

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, 23 MARCA 2014

Spis treści

1. Opis zadania	3
2. Opis języka akcji	3
2.1. Sygnatura języka	4
2.2. Opis domeny	4
2.3. Scenariusze działań	4
2.4. Semantyka	5
3. Opis języka kwerend	6
4. Przykłady	7

1 Opis zadania

Zadaniem projektu jest opracowanie i zaimplementowanie:

- języka akcji pewnej klasy systemów dynamicznych,
- język 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 znajduje się również rozdział 'Algorytmy wnioskowania', w którym zostały opisane algorytmy, które zostaną zaimplementowane. W ostatnim rozdziale znajdują się przykłady. Pokazują one konkretne przypadki użycia oraz oczekiwane wyniki działania programu.

2 Opis języka akcji

Język akcji zaprojektowany na potrzeby zadania, musi spełniać 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,
 - (b) dynamiczne.
9. Akcje mogą być niewykonalne.
10. Stany opisywane częściowo (obserwacje). (TODO wyjaśnić)
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

$\psi = (F, Ac, \mathbb{N})$

gdzie:

F – zbiór zmiennych inercji (fluentów)

Ac – zbiór akcji

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

2.2 Opis domeny

Rodzaje zdań występujących w projektowanym języku (domena języka):

Oznaczenia:

f – fluent

$Ac_i, Ac_j \in Ac$

$\pi \in Forms(F)$

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

- initially α
Określa stan początkowy fluentów w formule α .
- (Ac_i, d_i) causes α if π
Akcja Ac_i trwająca d_i chwil powoduje stan α , jeśli zachodzi warunek π .
- (Ac_i, d_i) invokes (Ac_j, d_j) after d if π
Akcja Ac_i trwająca d_i chwil powoduje wykonanie akcji Ac_j trwającej d_j chwil po d chwilach od zakończenia akcji Ac_i , jeśli zachodzi warunek π .
- (Ac_i, d_i) releases f if π
Akcja Ac_i trwająca d_i chwil powoduje uwolnienie f po zakończeniu akcji Ac_i , jeśli zachodzi warunek π .
- π triggers (Ac_i, d_i)
Akcja Ac_i trwająca d_i chwil jest wykonywana, jeśli zajdzie warunek π .

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:
 $m \geq 0$ – obserwacje, gdzie każda obserwacja jest stanem częściowym (stanem spełniającym warunek γ w pewnym punkcie czasu t).
 γ – zbiór (np. $x_1 = True, x_2 = True, x_3 = False$).
- $ACS = \{((Ac_1, d_1), t_1), \dots, ((Ac_n, d_n), t_n)\}$, gdzie:
 $n \geq 1$,
 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 $S = (H, O, E)$ taki, że:*

- $H : F \times \mathbb{N} \longrightarrow \{0, 1\}$ jest funkcją historii, pozwala ona stwierdzić, jaki stan ma pewny fluent w danej chwili czasu.
- $O : Ac \times \mathbb{N} \longrightarrow 2^F$ jest funkcją okluzji. Dla pewnej ustalonej akcji A i chwili czasu $t \in \mathbb{N}$ funkcja $O(A, t)$ zwraca zbiór fluentów, na który akcja A ma wpływ, jeśli zostanie wykonana od czasu $t-1$ do t .
- $E \subseteq Ac \times \mathbb{N}$ jest relacją wykonań akcji. Para (A, t) należy do relacji E jeśli akcja A jest wykonana w czasie t . W naszym modelu zakładamy warunek sekwencyjności działań. Oznacza on, że tylko jedną akcję możemy wykonać w danym czasie tak, więc jeśli $(A, t) \in E$ oraz $(B, t) \in E$, to $A = B$.

