Programowanie współbieżne

Lista zadań nr 7

Na ćwiczenia 2. grudnia 2021

Zadanie 1. Pokaż, że nie istnieje nieczekająca (ang. *wait-free*) implementacja protokołu binarnego[1] 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 2. Pokaż, że używając pewnej liczby obiektów implementujących binarny konsensus dla n wątków i pewnej liczby rejestrów atomowych można zaimplementować ogólny[2] protokół konsensusu dla n wątków.

Zadanie 3. 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: TAoMP 2e, rozdział 5.4, twierdzenie 5.4.3.

Zadanie 4. 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 konsensusu[3] dla rejestrów atomowych wynosi 1?

```
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;
    }
```

[1] W problemie binarnego konsensusu wszystkie wątki mają na wejściu wartości ze zbioru {0,1}.
[2] W problemie ogólnego konsensusu wątki przyjmują na wejściu wartości z dowolnego zbioru.
[3] poziom konsensusu to inaczej liczba konsensusu (ang. *consensus number*)

```
    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 5. Obiekty pewnej klasy **StickyBit** mają trzy możliwe stany: $\perp, 0, 1$. Początkowym stanem obiektu jest $\perp$. Wywołanie metody **write(v)**, gdzie **v** to 0 lub 1, ma następujące skutki:

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

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

1. pojedyńczy 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 pojedyńczy rejestr atomowy MRSW dla każdego wątku.

Zadanie 6. 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 7. Obiekt konsensusu zespołowego, podobnie jak obiekt "zwykłego" konsensusu, ma metodę **decide()**. Ten obiekt rozwiązuje problem konsensusu, jeśli tylko zbiór wartości podanych na wejściach wątków jest co najwyżej dwuelementowy. W przeciwnym przypadku wartości zwrócone przez **decide()** mogą być dowolne. Pokaż, jak rozwiązać problem konsensusu dla n wątków

z co najwyżej n różnymi wartościami na wejściu, mając danych dowolnie wiele obiektów zespołowego konsensusu.

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 ∞.