym przykładzie poprzez obserwację w chwili zero znana jest wartość funkcji historii dla dowolnego fluentu w momencie 0. Algorytm MNS dla chwili 0 zwróci zbiór $\{MOVE\_1, GO\_TO\_COMPUTER\_2, ADD\_TV\}$, ponieważ akcje atomowe nie posiadają warunków wykonania i wpływają na różne fluenty. Akcja ADD_TV spowoduje ustawienie wartości fluentu TV na **True**. Akcja $MOVE\_1$ spowoduje ustawienie wartości jednego z fluentów $NEAR\_X\_1$ na **True**, a pozostałych na **False**. Wartości pozostałych fluentów nie mogą ulec zmianie z racji zasady inercji.

W momencie 1 algorytm MNS zwróci zbiór $\{MOVE\_1, GO\_TO\_CONSOLE\_2, ADD\_CONSOLE\}$ lub $\{MOVE\_1, GO\_TO\_CONSOLE\_2, ADD\_CONSOLE, BUY\_TV\_1\}$ w zależności od wartości fluentu $NEAR\_TV\_1$. Akcja $MOVE\_1$ spowoduje ustawienie wartości jednego z fluentów $NEAR\_X\_1$ na **True**, a pozostałych na **False**. Akcja $GO\_TO\_CONSOLE\_2$ spowoduje ustawienie wartości fluentów

$NEAR\_TV\_2$ i $NEAR\_COMPUTER\_2$ na **False** oraz fluentu $NEAR\_CONSOLE\_2$ na **True**. Akcja $ADD\_CONSOLE$ spowoduje ustawienie wartości fluentu $CONSOLE$ na **True**. Jeżeli wartość fluentu $NEAR\_TV\_1$ to **True**, to wykonana zostanie akcja BUY_TV_1, która ustawi wartość fluentu TV na **False** oraz wartość fluentu $HAPPY\_1$ na **True**.

W momencie 2 algorytm MNS zwróci zbiór $\{BUY\_CONSOLE\_2\}$ lub $\{BUY\_CONSOLE\_2, BUY\_TV\_1\}$ w zależności od wartości fluentu $NEAR\_TV\_1$. Akcja $BUY\_CONSOLE\_2$ ustawi wartość fluentu $CONSOLE$ na **False** oraz fluentu $HAPPY\_2$ na **True**. Jeżeli wartość fluentu $NEAR\_TV\_1$ to **True**, to wykonana zostanie akcja BUY_TV_1, która ustawi wartość fluentu TV na **False** oraz wartość fluentu $HAPPY\_1$ na **True**.

Odpowiedź na kwerendę
necessarily executable $\{MOVE\_1, GO\_TO\_CONSOLE\_2, ADD\_CONSOLE, BUY\_TV\_1\}$ at **1** dla tego scenariusza to **False**, ponieważ dla modelu, w którym $H(NEAR\_TV\_1, 1) = 0$ akcja atomowa BUY_TV_1 jest niewykonywalna. Odpowiedź na kwerendę
possibly executable $\{MOVE\_1, GO\_TO\_CONSOLE\_2, ADD\_CONSOLE, BUY\_TV\_1\}$ at **1** dla tego scenariusza to **True**, ponieważ dla modelu, w którym $H(NEAR\_TV\_1, 1) = 1$ każda z akcji atomowych jest potencjalnie wykonywalna i wpływa na niezależne fluenty.



Rysunek 5: Timeline scenariusza *Promocja w supermarkecie*

5.3 Ewakuacja budynku

Mamy trzypiętrowy budynek. Na każdym piętrze pracują osoby. Na pierwszym piętrze - grupa A, na drugim piętrze - grupa B, na trzecim piętrze - grupa C. W budynku wybucha pożar. Wszystkie grupy są zagrożone. Trzeba ewakuować budynek. Ewakuacja grupy z piętra trzeciego odbywa się w krokach: na piętro 2., na pię-

