```
lremove_mem  THM

⊢  ∀l:St*, s:St, eq:(St→St→𝔹), sx:St.
|  mem_f(St;sx;l \ s) ⟺ if eq(s,sx)→ False else mem_f(St;sx;l) fi
| by RepD THEN (Analyze 0)
|
|\
| 1. St : Type
| 2. l : St*
| 3. s : St
| 4. eq : St→St→𝔹
| 5. sx : St
| ⊢  mem_f(St;sx;l \ s) ⇒ if eq(s,sx)→ False else mem_f(St;sx;l) fi
| | by ListInd 2
| |
| |\
| | ⊢  mem_f(St;sx;nil \ s) ⇒
| |    if eq(s,sx)→ False else mem_f(St;sx;nil) fi
| |   by RecUnfold ‘mem_f‘ 0 THEN RecUnfold ‘lremove‘ 0 THEN Reduce 0
| |
| .\
| . 6. u : St
| . 7. v : St*
| . 8. mem_f(St;sx;v \ s) ⇒ if eq(s,sx)→ False else mem_f(St;sx;v) fi
| . ⊢  mem_f(St;sx;(u.v) \ s) ⇒
| . |  if eq(s,sx)→ False else mem_f(St;sx;u.v) fi
| . | by RecUnfold ‘lremove‘ 0 THEN Reduce 0 THEN MoveToConcl -1
| . |
| . 7. v : St*
| . ⊢  (mem_f(St;sx;v \ s) ⇒
| . |   if eq(s,sx)→ False else mem_f(St;sx;v) fi)
| . |  ⇒
| . |
| . |  mem_f(St;sx;if eq(s,u)→ v \ s else u.(v \ s) fi) ⇒
| . |  if eq(s,sx)→ False else u = sx ∨ mem_f(St;sx;v) fi
| . | by SplitOnConclITE
| . |
| . |\
| . | .....truecase .....
| . | 8. eq(s,sx)
| . | ⊢  (mem_f(St;sx;v \ s) ⇒ False) ⇒
| . | |  mem_f(St;sx;if eq(s,u)→ v \ s else u.(v \ s) fi) ⇒ False
| . | | by (Analyze 0) THEN SplitOnConclITE
| . | |
| . | |\
| . | | .....truecase .....
| . | | 9. mem_f(St;sx;v \ s) ⇒ False
| . | | 10. eq(s,u)
| . | | ⊢  mem_f(St;sx;v \ s) ⇒ False  by NthHyp -2
| . | |
| . | .\
| . | . .....falsecase .....
| . | . 9. mem_f(St;sx;v \ s) ⇒ False
| . | . 10. ¬eq(s,u)
| . | . ⊢  mem_f(St;sx;u.(v \ s)) ⇒ False
```

```
| . | . | by RecUnfold `mem_f` 0 THEN Reduce 0
| . | . |
| . | . ⊢  u = sx ∨ mem_f(St;sx;v \ s) ⇒ False
| . | . | by (Analyze 0) THEN (Analyze -1)
| . | . |
| . | . |\
| . | . | 11. u = sx
| . | . | ⊢  False  by (RWH (HypC -1) -2) THEN ProveProp
| . | . |
| . | . .\
| . | . . 11. mem_f(St;sx;v \ s)
| . | . . ⊢  False  by ProveProp
| . |
| . .\
| . . .....falsecase .....
| . . 8. ¬eq(s,sx)
| . . ⊢  (mem_f(St;sx;v \ s) ⇒ mem_f(St;sx;v)) ⇒
| . . |  mem_f(St;sx;if eq(s,u)→ v \ s else u.(v \ s) fi) ⇒
| . . |  u = sx ∨ mem_f(St;sx;v)  by (Analyze 0) THEN SplitOnConclITE
| . . |
| . . |\
| . . | .....truecase .....
