Automaty skończone i weryfikacja programów współbieżnych

Maria Donten i Piotr Hofman

Automat

- etykiety (zbiór skończony),
- lokacje (zbiór skończony),
- wybrana lokacja początkowa,
- zmienne całkowitoliczbowe (zbiór skończony, istnieje ograniczenie na wartości przyjmowane przez zmienne),
- zegary (zbiór skończony, wartości z $\mathbb{R}_+$),
- relacja przejścia,
- funkcja przypisująca lokacjom warunki na zegary, które muszą być spełnione, żeby można było przebywać w lokacjach.

Przejście

- lokacja, z której wychodzimy,
- etykieta przejścia,
- warunek, który muszą spełniać zmienne, żeby można było użyć tego przejścia,
- warunek, który muszą spełniać zegary, żeby można było użyć tego przejścia,
- (ewentualnie) atrybut pilności przejścia,
- akcja — działania na zmiennych wykonywane przy tym przejściu,
- przypisania zegarów wykonywane przy tym przejściu,
- lokacja, do której wchodzimy.

(Czasami tylko przejścia synchronizujące mogą być pilne — UppAal.)

Semantyka — system tranzycyjny

Stan automatu: lokacja, wartościowanie zmiennych i wartościowanie zegarów. Jest wyróżniony stan początkowy.

Tranzycje między stanami: akcje operują na zmiennych tak, że z wartościowań stanu wejściowego powstały wartościowania stanu wyjściowego i żeby były spełnione warunki przejścia w automacie i warunek przebywania w lokacji stanu wyjściowego.

Tranzycje są etykietowane etykietami przejść automatu (zmiana lokacji) lub liczbami rzeczywistymi dodatnimi (pozostanie w pewnej lokacji przez pewien czas).

Synchronizacja automatów w sieci

Problem: Chcielibyśmy, żeby automaty obrazowały działanie programów współbieżnych. Wobec tego definiujemy automat, który jest siecią mniejszych automatów (n automatów, $\mathcal{A}_1, \ldots, \mathcal{A}_n$). O automatach $\mathcal{A}_i$ zakładamy, że mają rozłączne zbiory zegarów.

Musimy umieć wyrazić zależności między procesami, komunikację.

Etykiety

Synchronizacja odbywa się za pomocą etykiet. Dodajemy nowe możliwości etykietom w automatach $\mathcal{A}_i$.

Etykiety mają trzy warianty:

1. z wykrzyknikiem na końcu: $send\_data!$ — przesłanie wiadomości (output);

2. ze znakiem zapytania na końcu $send\_data?$ — odebranie wiadomości (input);

3. bez żadnego z tych znaków — zwykłe przejście.

Założenia dotyczące etykiet

Jeśli w którymś automacie $\mathcal{A}_i$ pojawia się etykieta l (trzeciego rodzaju), to w żadnym automacie tej sieci nie pojawiają się $l!$ ani $l?$. Etykieta l może wystąpić tylko w jednym automacie.

Jeśli w którymś automacie $\mathcal{A}_i$ pojawia się (być może wielokrotnie) etykieta $l!$, to nie może ona wystąpić w żadnym innym automacie, za to etykieta $l?$ musi gdzieś wystąpić, ale nie w tym automacie, w którym znajduje się $l!$.

