

Laboratorium Modelowania Procesów Współbieżnych

Ćwiczenie 2. Konstrukcja modeli systemów informatycznych.

1. Zgodnie z podanym opisem skonstruować model dwuprocesowego systemu współbieżnego implementującego wzajemne wykluczanie przy pomocy prostego arbitra pamięci (jednej zmiennej). Przesymulować model.

Modelowany system zawiera – zgodnie z warunkami zadania – zawiera dwa procesy P i Q wykonywane współbieżnie. Każdy z tych procesów wykonuje pewien cykl instrukcji 1,2,3 stanowiących jego sekcję lokalną. Na pewnym etapie swojego działania każdy proces musi uzyskać dostęp do danych współdzielonych, wykonując na nich operacje 4 i 5; operacje te stanowią sekcje krytyczne algorytmów obydwu procesów, a zatem muszą być wykonywane rozłącznie, z zapewnieniem wzajemnego wykluczania.

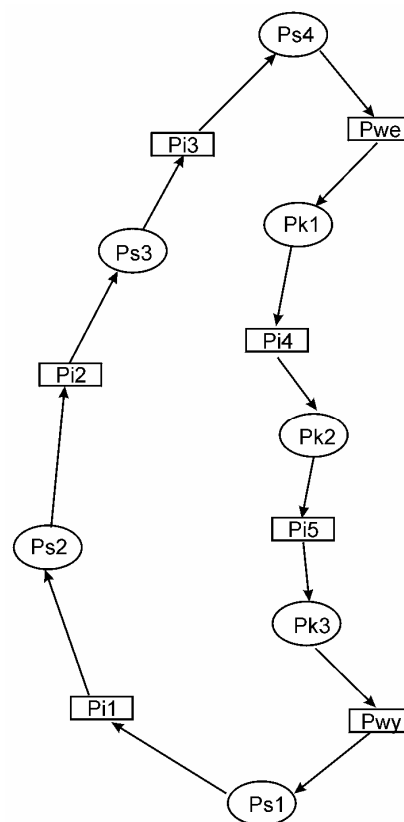
Modelowanie systemu rozpoczynamy od skonstruowania modelu każdego z procesów. Z uwagi na podobieństwo podany zostanie opis procesu P. Dla każdej z wykonywanych instrukcji n tworzymy przejście P_{in} , którego wykonanie będzie równoważne z realizacją odpowiadającej mu instrukcji. Dodatkowo należy wprowadzić przejście modelujące wejście do strefy krytycznej (o nazwie P_{we}) oraz wyjście z niej (P_{wy}). Z uwagi na reguły zawarte w definicji kolorowanych sieci Petriego niezbędne jest również wprowadzenie do modelu miejsc pomiędzy wszystkimi przejściami; modelują one stany, w jakich może znajdować się proces (zmiana stanu następuje na skutek wykonania poszczególnych instrukcji). Miejsca sekcji lokalnej nazwane zostały P_{sm} (m jest kolenym numerem stanu), sekcji krytycznej – P_{km} . Graf takiego procesu przedstawiono na rysunku 1.

W celu sygnalizacji aktualnego stanu procesu, do modelu wprowadzamy znacznik koloru *proces*, mogącego przyjmować wartości *proc*. Jest to najprostszy sposób modelowania – stan procesu modelowany jest przez obecność znacznika w odpowiednim miejscu modelu. Znacznik ten będzie taki sam dla procesu Q. A zatem wszystkie miejsca P_{sm} oraz P_{km} będą przyjmować znaczniki koloru *proces*. Do wszystkich łuków modelu będą natomiast przypisane wyrażenia $1 \text{ } proc$.

Mając model procesu P oraz analogiczny graf dla procesu Q, możemy przystąpić do modelowania arbitra pamięci. W naszym systemie będzie to zmienna *ap* mogąca przyjmować wartości *p* oraz *q* (wartość zmiennej wskazuje, który proces ma prawo wejść do strefy krytycznej).

W modelu stan zmiennej odwzorowany będzie przez wartość odpowiedniego znacznika koloru *ap* (o podanych wyżej wartościach). Znacznik ten będzie przechowywany w miejscu *arbiter*, połączonym z resztą modelu odpowiednimi łukami.