0 → 1	MOVE_1, GO_TO_COMPUTER_2, ADD_TV
1 → 2	BUY_TV_1, BUY_COMPUTER_2, MOVE_1, GO_TO_CONSOLE_2, ADD_CONSOLE
2 → 3	BUY_TV_1, BUY_CONSOLE_2

0	1	2	3
¬HAPPY_1	¬HAPPY_1	HAPPY_1	HAPPY_1
¬HAPPY_2	¬HAPPY_2	¬HAPPY_2	HAPPY_2
N_TV_1	N_TV_1	N_TV_1	N_TV_1
¬N_CONSOLE_1	¬N_CONSOLE_1	¬N_CONSOLE_1	¬N_CONSOLE_1
¬N_COMPUTER_1	¬N_COMPUTER_1	¬N_COMPUTER_1	¬N_COMPUTER_1
¬N_TV_2	¬N_TV_2	¬N_TV_2	¬N_TV_2
N_CONSOLE_2	¬N_CONSOLE_2	N_CONSOLE_2	N_CONSOLE_2
¬N_COMPUTER_2	N_COMPUTER_2	¬N_COMPUTER_2	¬N_COMPUTER_2
¬TV	TV	¬TV	¬TV
¬CONSOLE	¬CONSOLE	CONSOLE	¬CONSOLE
¬COMPUTER	¬COMPUTER	¬COMPUTER	¬COMPUTER

Rysunek 6: Timeline scenariusza *Promocja w supermarkecie* przy założeniu, że pierwszy klient zawsze podchodzi do stoiska z telewizorami

tro 1., poza budynek. Ewakuacja grupy z piętra drugiego odbywa się w krokach: na piętro 1., poza budynek. Ewakuacja grupy z piętra pierwszego odbywa się w jednym kroku: poza budynek. Nie można ewakuować grupy na piętro niższe, jeśli ktoś się tam znajduje. Podczas ewakuacji może nastąpić panika, która powoduje, że dana grupa nie daje się ewakuować i zostaje na piętrze, na którym była.

Dziedzina:
$Cfrom3to2\ causes\ (\neg Con3 \wedge Con2)\ if\ (Con3 \wedge \neg Bon2)$
$Cfrom2to1\ causes\ (\neg Con2 \wedge Con1)\ if\ (Con2 \wedge \neg Bon1)$
$Bfrom2to1\ causes\ (\neg Bon2 \wedge Bon1)\ if\ (Bon2 \wedge \neg Aon1)$
$EvacuateAfrom1\ causes\ (\neg Aon1 \wedge SafeA)\ if\ Aon1$
$EvacuateBfrom1\ causes\ (\neg Bon1 \wedge SafeB)\ if\ Bon1$
$EvacuateCfrom1\ causes\ (\neg Con1 \wedge SafeC)\ if\ Con1$
$PanicCon3\ causes\ Con3\ if\ Con3$
$PanicCon2\ causes\ Con2\ if\ Con2$
$PanicCon1\ causes\ Con1\ if\ Con1$
$PanicBon2\ causes\ Bon2\ if\ Bon2$
$PanicBon1\ causes\ Bon1\ if\ Bon1$
$PanicAon1\ causes\ Aon3\ if\ Aon3$

Rozważmy następujący scenariusz:

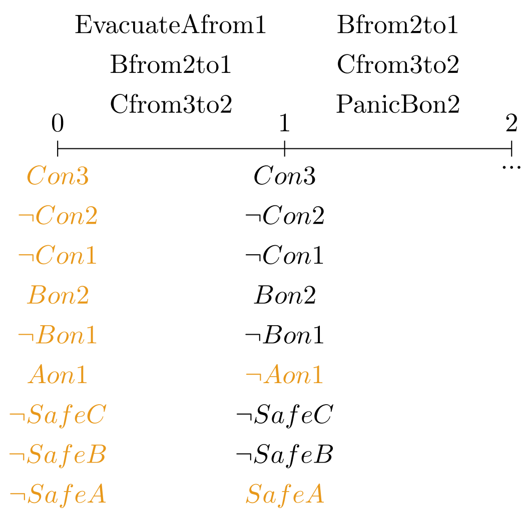
$Sc = (OBS, ACS)$
$OBS = \{$
$(Con3 \ \wedge Bon2 \ \wedge Aon1 \ \wedge \neg SafeA \ \wedge \neg SafeB \ \wedge \neg SafeC \ , 0)$
$\}$

