

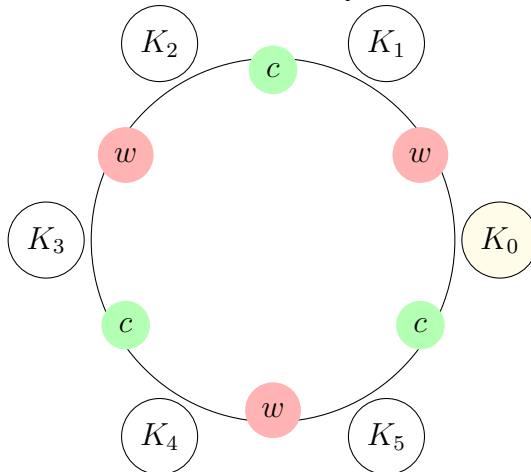
# Programowanie Równoległe – zadanie 2

19.03.2018

Zaprojektuj i zaimplementuj protokół synchronizacji dla poniższego problemu. Użyj monitorów (lock, conditional).

## Problem

1. Król Artur i jego Rycerze siedzą wokół Okrągłego Stołu. Liczba rycerzy wynosi  $n \geq 4$ , założmy, że  $n$  jest parzyste. W liczbie  $n$  uwzględniony jest też Król, jeśli nie jest wyraźnie powiedziane inaczej, Król jest także Rycerzem.
2. Każdy Rycerz na zmianę: śpi, pije i opowiada (głównie zmyślane) historie o swoich zwycięstwach. Spanie i opowiadanie historii zajmuje pewien losowy czas.
3. Jeśli ktoś opowiada historię, jego sąsiedzi (także Król) nie zaczną mówić, dopóki on nie skończy. Jeśli Król opowiada historię, nikt nie zacznie mówić (ale spanie i picie jest akceptowalne). Jeśli Rycerz zacznie opowiadać historię, może ją swobodnie dokończyć — nawet jeśli Król zacznie mówić.
4. Pomiędzy Rycerzami znajdują się kielichy na (bezalkoholowe) wino oraz talerzyki z ogórkami (patrz rysunek poniżej, kielichy są czerwone, a talerzyki zielone).



5. Aby zacząć pić, Rycerz musi trzymać zarówno kielich, jak i talerzyk (w tym samym momencie), zauważ, że kielichy i talerzyki są współdzielone przez sąsiadujących Rycerzy.
6. Rycerze nie mogą pić z pustych kielichów. Na stole znajduje się butelka wina, która może pomieścić  $w$  jednostek (kielichów). Kiedy Rycerz bierze kielich (i oczywiście talerzyk), nalewa wino z butelki do kielicha i wypija wino. Jeśli butelka jest pusta, Rycerz czeka.
7. Natychmiast po wypiciu, Rycerz zagryza ogórkiem. Jeśli na talerzyku nie ma ogórków, Rycerz nie zacznie picia. Talerzyk może pomieścić  $c$  ogórków.
8. Na sali jest też dwóch Służących, którzy pojawiają się w losowych odstępach czasu. Pierwszy Służący uzupełnia wino w butelce (nawet jeśli nie była zupełnie pusta), a drugi uzupełnia wszystkie talerzyki świeżymi ogórkami.
9. Po wypiciu 10 kielichów wina, Rycerz upada pod stół. To nie przesuwają pozostałych Rycerzy i nie zmienia ich sąsiedztwa.

## Aspekty techniczne

1. Rozwiązanie powinno oferować maksymalną równoległość. Jeśli wszyscy Rycerze chcą zacząć pić i zasoby są dostępne, około  $n/2$  rycerzy powinno być w stanie to zrobić.
2. Jeśli ogórki Rycerza nie są dostępne, nie powinno to blokować jego sąsiadów, którzy mogą mieć ogórki.
3. Unikaj aktywnego czekania i głodzenia. Kielichy i talerzyki powinny być brane tylko jeśli mamy pewność, że możemy rozpocząć picie. Nie bierz także wina z butelki i ogórków, jeśli nie możesz zacząć pić.
4. Wypisuj aktualny stan programu na konsolę.
5. Dopasuj parametry i losowe odstępy, aby było widać, co się dzieje. Dobór odstępów nie powinien mieć znaczenia dla protokołu synchronizacji.

**Termin wgrania rozwiązania: 09.04.2018 (7.00 rano)**