MODULE *Implementation*

ALGORITHM:

0: CHOOSE $i, j \in [1 \,..\, n]$: $Find\_p(i) \lor Unite\_p(i, j)$

$Find\_p(c)$:
$F1$: $u = c$
$F2$: if $F[u].bit = 1$ *goto* FR ELSE *goto* $F3$
$F3$: a $= READ(u)$; goto $F4$ or $F7$
$F4$: $b = READ(a.parent)$
$F5$: if $b.bit = 1$: $u = a.parent$; goto FR
$F6$: $CAS(F[u], [a.parent, a.rank, 0], [b.parent, a.rank, 0])$; goto $F2$
$F7$: $u = v$; goto $F2$
FR: return u

$Unite\_p(c, d)$:
$U1$: $u = c$; $v = d$
$u = Find\_p(u)$
$U2$: $v = Find\_p(v)$
$U3$: if $u = v$ goto UR
$U4$: a $= [u\_p, u\_r, u\_b] = READ(u)$
$U5$: $b = [v\_p, v\_r, v\_b] = READ(v)$
$U6$: if $a.rank < b.rank$ then $CAS(F[u], [a.parent, a.rank, 1], [v, a.rank, 0])$
$U6$: elif $u\_r > v\_r$ then $CAS(F[v], [b.parent, b.rank, 1], [u, b.rank, 0])$
$U6$: else:
$U6$: if $u < v$ then $CAS(F[u], [a.parent, a.rank, 1], [v, a.rank, \$])$
$U6$: else: $CAS(F[v], [b.parent, b.rank, 1], [u, b.rank, \$])$
$U7$: $u = Find\_p(u)$
$U8$: $v = Find\_p(v)$; goto $U3$
UR: return ACK

EXTENDS *FiniteSets*, *Integers*
CONSTANT BOT, ACK, $PROCESSES$, N
VARIABLES pc, F, u_F, a_F, b_F, u_U, v_U, a_U, b_U, c, d, M
$NodeSet \triangleq 1 \,..\, N$
ASSUME $NisNat \triangleq (N \in Nat) \land (N > 0)$
ASSUME $AckBotDef \triangleq BOT \notin NodeSet \land ACK \notin NodeSet \land BOT \neq ACK$
ASSUME $ExistProc \triangleq PROCESSES \neq \{\}$

Line Definitions
$varlist \triangleq \langle pc, F, u\_F, a\_F, b\_F, u\_U, v\_U, a\_U, b\_U, c, d, M \rangle$

