fischer-inv[11,40]

```
fischer-inv{i:l,
            $x:ut2,
            $try:ut2,
            $taken:ut2,
            $contending:ut2,
            $free:ut2,
            $mine:ut2,
            $wanted:ut2,
            $z:ut2}
           (es; L; del; e)
≡def  ((es-after(es; mkid{$x:ut2}; e) = mkid{$mine:ut2})
      ⇒ (l_all(L;
               Id;
               j.((¬(j = loc(e)))
               ⇒ ∃e′@j.f-event{$x:ut2}
                            (es; L; e′)
                        ∧ (f-round{i:l}
                                (mkid{$x:ut2}; mkid{$free:ut2}; es; e′)
                           =
                           f-round{i:l}
                                (mkid{$x:ut2}; mkid{$free:ut2}; es; e))
                        ∧ (es-after(es; mkid{$x:ut2}; e′) = mkid{$taken:ut2})
                        ∧ alle-at(es;
                                  j;
                                  e″.((f-round{i:l}
                                            (mkid{$x:ut2}; mkid{$free:ut2}; es; e″
                                             ) ≤ f-round{i:l}
                                                   (mkid{$x:ut2};
                                                    mkid{$free:ut2};
                                                    es;
                                                    e′))
                                  ⇒ es-le(es; e″; e′)))))
          ∧ (∀e′:es-E(es).
             f-event{$x:ut2}
                   (es; L; e′)
             ⇒ (f-round{i:l}
                      (mkid{$x:ut2}; mkid{$free:ut2}; es; e′) ≤ f-round{i:l}
                                                                 (mkid{$x:ut2};
                                                                  mkid{$free:ut2};
                                                                  es;
                                                                  e))
             ⇒ (qle((es-time(es; e′) + del); es-time(es; e)) ∨ (e′ = e)))))
```

$\wedge$ (f-newround{\$x:ut2, \$free:ut2, \$mine:ut2}
(es; L; e)
$\Rightarrow$ ($\forall e'$:es-E(es).
(loc(e') $\in$ L)
$\Rightarrow$ @e'(mkid{\$x:ut2}$\rightarrow$mkid{\$free:ut2})
$\Rightarrow$ ($\neg$(loc(e') = loc(e)))
$\Rightarrow$ ($\uparrow$es-isrcv(es; e'))
$\Rightarrow$ (es-tag(es; e') = mkid{\$free:ut2})
$\Rightarrow$ (es-lnk(es; e') = <loc(e), loc(e'), mkid{\$z:ut2}>)
$\Rightarrow$ (es-sender(es; e') = e)
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e')
=
f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e))))
$\wedge$ (@e(mkid{\$x:ut2}$\rightarrow$mkid{\$try:ut2})
$\Rightarrow$ ($\forall e'$:es-E(es).
($\neg$(loc(e') = loc(e)))
$\Rightarrow$ ($\uparrow$es-isrcv(es; e'))
$\Rightarrow$ (es-tag(es; e') = mkid{\$wanted:ut2})
$\Rightarrow$ (es-lnk(es; e') = <loc(e), loc(e'), mkid{\$z:ut2}>)
$\Rightarrow$ (es-sender(es; e') = e)
$\Rightarrow$ ((@e'(mkid{\$x:ut2}$\rightarrow$mkid{\$taken:ut2})
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e')
=
f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e)))
$\wedge$ (@e'(mkid{\$x:ut2}$\rightarrow$mkid{\$contending:ut2})
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e')
=
inc-snd(f-rank{i:l}(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e)))))))
$\wedge$ ($\forall e'$:es-E(es).
f-event{\$x:ut2}
(es; L; e')
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e')
=
f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; es; e))
$\Rightarrow$ qle(qdist(es-time(es; e); es-time(es; e')); del))

clarification:

