TAD ID:

obs dni : int
obs casta : int; \\0 es prole, 1 es bien

TAD Banco:

obs colaBien : seq(ID)
obs colaProle : seq(ID)
obs posEnCola(e:ID):($\mathbb{Z}$)

`proc Identificacion` (inout b: Banco, in p : ID)
`requiere` $\{(p.ID \notin colaBien \vee p.ID \notin colaProle) \wedge b_0 = b\}$
`asegura` $\{p.casta = 0 \longrightarrow_L b_c olaProle = concat(b_0.colaProle, [p]\}$
`asegura` $\{p.casta = 1 \longrightarrow_L b_c olaBien = concat(b_0.colaBien, [p]\}$

`proc AtenderCajaA` (inout b:Banco)
`requiere` $\{b_0 = b \wedge b.colaBien \neq []\}$
`asegura` $\{b.colaBien = subseq(b_0.colaBien, 1, |b_0.colaBien|)\}$
`asegura` $\{b.colaProple = b_0.colaProple\}$

`proc AtenderCajaB` (inout q: Banco)
`requiere` $\{q_0 = q \wedge (b.colaBien \neq [] \vee b.colaProle \neq [])\}$
`asegura` $\{q_0.colaBien = [] \longrightarrow_L (q_0.colaBien = q.colaBien \wedge q.colaProle = subseq(q_0.colaProle, 1, |q_0.colaBien|)\}$
`asegura` $\{q_0.colaBien \neq [] \longrightarrow_L (q.colaProple = q_0.colaProle \wedge q.colaBien = subseq(q_0.colaBien, 1, |q_0.colaBien)|\}$

`proc Cansarse` (inout q : Banco, in p : ID)
`requiere` $\{q = q_0 \wedge (p \in q.colaBien \vee p \in q.colaProle\}$
`asegura` $\{p \in q_0.colaBien \longrightarrow_L q.colaBien = concat(subseq(q_0.colaBien, 0, q.posEnCola(ID) - 1),$
$subseq(q_0.colaBien, q_0.posEnCola(ID) + 1, |q_o.colaBien|)\}$
`asegura` $\{p \in q_0.colaProle \longrightarrow_L q.colaProle = concat(subseq(q_0.colaProle, 0, q.posEnCola(ID) - 1),$
$subseq(q_0.colaProle, q_0.posEnCola(ID) + 1, |q_o.colaProle|)\}$