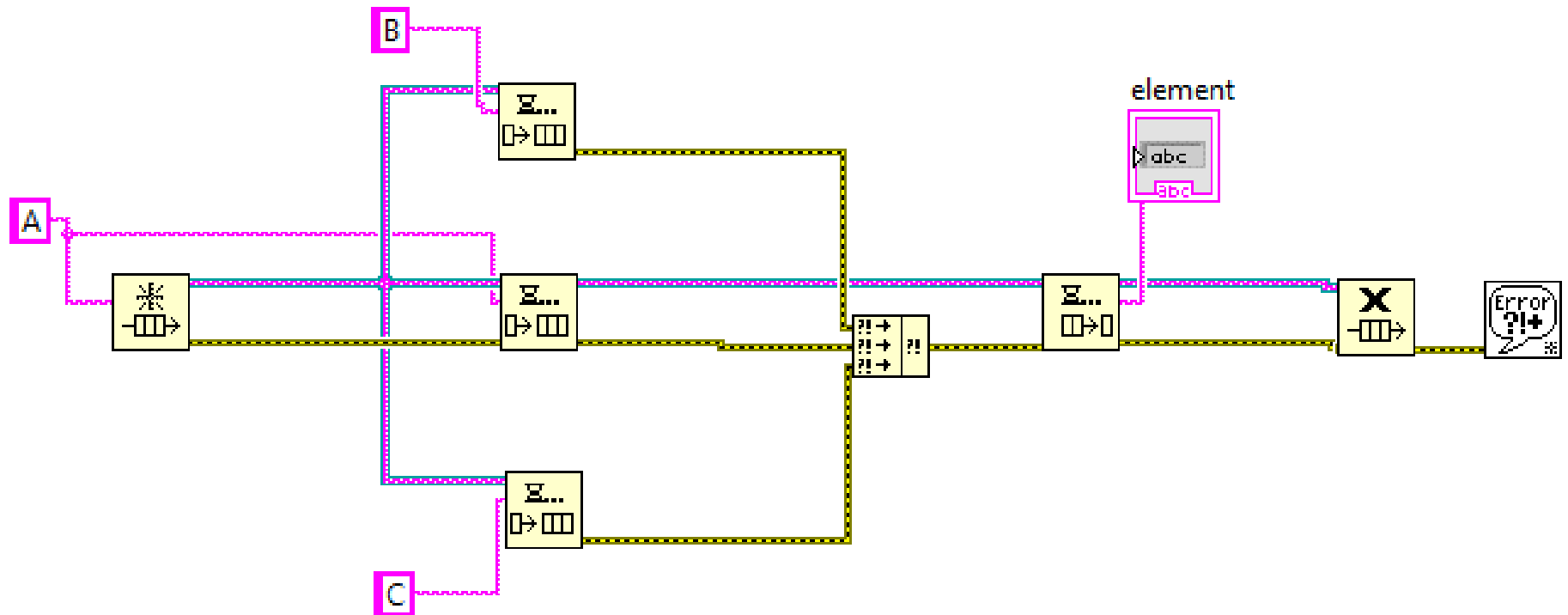


Race conditions i kako ih rešiti

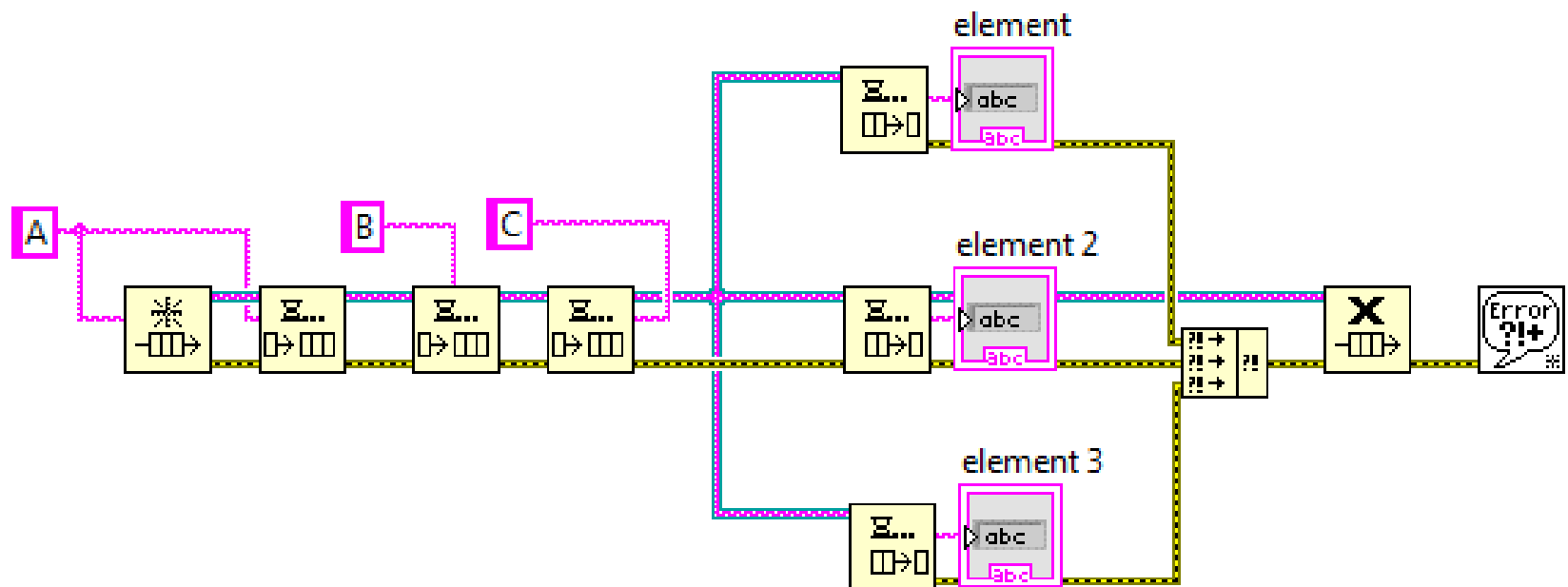
Uvod

- Problemi koji se javljaju kada ishod zavisi od redosleda izvršavanja delova koda



Uvod

