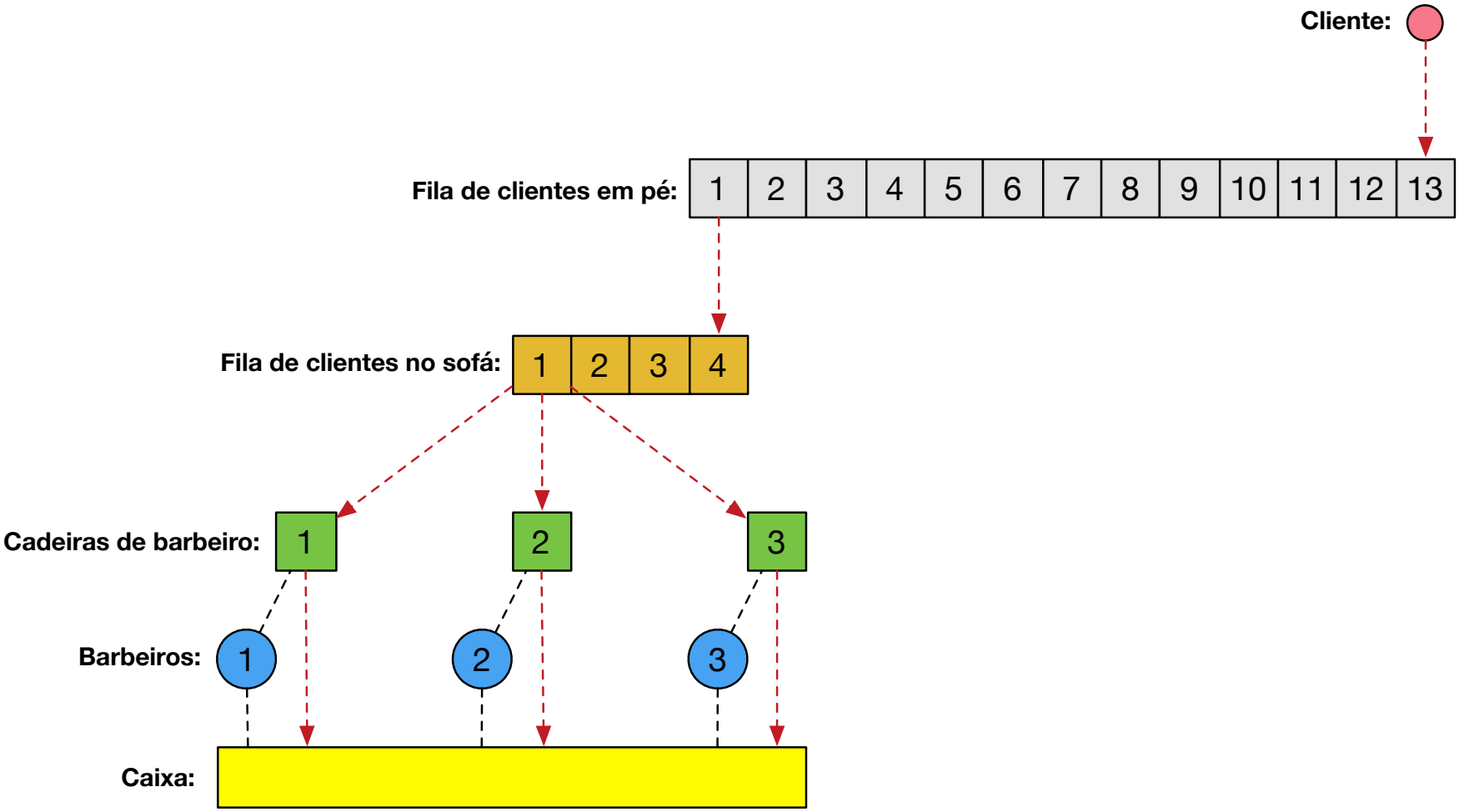
Problema da Barbearia de Hilzer

Problemas menos Clássicos

Caracterização

- A barbearia possui:
  - três cadeiras para corte de cabelo
  - três barbeiros
  - uma sala de espera com um sofá e espaço para alguns clientes ficarem em pé
  - o limite total de clientes na barbearia é 20
- Um cliente não entra na barbearia se esta já estiver no limite de clientes.
- Um cliente aguarda no sofá ou em pé, caso o sofá já esteja totalmente ocupado.
- Quando o barbeiro fica livre, o cliente há mais tempo no sofá é atendido. O cliente em pé há mais tempo na barbearia senta-se no sofá.
- Quando termina o corte de cabelo de um cliente, qualquer um dos barbeiros pode receber o pagamento. Mas, como há somente um caixa, somente um pagamento é feito por vez.
- O barbeiro divide o seu tempo entre cortar cabelo, receber pagamento e dormir na sua cadeira, esperando por um cliente.

Problema da Barbearia - Hilzer



Solução

- Limite de clientes na barbearia:

```
N := 20
```

- Contador de clientes na barbearia:

```
contadorClientes := 0
```

- Fila para o sofá:

```
filaSofa := Fila( Semáforo )
```

- Fila para as cadeiras:

```
filaCadeiras := Fila( Semáforo )
```

- Exclusão mútua para o contador de clientes e para as filas:

```
mutex := Semáforo( 1 )
```

- Controle do número de clientes no sofá:

```
sofa := Semáforo( 4 )
```

Solução

- Indicador de cliente na fila para o sofá:

```
clienteFilaSofa := Semáforo( 0 )
```

- Indicador de cliente na fila das cadeiras:

```
clienteFilaCadeiras := Semáforo( 0 )
```

- Indicador de pagamento por um cliente:

```
pagamento := Semáforo( 0 )
```

- Indicador de recebimento por um barbeiro:

```
recebimento := Semáforo( 0 )
```

- Indicador de barbeiro livre:

```
barbeiro := Semáforo( 0 )
```

Cliente *i* :

```
representanteFilaSofa[ i ] := Semáforo( 0 )
representanteFilaCadeiras[ i ] := Semáforo( 0 )
mutex.esperar( )
    se contadorClientes == N
        mutex.sinalizar( )
        desistir( )
    contadorClientes := contadorClientes + 1
    filaSofa.inserir( representanteFilaSofa[ i ] )
mutex.sinalizar( )
clienteFilaSofa.sinalizar( )
representanteFilaSofa[ i ].esperar( )
sofa.esperar( )
    mutex.esperar( )
        filaCadeiras.inserir( representanteFilaCadeiras[ i ] )
    mutex.sinalizar( )
    clienteFilaCadeiras.sinalizar( )
    representanteFilaCadeiras[ i ].esperar( )
sofa.sinalizar( )
terCabeloCortado( )
pagamento.sinalizar( )
recebimento.esperar( )
mutex.esperar( )
    contadorClientes := contadorClientes - 1
mutex.sinalizar( )
```

Barbeiro *k* :

```
repetir para sempre:
    clienteFilaSofa.esperar( )
    mutex.esperar( )
        sem := filaSofa.remover( )
    mutex.sinalizar( )
    sem.sinalizar( )

    clienteFilaCadeiras.esperar( )
    mutex.esperar( )
        sem := filaCadeiras.remover( )
    mutex.sinalizar( )
    sem.sinalizar( )

    barbeiro.sinalizar( )
    cortarCabelo( )

    pagamento.esperar( )
    recebimento.sinalizar( )
```