

Programowanie współbieżne

Lista zadań nr 7

Na ćwiczenia 7 grudnia 2022

Zadanie 1. Pokaż poprawność konstrukcji niewstrzymywanej¹ (ang. *lock-free*) atomowej migawki, używającej atomowych rejestrów MRSW.

Wskazówka: TAO_{MP} 2e, rozdział 4.3.1.

Zadanie 2. Pokaż, że nie istnieje nieczekająca (ang. *wait-free*) implementacja protokołu binarnego² konsensusu dla n wątków ($n > 2$), używająca jedynie rejestrów atomowych. Skorzystaj z faktu, że taka implementacja nie istnieje dla dwóch wątków oraz z tego, że rozważane implementacje muszą być nieczekające.

Zadanie 3. Pokaż, że używając pewnej liczby obiektów implementujących binarny konsensus dla n wątków i pewnej liczby rejestrów atomowych można zaimplementować ogólny³ protokół konsensusu dla n wątków.

Zadanie 4. Na wykładzie pokazaliśmy, że kolejki FIFO mają poziom konsensusu ≥ 2 . Udowodnij, że to ograniczenie jest dokładne, tzn. że nie istnieje nieczekająca implementacja protokołu binarnego konsensusu dla trzech wątków używająca tylko kolejek FIFO i rejestrów atomowych.

Wskazówka: TAO_{MP} 2e, rozdział 5.4, twierdzenie 5.4.3.

Zadanie 5. Pokaż, że poniższy obiekt implementuje protokół binarnego konsensusu dla dwóch wątków (wartość zwracana przez **decide()** jest jedną z wartości zaproponowanych przez wątki oraz metoda ta zwraca tę samą wartość obydwu wątkom). Załóż, że wszystkie komórki pamięci są atomowymi rejestrami. Dlaczego ten wynik nie jest sprzeczny z faktem, że poziom konsensusu⁴ dla rejestrów atomowych wynosi 1?

¹ W implementacji tej metoda *scan* jest również niehamowana (ang. *obstruction-free*): każdy wątek wykonujący metodę w izolacji, zakończy ją.

² W problemie binarnego konsensusu wszystkie wątki mają na wejściu wartości ze zbioru $\{0,1\}$.

³ W problemie ogólnego konsensusu wątki przyjmują na wejściu wartości z dowolnego zbioru.

⁴ poziom konsensusu to inaczej liczba konsensusu (ang. *consensus number*)

```

public class ConsensusProposal {
    Boolean proposed[] = new Boolean[2];
    Integer[] speed = new Integer[2];
    Integer[] position = new Integer[2];
    public ConsensusProposal() {
        position[0] = 0;
        position[1] = 0;
        speed[0] = 3;
        speed[1] = 1;
    }
    public Boolean decide(Boolean value) {
        int i = ThreadID.get(); //0 or 1
        int j = 1 - i;
        proposed[i] = value;
        while (true) {
            position[i] = position[i] + speed[i];
            if (position[i] > position[j] + speed[j]) // I am far ahead of you
                return proposed[i];
            else if (position[i] < position[j]) // I am behind you
                return proposed[j];
        }
    }
}

```

Zadanie 6. Obiekty pewnej klasy **StickyBit** mają trzy możliwe stany: 0, 1, 2. Początkowym stanem obiektu jest 0. Wywołanie metody **write(v)**, gdzie **v** to 0 lub 1, ma następujące skutki:

- jeśli stan obiektu to 0, wtedy zmienia się on na **v**,
- w przeciwnym przypadku stan nie podlega zmianie

Wywołanie **read()** zwraca bieżący stan obiektu. Pokaż, że:

1. pojedynczy obiekt klasy **StickyBit** wystarczy, by rozwiązać problem nieczekającego konsensusu binarnego dla dowolnej liczby wątków.
2. przy pomocy tablicy zawierającej $\log_2 m$ obiektów **StickyBit** i pewnej liczby atomowych rejestrów można podać nieczekającą implementację protokołu konsensusu dla dowolnej liczby wątków, gdy istnieje m możliwych wartości wejściowych.

Wskazówka: Może się przydać pojedynczy rejestr atomowy MRSW dla każdego wątku.

Zadanie 7. Dwuwątkowy protokół przybliżonej zgody (ang. *approximate agreement*) dla danego $\epsilon > 0$ ma następującą definicję: wątki A i B wywołują metody odpowiednio **decide**(x_A) oraz **decide**(x_B), gdzie x_A i x_B są liczbami rzeczywistymi nieujemnymi. Metody te zwracają dowolne wartości y_A i y_B z domkniętego przedziału $[\min(x_A, x_B), \max(x_A, x_B)]$ takie, że $|y_A - y_B| \leq \epsilon$. Jaki jest poziom konsensusu dla obiektów przybliżonej zgody?

Zadanie 8. Mamy dane wielowątkowe nieczekające kolejki FIFO, które oprócz metod **enq(x)** i **deq()** mają jeszcze metodę **peek()**, która zwraca pierwszy element kolejki, ale w odróżnieniu od **deq()**, nie usuwa go stamtąd. Pokaż, że takie kolejki mają poziom konsensusu równy ∞ .

Zadanie 9 (2pkt, bonusowe). Przedstaw konstrukcję i uzasadnij poprawność nieczekającej (ang. *wait-free*) atomowej migawki, używającej atomowych rejestrów MRSW.

Wskazówka: TAoMP 2e, rozdział 4.3.2.