- Problemi koji se javljaju kada ishod zavisi od redosleda izvršavanja delova koda

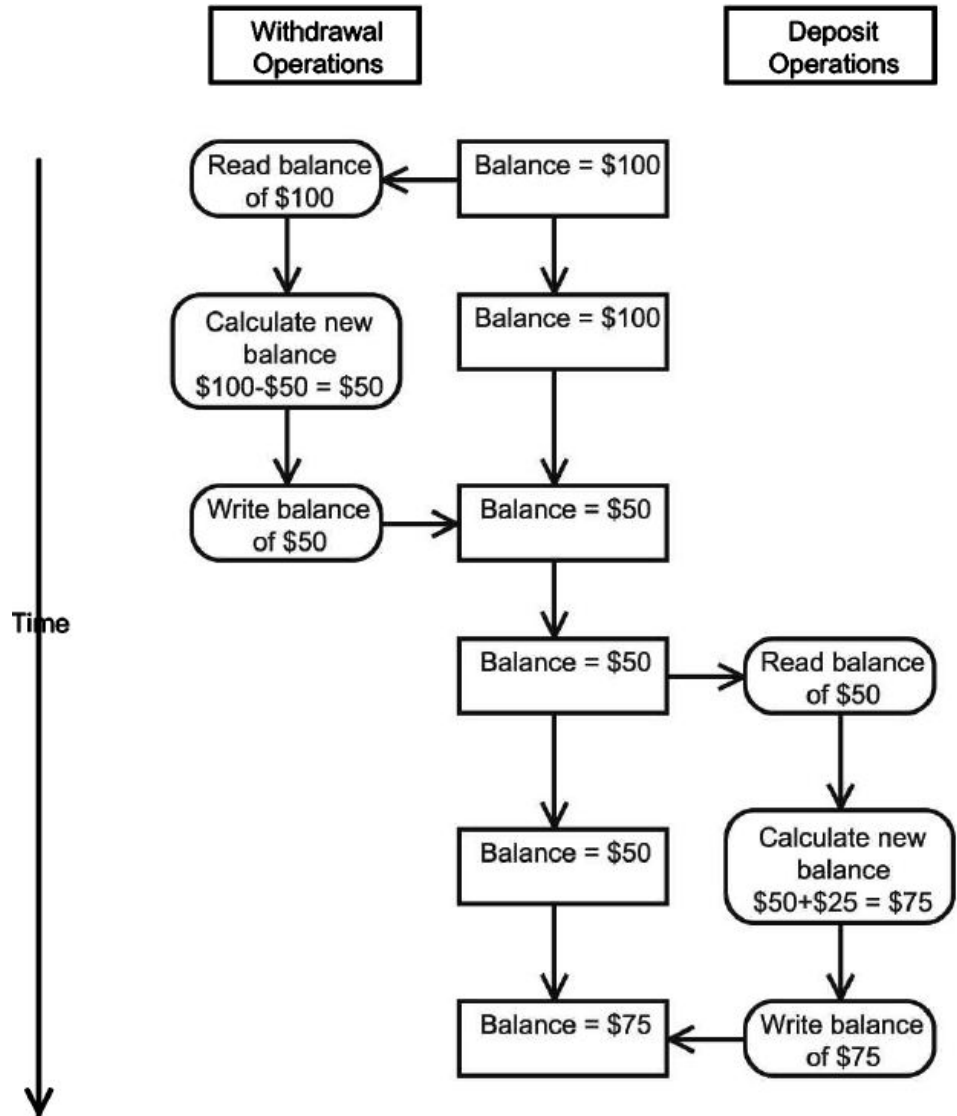


Uvod

- Problemi koji se javljaju kada ishod zavisi od redosleda izvršavanja delova koda
- Mogu izazvati ozbiljne probleme!

