Logiki temporalne

Andrzej Oszer

Seminarium Protokoły Komunikacyjne

Spis treści

Wprowadzenie

Logiki temporalne rozszerzają logikę pierwszego rzędu o symbole określające upływ czasu.

Własność modelu wyrażamy za pomocą:

atomowych zdarzeń „$n > 0$”, „$sekcja\_krytyczna_1$”, itd.

operatorów logicznych $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$, $\neg$, *prawda*, *fasz*

operatorów temporalnych X, F, G, U, W, A, G, $\overset{\infty}{G}$, $\overset{\infty}{F}$

Wprowadzenie

Logiki temporalne rozszerzają logikę pierwszego rzędu o symbole określające upływ czasu.
Własność modelu wyrażamy za pomocą:

atomowych zdarzeń „$n > 0$”, „$sekcja\_krytyczna_1$”, itd.

operatorów logicznych $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$, $\neg$, *prawda*, *fasz*

operatorów temporalnych X, F, G, U, W, A, G, $\overset{\infty}{G}$, $\overset{\infty}{F}$

Wprowadzenie

Logiki temporalne rozszerzają logikę pierwszego rzędu o symbole określające upływ czasu.
Własność modelu wyrażamy za pomocą:

atomowych zdarzeń „$n > 0$”, „$sekcja\_krytyczna_1$”, itd.

operatorów logicznych $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$, $\neg$, *prawda*, *fasz*

operatorów temporalnych X, F, G, U, W, A, G, $\overset{\infty}{G}$, $\overset{\infty}{F}$

Wprowadzenie

Logiki temporalne rozszerzają logikę pierwszego rzędu o symbole określające upływ czasu.
Własność modelu wyrażamy za pomocą:

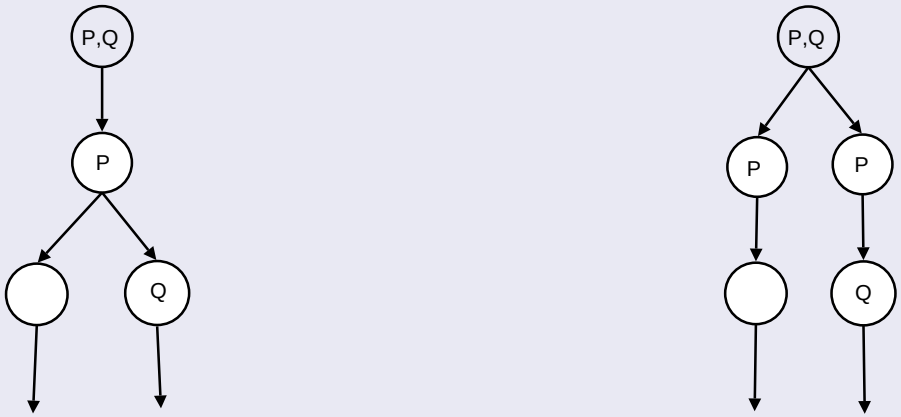
atomowych zdarzeń „$n > 0$”, „$sekcja\_krytyczna_1$”, itd.

operatorów logicznych $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$, $\neg$, *prawda*, *fasz*

operatorów temporalnych X, F, G, U, W, A, G, $\overset{\infty}{G}$, $\overset{\infty}{F}$

Propositional Linear Temporal Logic

Logika PLTL - bez rozgałęzień



Nierozróżnialne automaty w PLTL

Operatory temporalne

- X następny stan (np. $X\, x > y$)
- F przyszły stan (np. F *sekcja_krytyczna*)
- G wszystkie przyszłe stany (np. $G\neg(n < 5)$)

Operatory temporalne

- X następny stan (np. $X\, x > y$)
- F przyszły stan (np. $F\, sekcja\_krytyczna$)
- G wszystkie przyszłe stany (np. $G\neg(n < 5)$)

Operatory temporalne

- X następny stan (np. $X\, x > y$)
- F przyszły stan (np. $F\, sekcja\_krytyczna$)
- G wszystkie przyszłe stany (np. $G\neg(n < 5)$)

Operatory temporalne c.d.