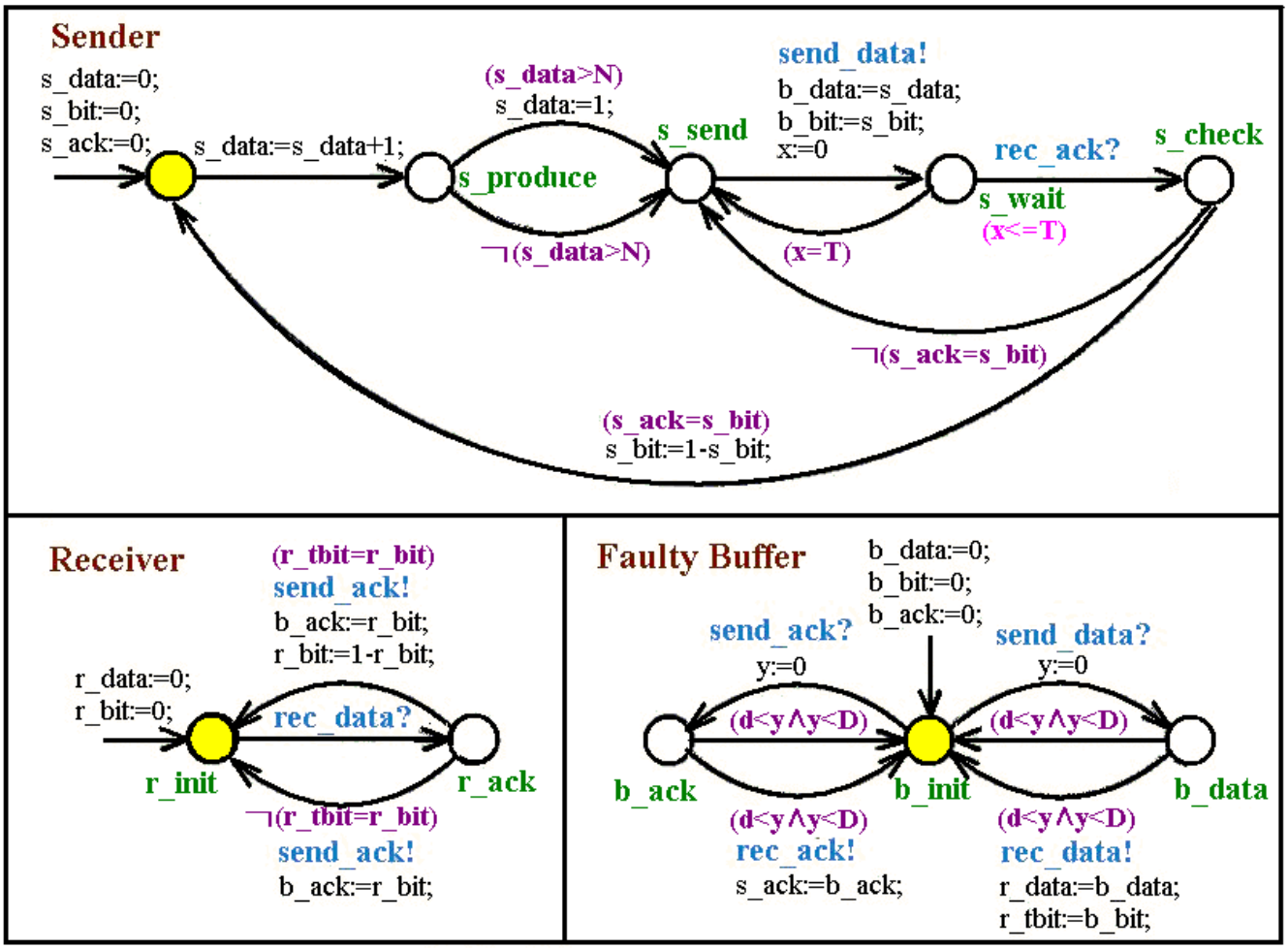
Tak samo, jeśli pojawia się $l?$, to musi wystąpić $l!$, ale l nie może.

Ponadto zakładamy, że w automatach $\mathcal{A}_i$ akcje przy przejściach z etykietą $l?$ nie modyfikują zmiennych dzielonych. Jest to potrzebne, aby nie musieć w żaden sposób uzależniać rezultatów od kolejności wykonania akcji przejść z etykietą $l?$. Akcja przejścia z $l!$ wykonuje się przed akcjami przejść z $l?$.

$\mathcal{A}$ — sieć automatów $\mathcal{A}_i$

- etykiety — suma zbiorów etykiet automatów $\mathcal{A}_i$,
- lokacje to n-tki lokacji automatów $\mathcal{A}_i$,
- lokacja początkowa — n-tka złożona z lokacji początkowych $\mathcal{A}_i$,
- zbiór zmiennych — suma zbiorów zmiennych dla poszczególnych $\mathcal{A}_i$,
- zbiór zegarów — suma zbiorów zegarów dla poszczególnych $\mathcal{A}_i$ (założyliśmy, że te zbiory są rozłączne),
- relacja przejścia,
- funkcja warunków możliwości przebywania w lokacjach — w każdej lokacji wartość funkcji to warunek będący koniunkcją wartości funkcji w odpowiednich lokacjach automatów $\mathcal{A}_i$ (po prostu chcemy, żeby dla każdego automatu składowego z osobna jego warunek był spełniony).

Przykład — Alternating Bit Protocol



s_init nazwa lokacji

(r_tbit=r_bit) warunek przejścia

send_data! etykieta

(x<=T) warunek możliwości pozostania w miejscu

b_data:=s_data;
b_bit:=s_bit;
x:=0
przypisania związane z przejściem

Przejście w $\mathcal{A}$ — idea

Przejście w $\mathcal{A}$ ma obrazować równoległe wykonanie kilku przejść w pewnej grupie automatów $\mathcal{A}_i$ (trzeba ustalić kolejność wykonywania akcji!). Być może zmiana lokacji następuje w tylko jednym $\mathcal{A}_i$, czyli w tym momencie nie ma synchronizacji automatów.

Przejście synchronizujące

Jeśli w automatach $\mathcal{A}_i$ mamy etykiety $l!$ i $l?$, to opis przejścia w $\mathcal{A}$ dla etykiety o treści l, wychodzącego z lokacji $(q_1, \ldots, q_n)$ jest następujący:

- jeśli $\mathcal{L}(l)$ jest zbiorem tych i, dla których z lokacji q_i wychodzi przejście z etykietą $l!$ lub $l?$, to docelowa lokacja definiowanego przejścia w $\mathcal{A}$ będzie się różniła od wyjściowej dokładnie na zbiorze indeksów $\mathcal{L}(l)$;

- na miejscach indeksowanych liczbami z $\mathcal{L}(l)$ będą się znajdować lokacje docelowe przejść z automatów składowych etykietowanych $l!$ lub $l?$;

- akcja przejścia to złożenie odpowiednich akcji automatów $\mathcal{A}_i$, w następującej kolejności: najpierw wykonuje się akcja odpowiadająca etykiecie $l!$, a później akcje odpowiadające $l?$ w kolejności dowolnej, bo założyliśmy, że nie modyfikują zmiennych dzielonych;

- z modyfikacją zegarów nie ma problemu, bo zbiory zegarów $\mathcal{A}_i$ są rozłączne.