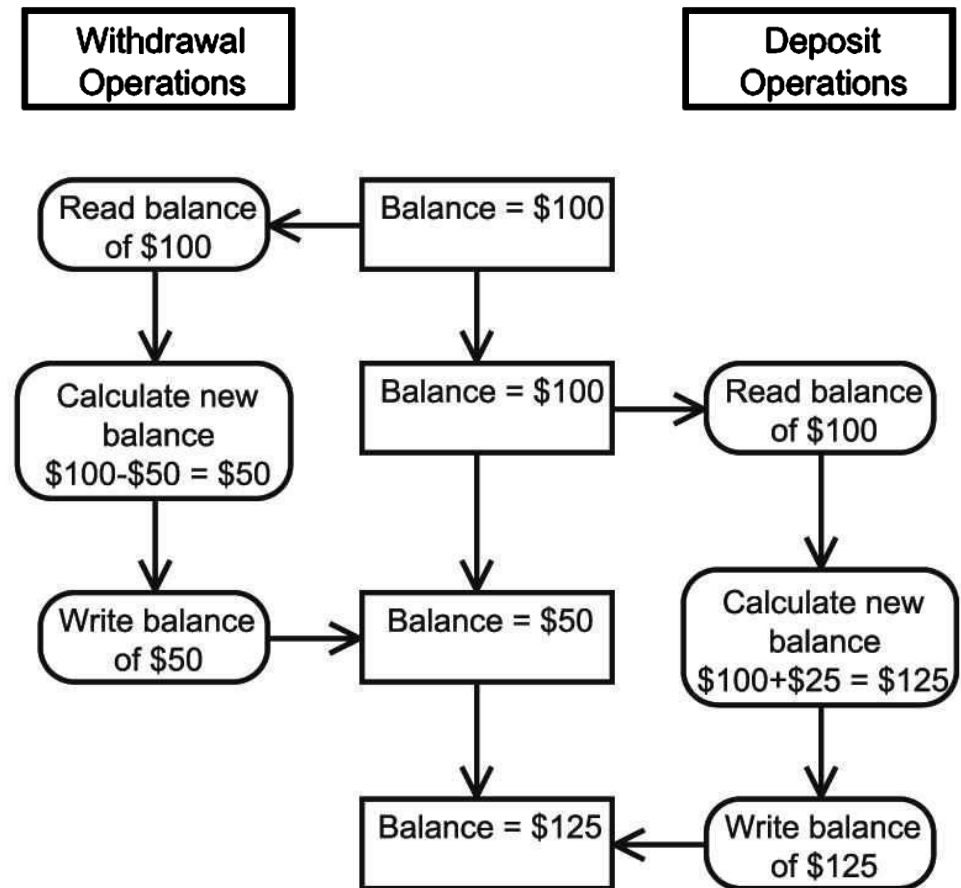
Primer

- Klijent ima na računu 100\$
- Želi da:
 - podigne 50\$ u gotovini i
 - da uloži na račun 25\$ preko čeka
- Jedan mogući scenario:
 - Podizanje gotovine
 - uplata
- Krajnje stanje na računu je 75\$
- Drugi mogući scenario:
 - Uplata
 - Podizanje gotovine
- Krajnje stanje na računu je opet 75\$