Niech: A, B będą akcjami, f - fluentem, α, π - literałami, d - liczbą naturalną oraz $fl(\alpha)$ będzie zbiorem fluentów występujących w α . Wtedy dla zdań języka AL muszą być spełnione następujące warunki:

- Dla każdego wyrażenia $(A \text{ causes } \alpha \text{ if } \pi) \in D$ i dla każdego momentu w czasie $t \in \mathbb{N}$, jeżeli $H(\pi, t) = 1$ oraz $(A, t) \in E$, wtedy $H(\alpha, t+1) = 1$ i $fl(\alpha) \subseteq O(A, t+1)$.
- Dla każdego wyrażenia $(A \text{ release } f \text{ if } \pi) \in D$ i dla każdego momentu czasu $t \in \mathbb{N}$, jeżeli $H(\pi, t) = 1$ oraz $(A, t) \in E$, wtedy $f \in O(A, t+1)$.
- Dla każdego wyrażenia $(\pi \text{ triggers } A) \in D$ i dla każdego momentu czasu $t \in \mathbb{N}$, jeżeli $H(\pi, t) = 1$, wtedy $(A, t) \in E$.
- Dla każdego wyrażenia $(A \text{ invokes } B \text{ after } d \text{ if } \pi) \in D$ i dla każdego momentu czasu $t \in \mathbb{N}$, jeżeli $H(\pi, t) = 1$ oraz $(A, t) \in E$, wtedy $(B, t+d+1) \in E$.

Definicja 2.2. *Niech $S = (H, O, E)$ będzie strukturą języka AL, $Sc = (OBS, ACS)$ będzie scenariuszem, oraz D domeną. Powiem, że S jest strukturą dla Sc zgodnym z opisem domeny D jeśli:*

- Dla każdej obserwacji $(\alpha, t) \in OBS$, $H(\alpha, t) = 1$
- $ACS \subseteq E$

Definicja 2.3. *Niech $O_1, O_2 : X \longrightarrow 2^Y$, 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 $S = (H, O, E)$ będzie strukturą dla scenariusza $Sc = (OBS, ACS)$ zgodną z opisem domeny D . Mówimy, że S jest O -minimalną strukturą, jeżeli nie istnieje struktura $S' = (H', O', E')$ dla tego samego scenariusza i domeny taka, że $O' \prec O$.*

Definicja 2.5. *Niech $S = (H, O, E)$ 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 $t \in \mathbb{N}$, $f \in F : H(f, t) \neq H(f, t+1) \subseteq O(A, t+1)$ dla pewnej akcji $A \in Ac$.

- 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 domeną 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 γ zachodzi zawsze/kiedykolwiek?

- *always/ever γ at t when Sc*
Oznacza, że zawsze/kiedykolwiek w chwili t realizacji scenariusza Sc zachodzi warunek γ .
- *always/ever γ when Sc*
Oznacza, że zawsze/kiedykolwiek w pewnej chwili t realizacji scenariusza Sc zachodzi warunek γ .

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

- *performing A at t when Sc*
Oznacza, że zawsze w chwili t realizacji scenariusza Sc zachodzi akcja A .
- *performing A when Sc*
Oznacza, że zawsze w pewnej chwili t realizacji scenariusza Sc zachodzi akcja A .
- *performing at t when Sc*
Oznacza, że zawsze w chwili t realizacji scenariusza Sc zachodzi pewna akcja A .

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

- *always/ever accessible γ when Sc*
Oznacza, że cel γ jest osiągalny zawsze/kiedykolwiek przy zadanym zbiorze obserwacji *OBS* przy realizacji scenariusza Sc .

Semantyka kwerend w języku

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

- zapytanie kwerendą Q postaci *γ at t when Sc*
zwróci wynik *TRUE* jeśli dla każdego modelu $S = (H, O, E)$ scenariusza Sc zgodnego z D zajdzie $H(\gamma, t) = 1$
- zapytanie kwerendą Q postaci *γ when Sc*
zwróci wynik *TRUE* jeśli dla każdego modelu $S = (H, O, E)$ scenariusza Sc zgodnego z D zajdzie $\exists_{t \in \mathbb{N}} H(\gamma, t) = 1$

