Programowanie współbieżne i rozproszone

WYKŁAD 9

Poprawność programów sekwencyjnych

Poprawność programu sekwencyjnego wyrażana jest za pomocą zdania o postaci:

$$\{p\}S\{q\}$$

gdzie:

- *S* jest programem,
- *p* i *q* są asercjami nazywanymi odpowiednio: warunkiem wstępnym i warunkiem końcowym.

Poprawność programów sekwencyjnych

- Warunek wstępny określa warunki jakie spełniają dane wejściowe.
- Warunek końcowy określa warunki jakie powinny spełniać wyniki działania programu.

Poprawność programów sekwencyjnych

Częściowa poprawność programu:

Każde kończące się wykonanie programu *S* z danymi wejściowymi spełniającymi asercję *p* kończy się z danymi wyjściowymi spełniającymi asercję *q*.

Całkowita (pełna) poprawność programu:

Każde wykonanie programu *S* z danymi wejściowymi spełniającymi asercję *p* kończy się a dane wyjściowe spełniają asercję *q*.

Poprawność programów sekwencyjnych

- W przypadku częściowej poprawności nie bierze się pod uwagę, czy obliczenia programu *S* kończą się, czy nie.
- Kończenie się obliczeń programu dla wszystkich prawidłowych danych wejściowych, tj. spełniających asercję *p*, nazywane jest warunkiem stopu.

Poprawność programów sekwencyjnych

Dowód pełnej poprawności programu sekwencyjnego:

- Udowodnienie, że program jest częściowo poprawny.
- Udowodnienie, że program spełnia warunek stopu.

Poprawność programów współbieżnych

Poprawność programu współbieżnego jest trudniejsza do udowodnienia niż programu sekwencyjnego

Poprawność programów współbieżnych

W przypadku programu współbieżnego należy wykazać dodatkowe własności:

- Własności bezpieczeństwa.
- Własność żywotności.

Poprawność programów współbieżnych

Własności bezpieczeństwa obejmują warunki, które powinny być zawsze spełnione, tj. dla wszystkich możliwych realizacji procesów współbieżnych.

Własności bezpieczeństwa:

- wzajemne wykluczanie,
- brak blokady/zakleszczenia.

Poprawność programów współbieżnych

Blokada – sytuacja, w której jeden lub większa liczba procesów współbieżnych się nie kończy.

Zakleszczenie – rodzaj blokady dotyczącej zbioru procesów. Zakleszczenie pojawia się wtedy, gdy każdy z procesów należących do danego zbioru procesów jest wstrzymany w oczekiwaniu na zdarzenie, które może być spowodowane tylko przez jakiś inny proces z tego zbioru.

Poprawność programów współbieżnych

Własność żywotności obejmuje warunki, które powinny być ostatecznie spełnione, tj. jeśli powinno zajść pewne zdarzenie, to dla każdej możliwej realizacji procesów współbieżnych w pewnym momencie ono rzeczywiście zachodzi.

Własność żywotności:

- brak zagłodzenia.

Poprawność programów współbieżnych

Z własnością żywotności związana jest tzw. uczciwość.

Rodzaje uczciwości:

- Uczciwość słaba: jeśli proces nieprzerwanie zgłasza żądanie, to kiedyś będzie ono obsłużone.
- Uczciwość mocna: jeśli proces zgłasza żądanie nieskończenie wiele razy, to kiedyś będzie ono obsłużone.

Dowodzenie poprawności za pomocą diagramów stanów

Zbiór stanów osiągalnych – zbiór zawierający te i tylko te stany, które mogą wystąpić w dowolnym obliczeniu programu współbieżnego.

Diagram stanów konstruuje się stopniowo, począwszy od stanu początkowego, analizując wszystkie możliwe kolejne stany. Jeśli otrzymany w ten sposób stan już występuje w diagramie, to nie jest on powtarzany.

Dowodzenie poprawności za pomocą diagramów stanów

Proces *A*	Proces *B*

pętla
{
  a1: sekcja lokalna
  a2: **semafor**.wait()
  a3: sekcja krytyczna
  a4: **semafor**.signal()
}
	pętla
	{
	  b1: sekcja lokalna
	  b2: **semafor**.wait()
	  b3: sekcja krytyczna
	  b4: **semafor**.signal()
	}

a1, a2, a3, a4 – stany procesu *A*
b1, b2, b3, b4 – stany procesu *B*

Specyfikacja własności poprawności w języku logiki

Przykład:

Proces *A*	Proces *B*

pętla
{
a1: sekcja lokalna
a2: **semafor**.wait()
a3: sekcja krytyczna
a4: **semafor**.signal()
}
	pętla
	{
	b1: sekcja lokalna
	b2: **semafor**.wait()
	b3: sekcja krytyczna
	b4: **semafor**.signal()
	}

Program spełnia własność wzajemnego wykluczania w tych stanach, w których prawdziwa jest formuła:

$$\neg(a3 \wedge b3)$$