$ACS = \{$
$(\{EvacuateAfrom1,\ Bfrom2to1,\ Cfrom3to2\}, 0),$
$(\{Bfrom2to1,\ Cfrom3to2,\ PanicBon2\}, 1),$
$(\{Bfrom2to1,\ EvacuateBfrom1,\ Cfrom3to2\}, 2),$
$(\{EvacuateBfrom1,\ Cfrom3to2,\ Cfrom2to1\}, 3),$
$(\{EvacuatCfrom1,\ Cfrom2to1\}, 4),$
$(\{EvacuatCfrom1\}, 5),$
$\}$

Prześledźmy przebieg scenariusza. W chwili 0. wykona się jedynie akcja $EvacuateAfrom1$, gdyż warunki wstępne pozostałych akcji nie są spełnione. W chwili 1. następuje konflikt. Konfliktują ze sobą akcje $Bfrom2to1$ oraz $PanicBon2$. W tym momencie akcja $\{Bfrom2to1,\ Cfrom3to2,\ PanicBon2\}$ zostanie rozłożona za pomocą algorytmu MNS na maksymalne pod względem inkluzji niekonfliktowe akcje złożone:

- $\{Bfrom2to1,\ Cfrom3to2\}$
- $\{PanicBon2,\ Cfrom3to2\}$

W tym momencie obserwujemy współbieżność - scenariusz rozdziela się na dwie ścieżki wykonania.



Rysunek 7: Timeline scenariusza „Ewakuacja budynku"do momentu rozdzielenia się ścieżek wykonania

Pytanie, które można postawić w tej historyjce, to: „Czy uda się uratować wszystkich ludzi w danym czasie"? Warunek, który świadczy o tym, że "wszyscy zostali uratowani", to: $\gamma = SafeA \wedge SafeB \wedge SafeC$

Zauważmy, że odpowiedź na kwerendę **possibly accessible** γ **at 4 when** Sc będzie negatywna, odpowiedź na kwerendę **possibly accessible** γ **at 5 when** Sc będzie pozytywna, odpowiedź na kwerendę **necessarily accessible** γ **at 5 when** Sc będzie negatywna.

0–1	1–2	2–3	3–4	4–5	5–6
EvacuateAfrom1	Bfrom2to1	Bfrom2to1	EvacuateBfrom1	EvacuateCfrom1	EvacuateCfrom1
Bfrom2to1	Cfrom3to2	EvacuateBfrom1	Cfrom3to2	Cfrom2to1	
Cfrom3to2	~~PanicBon2~~	Cfrom3to2	Cfrom2to1		

0	1	2	3	4	5	6
$Con3$	$Con3$	$Con3$	$\neg Con3$	$\neg Con3$	$\neg Con3$	$\neg Con3$
$\neg Con2$	$\neg Con2$	$\neg Con2$	$Con2$	$\neg Con2$	$\neg Con2$	$\neg Con2$
$\neg Con1$	$\neg Con1$	$\neg Con1$	$\neg Con1$	$Con1$	$\neg Con1$	$\neg Con1$
$Bon2$	$Bon2$	$\neg Bon2$	$\neg Bon2$	$\neg Bon2$	$\neg Bon2$	$\neg Bon2$
$\neg Bon1$	$\neg Bon1$	$Bon1$	$\neg Bon1$	$\neg Bon1$	$\neg Bon1$	$\neg Bon1$
$Aon1$	$\neg Aon1$	$\neg Aon1$	$\neg Aon1$	$\neg Aon1$	$\neg Aon1$	$\neg Aon1$
$\neg SafeC$	$\neg SafeC$	$\neg SafeC$	$\neg SafeC$	$\neg SafeC$	$SafeC$	$SafeC$
$\neg SafeB$	$\neg SafeB$	$\neg SafeB$	$SafeB$	$SafeB$	$SafeB$	$SafeB$
$\neg SafeA$	$SafeA$	$SafeA$	$SafeA$	$SafeA$	$SafeA$	$SafeA$

Rysunek 8: Timeline scenariusza *Ewakuacja budynku*, kiedy w momencie 1. zostanie wykonane $\{Bfrom2to1, Cfrom3to2\}$. Akcja w momencie 5 nic nie zmienia.