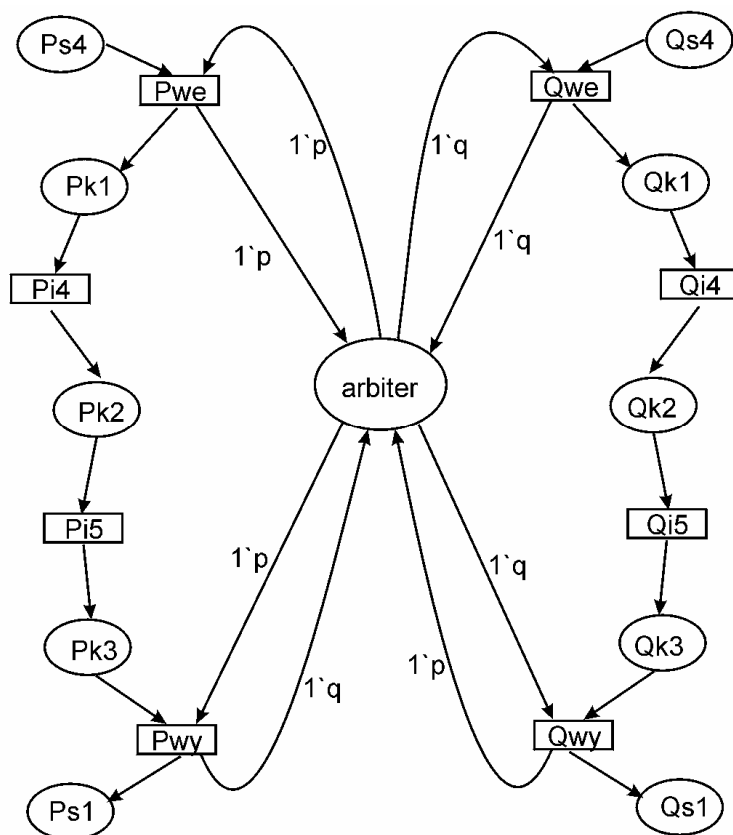
Jak wiadomo, stan zmiennej-arbitra sprawdzany jest przez procesy przed wejściem do strefy krytycznej. A zatem wykonanie przejść P_{we} oraz Q_{we} powinno być uzależnione od stanu zmiennej *ap* (wartości znacznika *ap*). Uzależnienie takie może być wykonane za pomocą odpowiednich wyrażeń łuków (jak na rysunku 2), bądź za pomocą odpowiednich wyrażeń dozorów tych przejść. W obu przypadkach należy zapewnić zwrot znacznika *ap* o tej samej wartości do miejsca *arbiter*.



Rysunek 1. Graf dla procesu P.

Podobnie modeluje się wyjście ze strefy krytycznej – jest ono jednak bezwarunkowe, należy jedynie pamiętać o zmianie wartości znacznika na przeciwną.

Na rysunku 2 przedstawiono model arbitra pamięci opisany powyżej, wraz z fragmentami modeli procesów P i Q zawierającymi ich sekcje krytyczne. Dla czytelności rysunku pominięto wyrażenia łuków w grafach procesów oraz ich sekcje lokalne.



Rysunek 2. Arbiter pamięci do implementacji wzajemnego wykluczania

2. Przekształcić model z zadania 1 tak, aby sekcje krytyczne obydwu procesów były modelowane za pomocą jednego fragmentu sieci. Wykonać symulację.

3. W oparciu o model z zadania 2 skonstruować model systemu o dwóch procesach współbieżnych implementującego wzajemne wykluczanie przy pomocy arbitra pamięci drugiego rodzaju (do każdego procesu przypisana jest zmienna arbitra; wartość zero wskazuje, że proces jest w strefie krytycznej; przed wejściem do strefy proces sprawdza zmienną drugiego procesu). Przyjąć, że jeden z procesów ma znacznie dłuższą sekcję lokalną niż drugi (np. odpowiednio 3 instrukcje i 10). Wykonać symulację.

4. Skonstruować model systemu informatycznego o dwóch procesach współbieżnych implementującego wzajemne wykluczanie w oparciu o algorytm Dekkera. Przesymulować model.

5. W terminach sieci kolorowanych skonstruować model automatu sprzedającego cukierki. Przyjmujemy, że automat przyjmuje monety 5c i 10c, wydaje cukierki po 15c i 20c, zaś użytkownik ma prawo w każdej chwili zrezygnować z zakupu i uzyskać zwrot pieniędzy; automat nie musi wydawać reszty. Struktura modelu powinna być możliwie prosta (tzn. posiadać jak najmniej węzłów).