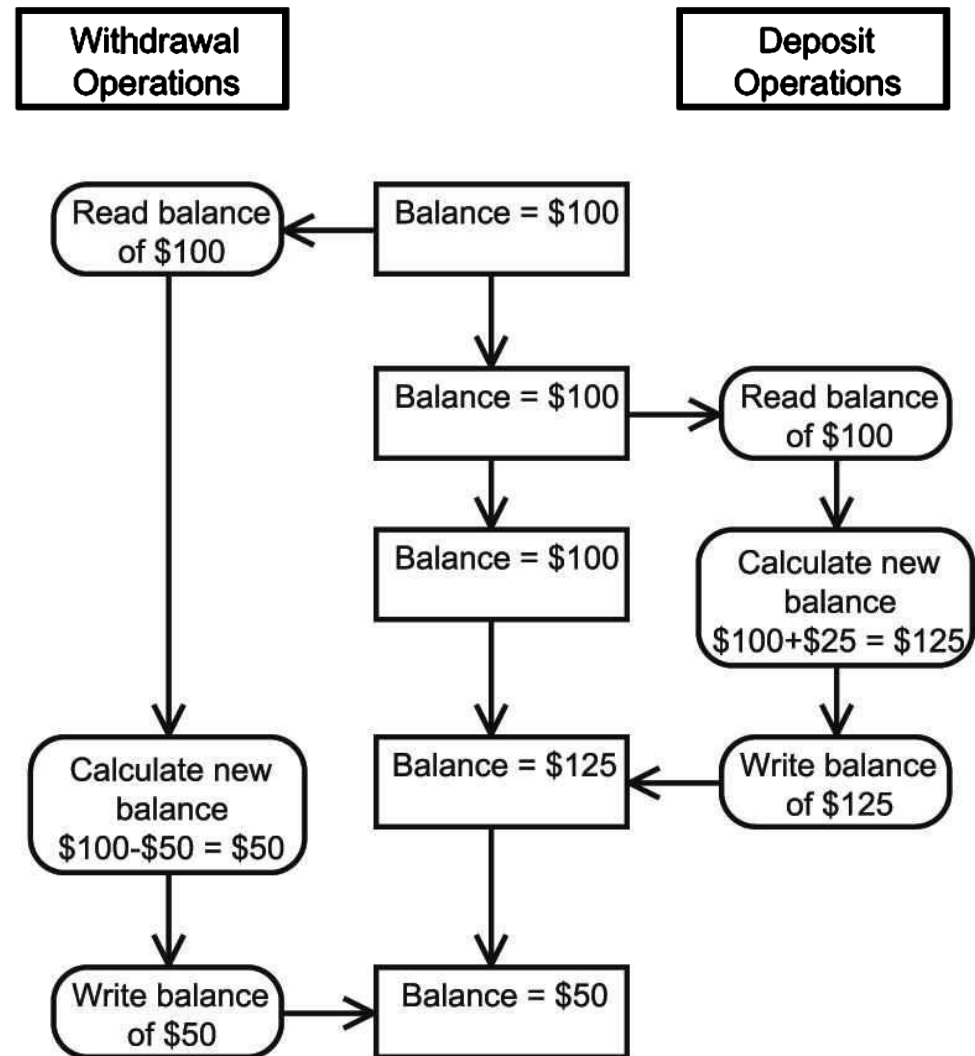
Primer

- Treći scenario:
 - Obe operacije istovremeno pristupaju računu
 - Stanje na kraju je 125\$
 - Pogrešno!



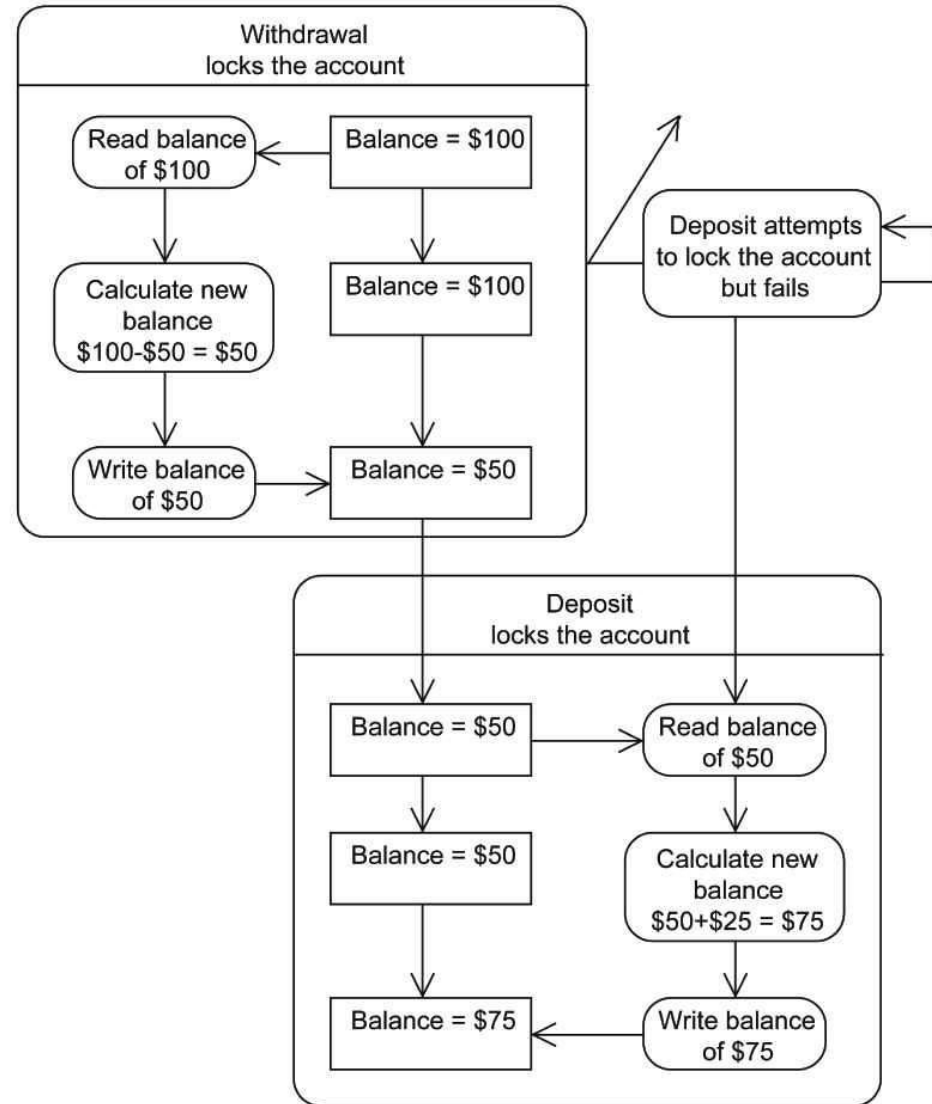
Primer

- Četvrti scenario:
 - Obe operacije istovremeno pristupaju računu
 - Stanje na kraju je 50\$
 - Pogrešno!
- Klasičan primer za Race condition!
- Problem je što se operacije međusobno ne blokiraju!