fischer-inv{i:l,

```
            $x:ut2,
            $try:ut2,
            $taken:ut2,
            $contending:ut2,
            $free:ut2,
            $mine:ut2,
            $wanted:ut2,
            $z:ut2}
          (es; L; del; e)
≡def  ((es-after(es; mkid{$x:ut2}; e) = mkid{$mine:ut2} ∈ Id)
      ⇒ (l_all(L;
               Id;
               j.((¬(j = es-loc(es; e) ∈ Id))
               ⇒ existse-at(es;
                            j;
                            e′.(f-event{$x:ut2}
                                      (es; L; e′)
                            ∧ (f-round{i:l}
                                      (mkid{$x:ut2}; mkid{$free:ut2}; es; e′)
                               =
                               f-round{i:l}
                                      (mkid{$x:ut2}; mkid{$free:ut2}; es; e)
                               ∈ ℕ)
                            ∧ (es-after(es; mkid{$x:ut2}; e′) = mkid{$taken:ut2} ∈ Id)
                            ∧ alle-at(es;
                                      j;
                                      e″.((f-round{i:l}
                                                  (mkid{$x:ut2};
                                                   mkid{$free:ut2};
                                                   es;
                                                   e″) ≤ f-round{i:l}
                                                            (mkid{$x:ut2};
                                                             mkid{$free:ut2};
                                                             es;
                                                             e′))
                                      ⇒ es-le(es; e″; e′)))))))
          ∧ (∀e′:es-E(es).
            f-event{$x:ut2}
                  (es; L; e′)
            ⇒ (f-round{i:l}
                      (mkid{$x:ut2}; mkid{$free:ut2}; es; e′) ≤ f-round{i:l}
                                                                (mkid{$x:ut2};
                                                                 mkid{$free:ut2};
                                                                 es;
                                                                 e))
            ⇒ (qle((es-time(es; e′) + del); es-time(es; e)) ∨ (e′ = e ∈ es-E(es))))))
```

$\wedge$ (f-newround{\$x:ut2, \$free:ut2, \$mine:ut2}
(*es*; *L*; *e*)
$\Rightarrow$ ($\forall e'$:es-E(*es*).
(es-loc(*es*; e') $\in$ *L* $\in$ Id)
$\Rightarrow$ es-change-to(*es*;Id;mkid{\$x:ut2};e';mkid{\$free:ut2})
$\Rightarrow$ ($\neg$(es-loc(*es*; e') = es-loc(*es*; *e*) $\in$ Id))
$\Rightarrow$ ($\uparrow$es-isrcv(*es*; e'))
$\Rightarrow$ (es-tag(*es*; e') = mkid{\$free:ut2} $\in$ Id)
$\Rightarrow$ (es-lnk(*es*; e') = <es-loc(*es*; *e*), es-loc(*es*; e'), mkid{\$z:ut2}> $\in$ IdLnk)
$\Rightarrow$ (es-sender(*es*; e') = *e* $\in$ es-E(*es*))
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; e')
=
f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; *e*)
$\in$ (:$\mathbb{N} \times \mathbb{N}$))))
$\wedge$ (es-change-to(*es*;Id;mkid{\$x:ut2};*e*;mkid{\$try:ut2})
$\Rightarrow$ ($\forall e'$:es-E(*es*).
($\neg$(es-loc(*es*; e') = es-loc(*es*; *e*) $\in$ Id))
$\Rightarrow$ ($\uparrow$es-isrcv(*es*; e'))
$\Rightarrow$ (es-tag(*es*; e') = mkid{\$wanted:ut2} $\in$ Id)
$\Rightarrow$ (es-lnk(*es*; e') = <es-loc(*es*; *e*), es-loc(*es*; e'), mkid{\$z:ut2}> $\in$ IdLnk)
$\Rightarrow$ (es-sender(*es*; e') = *e* $\in$ es-E(*es*))
$\Rightarrow$ ((es-change-to(*es*;Id;mkid{\$x:ut2};e';mkid{\$taken:ut2})
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; e')
=
f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; *e*)
$\in$ (:$\mathbb{N} \times \mathbb{N}$)))
$\wedge$ (es-change-to(*es*;Id;mkid{\$x:ut2};e';mkid{\$contending:ut2})
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; e')
=
inc-snd(f-rank{i:l}(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; *e*))
$\in$ (:$\mathbb{N} \times \mathbb{N}$))))))
$\wedge$ ($\forall e'$:es-E(*es*).
f-event{\$x:ut2}
(*es*; *L*; e')
$\Rightarrow$ (f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; e')
=
f-rank{i:l}
(mkid{\$x:ut2}; mkid{\$free:ut2}; *es*; *e*)
$\in$ (:$\mathbb{N} \times \mathbb{N}$))
$\Rightarrow$ qle(qdist(es-time(*es*; *e*); es-time(*es*; e')); *del*))

http://www.nuprl.org/FDLcontent/p0_963683_/p74_433552_{fischer-inv}.html