5.4 Myślał indyk o niedzieli

Fred biega bardzo szybko i nie da się go zastrzelić. Jeśli Fred myśli o niedzieli, to zatrzymuje się. Jeśli indyk się nie porusza i Bill strzeli w niego, indyk zostaje zabity. W okolicy znajduje się Adam - przedstawiciel organizacji, zajmującej się prawami zwierząt. Adam potrafi wydawać przeraźliwy okrzyk, który powoduje, że indyk zaczyna biegać.

Dziedzina:

$THINK\_ABOUT\_SUNDAY$ *causes* $\neg MOVING$

$ALARM$ *causes* $MOVING$ if $ALIVE$

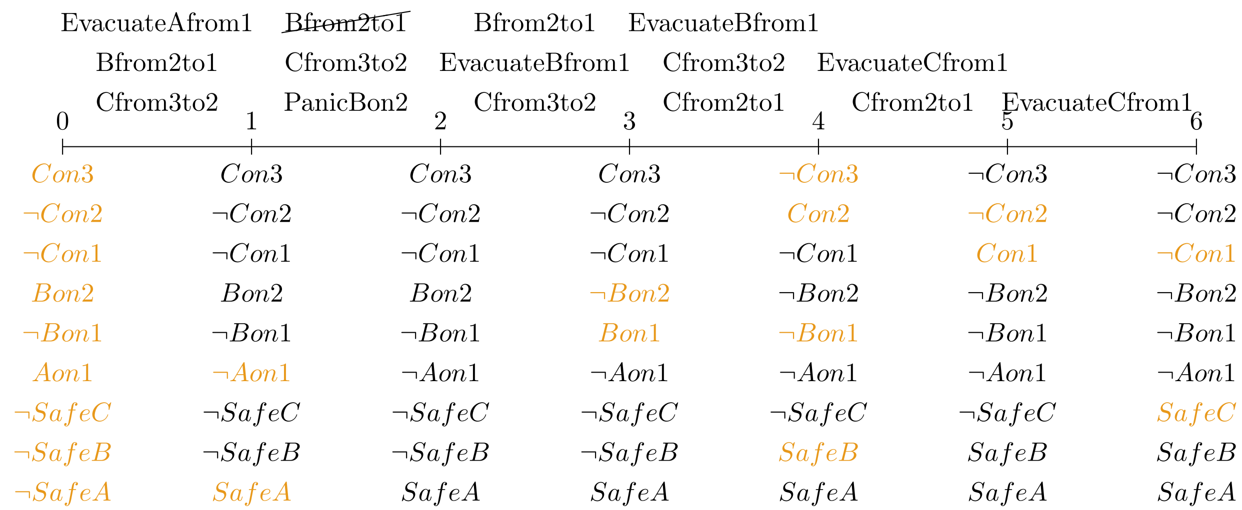
$SHOOT$ *causes* $(\neg ALIVE \wedge \neg MOVING)$ if $\neg MOVING$

$Sc = (OBS, ACS)$

$OBS = \{(ALIVE \wedge MOVING, 0)\}$

$ACS = \{$

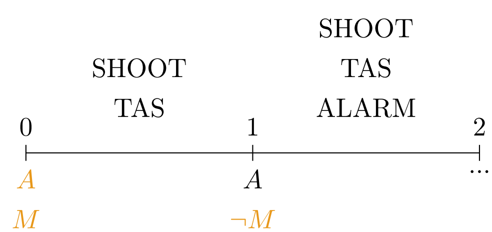
$(\{SHOOT, THINK\_ABOUT\_SUNDAY, 0),$



Rysunek 9: Timeline scenariusza „Ewakuacja budynku", kiedy w momencie 1. zostanie wykonane $\{PanicBon2,\ Cfrom3to2\}$

$(\{SHOOT,\ THINK\_ABOUT\_SUNDAY,\ ALARM\}, 1)$,
$(\{SHOOT\}, 2)$,
$\}$

Przeanalizujmy wykonanie danego scenariusza. W chwili 0, wykonana zostanie jedna akcja: THINK_ABOUT_SUNDAY, ponieważ warunek wstępny akcji SHOOT nie jest spełniony, zatem nie będzie żadnych konfliktów.