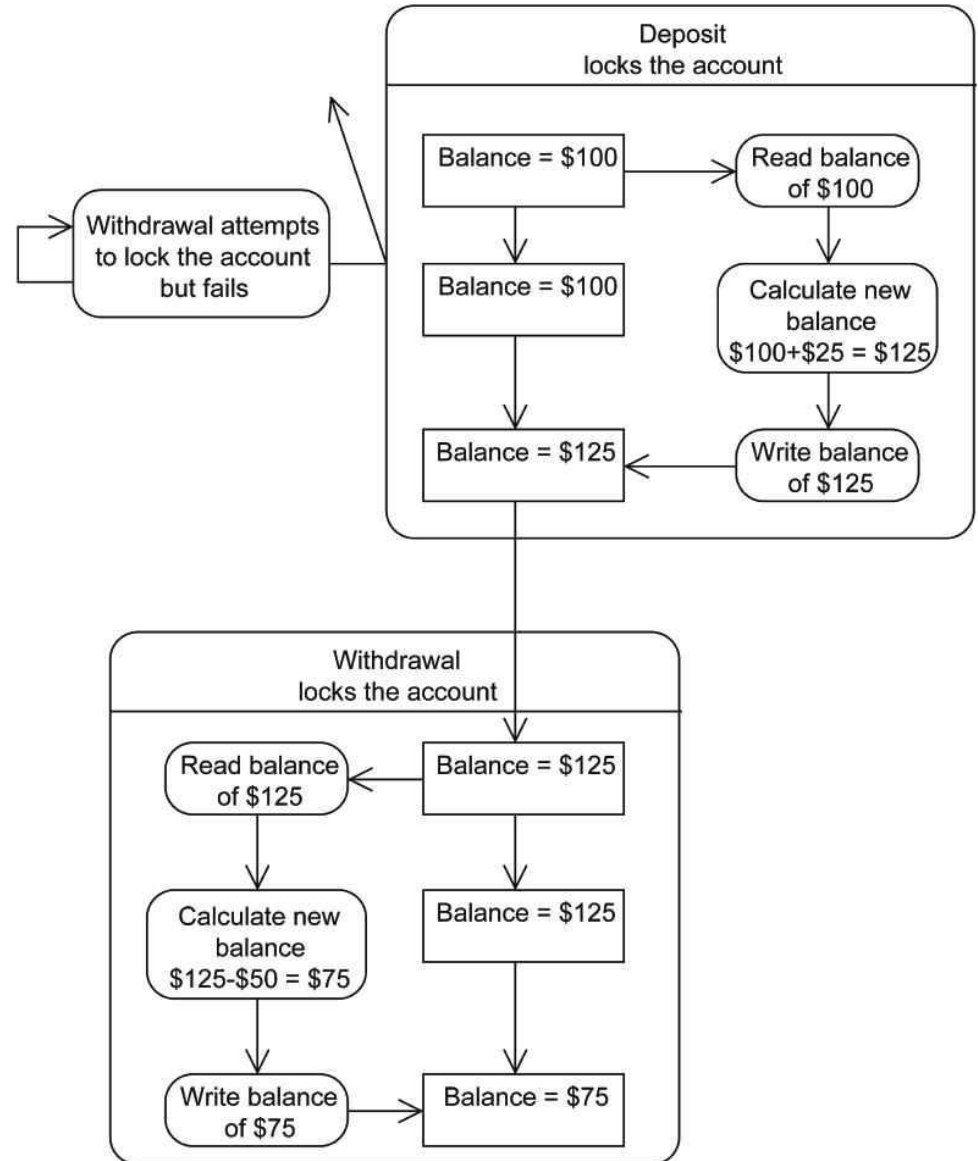
Primer

- Pravilan pristup:
 - Podizanje gotovine započinje
 - Blokira se pristup računu za uplate
 - Novo stanje je 50\$
 - Odblokira se pristup
 - Uplata od 25\$, blokiran pristup za podizanje
 - Konačno stanje 75\$



