```
select_concat  THM

⊢  ∀T:Type, ll:(T List) List, n:ℕ||concat(ll)||.
|  ∃m:ℕ||ll||.
|  ||concat(firstn(m;ll))||≤n
|  & n-||concat(firstn(m;ll))||<||ll[m]||
|  & concat(ll)[n] = ll[m][(n-||concat(firstn(m;ll))||)]
| by InductionOnList
|
|\
| 1. T : Type
| 2. (T List) List
| ⊢  ∀n:ℕ||concat(nil)||.
|    ∃m:ℕ||nil||.
|    ||concat(firstn(m;nil))||≤n
|    & n-||concat(firstn(m;nil))||<||nil[m]||
|    & concat(nil)[n] = nil[m][(n-||concat(firstn(m;nil))||)]
|   by Unfold ‘concat‘ 0 THEN Reduce 0
|
.\
. 1. T : Type
. 2. (T List) List
. 3. u : T List
. 4. v : (T List) List
. 5. ∀n:ℕ||concat(v)||.
. |  ∃m:ℕ||v||.
. |  ||concat(firstn(m;v))||≤n
. |  & n-||concat(firstn(m;v))||<||v[m]||
. |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. ⊢  ∀n:ℕ||concat([u / v])||.
. |  ∃m:ℕ||[u / v]||.
. |  ||concat(firstn(m;[u / v]))||≤n
. |  & n-||concat(firstn(m;[u / v]))||<||[u / v][m]||
. |  & concat([u / v])[n]
. |    =
. |    [u / v][m][(n-||concat(firstn(m;[u / v]))||)]
. | by Unfold ‘concat‘ 0 THEN Reduce 0 THEN Try (Fold ‘concat‘ 0)
. |
. 1. T : Type
. 2. (T List) List
. 3. u : T List
. 4. v : (T List) List
. 5. ∀n:ℕ||concat(v)||.
. |  ∃m:ℕ||v||.
. |  ||concat(firstn(m;v))||≤n
. |  & n-||concat(firstn(m;v))||<||v[m]||
. |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. ⊢  ∀n:ℕ||u @ concat(v)||.
. |  ∃m:ℕ(||v||+1).
. |  ||concat(firstn(m;[u / v]))||≤n
. |  & n-||concat(firstn(m;[u / v]))||<||[u / v][m]||
. |  & (u @ concat(v))[n]
. |    =
. |    [u / v][m][(n-||concat(firstn(m;[u / v]))||)]
. | by Analyze 0 THEN Decide (n<||u||)
```

```
. |
. |\
. | 1. T : Type
. | 2. (T List) List
. | 3. u : T List
. | 4. v : (T List) List
. | 5. ∀n:ℕ||concat(v)||.
. | |  ∃m:ℕ||v||.
. | |  ||concat(firstn(m;v))||≤n
. | |  & n-||concat(firstn(m;v))||<||v[m]||
. | |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. | 6. n : ℕ||u @ concat(v)||
. | 7. n<||u||
. | ⊢  ∃m:ℕ(||v||+1).
. | |  ||concat(firstn(m;[u / v]))||≤n
. | |  & n-||concat(firstn(m;[u / v]))||<||[u / v][m]||
. | |  & (u @ concat(v))[n]
. | |    =
. | |    [u / v][m][(n-||concat(firstn(m;[u / v]))||)]
. | | by InstConcl [0]
. | |
. | |\
. | | .....wf .....
. | | 1. T : Type
. | | 2. (T List) List
. | | 3. u : T List
. | | 4. v : (T List) List
. | | 5. ∀n:ℕ||concat(v)||.
. | |    ∃m:ℕ||v||.
. | |    ||concat(firstn(m;v))||≤n
. | |    & n-||concat(firstn(m;v))||<||v[m]||
. | |    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. | | 6. n : ℕ||u @ concat(v)||
. | | 7. n<||u||
. | | ⊢  0 ∈ ℕ(||v||+1)  by Auto
. | |
. | |\
. | | 1. T : Type
. | | 2. (T List) List
. | | 3. u : T List
. | | 4. v : (T List) List
. | | 5. ∀n:ℕ||concat(v)||.
