```
better-sframe-rule  THM

⊢  ∀L:Knd List, l:IdLnk, tg:Id.
|  @source(l): only L sends on (l with tg) ∈ Dsys
|  & (∀D:Dsys.
|     @source(l): only L sends on (l with tg) ⊆ D
|     ⇒
|     D
|     realizes es.∀e:E.
|                 loc(e) = destination(l)
|                 ⇒
|                 kind(e) = rcv(l; tg) ⇒ (kind(sender(e)) ∈ L))
| by UnivCD THEN Inst Thm: s_DASH_sframe_DASH_rule [source(l);L;l;tg]
|    THEN
|    Analyze -1
|    THEN
|    Analyze 0
|    THEN
|    Try Trivial
|    THEN
|    ParallelOp -2
|    THEN
|    RepeatFor 7 ParallelLast
|    THEN
|    BackThruSomeHyp
|    THEN
|    MaAuto
|
1. L : Knd List
2. l : IdLnk
3. tg : Id
4. @source(l): only L sends on (l with tg) ∈ Dsys
5. ∀D:Dsys.
|  @source(l): only L sends on (l with tg) ⊆ D
|  ⇒
|  D
|  realizes es.∀e:E.
|              loc(e) = source(l)
|              ⇒
|              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
6. @source(l): only L sends on (l with tg) ∈ Dsys
7. D : Dsys
8. @source(l): only L sends on (l with tg) ⊆ D
9. D' : Dsys
10. D ⊆ D'
11. w : World
12. p : FairFifo
13. PossibleWorld(D';w)
14. ∀e:E. loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
15. e : E
16. loc(e) = destination(l)
17. kind(e) = rcv(l; tg)
⊢  ¬null(sends(l,tg,sender(e)))  by RW assert_pushdownC 0 THEN MaAuto
|
1. L : Knd List
```

```
2. l : IdLnk
3. tg : Id
4. @source(l): only L sends on (l with tg) ∈ Dsys
5. ∀D:Dsys.
|  @source(l): only L sends on (l with tg) ⊆ D
|  ⇒
|  D
|  realizes es.∀e:E.
|              loc(e) = source(l)
|              ⇒
|              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
6. @source(l): only L sends on (l with tg) ∈ Dsys
7. D : Dsys
8. @source(l): only L sends on (l with tg) ⊆ D
9. D' : Dsys
10. D ⊆ D'
11. w : World
12. p : FairFifo
13. PossibleWorld(D';w)
14. ∀e:E. loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
15. e : E
16. loc(e) = destination(l)
17. kind(e) = rcv(l; tg)
⊢  ¬sends(l,tg,sender(e)) = nil  by Analyze 0 THENA MaAuto
|
1. L : Knd List
2. l : IdLnk
3. tg : Id
4. @source(l): only L sends on (l with tg) ∈ Dsys
5. ∀D:Dsys.
|  @source(l): only L sends on (l with tg) ⊆ D
|  ⇒
|  D
|  realizes es.∀e:E.
|              loc(e) = source(l)
|              ⇒
|              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
6. @source(l): only L sends on (l with tg) ∈ Dsys
7. D : Dsys
8. @source(l): only L sends on (l with tg) ⊆ D
9. D' : Dsys
10. D ⊆ D'
11. w : World
12. p : FairFifo
13. PossibleWorld(D';w)
14. ∀e:E. loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
15. e : E
16. loc(e) = destination(l)
17. kind(e) = rcv(l; tg)
18. sends(l,tg,sender(e)) = nil
⊢  False  by InstESAxiom ES(w) [e] 9 THEN MaAuto
|
1. L : Knd List
2. l : IdLnk
3. tg : Id
```

```
4. @source(l): only L sends on (l with tg) ∈ Dsys
5. ∀D:Dsys.
|  @source(l): only L sends on (l with tg) ⊆ D
|  ⇒
|  D
|  realizes es.∀e:E.