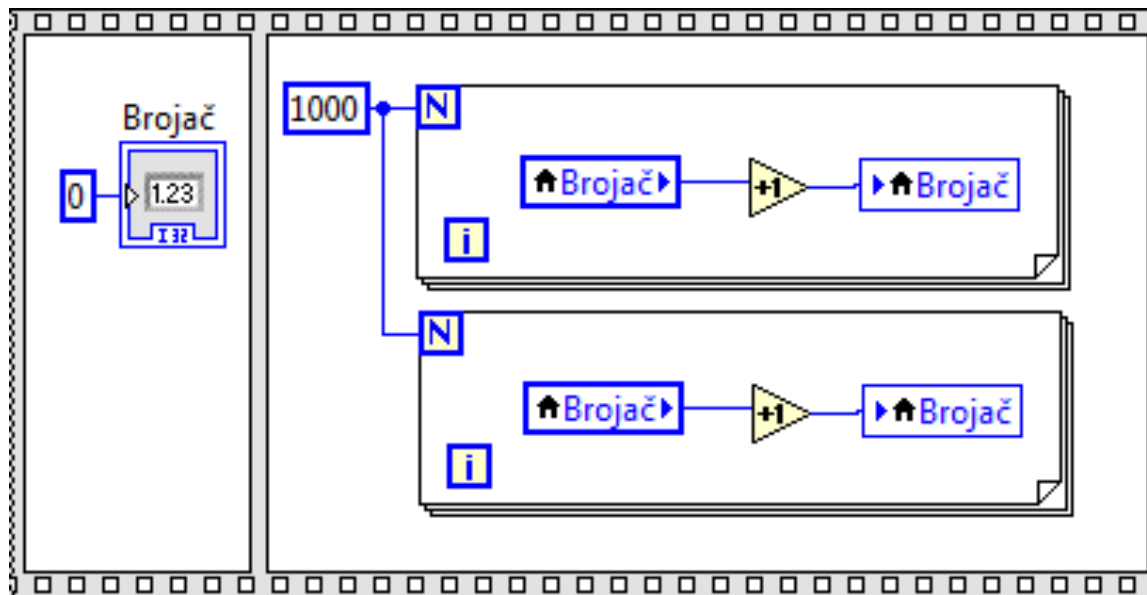
Primer

- ILI:
 - Uplata započinje
 - Blokira se pristup računu za podizanje
 - Novo stanje je 125\$
 - Odblokira se pristup
 - Podizanje 50\$, blokiran pristup za uplate
 - Konačno stanje 75\$
- Zaključak
 - Mora se obezbediti blokada istovremenog pristupa jednom istom, deljenom elementu (resursu)!



Primer - LabView

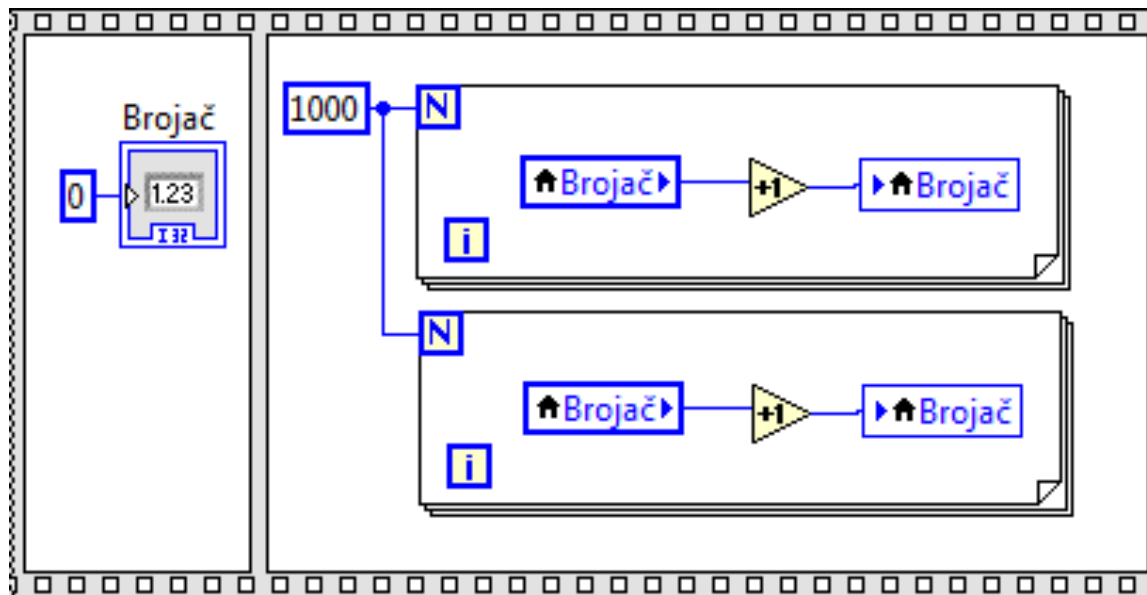
- Dve paralelne for petlje
- U svakoj se inkrementira vrednosti brojača



- Kolika je vrednost brojača na kraju?

Primer - LabView

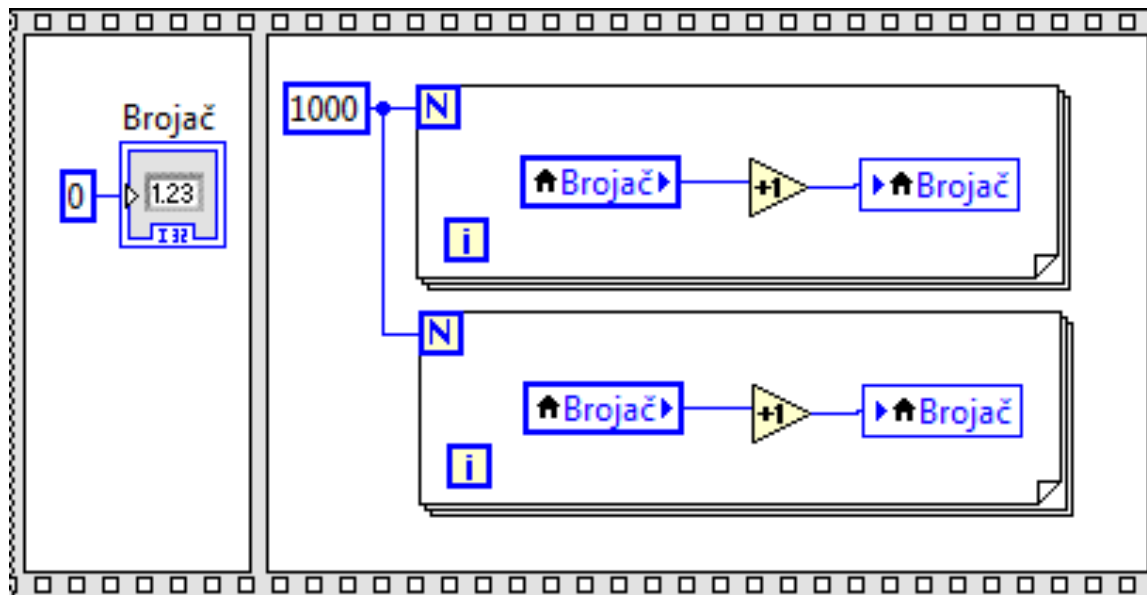
- Dve paralelne for petlje
- U svakoj se inkrementira vrednosti brojača



- Kolika je vrednost brojača na kraju?
- 2000?

