```
better-w-sends-wf  THM

⊢  ∀the_w:World, l:IdLnk, e:E. sends(l;e) ∈ Msg_sub(l; the_w.M) List
| by Unfold ‘w-sends‘ 0 THEN UnivCD
|    THEN
|    Unfolds [‘Msg_sub‘;‘w-onlnk‘] 0
|    THEN
|    Unfold ‘haslink‘ 0
|
1. the_w : World
2. l : IdLnk
3. e : E
⊢  filter(λms.mlnk(ms) = l;m(loc(e);time(e)))
|  ∈ {m:Msg(the_w.M)| mlnk(m) = l } List
| by GenConclAtAddr [2;2] THEN Assert (v ∈ Msg(the_w.M) List)
|
|\
| .....assertion .....
| 1. the_w : World
| 2. IdLnk
| 3. e : E
| 4. v : {m:Msg(the_w.M)| source(mlnk(m)) = loc(e) } List
| 5. m(loc(e);time(e)) = v
| ⊢  v ∈ Msg(the_w.M) List  by Auto
|
.\
. 1. the_w : World
. 2. l : IdLnk
. 3. e : E
. 4. v : {m:Msg(the_w.M)| source(mlnk(m)) = loc(e) } List
. 5. m(loc(e);time(e)) = v
. 6. v ∈ Msg(the_w.M) List
. ⊢  filter(λms.mlnk(ms) = l;v) ∈ {m:Msg(the_w.M)| mlnk(m) = l } List
. | by SubsumeC ({m:Msg(the_w.M)| (λms.mlnk(ms) = l)(m) } List)
. |
. |\
. | 1. the_w : World
. | 2. l : IdLnk
. | 3. e : E
. | 4. v : {m:Msg(the_w.M)| source(mlnk(m)) = loc(e) } List
. | 5. m(loc(e);time(e)) = v
. | 6. v ∈ Msg(the_w.M) List
. | ⊢  filter(λms.mlnk(ms) = l;v)
. |    ∈ {m:Msg(the_w.M)| (λms.mlnk(ms) = l)(m) } List
. |   by BackThru
. |      Thm* ∀T:Type, P:(T→𝔹), l:T List.
. |           filter(P;l) ∈ {x:T| P(x) } List
. |      {filter_type}
. |
. .\
. . .....subtype_rel .....
. . 1. the_w : World
. . 2. l : IdLnk
. . 3. e : E
. . 4. v : {m:Msg(the_w.M)| source(mlnk(m)) = loc(e) } List
```

```
. . 5. m(loc(e);time(e)) = v
. . 6. v ∈ Msg(the_w.M) List
. . ⊢  ({m:Msg(the_w.M)
. . |    | (λms.mlnk(ms) = l)(m) } List) ⊆r ({m:Msg(the_w.M)
. . |                                        | mlnk(m) = l } List)
. . | by Reduce 0 THEN UseSubtypeRelLemmas THEN Analyze 0
. . |    THEN
. . |    Analyze -1
. . |    THEN
. . |    Analyze
. . |
. . .....set predicate .....
. . 1. the_w : World
. . 2. l : IdLnk
. . 3. e : E
. . 4. v : {m:Msg(the_w.M)| source(mlnk(m)) = loc(e) } List
. . 5. m(loc(e);time(e)) = v
. . 6. v ∈ Msg(the_w.M) List
. . 7. x : Msg(the_w.M)
. . 8. mlnk(x) = l
. . ⊢  mlnk(x) = l
. .   by RWO
. .      Thm* ∀a,b:IdLnk. a = b ⟺ a = b    {assert_DASH_eq_DASH_lnk}
. .      -1
END
```