EE 382C/361C: Multicore Computing **Fall 2016**

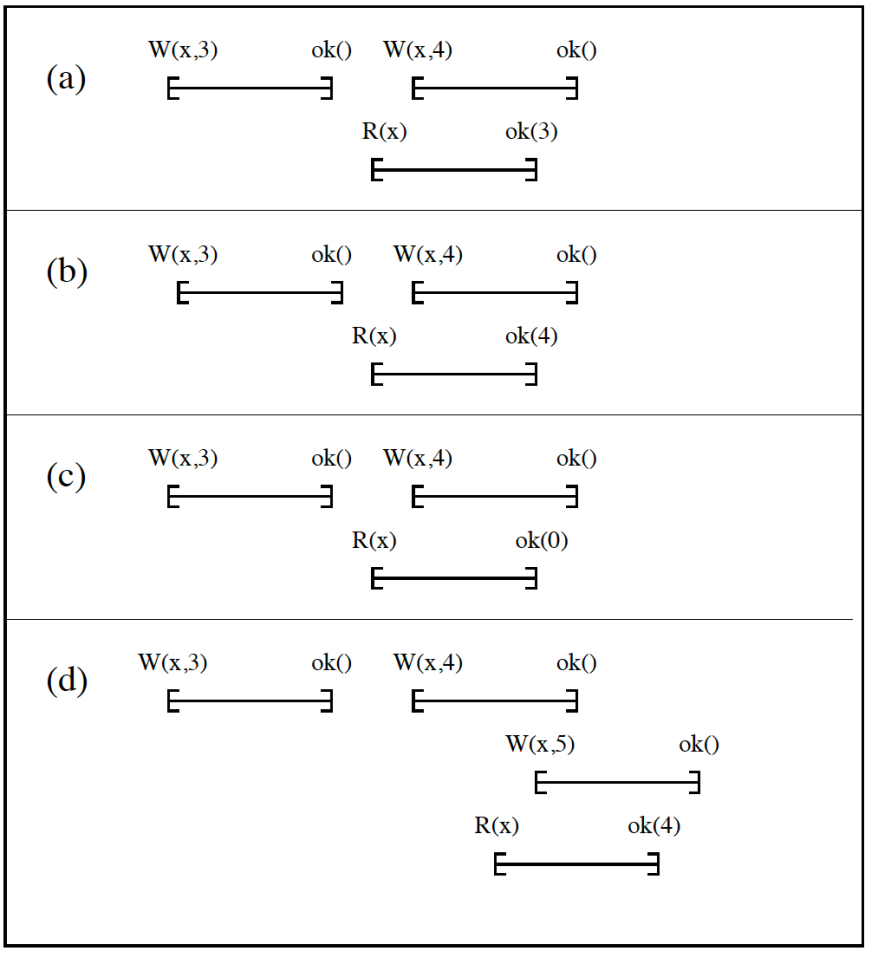
Lecture 24: November 17th

Lecturer: Vijay Garg *Scribe: Shuai QIN*

24.1 Introduction

This lecture, I will give 8 choice questions for the chapter 5 to help you review this chapter.

Q1



Which register is *safe* in the figure above?

A. (a)

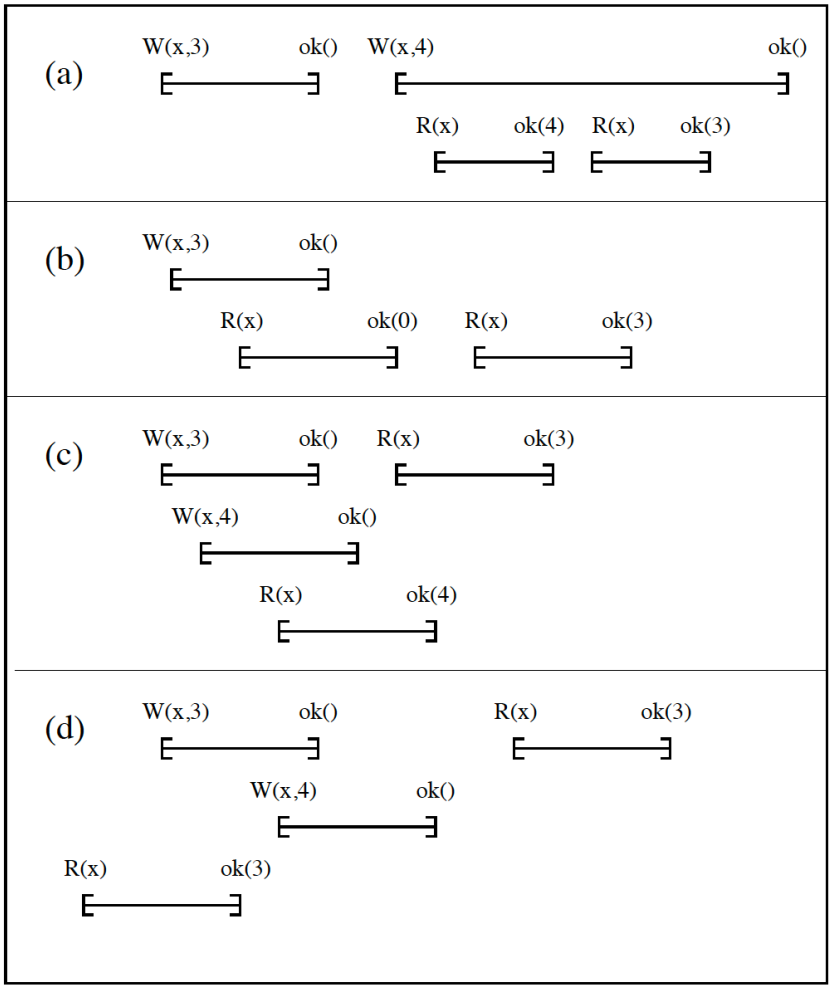
B. (a) and (b)