Type Sets
$PCSet \triangleq$ {"0", "F1", "F2", "F3", "F4", "F5", "F6", "F7", "FR",
"U1", "U2", "U3", "U4", "U5", "U6", "U7", "U8", "UR",

```
                                "F1U1", "F2U1", "F3U1", "F4U1", "F5U1", "F6U1", "F7U1", "F8U1", "FRU1",
                                "F1U2", "F2U2", "F3U2", "F4U2", "F5U2", "F6U2", "F7U2", "F8U2", "FRU2",
                                "F1U7", "F2U7", "F3U7", "F4U7", "F5U7", "F6U7", "F7U7", "F8U7", "FRU7",
                                "F1U8", "F2U8", "F3U8", "F4U8", "F5U8", "F6U8", "F7U8", "F8U8", "FRU8" }
FieldSet  ≜   [parent : NodeSet, rank : Nat, bit : {0, 1}]
StateSet  ≜   {A ∈ [NodeSet → NodeSet] : ∀ i ∈ NodeSet : A[A[i]] = A[i]}
ReturnSet ≜   [PROCESSES → NodeSet ∪ {BOT} ∪ {ACK}]
OpSet     ≜   [PROCESSES → {"F", "U", BOT}]
ArgSet    ≜   [PROCESSES → {BOT} ∪ NodeSet ∪ NodeSet × NodeSet]
Configs   ≜   [sigma : StateSet, ret : ReturnSet, op : OpSet, arg : ArgSet]

 InitStates
InitState ≜   [i ∈ NodeSet ↦ i]
InitF     ≜   [i ∈ NodeSet ↦ [parent ↦ i, rank ↦ 0, bit ↦ 1]]
InitRet   ≜   [p ∈ PROCESSES ↦ BOT]
InitOp    ≜   [p ∈ PROCESSES ↦ BOT]
InitArg   ≜   [p ∈ PROCESSES ↦ BOT]

 Initial state of algorithm
Init ≜   ∧ pc = [p ∈ PROCESSES ↦ "0"]
         ∧ F  = InitF
         ∧ a_F ∈ [PROCESSES → FieldSet]
         ∧ b_F ∈ [PROCESSES → FieldSet]
         ∧ u_F ∈ [PROCESSES → NodeSet]
         ∧ a_U ∈ [PROCESSES → FieldSet]
         ∧ b_U ∈ [PROCESSES → FieldSet]
         ∧ u_U ∈ [PROCESSES → NodeSet]
         ∧ v_U ∈ [PROCESSES → NodeSet]
         ∧ c   ∈ [PROCESSES → NodeSet]
         ∧ d   ∈ [PROCESSES → NodeSet]
         ∧ M = {[sigma ↦ InitState,  ret ↦ InitRet, op ↦ InitOp, arg ↦ InitArg]}

 Find operation
F1(p) ≜   ∧ u_F' = [u_F EXCEPT ![p] = c[p]]
          ∧  ∨  pc[p] = "F1"     ∧ pc' = [pc EXCEPT ![p] = "F2"]
             ∨  pc[p] = "F1U1"   ∧ pc' = [pc EXCEPT ![p] = "F2U1"]
             ∨  pc[p] = "F1U2"   ∧ pc' = [pc EXCEPT ![p] = "F2U2"]
             ∨  pc[p] = "F1U7"   ∧ pc' = [pc EXCEPT ![p] = "F2U7"]
             ∨  pc[p] = "F1U8"   ∧ pc' = [pc EXCEPT ![p] = "F2U8"]
          ∧ UNCHANGED ⟨F, a_F, b_F, u_U, v_U, a_U, b_U, c, d, M⟩

F2(p) ≜   ∧ IF F[u_F[p]].bit = 1
            THEN   ∨ pc[p] = "F2"     ∧ pc' = [pc EXCEPT ![p] = "FR"]
                   ∨ pc[p] = "F2U1"   ∧ pc' = [pc EXCEPT ![p] = "FRU1"]
                   ∨ pc[p] = "F2U2"   ∧ pc' = [pc EXCEPT ![p] = "FRU2"]
```

```
                                  ∨ pc[p] = "F2U7"   ∧ pc' = [pc EXCEPT ![p] = "FRU7"]
                                  ∨ pc[p] = "F2U8"   ∧ pc' = [pc EXCEPT ![p] = "FRU8"]
                          ELSE    ∨ pc[p] = "F2"     ∧ pc' = [pc EXCEPT ![p] = "F3"]
                                  ∨ pc[p] = "F2U1"   ∧ pc' = [pc EXCEPT ![p] = "F3U1"]
                                  ∨ pc[p] = "F2U2"   ∧ pc' = [pc EXCEPT ![p] = "F3U2"]
                                  ∨ pc[p] = "F2U7"   ∧ pc' = [pc EXCEPT ![p] = "F3U7"]
                                  ∨ pc[p] = "F2U8"   ∧ pc' = [pc EXCEPT ![p] = "F3U8"]
    ∧ IF F[u_F[p]].bit = 1 ∧ pc[p] = "F2"
          THEN M' =   {t ∈ Configs : ∃ told ∈ M : ∧ told.ret[p] = BOT
                                                  ∧ t.sigma = told.sigma
                                                  ∧ t.ret = [told.ret EXCEPT ![p] = u_F[p]
                                                  ∧ t.op = told.op
                                                  ∧ t.arg = told.arg}
          ELSE  M' = M
    ∧ UNCHANGED ⟨F, a_F, b_F, u_F, u_U, v_U, a_U, b_U, c, d⟩

F3(p) ≜   ∧  a_F' = [a_F EXCEPT ![p] = F[u_F[p]]]
          ∧  ∨ pc[p] = "F3"     ∧ (pc' = [pc EXCEPT ![p] = "F4"]     ∨ pc' = [pc EXCEPT ![p] = "F7"]