U until (np. $(n > 0)\, U\, (g > 3)$)

W while not (np. $(n > 0)\, W\, (g > 3)$)
$(\phi_1 W \phi_2 = (\phi_1 U \phi_2) \vee G\phi_1)$

$\overset{\infty}{F}$ nieskończenie wiele przyszłych stanów
(np. $\overset{\infty}{F}\, (x > 0 \vee y > 0)$)
$(\overset{\infty}{F} = GF)$

$\overset{\infty}{G}$ cały czas od pewnego stanu (np. $\overset{\infty}{G}\, koniec$) $(\overset{\infty}{G} = FG)$

Operatory temporalne c.d.

U until (np. $(n > 0)\, U\, (g > 3)$)

W while not (np. $(n > 0)\, W\, (g > 3)$)
$(\phi_1 W \phi_2 = (\phi_1 U \phi_2) \vee G\phi_1)$

$\overset{\infty}{F}$ nieskończenie wiele przyszłych stanów
(np. $\overset{\infty}{F}\, (x > 0 \vee y > 0)$)
$(\overset{\infty}{F} = GF)$

$\overset{\infty}{G}$ cały czas od pewnego stanu (np. $\overset{\infty}{G}\, koniec$) $(\overset{\infty}{G} = FG)$

Operatory temporalne c.d.

U until (np. $(n > 0)\, U\, (g > 3)$)

W while not (np. $(n > 0)\, W\, (g > 3)$)
$(\phi_1 W \phi_2 = (\phi_1 U \phi_2) \vee G\phi_1)$

$\overset{\infty}{F}$ nieskończenie wiele przyszłych stanów
(np. $\overset{\infty}{F}\, (x > 0 \vee y > 0)$)
$(\overset{\infty}{F} = GF)$

$\overset{\infty}{G}$ cały czas od pewnego stanu (np. $\overset{\infty}{G}\, koniec$) $(\overset{\infty}{G} = FG)$

Operatory temporalne c.d.

U until (np. $(n > 0)\, U\, (g > 3)$)

W while not (np. $(n > 0)\, W\, (g > 3)$)
$(\phi_1 W \phi_2 = (\phi_1 U \phi_2) \vee G\phi_1)$

$\overset{\infty}{F}$ nieskończenie wiele przyszłych stanów
(np. $\overset{\infty}{F}\, (x > 0 \vee y > 0)$)
$(\overset{\infty}{F} = GF)$

$\overset{\infty}{G}$ cały czas od pewnego stanu (np. $\overset{\infty}{G}\, koniec$) $(\overset{\infty}{G} = FG)$

CTL - Computation Tree Logic

Computation Tree Logic - rozróżnianie ścieżek

W CTL każdy operator temporalny jest poprzedzony kwantyfikatorem temporalnym.

$\overset{\infty}{F}$ i $\overset{\infty}{G}$ nie można używać w CTL

Kwantyfikatory temporalne

- A we wszystkich ścieżkach (np. $AF\,koniec$)
- E istnieje ścieżka (np. $EG\,x > 0 \Rightarrow y > 0$)

Kwantyfikatory temporalne

- A we wszystkich ścieżkach (np. $AF\,koniec$)
- E istnieje ścieżka (np. $EG\,x > 0 \Rightarrow y > 0$)

CTL* - Computation Tree Logic

CTL* to abstrakcyjny nadzbiór logik temporalnych.

Możemy używać wszystkich operatorów temporalnych w dowolnej kolejności

Implementacje są ograniczone do konkretnych podzbiorów.

CTL* - Computation Tree Logic

CTL* to abstrakcyjny nadzbiór logik temporalnych.
Możemy używać wszystkich operatorów temporalnych w dowolnej kolejności
Implementacje są ograniczone do konkretnych podzbiorów.

Osiągalność

Osiągalność oznacza, że pewna sytuacja może zajść w modelu.