|              loc(e) = source(l)
|              ⇒
|              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
6. @source(l): only L sends on (l with tg) ∈ Dsys
7. D : Dsys
8. @source(l): only L sends on (l with tg) ⊆ D
9. D' : Dsys
10. D ⊆ D'
11. w : World
12. p : FairFifo
13. PossibleWorld(D';w)
14. ∀e:E. loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
15. e : E
16. loc(e) = destination(l)
17. kind(e) = rcv(l; tg)
18. sends(l,tg,sender(e)) = nil
19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
⊢  False
| by Assert (msg(lnk(e);tag(e);val(e)) ∈ sends(l,tg,sender(e)))
|
|\
| .....assertion .....
| 1. L : Knd List
| 2. l : IdLnk
| 3. tg : Id
| 4. @source(l): only L sends on (l with tg) ∈ Dsys
| 5. ∀D:Dsys.
| |  @source(l): only L sends on (l with tg) ⊆ D
| |  ⇒
| |  D
| |  realizes es.∀e:E.
| |              loc(e) = source(l)
| |              ⇒
| |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| 6. @source(l): only L sends on (l with tg) ∈ Dsys
| 7. D : Dsys
| 8. @source(l): only L sends on (l with tg) ⊆ D
| 9. D' : Dsys
| 10. D ⊆ D'
| 11. w : World
| 12. p : FairFifo
| 13. PossibleWorld(D';w)
| 14. ∀e:E.
| |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| 15. e : E
| 16. loc(e) = destination(l)
| 17. kind(e) = rcv(l; tg)
| 18. sends(l,tg,sender(e)) = nil
| 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
```

```
| ⊢  (msg(lnk(e);tag(e);val(e)) ∈ sends(l,tg,sender(e)))
| | by Unfold ‘es-tg-sends‘ 0
| |    THEN
| |    RW0
| |    Thm* ∀P:(T→ℙ), L:T List, x:T.
| |         (x ∈ filter(P;L)) ⟺ (x ∈ L) & P(x)
| |    {member_filter}
| |    0
| |    THEN
| |    MaAuto
| |
| |\
| | 1. L : Knd List
| | 2. l : IdLnk
| | 3. tg : Id
| | 4. @source(l): only L sends on (l with tg) ∈ Dsys
| | 5. ∀D:Dsys.
| | |  @source(l): only L sends on (l with tg) ⊆ D
| | |  ⇒
| | |  D
| | |  realizes es.∀e:E.
| | |              loc(e) = source(l)
| | |              ⇒
| | |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | 6. @source(l): only L sends on (l with tg) ∈ Dsys
| | 7. D : Dsys
| | 8. @source(l): only L sends on (l with tg) ⊆ D
| | 9. D’ : Dsys
| | 10. D ⊆ D’
| | 11. w : World
| | 12. p : FairFifo
| | 13. PossibleWorld(D’;w)
| | 14. ∀e:E.
| | |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | 15. e : E
| | 16. loc(e) = destination(l)
| | 17. kind(e) = rcv(l; tg)
| | 18. sends(l,tg,sender(e)) = nil
| | 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
| | ⊢  (msg(lnk(e);tag(e);val(e)) ∈ sends(l;sender(e)))
| | | by Subst’ (l = lnk(e)) 0 THEN MaAuto
| | |
| | |\
| | | 1. L : Knd List
| | | 2. l : IdLnk
| | | 3. tg : Id
| | | 4. @source(l): only L sends on (l with tg) ∈ Dsys
| | | 5. ∀D:Dsys.
| | | |  @source(l): only L sends on (l with tg) ⊆ D
| | | |  ⇒
| | | |  D
| | | |  realizes es.∀e:E.
| | | |              loc(e) = source(l)
| | | |              ⇒
| | | |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
```

```
| | | 6. @source(l): only L sends on (l with tg) ∈ Dsys
| | | 7. D : Dsys
| | | 8. @source(l): only L sends on (l with tg) ⊆ D
| | | 9. D' : Dsys
| | | 10. D ⊆ D'
| | | 11. w : World
| | | 12. p : FairFifo
| | | 13. PossibleWorld(D';w)
| | | 14. ∀e:E.