| . . | 9. mem_f(St;sx;v \ s) ⇒ mem_f(St;sx;v)
| . . | 10. eq(s,u)
| . . | ⊢  mem_f(St;sx;v \ s) ⇒ u = sx ∨ mem_f(St;sx;v)
| . . |   by ProveProp
| . . |
| . . .\
| . . . .....falsecase .....
| . . . 9. mem_f(St;sx;v \ s) ⇒ mem_f(St;sx;v)
| . . . 10. ¬eq(s,u)
| . . . ⊢  mem_f(St;sx;u.(v \ s)) ⇒ u = sx ∨ mem_f(St;sx;v)
| . . . | by ((Analyze 0) THEN (RecUnfold `mem_f` -1)) THEN Reduce -1
| . . . |
| . . . 11. u = sx ∨ mem_f(St;sx;v \ s)
| . . . ⊢  u = sx ∨ mem_f(St;sx;v)  by ProveProp
|
.\
. 1. St : Type
. 2. l : St*
. 3. s : St
. 4. eq : St→St→𝔹
. 5. sx : St
. ⊢  mem_f(St;sx;l \ s) ⇐ if eq(s,sx)→ False else mem_f(St;sx;l) fi
. | by ListInd 2
. |
. |\
. | ⊢  mem_f(St;sx;nil \ s) ⇐ if eq(s,sx)→ False
. | |                          else mem_f(St;sx;nil) fi
. | | by SplitOnConclITE THEN (Analyze 0)
. | |
. | |\
. | | 6. eq(s,sx)
. | | 7. False
. | | ⊢  mem_f(St;sx;nil \ s)  by Auto
```

```
. | |
. | .\
. | . 6. ¬eq(s,sx)
. | . 7. mem_f(St;sx;nil)
. | . ⊢  mem_f(St;sx;nil \ s)  by RecUnfold ‘mem_f‘ -1 THEN Reduce -1
. |
. .\
. . 6. u : St
. . 7. v : St*
. . 8. mem_f(St;sx;v \ s) ⇐ if eq(s,sx)→ False else mem_f(St;sx;v) fi
. . ⊢  mem_f(St;sx;(u.v) \ s) ⇐ if eq(s,sx)→ False
. . |                            else mem_f(St;sx;u.v) fi
. . | by (MoveToConcl -1 THEN SplitOnConclITE) THEN (Analyze 0)
. . |
. . |\
. . | 8. eq(s,sx)
. . | 9. mem_f(St;sx;v \ s) ⇐ False
. . | ⊢  mem_f(St;sx;(u.v) \ s) ⇐ False  by Auto
. . |
. . .\
. . . 8. ¬eq(s,sx)
. . . 9. mem_f(St;sx;v \ s) ⇐ mem_f(St;sx;v)
. . . ⊢  mem_f(St;sx;(u.v) \ s) ⇐ mem_f(St;sx;u.v)
. . . | by RecUnfold ‘lremove‘ 0 THEN Reduce 0 THEN SplitOnConclITE
. . . |
. . . |\
. . . | .....truecase .....
. . . | 10. eq(s,u)
. . . | ⊢  mem_f(St;sx;v \ s) ⇐ (u = sx ∨ mem_f(St;sx;v))
. . . | | by (Analyze 0) THEN (Analyze -1)
. . . | |
. . . | |\
. . . | | 11. u = sx
. . . | | ⊢  mem_f(St;sx;v \ s)  by RWH (HypC -1) -2
. . . | |
. . . | .\
. . . | . 11. mem_f(St;sx;v)
. . . | . ⊢  mem_f(St;sx;v \ s)  by ProveProp
. . . |
. . . .\
. . . . .....falsecase .....
. . . . 10. ¬eq(s,u)
. . . . ⊢  mem_f(St;sx;u.(v \ s)) ⇐ (u = sx ∨ mem_f(St;sx;v))
. . . .   by (Analyze 0) THEN (RecUnfold ‘mem_f‘ 0) THEN (Reduce 0)
. . . .      THEN
. . . .      ProveProp
END
```