Osiągalność prosta:

- „może zajść, że $n < 0$”,
- „możemy wejść do sekcji krytycznej”

Osiągalność warunkowa:

- „możemy wejść do sekcji krytycznej bez przechodzenia przez $n = 0$”

Osiągalność

Osiągalność oznacza, że pewna sytuacja może zajść w modelu.
Osiągalność prosta:

- „może zajść, że $n < 0$”,
- „możemy wejść do sekcji krytycznej”

Osiągalność warunkowa:

- „możemy wejść do sekcji krytycznej bez przechodzenia przez $n = 0$”

Osiągalność

Osiągalność oznacza, że pewna sytuacja może zajść w modelu.
Osiągalność prosta:

- „może zajść, że $n < 0$”,
- „możemy wejść do sekcji krytycznej”

Osiągalność warunkowa:

- „możemy wejść do sekcji krytycznej bez przechodzenia przez $n = 0$”

Osiągalność w CTL

W CTL osiągalność prosta formuły ϕ to $EF\phi$,
czyli „istnieje taka ścieżka, że przechodzimy przez stan w którym ϕ”.

$EF\, n < 0$ „może zajść, że $n < 0$”

EF *sekcja_krytyczna* „możemy wejść do sekcji krytycznej”

Możemy też napisać:

$AG(EF\, n > 0)$ „zawsze możemy przejść do stanu z $n > 0$”

Osiągalność warunkowa może być zapisana przy użyciu U:

$E\,(n \neq 0)\, U$ *sekcja_krytyczna* „możemy wejść do sekcji krytycznej bez przechodzenia przez $n = 0$”

Osiągalność w CTL

W CTL osiągalność prosta formuły ϕ to $EF\phi$,
czyli „istnieje taka ścieżka, że przechodzimy przez stan w którym ϕ”.

$EF\, n < 0$ „może zajść, że $n < 0$”

$EF\, sekcja\_krytyczna$ „możemy wejść do sekcji krytycznej”

Możemy też napisać:

$AG(EF\, n > 0)$ „zawsze możemy przejść do stanu z $n > 0$”

Osiągalność warunkowa może być zapisana przy użyciu U:

$E\,(n \neq 0)\, U\, sekcja\_krytyczna$ „możemy wejść do sekcji krytycznej bez przechodzenia przez $n = 0$”

Osiągalność w CTL

W CTL osiągalność prosta formuły ϕ to $EF\phi$,
czyli „istnieje taka ścieżka, że przechodzimy przez stan w którym ϕ”.

$EF\, n < 0$ „może zajść, że $n < 0$”

EF *sekcja_krytyczna* „możemy wejść do sekcji krytycznej”

Możemy też napisać:

$AG(EF\, n > 0)$ „zawsze możemy przejść do stanu z $n > 0$”

Osiągalność warunkowa może być zapisana przy użyciu U:

$E\,(n \neq 0)\, U$ *sekcja_krytyczna* „możemy wejść do sekcji krytycznej bez przechodzenia przez $n = 0$”

Osiągalność w PLTL

W PLTL nie da się wyrazić większości własności osiągalności.
Możemy tylko wyrazić jej negację.

Bezpieczeństwo

Własność bezpieczeństwa oznacza, że pewna sytuacja nigdy nie zajdzie.

Bezpieczeństwo w CTL

$AG\neg(sekcja\_krytyczna_1 \wedge sekcja\_krytyczna_2)$ dwa procesy nie będą naraz w sekcji krytycznej

$AG\neg memory\_overflow$ nie będzie przepełnienia

Bezpieczeństwo z warunkami

Przykład

„Samochód nie ruszy bez kierowcy”

$A\neg samochod\_rusza\, W\, kierowca\_w\_srodku$ CTL

$\neg samochod\_rusza W\, kierowca\_w\_srodku$ PLTL

Bezpieczeństwo z warunkami

Przykład

„Samochód nie ruszy bez kierowcy”