Przejście niesynchronizujące

Jeśli natomiast w automatach $\mathcal{A}_i$ występuje etykieta l (bez znaków specjalnych), to przejście w $\mathcal{A}$ dla etykiety l definiuje się tak:

- lokacja wyjściowa $(q_1, \ldots, q_n)$ różni się od docelowej tylko dla jednego indeksu i - tego, dla którego w $\mathcal{A}_i$ z q_i wychodzi przejście z etykietą l;

- ponieważ zmiana lokacji następuje tylko w jednym A_i, to nie ma problemu z szeregowaniem akcji.

Zmienne zdaniowe

W celach weryfikacji dla sieci automatów $\mathcal{A} = \mathcal{T}(\mathcal{A}_i)$ wprowadzamy następujące zmienne zdaniowe:

- $\mathcal{A}_i.l$ — będzie ona oznaczać, że w pewnym stanie automat $\mathcal{A}_i$ jest w lokacji l,

- $e_1 \sim e_2$, gdzie e_1 i e_2 są wyrażeniami, a $\sim$ to któraś z dopuszczonych relacji porównania ich wartości, przy jakimś wartościowaniu będzie oznaczać, że wartości e_1 i e_2 po podstawieniu określonym przez to wartościowanie spełniają relację $\sim$.

Ponieważ stan automatu $\mathcal{A}$ to trójka: lokacja (czyli n-tka lokacji automatów $\mathcal{A}_i$) oraz wartościowania zmiennych i zegarów, to każdemu stanowi możemy przyporządkować zbiór zmiennych zdaniowych spełnionych w tym stanie.

Model dla automatu $\mathcal{A}$

Jeśli automat $\mathcal{A}$ jest siecią automatów $\mathcal{A}_i$, to para:

- system tranzycyjny będący semantyką automatu $\mathcal{A}$,
- opisane powyżej przyporządkowanie stanom zbiorów zmiennych zdaniowych prawdziwych w tych stanach

to *model* dla automatu $\mathcal{A}$.

Co będziemy weryfikować?

Ogólnie, chcemy umieć sprawdzić, czy przy jakimś wykonaniu programu znajdziemy się w pewnej określonej przez nas sytuacji, zazwyczaj niepożądanej dla poprawnego wykonania programu.

Ta sytuacja ma być opisana za pomocą formuły logicznej zawierającej wprowadzone wcześniej zmienne zdaniowe $\mathcal{A}_i.l$ i $e_1 \sim e_2$.

Dla kolejnych $k \in \mathbb{N}$ będziemy sprawdzać, czy w semantyce automatu $\mathcal{A}$, w którymś stanie na ścieżce o długości k, zaczynającej się w stanie początkowym, ta formuła będzie spełniona.

Jak to się robi?

1. Wprowadza się nowe zmienne zdaniowe, pozwalające tworzyć formuły oznaczające przechodzenie po ścieżkach i bycie w konkretnych stanach.

2. Dla kolejnych $k \in \mathbb{N}$, za pomocą tych zmiennych zapisuje się formułę mówiącą, że w którymś stanie na ścieżce o długości k, zaczynającej się w stanie początkowym, będzie spełniona ta własność, której osiągalność chcemy sprawdzić.

3. *SAT-solver* sprawdza spełnialność tej formuły dla kolejnych $k \in \mathbb{N}$.

4. Jeśli dla pewnego k znajdzie wartościowanie, dla którego ona jest spełniona, to znaczy, że testowana własność jest osiągalna.

5. W przeciwnym przypadku k zostaje zwiększone.

Kiedy zakończyć?

Sprawdzanie kończy się, gdy zachodzi jeden z warunków:

- znaleziono wartościowanie, przy którym formuła dla pewnego k jest spełniona,
- k osiągnie wartość średnicy systemu tranzycyjnego (czyli maksymalną długość najkrótszej ze ścieżek łączących dwa stany),
- k jest na tyle duże, że sprawdzenie spełnialności formuły przekracza możliwości *SAT-solvera*.

Problemy

Tranzycji jest zazwyczaj nieprzeliczalnie wiele, ze względu na to, że mogą być etykietowane wartościami z pewnych przedziałów w $\mathbb{R}_+$, gdy oznaczają pozostanie przez pewien czas w jakiejś lokacji.

Wobec tego przeprowadza się dyskretyzację modelu sieci $\mathcal{A}$.