C. (a), (b) and (d)

D. All of above

Answer: D

Q2



Which register is *regular* in the figure above (The initial value is 0)?

A. (a)

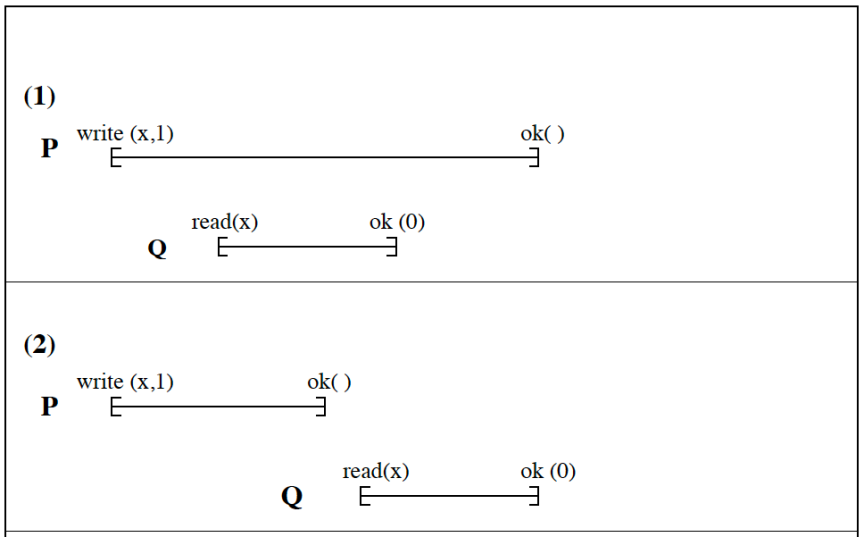
B. (a) and (c)

C. (a), (c) and (d)

D. All of above

Answer: D

Q3



Which register is *atomic* in the figure above (The initial value is 0)?

A. (1)

B. (2)

C. Both

D. Neither

Answer: A

Q4

Consider the following concurrent program:
Initially, a = 0, b = 0
P1: a = 1; print a; print b;
P2: b = 2; print a; print b;
Assume the output is P1: 12, P2: 02; Is it possible for both of the two to be atomic? If possible, please give a history to show that.

Answer: it's possible for the following sequence:
P2(w b 2) - P2(r a 0) - P2(r b 2) - P1(w a 1) - P1(r a 1) - P2(r b 2),
which is linearizable, so both a and b are atomic.

Q5

Briefly introduce the key method which is use to construct from *SRSW safe* bit to **SRSW** regular bit then to **SRSW** Multivalued Register?

Answer: from SRSW safe, we add a *prev* field to help check, and from SRSW regular to SRSW multivalue, we use a array from 0 to maxVal to allow values in the range 0 ... maxVal-1.

Q6

Briefly explain to construct the multi-value SRSW register, why we need to double scan when read?

Answer: the aim is to guarantee the linearizability. For example, if we set a[1] and then a[4], and then set 1 false, if we have two read occur sequentially, it may read 4 first and then read 1, which is not allow for a linearizable register.

Q7

Why we need communication matrix to implement the MRSW register?

Answer: it essentially allows each register to share information. Comm[i][j] is used by the reader i to inform the value it read to the reader j, so that j will not read the value before the reader i read.

Q8

Which algorithm we have learned before can help us to help MRMW register to decide on timestamps?

Answer: Lamport's Bakery Algorithm.

References

[1] V.K. Garg, Introduction to Multicore Computing