             ∨ pc[p] = "F3U1"   ∧ (pc' = [pc EXCEPT ![p] = "F4U1"]   ∨ pc' = [pc EXCEPT ![p] = "F7U
             ∨ pc[p] = "F3U2"   ∧ (pc' = [pc EXCEPT ![p] = "F4U2"]   ∨ pc' = [pc EXCEPT ![p] = "F7U
             ∨ pc[p] = "F3U7"   ∧ (pc' = [pc EXCEPT ![p] = "F4U7"]   ∨ pc' = [pc EXCEPT ![p] = "F7U
             ∨ pc[p] = "F3U8"   ∧ (pc' = [pc EXCEPT ![p] = "F4U8"]   ∨ pc' = [pc EXCEPT ![p] = "F7U
    ∧ UNCHANGED ⟨F, u_F, b_F, u_U, v_U, a_U, b_U, c, d, M⟩

F4(p) ≜   ∧ b_F' = [b_F EXCEPT ![p] = F[a_F[p].parent]]
          ∧  ∨ pc[p] = "F4"     ∧ pc' = [pc EXCEPT ![p] = "F5"]
             ∨ pc[p] = "F4U1"   ∧ pc' = [pc EXCEPT ![p] = "F5U1"]
             ∨ pc[p] = "F4U2"   ∧ pc' = [pc EXCEPT ![p] = "F5U2"]
             ∨ pc[p] = "F4U7"   ∧ pc' = [pc EXCEPT ![p] = "F5U7"]
             ∨ pc[p] = "F4U8"   ∧ pc' = [pc EXCEPT ![p] = "F5U8"]
    ∧ UNCHANGED ⟨F, u_F, a_F, u_U, v_U, a_U, b_U, c, d, M⟩

F5(p) ≜   ∧ IF b_F[p].bit = 1
              THEN  ∧ u_F' = [u_F EXCEPT ![p] = a_F[p].parent]
                    ∧  ∨ pc[p] = "F5"     ∧ pc' = [pc EXCEPT ![p] = "FR"]
                       ∨ pc[p] = "F5U1"   ∧ pc' = [pc EXCEPT ![p] = "FRU1"]
                       ∨ pc[p] = "F5U2"   ∧ pc' = [pc EXCEPT ![p] = "FRU2"]
                       ∨ pc[p] = "F5U7"   ∧ pc' = [pc EXCEPT ![p] = "FRU7"]
                       ∨ pc[p] = "F5U8"   ∧ pc' = [pc EXCEPT ![p] = "FRU8"]
              ELSE  ∧ u_F' = u_F
                    ∧  ∨ pc[p] = "F5"     ∧ pc' = [pc EXCEPT ![p] = "F6"]
                       ∨ pc[p] = "F5U1"   ∧ pc' = [pc EXCEPT ![p] = "F6U1"]
                       ∨ pc[p] = "F5U2"   ∧ pc' = [pc EXCEPT ![p] = "F6U2"]
                       ∨ pc[p] = "F5U7"   ∧ pc' = [pc EXCEPT ![p] = "F6U7"]
                       ∨ pc[p] = "F5U8"   ∧ pc' = [pc EXCEPT ![p] = "F6U8"]
          ∧ IF b_F[p].bit = 1 ∧ pc[p] = "F5"
```

```
                    THEN M' = {t ∈ Configs : ∃ told ∈ M : ∧ told.ret[p] = BOT
                                                          ∧ t.sigma = told.sigma
                                                          ∧ t.ret = [told.ret EXCEPT ![p] = a_F[p].pc
                                                          ∧ t.op = told.op
                                                          ∧ t.arg = told.arg}
                    ELSE  M' = M
                ∧ UNCHANGED ⟨F, a_F, b_F, u_F, u_U, v_U, a_U, b_U, c, d⟩

F6(p) ≜     ∧ IF (F[u_F[p]] = [parent ↦ a_F[p].parent, rank ↦ a_F[p].rank, bit ↦ 0])
                THEN   ∧ F' = [F EXCEPT ![u_F[p]] = [parent ↦ b_F[p].parent, rank ↦ a_F[p].r
                ELSE   ∧ F' = F
            ∧   ∨ pc[p] = "F6"      ∧ pc' = [pc EXCEPT ![p] = "F2"]
                ∨ pc[p] = "F6U1"    ∧ pc' = [pc EXCEPT ![p] = "F2U1"]
                ∨ pc[p] = "F6U2"    ∧ pc' = [pc EXCEPT ![p] = "F2U2"]
                ∨ pc[p] = "F6U7"    ∧ pc' = [pc EXCEPT ![p] = "F2U7"]
                ∨ pc[p] = "F6U8"    ∧ pc' = [pc EXCEPT ![p] = "F2U8"]
            ∧ UNCHANGED ⟨a_F, b_F, u_F, u_U, v_U, a_U, b_U, c, d, M⟩

F7(p) ≜     ∧ u_F' = [u_F EXCEPT ![p] = a_F[p].parent]
            ∧   ∨ pc[p] = "F7"      ∧ pc' = [pc EXCEPT ![p] = "F2"]
                ∨ pc[p] = "F7U1"    ∧ pc' = [pc EXCEPT ![p] = "F2U1"]
                ∨ pc[p] = "F7U2"    ∧ pc' = [pc EXCEPT ![p] = "F2U2"]
                ∨ pc[p] = "F7U7"    ∧ pc' = [pc EXCEPT ![p] = "F2U7"]
                ∨ pc[p] = "F7U8"    ∧ pc' = [pc EXCEPT ![p] = "F2U8"]
            ∧ UNCHANGED ⟨F, a_F, b_F, u_U, v_U, a_U, b_U, c, d, M⟩

FR(p) ≜     ∧ pc' = [pc EXCEPT ![p] = "0"]
            ∧   ∨ pc[p] = "FR"      ∧ pc'  = [pc EXCEPT ![p] = "0"]
                                    ∧ u_U' = u_U
                                    ∧ M'   = {t ∈ Configs :   ∃ told ∈ M :  ∧ told.ret[p] = u_F[p]
