

Programowanie współbieżne

Lista zadań nr 6

Na ćwiczenia 30 listopada 2022

Zadanie 1. Zdefiniuj następujące własności postępu: **nieczekanie** (ang. *wait-freeness*), **niewstrzymywanie** (ang. *lock-freeness*). Dlaczego te własności nazywane są **nieblokującymi** (ang. *non-blocking*) i **niezależnymi** (ang. *independent*). Dlaczego **niezakleszczenie** i **niezagłodzenie** są **blokujące** i **zależne**? Czy współbieżna metoda w nietrywialny sposób wykorzystująca zamki może być nieczekająca? Dana jest metoda **weird()**, której każde i -te wywołanie powraca po wykonaniu 2^i instrukcji. Czy ta metoda jest nieczekająca? A **nieczekająca z limitem kroków** (ang. *bounded wait-free*)?

Wskazówka: The Art of Multiprocessor Programming 2e, rozdział 3.8.

Zadanie 2. W 32-bitowym systemie komputerowym wszystkie komórki pamięci są atomowymi 32-bitowymi rejestrami MRMW. Chcesz w tym systemie zasymulować programowo 64-bitowy rejestr za pomocą dwóch rejestrów 32-bitowych. W tym celu implementujesz metody **read()** i **write()**, które odczytują i zapisują te dwa rejestry po kolei. Czy otrzymany 64-bitowy rejestr jest bezpieczny, regularny, atomowy czy też nie spełnia żadnego z tych warunków?

Zadanie 3. Algorytm Petersona spełnia warunki wzajemnego wykluczania i niezagłodzenia, jeśli zmienne **flag[0]**, **flag[1]** oraz **victim** oznaczają rejestry atomowe. Czy te warunki zostaną zachowane, jeśli dla **flag[0]** i **flag[1]** użyjemy rejestrów regularnych?

Zadanie 4. Rejestr typu SRSW nazywamy **1-regularnym**, jeśli spełnia następujące warunki:

- wywołanie **read()** która nie jest współbieżne z żadnym wywołaniem **write()** zawsze zwraca wartość umieszczoną tam przez ostatnie wcześniejsze wywołanie **write()**,
- wywołanie **read()** współbieżne z dokładnie jednym wywołaniem **write()** zwraca wartość zapisaną przez to wywołanie, lub przez ostatnie wcześniejsze wywołanie **write()**,

- w przeciwnym przypadku (wywołanie **read()** jest współbieżne z wieloma wywołaniami **write()**) wartość zwracana przez **read()** jest dowolna.

Skonstruuj M-wartościowy 1-regularny rejestr SRSW używając $O(\log M)$ Boolowskich regularnych rejestrów SRSW. Uzasadnij poprawność swojej konstrukcji.

Def. Rejestr nazwiemy **dobrym**, jeśli dla każdego ciągu współbieżnych dostępów do tego rejestru (zapisów i odczytów) każda wartość odczytana występuje wśród wartości zapisanych (tzn. wartości odczytane nie biorą się "z powietrza").

Zadanie 5. Mamy dobry rejestr typu MRSW. Dla dowolnej historii współbieżnych dostępów do tego rejestru, porządek zapisów jest jednoznacznie wyznaczony (bo jest tylko jeden wątek zapisujący). Możemy więc wszystkie zapisy ponumerować, oznaczając i -ty zapis przez W^i . Przez R^i oznaczmy odczyt rejestru, który zwrócił wartość zapisaną tam przez zapis W^i . Zauważmy, w każdym ciągu dostępów do rejestru może być co najwyżej jeden zapis W^i , oraz że istnieją takie ciągi dostępów, w których występuje wiele odczytów R^i . Pokaż, że dobry rejestr MRSW jest **regularny** wtedy i tylko wtedy, gdy dla każdego ciągu współbieżnych dostępów:

- dla każdego i nie jest prawdą, że $R^i \rightarrow W^i$ (*), oraz
- dla każdych i oraz j nie jest prawdą, że $W^i \rightarrow W^j \rightarrow R^i$ (**).

Strzałka \rightarrow oznacza relację *happens before* na metodach odczytu i zapisu rejestru.

Zadanie 6. Mamy dobry rejestr typu MRMW. Dla dowolnej historii współbieżnych dostępów do tego rejestru, przez W^i oznaczamy punkt w czasie, w którym wystąpił efekt i -tego zapisu. Pokaż, że dobry rejestr typu MRMW jest **atomowy** wtedy i tylko wtedy każda historia dostępów do tego rejestru spełnia warunki (*), (**) oraz

- dla dla każdych i oraz j jeśli $R^i \rightarrow R^j$ to $i \leq j$ (**).

Podobnie jak w poprzednim zadaniu, strzałka \rightarrow oznacza relację *happens before*, a R^i to odczyt rejestru, który zwrócił wartość zapisaną tam przez zapis W^i .

Zadanie 7. Pokaż poprawność konstrukcji regularnego M-wartościowego rejestru MRSW używającej regularnych rejestrów Boolowskich MRSW.

Wskazówka: The Art of Multiprocessor Programming 2e, rozdział 4.2.3.

Zadanie 8. Pokaż poprawność konstrukcji atomowego rejestru MRSW używającej atomowych rejestrów SRSW.

Wskazówka: TAoMP 2e, rozdział 4.2.5.

Zadanie 9. Pokaż poprawność konstrukcji atomowego rejestru MRMW używającej atomowych rejestrów MRSW.

Wskazówka: TAoMP 2e, rozdział 4.2.6.