Primer - LabView

- Dve paralelne for petlje
- U svakoj se inkrementira vrednosti brojača



- Kolika je vrednost brojača na kraju?
- 2000?
- Tačan odgovor - bilo koja vrednost između 1000 i 2000!

Rezime

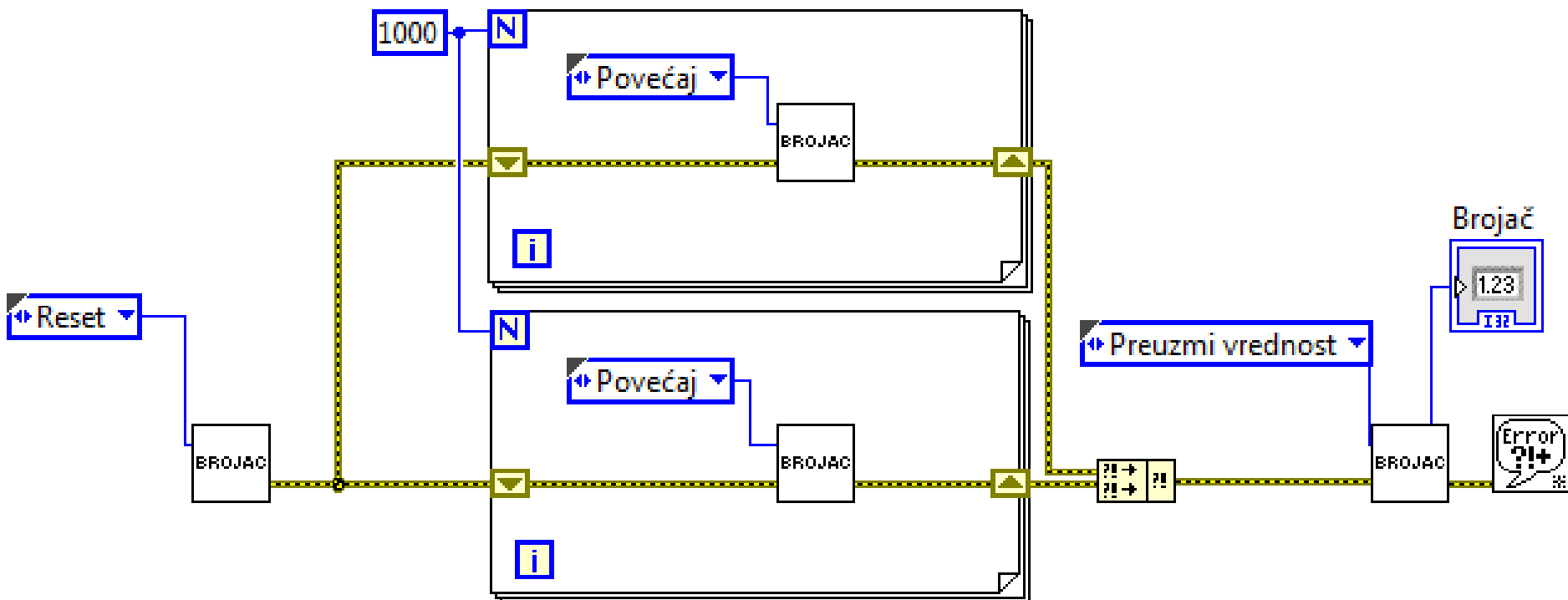
- Race conditions mogu napraviti problem kod izvršavanja koda
- Obično je problematičan samo deo koda
- Veoma ih je teško otkriti; u nekim fazama testiranja kod prolazi kao tačan
- Teško je i smisliti test koji će ih otkriti
- Rešenje – prepoznati problematične delove koda i preduprediti pojavu ovih uslova!

Rešavanje problema

- Treba prepoznati problematične delove koda
- Kod paralelnih petlji to je inkrementiranje
- U ovoj sekvenci treba blokirati pristup lokalnoj promenljivoj
- Jedno moguće rešenje – tzv. non-reentrant sub VI
- Sub VI-jevi su podrazumevano “non-reentrant”; samo jedna kopija može se izvršavati u jednom trenutku
- Sub VI ne može biti u paraleli sa samim sobom!
- Action engine je primer za “non-reentrant” Sub VI

Rešavanje problema – Sub VI

- Sub VI “brojač”
- Action engine

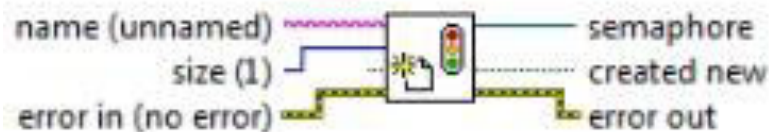


Rešavanje problema – Semafori

- Fizički – semafor služi za signalizaciju; blokira saobraćaj u jednom pravcu, dozvoljava u drugom
- Softverski – sličan princip
- Kritičan deo koda se stavlja između funkcija Acquire Semaphore i Release Semaphore
- Na taj način se taj deo koda štiti i omogućava njegovo ekskluzivno izvršavanje (blokirani pristup deljenim resursima za druge delove koda)