                                                                            ∧ t.sigma = told.sigma
                                                                            ∧ t.ret = [told.ret EXCEPT !
                                                                            ∧ t.op = [told.op EXCEPT ![
                                                                            ∧ t.arg = [told.arg EXCEPT
                ∨ pc[p] = "FRU1"    ∧ pc'  = [pc EXCEPT ![p] = "U2"]
                                    ∧ u_U' = [u_U EXCEPT ![p] = u_F[p]]
                                    ∧ M'   = M
                ∨ pc[p] = "FRU2"    ∧ pc'  = [pc EXCEPT ![p] = "U3"]
                                    ∧ v_U' = [v_U EXCEPT ![p] = u_F[p]]
                                    ∧ M'   = M
                ∨ pc[p] = "FRU7"    ∧ pc'  = [pc EXCEPT ![p] = "U8"]
                                    ∧ u_U' = [u_U EXCEPT ![p] = u_F[p]]
                                    ∧ M'   = M
                ∨ pc[p] = "FRU8"    ∧ pc'  = [pc EXCEPT ![p] = "U3"]
                                    ∧ v_U' = [v_U EXCEPT ![p] = u_F[p]]
```

```
                                        ∧ M′    = M
                    ∧ UNCHANGED ⟨F, a_F, b_F, u_F, u_U, v_U, a_U, b_U, c, d⟩

U1(p) ≜             ∧ pc[p] = "U1"
                    ∧ pc′    = [pc EXCEPT ![p] = "F1U1"]
                    ∧ u_U′ = [u_U EXCEPT ![p] = c[p]]
                    ∧ v_U′ = [v_U EXCEPT ![p] = d[p]]
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, a_U, b_U, c, d, M⟩

U2(p) ≜             ∧ pc[p] = "U2"
                    ∧ pc′   = [pc EXCEPT ![p] = "F1U2"]
                    ∧ c′    = [c EXCEPT ![p] = v_U[p]]
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, a_U, b_U, u_U, v_U, d, M⟩

U3(p) ≜             ∧ pc[p] = "U3"
                    ∧ IF u_U[p] = v_U[p]
                          THEN    ∧ pc′ = [pc EXCEPT ![p] = "UR"]
                                  ∧ M′ = {t ∈ Configs : ∃ told ∈ M :
                                                  ∨ ∧ told.ret[p] = BOT
                                                    ∧ t.sigma = told.sigma
                                                    ∧ t.ret = [told.ret EXCEPT ![p] = ACK]
                                                    ∧ t.op = told.op
                                                    ∧ t.arg = told.arg
                                                  ∨ ∧ told.ret[p] = ACK
                                                    ∧ t.sigma = told.sigma
                                                    ∧ t.ret = told.ret
                                                    ∧ t.op = told.op
                                                    ∧ t.arg = told.arg}
                          ELSE    ∧ pc′ = [pc EXCEPT ![p] = "U4"]
                                  ∧ M′ = M
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, b_U, u_U, v_U, c, d⟩

U4(p) ≜             ∧ pc[p] = "U4"
                    ∧ pc′ = [pc EXCEPT ![p] = "U5"]
                    ∧ a_U′ = [a_U EXCEPT ![p] = F[u_U[p]]]
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, b_U, c, d, M⟩

U5(p) ≜             ∧ pc[p] = "U5"
                    ∧ pc′ = [pc EXCEPT ![p] = "U6"]
                    ∧ b_U′ = [b_U EXCEPT ![p] = F[v_U[p]]]
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, c, d, M⟩

U6(p) ≜             ∧ pc[p] = "U6"
                    ∧  IF a_U[p].rank < b_U[p].rank
                          THEN IF F[u_U[p]] = [parent ↦ a_U[p].parent, rank ↦ a_U[p].rank, bit ↦ 1]
                                  THEN    ∧ F′ = [F EXCEPT ![u_U[p]] = [parent ↦ v_U[p], rank ↦
                                          ∧ M′ = {t ∈ Configs : ∃ told ∈ M : ∧ t.ret = [told.ret EXCE
```