$A\neg samochod\_rusza\, W\, kierowca\_w\_srodku$ CTL

$\neg samochod\_rusza W\, kierowca\_w\_srodku$ PLTL

Żywotność

Własność żywotności oznacza, że pewna sytuacja kiedyś zajdzie

Żywotność w CTL*

Przykład

„Każde żądanie będzie w końcu spełnione”:

W CLT:

$AG(request \Rightarrow AF\, satisfy)$

W PLTL:

$G(request \Rightarrow F\, satisfy)$

Przykład

„zawsze możemy wrócić do inicjalizacji”

W CTL:

$AGEF\, init$

Żywotność w CTL*

Przykład

„Każde żądanie będzie w końcu spełnione”:

W CLT:

$AG(request \Rightarrow AF\, satisfy)$

W PLTL:

$G(request \Rightarrow F\, satisfy)$

Przykład

„zawsze możemy wrócić do inicjalizacji”

W CTL:

$AGEF\, init$

Żywotność w CTL*

Przykład

„Każde żądanie będzie w końcu spełnione”:
W CLT:
$AG(request \Rightarrow AF\, satisfy)$
W PLTL:
$G(request \Rightarrow F\, satisfy)$

Przykład

„zawsze możemy wrócić do inicjalizacji”
W CTL:
$AGEF\, init$

Żywotność w CTL*

Przykład

„Każde żądanie będzie w końcu spełnione”:
W CLT:
$AG(request \Rightarrow AF\, satisfy)$
W PLTL:
$G(request \Rightarrow F\, satisfy)$

Przykład

„zawsze możemy wrócić do inicjalizacji”
W CTL:
$AGEF\, init$

Żywotność w CTL*

Przykład

„Każde żądanie będzie w końcu spełnione”:
W CLT:
$AG(request \Rightarrow AF\, satisfy)$
W PLTL:
$G(request \Rightarrow F\, satisfy)$

Przykład

„zawsze możemy wrócić do inicjalizacji”
W CTL:
$AGEF\, init$

Brak zakleszczeń

AGEX true zawsze można wykonać kolejny krok

Sprawiedliwość

Sprawiedliwość oznacza, że pewna sytuacja zdarza się nieskończenie często

Przykład

„jeżeli żądanie dostępu będzie zgłaszane nieskończenie wiele razy, to dostęp zostanie udzielony nieskończenie wiele razy”

Sprawiedliwość

Sprawiedliwość oznacza, że pewna sytuacja zdarza się nieskończenie często

Przykład

„jeżeli żądanie dostępu będzie zgłaszane nieskończenie wiele razy, to dostęp zostanie udzielony nieskończenie wiele razy”

Sprawiedliwość w CTL*

$A(\overset{\infty}{F}\, access\_requested \Rightarrow \overset{\infty}{F}\, access\_granted)$, albo:

$A(\overset{\infty}{F}\, access\_granted \vee \overset{\infty}{G} \neg access\_requested)$

Rozszerzenie CTL o $\overset{\infty}{F}$ i $\overset{\infty}{G}$- $ECTL^{+}$ i $FCTL$.

Sprawiedliwość w CTL*

$A(\overset{\infty}{F}\, access\_requested \Rightarrow \overset{\infty}{F}\, access\_granted)$, albo:

$A(\overset{\infty}{F}\, access\_granted \vee \overset{\infty}{G} \neg access\_requested)$

Rozszerzenie CTL o $\overset{\infty}{F}$ i $\overset{\infty}{G}$- $ECTL^{+}$ i $FCTL$.

Sprawiedliwość w CTL*

$A(\overset{\infty}{F}\, access\_requested \Rightarrow \overset{\infty}{F}\, access\_granted)$, albo:

$A(\overset{\infty}{F}\, access\_granted \vee \overset{\infty}{G} \neg access\_requested)$

Rozszerzenie CTL o $\overset{\infty}{F}$ i $\overset{\infty}{G}$- $ECTL^{+}$ i $FCTL$.

Bibliografia

1. B.Berard, M.Bidoit, A.Finkel, F.Larousssinie, A.Petit, L.Petrucci, Ph. Schnoebelen i P.McKenzie „Systems and Software Verification”