```
add_action_mem  THM

⊢  ∀l:St*, f:(St→St), s:St.
|  mem_f(St;s;(+f)(l))
|  ⟺
|  mem_f(St;s;l) ∨ (∃s':St. mem_f(St;s';l) & f(s') = s)  by GenExRepD
|
|\
| 1. St : Type
| 2. l : St*
| 3. f : St→St
| 4. s : St
| 5. mem_f(St;s;(+f)(l))
| ⊢  mem_f(St;s;l) ∨ (∃s':St. mem_f(St;s';l) & f(s') = s)
| | by ListInd 2
| |
| |\
| | 4. s : St
| | ⊢  mem_f(St;s;(+f)(nil)) ⇒
| |    mem_f(St;s;nil) ∨ (∃s':St. mem_f(St;s';nil) & f(s') = s)
| |   by RecUnfold `mem_f` 0 THEN RecUnfold `add_action` 0
| |      THEN
| |      Reduce 0
| |
| .\
| . 5. u : St
| . 6. v : St*
| . 7. mem_f(St;s;(+f)(v)) ⇒
| . |  mem_f(St;s;v) ∨ (∃s':St. mem_f(St;s';v) & f(s') = s)
| . ⊢  mem_f(St;s;(+f)(u.v)) ⇒
| . |  mem_f(St;s;u.v) ∨ (∃s':St. mem_f(St;s';u.v) & f(s') = s)
| . | by RecUnfold `mem_f` 0 THEN RecUnfold `add_action` 0
| . |    THEN
| . |    Reduce 0
| . |
| . ⊢  u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v)) ⇒
| . |  u = s ∨ mem_f(St;s;v)
| . |  ∨ (∃s':St. u = s' ∨ mem_f(St;s';v) & f(s') = s)
| . | by (Analyze 0) THEN (Analyze -1)
| . |
| . |\
| . | 8. u = s
| . | ⊢  u = s ∨ mem_f(St;s;v)
| . |    ∨ (∃s':St. u = s' ∨ mem_f(St;s';v) & f(s') = s)
| . |   by ProveProp
| . |
| . .\
| . . 8. f(u) = s ∨ mem_f(St;s;(+f)(v))
| . . ⊢  u = s ∨ mem_f(St;s;v)
| . . |  ∨ (∃s':St. u = s' ∨ mem_f(St;s';v) & f(s') = s)
| . . | by Analyze -1
| . . |
| . . |\
| . . | 8. f(u) = s
| . . | ⊢   by Sel 2 (Analyze 0)
```

```
| . . | |
| . . | ⊢  ∃s':St. u = s' ∨ mem_f(St;s';v) & f(s') = s
| . . | | by (InstConcl [u]) THEN (Analyze 0)
| . . | |
| . . | |\
| . . | | ⊢  u = u ∨ mem_f(St;u;v)  by Sel 1 (Analyze 0)
| . . | |
| . . | .\
| . . | . ⊢  f(u) = s  by NthHyp 8
| . . |
| . . .\
| . . . 8. mem_f(St;s;(+f)(v))
| . . . ⊢   by FwdThru 7 [8] THEN Analyze -1
| . . . |
| . . . |\
| . . . | 9. mem_f(St;s;v)
| . . . | ⊢   by ProveProp
| . . . |
| . . . .\
| . . . . 9. ∃s':St. mem_f(St;s';v) & f(s') = s
| . . . . ⊢   by Sel 2 (Analyze 0)
| . . . . |
| . . . . ⊢  ∃s':St. u = s' ∨ mem_f(St;s';v) & f(s') = s
| . . . .   by (Analyze -1 THEN InstConcl [s']) THEN ProveProp
|
|\
| 1. St : Type
| 2. l : St*
| 3. f : St→St
| 4. s : St
| 5. mem_f(St;s;l)
| ⊢  mem_f(St;s;(+f)(l))  by ListInd 2
| |
| |\
| | 4. s : St
| | ⊢  mem_f(St;s;nil) ⇒ mem_f(St;s;(+f)(nil))
| |   by RecUnfold `mem_f` 0 THEN Reduce 0
| |
| .\
| . 5. u : St
| . 6. v : St*
| . 7. mem_f(St;s;v) ⇒ mem_f(St;s;(+f)(v))
| . ⊢  mem_f(St;s;u.v) ⇒ mem_f(St;s;(+f)(u.v))
| . | by RecUnfold `mem_f` 0 THEN RecUnfold `add_action` 0
| . |    THEN
| . |    Reduce 0
| . |
| . ⊢  u = s ∨ mem_f(St;s;v) ⇒ u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v))
| . | by GenExRepD
| . |
| . |\
| . | 8. u = s
| . | ⊢  u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v))  by ProveProp
| . |
| . .\
```

```
| . . 8. mem_f(St;s;v)
| . . ⊢  u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v))
| . .   by FwdThru 7 [8] THEN ProveProp
|
.\
. 1. St : Type
. 2. l : St*
. 3. f : St→St
. 4. s : St
. 5. s' : St
. 6. mem_f(St;s';l)
. 7. f(s') = s
. ⊢  mem_f(St;s;(+f)(l))  by ListInd 2
. |
. |\
. | 6. f(s') = s
. | ⊢  mem_f(St;s';nil) ⇒ mem_f(St;s;(+f)(nil))
. |   by RecUnfold `mem_f` 0 THEN Reduce 0
. |
. .\
. . 6. f(s') = s
. . 7. u : St
. . 8. v : St*
. . 9. mem_f(St;s';v) ⇒ mem_f(St;s;(+f)(v))
. . ⊢  mem_f(St;s';u.v) ⇒ mem_f(St;s;(+f)(u.v))
. . | by RecUnfold `mem_f` 0 THEN RecUnfold `add_action` 0
. . |    THEN
. . |    Reduce 0
. . |
. . ⊢  u = s' ∨ mem_f(St;s';v) ⇒
. . |  u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v))
. . | by (Analyze 0) THEN (Analyze -1)
. . |
. . |\
. . | 10. u = s'
. . | ⊢  u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v))
. . |   by (RWH (RevHypC 10) 6) THEN ProveProp
. . |
. . .\
. . . 10. mem_f(St;s';v)
. . . ⊢  u = s ∨ f(u) = s ∨ mem_f(St;s;(+f)(v))  by ProveProp
END
```