**Algorytm Ricarta i Agrawali**

Algorytm Ricarta i Agrawali jest zoptymalizowana wersją algorytmu Lamporta. Każdy proces ma swój zegar logiczny. Jeżeli proces chce wejść do sekcji krytycznej, wysyła do wszystkich innych komunikat żądania ze swoim numerem, oraz zapamiętuje czas swojego żądania. Do każdego komunikatu dołączony jest stempel czasowy. W przypadku gdy proces otrzyma żądanie to: jeżeli nie chce wejść do sekcji krytycznej, wysyła zgodę żądającemu lub jeśli chciał wejść to porównuje stempel czasowy żądania z czasem swojego żądania, jeśli jego jest wyższy(chciał później uzyskać dostęp) to wysyła żądającemu zgodę.   
Jeśli nie, dodaje żądającego do kolejki. Jeżeli stempel żądającego jest taki sam jak czas żądania odbiorcy,   
to odbiorca wysyła zgodę jeśli proces żądający ma mniejszy numer od jego numeru (bez tego byłyby błędy - zakleszczenie lub naruszenie bezpieczeństwa). Gdy proces otrzyma zgody od wszystkich procesów, wchodzi do sekcji krytycznej. Podczas wychodzenia z sekcji krytycznej, proces wysyła zgodę wszystkim procesom, które umieścił wcześniej w kolejce oczekujących. W sekcji wyjściowej roześle zgodę do wszystkich zapamiętanych na liście oczekujących.

**Algorytm** **Maekawy**

W pierwszym kroku algorytmu Maekawy proces Pi, który ubiega się o wejście do sekcji krytycznej, rozsyła żądanie do wszystkich procesów, od których wymaga pozwolenia. Kiedy proces Pj otrzyma komunikat żądania, wysyła odpowiedź do procesu Pi pod warunkiem, że nie wysłał już komunikatu z odpowiedzią od czasu otrzymania ostatniej wiadomości zwolnij. W przeciwnym wypadku komunikat z żądaniem trafia do kolejki, w celu jego późniejszego rozpatrzenia. W momencie kiedy proces Pi otrzyma komunikaty typu odpowiedź od wszystkich procesów ze zbioru Ri, może uruchomić swoją sekcję krytyczną. Po wykonaniu sekcji krytycznej, proces Pi wysyła wiadomości zwolnij do wszystkich procesów w Ri. W wypadku kiedy proces Pj otrzyma komunikat zwolnij od procesu Pi, wysyła odpowiedź do następnego oczekującego procesu, który jest w jego kolejce i usuwa go z niej. Jeżeli kolejka jest pusta, wtedy proces aktualizuje swój stan.

**Algorytm** **Raymonda**Algorytm Raymonda używa struktury drzewa którego korzeniem jest proces. Kiedy proces Pi chce wejść do sekcji krytycznej, wysyła żądania wzdłuż ścieżki do korzenia i dodaje żądania do kolejki. Następuje odebranie żądania, wstawienie go do kolejki i przesłanie dalej wzdłuż ścieżki do korzenia. Następnie zostaje wysłany żeton przez korzeń do ubiegającego się procesu oraz aktualizacja zmiennej posiadacz. Żeton zostaje odebrany, następuje wyjęcie żądania z kolejki i wysłanie do proces wskazanego w tym żądaniu, a zmienna posiadacz zostaje zaktualizowana. Proces wchodzi do sekcji krytycznej pod warunkiem, że otrzymał żeton,   
a jego żądanie jest na szczycie jego kolejki żądań. Zwalnianie sekcji krytycznej następuje poprzez przesłanie żetonu i aktualizacji kolejki oraz zmiennej posiadacz, ewentualnie przesłanie żądania jeżeli kolejka żądań jest niepusta.