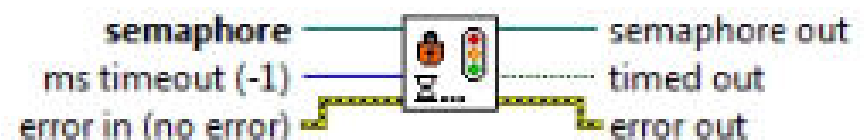
Funkcije za rad sa semaforima

Obtain Semaphore Reference.vi



Obtains a reference to an existing semaphore or creates a new semaphore and returns a reference to that semaphore.

Acquire Semaphore.vi



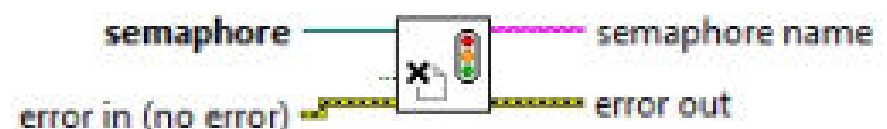
Acquires access to a semaphore.

Release Semaphore.vi



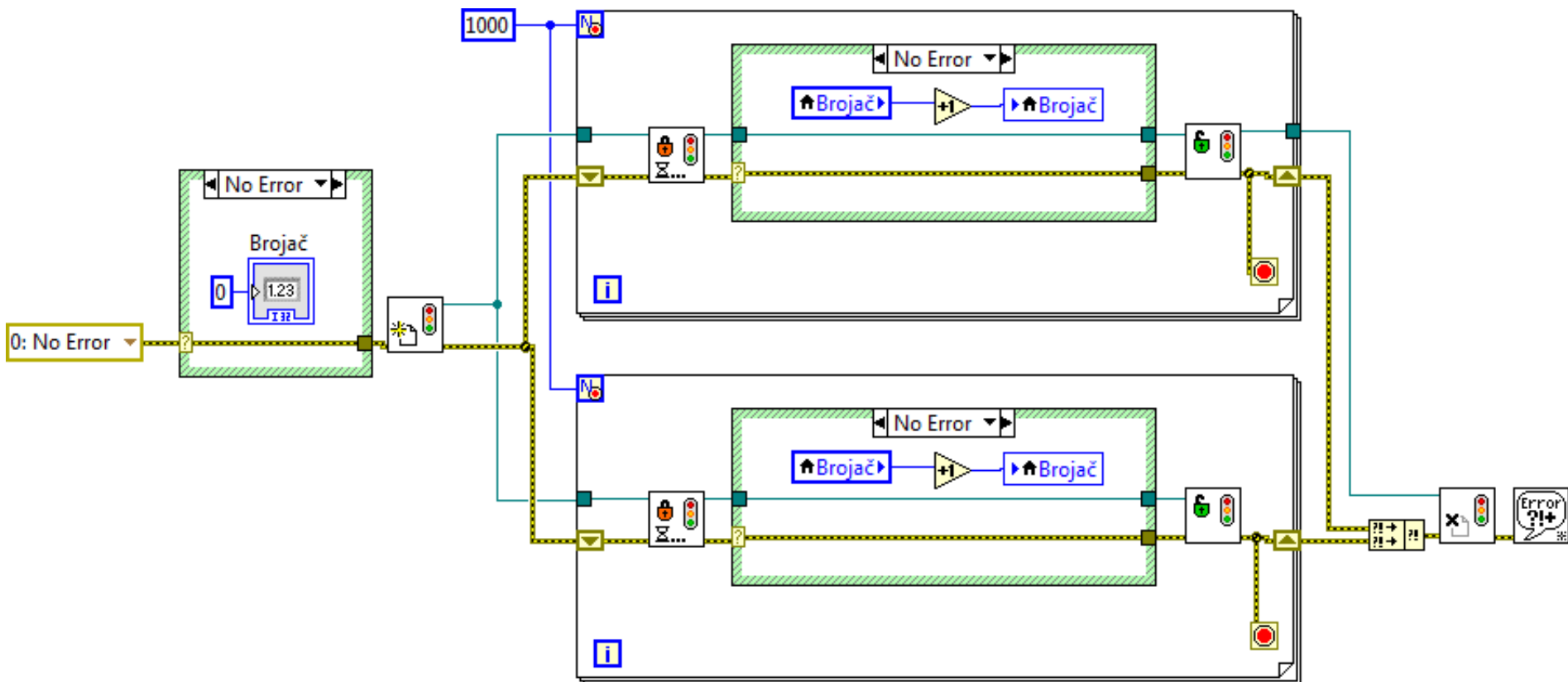
Releases access to a semaphore. If the Acquire Semaphore VI is waiting for the semaphore this VI releases, it stops waiting and continues execution.

Release Semaphore Reference.vi



Releases a reference to a semaphore.

Rešenje sa semaforima



Rešenje sa semaforima