$\land\ t.sigma = [i \in Node$

$\land\ t.op = told.op$
$\land\ t.arg = told.arg\}$

ELSE $\land\ F' = F$
$\land\ M' = M$

ELSE IF $a\_U[p].rank < b\_U[p].rank$

THEN IF $F[v\_U[p]] = [parent \mapsto b\_U[p].parent,\ rank \mapsto b\_U[p].rank,\ bit \mapsto 1]$

THEN $\land\ F' = [F$ EXCEPT $![v\_U[p]] = [parent \mapsto u\_U[p],\ rank \mapsto$
$\land\ M' = \{t \in Configs : \exists\, told \in M :\ \land\ t.ret = [told.ret$ EXC
$\land\ t.sigma = [i \in Nod$

$\land\ t.op = told.op$
$\land\ t.arg = told.arg\}$

ELSE $\land\ F' = F$
$\land\ M' = M$

ELSE

IF $u\_U[p] < v\_U[p]$ ranks are equal

THEN IF $F[u\_U[p]] = [parent \mapsto a\_U[p].parent,\ rank \mapsto a\_U[p].rank,\ b$

THEN $\lor\ \land\ F' = [F$ EXCEPT $![u\_U[p]] = [parent \mapsto v\_U[$
$\land\ M' = \{t \in Configs : \exists\, told \in M :\ \land\ t.ret = [to$
$\land\ t.sigma =$

$\land\ t.op = to$
$\land\ t.arg = t$

$\lor\ \land\ F' = [F$ EXCEPT $![u\_U[p]] = [parent \mapsto v\_U[$
$\land\ M' = M$

ELSE $\land\ F' = F$
$\land\ M' = M$

ELSE IF $F[v\_U[p]] = [parent \mapsto b\_U[p].parent,\ rank \mapsto b\_U[p].rank,\ b$

THEN $\lor\ \land\ F' = [F$ EXCEPT $![v\_U[p]] = [parent \mapsto u\_U[$
$\land\ M' = \{t \in Configs : \exists\, told \in M :\ \land\ t.ret = [to$
$\land\ t.sigma =$