- 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 zajdzie $(A, t) \in E$
- zapytanie kwerendą Q postaci *performing A when Sc*
zwróci wynik *TRUE* jeśli dla każdego modelu $S = (H, O, E)$ scenariusza Sc zgodnego z D zajdzie $\exists_{t \in N} (A, t) \in E$
- 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 $\exists_{A \in Ac} (A, t) \in E$
- zapytanie kwerendą Q postaci *accessible γ when Sc*
zwróci wynik *TRUE* jeśli dla każdego modelu $S = (H, O, E)$ scenariusza Sc zgodnego z D zajdzie $\exists_{t \in N} \exists_{A \in Ac} \gamma \in O(A, t)$

jeśli warunek nie zajdzie program zwróci wartość *FALSE*.

4 Przykłady

1. Pytanie czy scenariusz może wystąpić

- (a) Historia
- (b) Opis akcji
- (c) Scenariusz
- (d) Kwerendy
- (e) Analiza

2. Pytanie czy dany warunek zachodzi w danym czasie

- (a) Historia
- (b) Opis akcji
- (c) Scenariusz
- (d) Kwerendy
- (e) Analiza

3. Pytanie czy dana akcja jest wykonywana w danym czasie

- (a) Historia
- (b) Opis akcji
- (c) Scenariusz
- (d) Kwerendy
- (e) Analiza

4. Pytanie czy cel jest osiągalny

- (a) Historia
- (b) Opis akcji
- (c) Scenariusz
- (d) Kwerendy
- (e) Analiza

5. Brak integralności Przykład *Brak integralności* pokazuje scenariusz, który mimo zgodności z warunkami zadania, jest sprzeczny z logiką *common sense* (z powodu braku warunków integralności).

- (a) Historia
Mamy Billa oraz komputer. Bill może nacisnąć przycisk *Włącz* lub odłączyć komputer od zasilania. Komputer jest wyłączony i podłączony do zasilania. Jeżeli zostanie naciśnięty jego przycisk *Włącz*, to komputer włącza się.
- (b) Opis akcji
initially $\neg on_computer$ and $connects_power_computer$ **and** $\neg swithing_on_computer$
 $(click_button_on, 1)$ **causes** $switching_on_computer$
 $(click_button_on, 1)$ **invokes** $(switch_on_computer, 2)$ **after** 1
 $(switch_on_computer, 1)$ **causes** $on_computer$
 $(disconnect_power, 1)$ **causes** $on_computer$ **and** $\neg swithing_on_computer$
- (c) Scenariusz
 $Sc = (OBS, ACS)$
 $OBS = \emptyset$
 $ACS = (click_button_on, 0 + 1), (disconnect_power,), 3 + 1), (click_button_on, 4 + 1)$
- (d) Kwerendy

- i. *swithing_on_computer* **at** $6 + 2$ **when** *Sc*
 - ii. *swithing_on_computer* **and** $\neg on_computer$ **at** $6 + 2$ **when** *S***c**
- (e) Analiza
- Powyższy scenariusz jest prawidłowy, lecz zawiera pewną niezgodność. W chwili $t = 4 + 1$ komputer zostaje odcięty od zasilania. Powinien więc wyłączyć się. Bill chwili $t = 5 + 1$ naciska przycisk *Włącz*. Komputer zacznie włączać się mimo iż jest odcięty od zasilania. Zachodzą dwa sprzeczne ze sobą stany, tj. *swithing_on_computer* = *T* i *on_computer* = *T*. Odpowiedzi na powyższe kwerendy będą odpowiednio: 1.*True* i 2.*False*. Należy zaznaczyć, że odpowiedzi zgodnie z logiką *commonsense* powinny być sobie równe.

[!h]								
		click_button	switching_on_computer	disconnet_power	click_button	switching_on_computer		
0	1	2	3	4	5	6	7	8
on_computer	F	F	F	F	-F	?F	?F	?F
connects_power_computer	T	T	T	T	-T	-T	-T	-T
switching_on_computer	G	G	-G	-G	G	G	G	G
okluzja	{}	{}	{}	{}	{}	{}	{}	{}