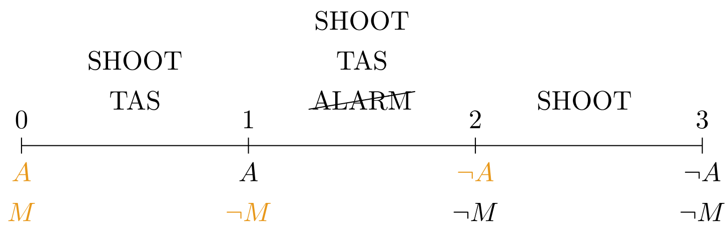
Rysunek 10: Początkowa część timeline'u do momentu rozłączenia

W chwili 1 mamy do wykonania akcję złożoną z trzech akcji atomowych. Konfliktowe są akcje TAS i ALARM oraz akcje SHOOT i ALARM. Algorytmu MNS wyznacza zbiór całkowicie wykonywalnych akcji złożonych, składający się z dwóch elementów:

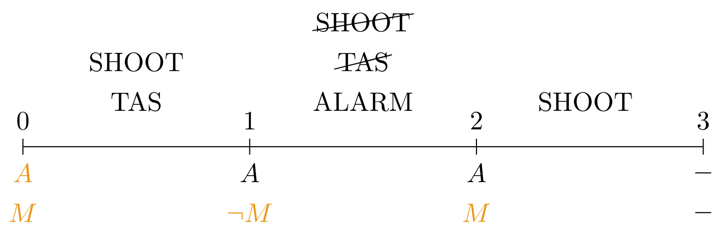
- $\{SHOOT,\ THINK\_ABOUT\_SUNDAY\}$
- $\{ALARM\}$

Zauważmy, że do zbioru, wyznaczonego przez algorytm nie będzie należała akcja $\{SHOOT\}$, gdyż istnieje większa pod względem inkluzji akcja: $\{SHOOT,\ THINK\_ABOUT\_SUNDAY\}$.

W tym miejscu następuje rozbicie na dwie alternatywe ścieżki wykonania:



Rysunek 11: Timeline scenariusza "Myślał indyk o niedzieli", kiedy w momencie 1. zostanie wykonane $\{SHOOT, THINK\_ABOUT\_SUNDAY\}$



Rysunek 12: Timeline wykonania scenariusza "Myślał indyk o niedzieli", kiedy w momencie 1. zostanie wykonane $\{SHOOT, ALARM\}$. Scenariusz nie wykona się, urwie się na chwili 2.

Zwróćmy szczególnie uwagę na drugą możliwość wykonania scenariusza. W momencie 2., indyk biega. Nie są spełnione warunki wstępne akcji $SHOOT$, zatem nie będzie ona mogła być wykonana. Model się urywa, nie można wykonać niepustego podzbioru akcji złożonej tutaj składającej się wyłącznie z $SHOOT$.

Z powyższych rozważań odpowiedź na kwerendę **possibly executable** Sc będzie pozytywna, bo istnieje ścieżka wykonania tego scenariusza. Odpowiedź na kwerendę **necessarily executable** Sc będzie negatywna, bo druga ścieżka wykonania nie będzie mogła być spełniona.

Odpowiedź na kwerendę **possibly accessible** $ALIVE$ **at** 3 **when** Sc będzie negatywna, bo w żadnej ścieżce wykonania nie dojdziemy do takiego stanu. Odpowiedź na kwerendę **necessarily accessible** $ALIVE$ **at** 3 **when** Sc będzie oczywiście negatywna.