$\land\ t.op = to$
$\land\ t.arg = t$

$\lor\ \land\ F' = F$
$\land\ M' = M$

ELSE $\land\ F' = F$
$\land\ M' = M$

$\land\ pc' = [pc$ EXCEPT $![p] =$ "U7"]

```
                    ∧ UNCHANGED ⟨u_F, a_F, b_F, u_U, v_U, a_U, b_U, c, d⟩

U7(p) ≜             ∧ pc[p] = "U7"
                    ∧ pc′ = [pc EXCEPT ![p] = "F1U7"]
                    ∧ c′ = [c EXCEPT ![p] = u_U[p]]
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, b_U, d, M⟩

U8(p) ≜             ∧ pc[p] = "U8"
                    ∧ pc′ = [pc EXCEPT ![p] = "F1U8"]
                    ∧ c′ = [c EXCEPT ![p] = v_U[p]]
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, b_U, d, M⟩

UR(p) ≜             ∧ pc[p] = "UR"
                    ∧ pc′ = [pc EXCEPT ![p] = "0"]
                    ∧ M′ = {t ∈ Configs : ∃ told ∈ M : ∧ told.ret[p] = ACK
                                                       ∧ t.sigma = told.sigma
                                                       ∧ t.ret = [told.ret EXCEPT ![p] = BOT]
                                                       ∧ t.op = [told.op EXCEPT ![p] = BOT]
                                                       ∧ t.arg = [told.arg EXCEPT ![p] = BOT]}
                    ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, b_U, c, d⟩

Decide(p) ≜         ∧ pc[p] = "0"
                    ∧  ∨  ∧ pc′ = [pc EXCEPT ![p] = "F1"]
                          ∧ ∃ i ∈ NodeSet :    ∧ c′  = [c EXCEPT ![p] = i]
                                               ∧ M′ = {t ∈ Configs : ∃ told ∈ M : ∧ told.ret[p] = BOT
                                                                                  ∧ told.op[p] = BOT
                                                                                  ∧ told.arg[p] = BOT
                                                                                  ∧ t.sigma = told.sigma
                                                                                  ∧ t.op = [told.op EXCEPT
                                                                                  ∧ t.arg = [told.arg EXCEPT
                                                                                  ∧ t.ret = told.ret}
                          ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, b_U, d⟩
                       ∨  ∧ pc′ = [pc EXCEPT ![p] = "U1"]
                          ∧ ∃ i ∈ NodeSet : ∃ j ∈ NodeSet :
                               ∧ c′  = [c EXCEPT ![p] = i]
                               ∧ d′  = [d EXCEPT ![p] = j]
                               ∧ M′ = {t ∈ Configs : ∃ told ∈ M : ∧ told.ret[p] = BOT
                                                                  ∧ told.op[p] = BOT
                                                                  ∧ told.arg[p] = BOT
                                                                  ∧ t.sigma = told.sigma
                                                                  ∧ t.op = [told.op EXCEPT ![p] = "U"]
                                                                  ∧ t.arg = [told.arg EXCEPT ![p] = ⟨i, j⟩]
                                                                  ∧ t.ret = told.ret}
                          ∧ UNCHANGED ⟨F, u_F, a_F, b_F, u_U, v_U, a_U, b_U⟩

Step(p) ≜    ∨ F1(p)
```

$$\lor\ F2(p)$$
$$\lor\ F3(p)$$
$$\lor\ F4(p)$$
$$\lor\ F5(p)$$
$$\lor\ F6(p)$$
$$\lor\ F7(p)$$
$$\lor\ FR(p)$$
$$\lor\ U1(p)$$
$$\lor\ U2(p)$$
$$\lor\ U3(p)$$
$$\lor\ U4(p)$$
$$\lor\ U5(p)$$
$$\lor\ U6(p)$$
$$\lor\ U7(p)$$
$$\lor\ U8(p)$$
$$\lor\ UR(p)$$
$$\lor\ Decide(p)$$

$$Next \triangleq \exists\, p \in PROCESSES : Step(p)$$

$$Spec \triangleq Init \land \Box[Next]_{varlist}$$

* Modification History
* Last modified *Tue Apr* 22 17:47:15 *EDT* 2025 by *karunram*
* Created *Thu Apr* 03 12:26:37 *EDT* 2025 by *karunram*