

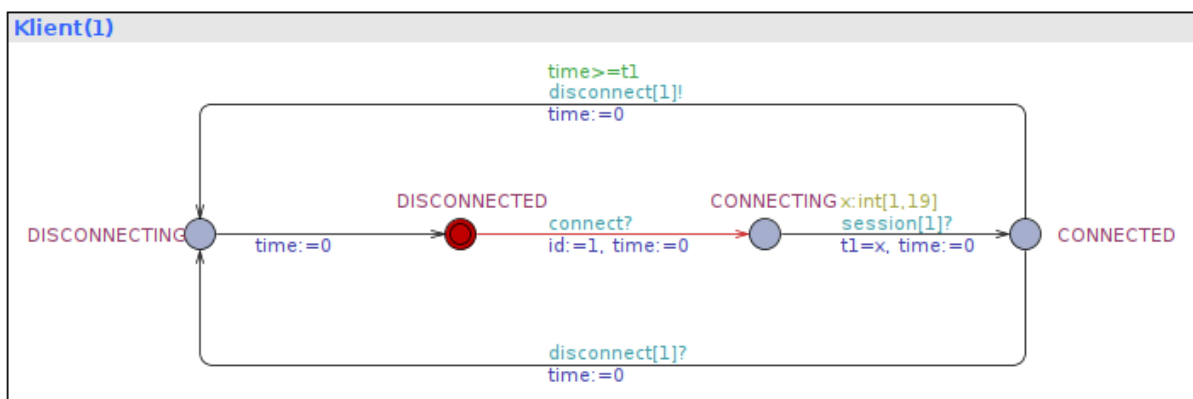
MiASI laboratorium 13

Konstrukcja i weryfikacja zsynchronizowanych czasowych
automatów UPPAAL (cz. 2)

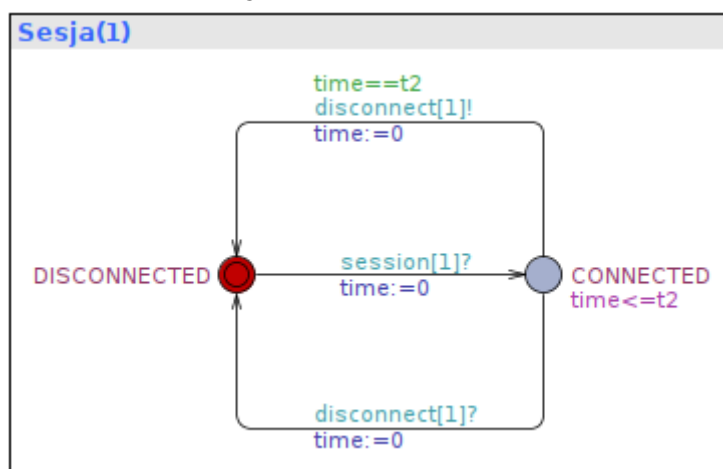
Prowadzący: dr inż. Paweł Głuchowski

Termin zajęć: Pn 17:05

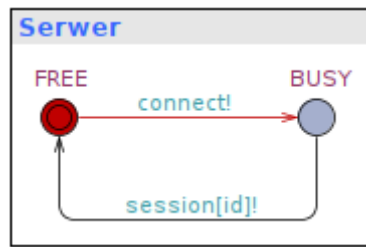
Zad1.



rys. 1 Automat Klient



rys. 2 Automat Sesja



rys. 3 Automat Server

Zad 2.

(czy serwer obsługuje tylko 1 klienta na raz)

UPPAAL: $A[] \text{ forall } (i:\text{int}[1,3]) \text{ forall } (j:\text{int}[1,3]) (Klient(i).CONNECTING \text{ and } Klient(j).CONNECTING) \text{ imply } i=j$

CTL: $AG Klient(i).CONNECTING \wedge Klient(j).CONNECTING \Rightarrow i=j$

Opis: Sprawdza każdego klienta z jednej tablicy z każdym z drugiej, jeżeli obaj są w trakcie łączenia się to znaczy, że to jest ten sam klient

Wynik: Prawda

UPPAAL: $E<> \text{ forall } (i:\text{int}[1,3]) Klient(i).CONNECTING$

CTL: $EF Klient(i).CONNECTING$

Opis: Czy wszyscy klienci mogą łączyć się w tym samym czasie

Wynik: Fałsz

(czy połączenie klient-sesja jest możliwe dla wszystkich klientów i sesji na raz)

UPPAAL: $E<> \text{ forall } (i:\text{int}[1,3]) Klient(i).CONNECTED \text{ and } Sesja(i).CONNECTED$

CTL: $EF Klient(i).CONNECTED \wedge Sesja(i).CONNECTED$

Opis: Czy wszyscy klienci i sesje mogą być jednocześnie połączeni

Wynik: Prawda

(czy połączenie klient-sesja trwa od t1 do t2 i tylko tyle)

UPPAAL: $E<> \text{ forall } (i:\text{int}[1,3]) Klient(i).CONNECTED \text{ and } Sesja(i).CONNECTED \text{ and } Klient(i).time > 20$

CTL: $EF Klient(i).CONNECTED \wedge Sesja(i).CONNECTED \wedge Klient(i).time \geq 20$

Opis: Czy jakiś klient lub sesja może być połączony po czasie t2 (20s)

Wynik: Fałsz

UPPAAL: $E<> \text{ exists } (i:\text{int}[1,3]) Klient(i).CONNECTED \text{ and } Klient(i).t1 = 20$

CTL: $EF Klient(i).CONNECTED \wedge Klient(i).t1 = 20$

Opis: Czy jakiś klient gdzie połączenie z sesją trwa 20 sekund

Wynik: Fałsz

UPPAAL: Klient(1).CONNECTED --> Klient(1).DISCONNECTING

CTL: AG Klient(1).CONNECTED => Klient(1). DISCONNECTING

Opis: Czy połączenie kiedyś się zakończy

Wynik: Prawda

UPPAAL: Klient(1).CONNECTED --> (Klient(1).DISCONNECTED and (Klient(1).time < Klient(1).t1))

CTL: AG Klient(1).CONNECTED => Klient(1). DISCONNECTED \wedge (Klient(i).time < Klient(i).t1)

Opis: Czy klient może przejść do stanu DISCONNECTED przed t1.

Wynik: Fałsz