| | | |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | | 15. e : E
| | | 16. loc(e) = destination(l)
| | | 17. kind(e) = rcv(l; tg)
| | | 18. sends(l,tg,sender(e)) = nil
| | | 19. sends(lnk(e);sender(e))[index(e)]
| | | |   =
| | | |   msg(lnk(e);tag(e);val(e))
| | | ⊢  l = lnk(e)  by Unfold `es-lnk` 0 THEN HypSubst' -3 0
| | | |
| | | 1. L : Knd List
| | | 2. l : IdLnk
| | | 3. tg : Id
| | | 4. @source(l): only L sends on (l with tg) ∈ Dsys
| | | 5. ∀D:Dsys.
| | |    @source(l): only L sends on (l with tg) ⊆ D
| | |    ⇒
| | |    D
| | |    realizes es.∀e:E.
| | |                loc(e) = source(l)
| | |                ⇒
| | |                ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | | 6. @source(l): only L sends on (l with tg) ∈ Dsys
| | | 7. D : Dsys
| | | 8. @source(l): only L sends on (l with tg) ⊆ D
| | | 9. D' : Dsys
| | | 10. D ⊆ D'
| | | 11. w : World
| | | 12. p : FairFifo
| | | 13. PossibleWorld(D';w)
| | | 14. ∀e:E.
| | |     loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | | 15. e : E
| | | 16. loc(e) = destination(l)
| | | 17. kind(e) = rcv(l; tg)
| | | 18. sends(l,tg,sender(e)) = nil
| | | 19. sends(lnk(e);sender(e))[index(e)]
| | |     =
| | |     msg(lnk(e);tag(e);val(e))
| | | ⊢  l = lnk(rcv(l; tg))
| | |   by Unfolds [`lnk`;`rcv`] 0 THEN Reduce 0
| | |
| | .\
| | . 1. L : Knd List
| | . 2. l : IdLnk
| | . 3. tg : Id
```

```
| | . 4. @source(l): only L sends on (l with tg) ∈ Dsys
| | . 5. ∀D:Dsys.
| | . |  @source(l): only L sends on (l with tg) ⊆ D
| | . |  ⇒
| | . |  D
| | . |  realizes es.∀e:E.
| | . |              loc(e) = source(l)
| | . |              ⇒
| | . |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | . 6. @source(l): only L sends on (l with tg) ∈ Dsys
| | . 7. D : Dsys
| | . 8. @source(l): only L sends on (l with tg) ⊆ D
| | . 9. D' : Dsys
| | . 10. D ⊆ D'
| | . 11. w : World
| | . 12. p : FairFifo
| | . 13. PossibleWorld(D';w)
| | . 14. ∀e:E.
| | . |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | . 15. e : E
| | . 16. loc(e) = destination(l)
| | . 17. kind(e) = rcv(l; tg)
| | . 18. sends(l,tg,sender(e)) = nil
| | . 19. sends(lnk(e);sender(e))[index(e)]
| | . |   =
| | . |   msg(lnk(e);tag(e);val(e))
| | . ⊢  (msg(lnk(e);tag(e);val(e)) ∈ sends(lnk(e);sender(e)))

| | . | by Unfold `l_member` 0 THEN InstConcl [index(e)] THEN MaAuto
| | . |
| | . 1. L : Knd List
| | . 2. l : IdLnk
| | . 3. tg : Id
| | . 4. @source(l): only L sends on (l with tg) ∈ Dsys
| | . 5. ∀D:Dsys.
| | .    @source(l): only L sends on (l with tg) ⊆ D
| | .    ⇒
| | .    D
| | .    realizes es.∀e:E.
| | .                loc(e) = source(l)
| | .                ⇒
| | .                ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | . 6. @source(l): only L sends on (l with tg) ∈ Dsys
| | . 7. D : Dsys
| | . 8. @source(l): only L sends on (l with tg) ⊆ D
| | . 9. D' : Dsys
| | . 10. D ⊆ D'
| | . 11. w : World
| | . 12. p : FairFifo
| | . 13. PossibleWorld(D';w)
| | . 14. ∀e:E.
