Drinking Philosophers (III)
ModularDP Algorithmus (I)

D_i automaton:

Signature:

Input:
- $try(B)_i$, $B \subseteq B_i$
- $exit_i$
- $internal\text{-}crit_i$
- $internal\text{-}rem_i$
- $receive(m)_{j,i}$, $m \in \{$ *"request"*, *"bottle"*, *"demand"*$\} \times (B_i \cap B_j)$, $j \in nbrs$

Output:
- $crit_i$
- rem_i
- $internal\text{-}try_i$
- $internal\text{-}exit_i$
- $send(m)_{i,j}$, $m \in \{$ *"request"*, *"bottle"*, *"demand"*$\} \times (B_i \cap B_j)$, $j \in nbrs$

States:

region $\in \{R, T, C, E\}$, initially R
internal-region $\in \{R, T, C, E\}$, initially R
need $\subseteq B_i$, initially $\emptyset$
bottles $\subseteq B_i$; initially arbitrary, subject to the global restriction that the *bottle* sets for all processes partition the set of all bottles of $\mathcal{R}$.
deferred $\subseteq B_i$, initially $\emptyset$
current, a Boolean, initially *false*
for every $j \in nbrs$:
- *send-buffer*(j), a FIFO queue of messages in $\{$ *"request"*, *"bottle"*, *"demand"*$\} \times (B_i \cap B_j)$, initially empty

Drinking Philosophers (III)
ModularDP Algorithmus (II)

Transitions:

$try(B)_i$
  Effect:
    $region := T$
    $need := B$
    for all $j \in nbrs$,
      all $b \in (need \cap B_j) - bottles$, do
      add ("*request*", b) to $send\text{-}buffer(j)$

$send(m)_{i,j}$
  Precondition:
    m is first on $send\text{-}buffer(j)$
  Effect:
    remove first element of $send\text{-}buffer(j)$

$internal\text{-}try_i$
  Precondition:
    $region = T$
    $internal\text{-}region = R$
  Effect:
    $internal\text{-}region := T$

$receive(\text{“demand”}, b)_{j,i}$
  Effect:
    if $b \in bottles$ and ($region \neq C$ or $b \notin need$)
    then
      add ("*bottle*", b) to $send\text{-}buffer(j)$
      $bottles := bottles - \{b\}$
      $deferred := deferred - \{b\}$

$receive(\text{“bottle”}, b)_{j,i}$
  Effect:
    $bottles := bottles \cup \{b\}$

$crit_i$
  Precondition:
    $region = T$
    $need \subseteq bottles$

  Effect:
    $region := C$
    $current := false$

$receive(\text{“request”}, b)_{j,i}$
  Effect:
    if $region \in \{T, C\}$ and $b \in need$ then
      $deferred := deferred \cup \{b\}$
    else
      add ("*bottle*", b) to $send\text{-}buffer(j)$
      $bottles := bottles - \{b\}$

$internal\text{-}crit_i$
  Effect:
    $internal\text{-}region := C$
    if $region = T$ then
      $current := true$
      for all $j \in nbrs$,
        all $b \in (need \cap B_j) - bottles$, do
        add ("*demand*", b) to $send\text{-}buffer(j)$

$internal\text{-}exit_i$
  Precondition:
    $internal\text{-}region = C$
    $current = false$
  Effect:
    $internal\text{-}region = E$

$internal\text{-}rem_i$
  Effect:
    $internal\text{-}region := R$

$exit_i$
  Effect:
    $region := E$
    for all $j \in nbrs$,
      all $b \in deferred \cap B_j$ do
      add ("*bottle*", b) to $send\text{-}buffer(j)$
    $bottles := bottles - deferred$
    $deferred := \emptyset$

rem_i
  Precondition:
    $region = E$
  Effect:
    $region := R$