| | .     loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| | . 15. e : E
| | . 16. loc(e) = destination(l)
| | . 17. kind(e) = rcv(l; tg)
| | . 18. sends(l,tg,sender(e)) = nil
```

```
| | . 19. sends(lnk(e);sender(e))[index(e)]
| | .     =
| | .     msg(lnk(e);tag(e);val(e))
| | . ⊢  index(e)<||sends(lnk(e);sender(e))||
| | .   by Inst
| | .      Thm* ∀the_es:ES, e:E.
| | .           isrcv(e) ⇒ index(e) ∈ ℕ||sends(lnk(e);sender(e))||
| | .      {es_DASH_index_wf}
| | .      [ES(w);e]
| | .      THEN
| | .      MaAuto
| | .      THEN
| | .      MaAuto
| |
| .\
| . 1. L : Knd List
| . 2. l : IdLnk
| . 3. tg : Id
| . 4. @source(l): only L sends on (l with tg) ∈ Dsys
| . 5. ∀D:Dsys.
| . |  @source(l): only L sends on (l with tg) ⊆ D
| . |  ⇒
| . |  D
| . |  realizes es.∀e:E.
| . |              loc(e) = source(l)
| . |              ⇒
| . |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| . 6. @source(l): only L sends on (l with tg) ∈ Dsys
| . 7. D : Dsys
| . 8. @source(l): only L sends on (l with tg) ⊆ D
| . 9. D' : Dsys
| . 10. D ⊆ D'
| . 11. w : World
| . 12. p : FairFifo
| . 13. PossibleWorld(D';w)
| . 14. ∀e:E.
| . |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| . 15. e : E
| . 16. loc(e) = destination(l)
| . 17. kind(e) = rcv(l; tg)
| . 18. sends(l,tg,sender(e)) = nil
| . 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
| . ⊢  (λm.mtag(m) = tg)(msg(lnk(e);tag(e);val(e)))
| . | by Repeat
| . |    (Unfolds [`msg`;`es-mtag`;`mtag`;`es-tag`] 0 THEN Reduce 0)
| . |    THEN
| . |    HypSubst' -3 0
| . |
| . 1. L : Knd List
| . 2. l : IdLnk
| . 3. tg : Id
| . 4. @source(l): only L sends on (l with tg) ∈ Dsys
| . 5. ∀D:Dsys.
| . |  @source(l): only L sends on (l with tg) ⊆ D
| . |  ⇒
```

```
| . |  D
| . |  realizes es.∀e:E.
| . |              loc(e) = source(l)
| . |              ⇒
| . |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| . 6. @source(l): only L sends on (l with tg) ∈ Dsys
| . 7. D : Dsys
| . 8. @source(l): only L sends on (l with tg) ⊆ D
| . 9. D' : Dsys
| . 10. D ⊆ D'
| . 11. w : World
| . 12. p : FairFifo
| . 13. PossibleWorld(D';w)
| . 14. ∀e:E.
| . |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| . 15. e : E
| . 16. loc(e) = destination(l)
| . 17. kind(e) = rcv(l; tg)
| . 18. sends(l,tg,sender(e)) = nil
| . 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
| . ⊢  tag(rcv(l; tg)) = tg
| . | by Unfolds [‘tagof‘;‘rcv‘] 0 THEN Reduce 0
| . |
| . 1. L : Knd List
| . 2. l : IdLnk
| . 3. tg : Id
| . 4. @source(l): only L sends on (l with tg) ∈ Dsys
| . 5. ∀D:Dsys.
| .    @source(l): only L sends on (l with tg) ⊆ D
| .    ⇒
| .    D
| .    realizes es.∀e:E.
| .                loc(e) = source(l)
| .                ⇒
| .                ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| . 6. @source(l): only L sends on (l with tg) ∈ Dsys
| . 7. D : Dsys
| . 8. @source(l): only L sends on (l with tg) ⊆ D
| . 9. D' : Dsys
| . 10. D ⊆ D'
| . 11. w : World
| . 12. p : FairFifo
| . 13. PossibleWorld(D';w)
| . 14. ∀e:E.
| .     loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
| . 15. e : E
| . 16. loc(e) = destination(l)
| . 17. kind(e) = rcv(l; tg)
| . 18. sends(l,tg,sender(e)) = nil
| . 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
| . ⊢  tg = tg  by RW assert_pushdownC 0
|
.\
. 1. L : Knd List
. 2. l : IdLnk
```

```
. 3. tg : Id
. 4. @source(l): only L sends on (l with tg) ∈ Dsys
. 5. ∀D:Dsys.
. |  @source(l): only L sends on (l with tg) ⊆ D
. |  ⇒
. |  D
. |  realizes es.∀e:E.
. |              loc(e) = source(l)
. |              ⇒
. |              ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
. 6. @source(l): only L sends on (l with tg) ∈ Dsys
. 7. D : Dsys
. 8. @source(l): only L sends on (l with tg) ⊆ D
. 9. D' : Dsys
. 10. D ⊆ D'
. 11. w : World
. 12. p : FairFifo
. 13. PossibleWorld(D';w)
. 14. ∀e:E.
. |   loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
. 15. e : E
. 16. loc(e) = destination(l)
. 17. kind(e) = rcv(l; tg)
. 18. sends(l,tg,sender(e)) = nil
. 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
. 20. (msg(lnk(e);tag(e);val(e)) ∈ sends(l,tg,sender(e)))
. ⊢  False  by HypSubst -3 -1
. |
. |\
. | .....unproved Inclusion subgoal .....
. | 1. L : Knd List
. | 2. l : IdLnk
. | 3. tg : Id
. | 4. @source(l): only L sends on (l with tg) ∈ Dsys
. | 5. ∀D:Dsys.
. |    @source(l): only L sends on (l with tg) ⊆ D
. |    ⇒
. |    D
. |    realizes es.∀e:E.
. |                loc(e) = source(l)
. |                ⇒
. |                ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
. | 6. @source(l): only L sends on (l with tg) ∈ Dsys
. | 7. D : Dsys
. | 8. @source(l): only L sends on (l with tg) ⊆ D
. | 9. D' : Dsys
. | 10. D ⊆ D'
. | 11. w : World
. | 12. p : FairFifo
. | 13. PossibleWorld(D';w)
. | 14. ∀e:E.
. |     loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
. | 15. e : E
. | 16. loc(e) = destination(l)
. | 17. kind(e) = rcv(l; tg)
```

```
. | 18. sends(l,tg,sender(e)) = nil
. | 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
. | 20. (msg(lnk(e);tag(e);val(e)) ∈ sends(l,tg,sender(e)))
. | 21. (Msg on l) List
. | 22. (Msg on l) List
. | 23. u : (Msg on l)
. | ⊢  u ∈ Msg  by Analyze -1

. |
. .\
. . 1. L : Knd List
. . 2. l : IdLnk
. . 3. tg : Id
. . 4. @source(l): only L sends on (l with tg) ∈ Dsys
. . 5. ∀D:Dsys.
. .    @source(l): only L sends on (l with tg) ⊆ D
. .    ⇒
. .    D
. .    realizes es.∀e:E.
. .                loc(e) = source(l)
. .                ⇒
. .                ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
. . 6. @source(l): only L sends on (l with tg) ∈ Dsys
. . 7. D : Dsys
. . 8. @source(l): only L sends on (l with tg) ⊆ D
. . 9. D' : Dsys
. . 10. D ⊆ D'
. . 11. w : World
. . 12. p : FairFifo
. . 13. PossibleWorld(D';w)
. . 14. ∀e:E.
. .     loc(e) = source(l) ⇒ ¬null(sends(l,tg,e)) ⇒ (kind(e) ∈ L)
. . 15. e : E
. . 16. loc(e) = destination(l)
. . 17. kind(e) = rcv(l; tg)
. . 18. sends(l,tg,sender(e)) = nil
. . 19. sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e))
. . 20. (msg(lnk(e);tag(e);val(e)) ∈ nil)
. . ⊢  False  by Unfold `l_member` -1 THEN Reduce -1 THEN ExRepD
END
```