. | |    ∃m:ℕ||v||.
. | |    ||concat(firstn(m;v))||≤n
. | |    & n-||concat(firstn(m;v))||<||v[m]||
. | |    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. | | 6. n : ℕ||u @ concat(v)||
. | | 7. n<||u||
. | | ⊢  ||concat(firstn(0;[u / v]))||≤n
. | |    & n-||concat(firstn(0;[u / v]))||<||[u / v][0]||
. | |    & (u @ concat(v))[n]
. | |      =
. | |      [u / v][0][(n-||concat(firstn(0;[u / v]))||)]
. | |   by RWO Thm* ∀L:Top List. firstn(0;L) ~ nil    {first0} 0
. | |      THEN
```

```
. | |      Reduce 0
. | |      THEN
. | |      Subst' (n-0 = n) 0
. | |      THEN
. | |      BackThru
. | |      Thm* ∀as,bs:T List, i:ℕ||as||. (as @ bs)[i] = as[i]
. | |      {select_append_front}
. | |
. | .\
. | . .....wf .....
. | . 1. T : Type
. | . 2. (T List) List
. | . 3. u : T List
. | . 4. v : (T List) List
. | . 5. ∀n:ℕ||concat(v)||.
. | .    ∃m:ℕ||v||.
. | .    ||concat(firstn(m;v))||≤n
. | .    & n-||concat(firstn(m;v))||<||v[m]||
. | .    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. | . 6. n : ℕ||u @ concat(v)||
. | . 7. n<||u||
. | . 8. m : ℕ(||v||+1)
. | . ⊢  (||concat(firstn(m;[u / v]))||≤n
. | .     & n-||concat(firstn(m;[u / v]))||<||[u / v][m]||
. | .     & (u @ concat(v))[n]
. | .       =
. | .       [u / v][m][(n-||concat(firstn(m;[u / v]))||)])
. | .    ∈ Prop  by Auto THEN Reduce 0
. |
. .\
. . 1. T : Type
. . 2. (T List) List
. . 3. u : T List
. . 4. v : (T List) List
. . 5. ∀n:ℕ||concat(v)||.
. . |  ∃m:ℕ||v||.
. . |  ||concat(firstn(m;v))||≤n
. . |  & n-||concat(firstn(m;v))||<||v[m]||
. . |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . 6. n : ℕ||u @ concat(v)||
. . 7. ¬n<||u||
. . ⊢  ∃m:ℕ(||v||+1).
. . |  ||concat(firstn(m;[u / v]))||≤n
. . |  & n-||concat(firstn(m;[u / v]))||<||[u / v][m]||
. . |  & (u @ concat(v))[n]
. . |    =
. . |    [u / v][m][(n-||concat(firstn(m;[u / v]))||)]
. . | by InstHyp [n-||u||] 5 THEN ExRepD
. . |
. . 1. T : Type
. . 2. (T List) List
. . 3. u : T List
. . 4. v : (T List) List
. . 5. ∀n:ℕ||concat(v)||.
. . |  ∃m:ℕ||v||.
```

```
. . |  ||concat(firstn(m;v))||≤n
. . |  & n-||concat(firstn(m;v))||<||v[m]||
. . |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . 6. n : ℕ||u @ concat(v)||
. . 7. ¬n<||u||
. . 8. m : ℕ||v||
. . 9. ||concat(firstn(m;v))||≤n-||u||
. . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . 11. concat(v)[(n-||u||)] = v[m][(n-||u||-||concat(firstn(m;v))||)]
. . ⊢  ∃m:ℕ(||v||+1).
. . |  ||concat(firstn(m;[u / v]))||≤n
. . |  & n-||concat(firstn(m;[u / v]))||<||[u / v][m]||
. . |  & (u @ concat(v))[n]
. . |    =
. . |    [u / v][m][(n-||concat(firstn(m;[u / v]))||)]
. . | by InstConcl [m+1]
. . |
. . 1. T : Type
. . 2. (T List) List
. . 3. u : T List
. . 4. v : (T List) List
. . 5. ∀n:ℕ||concat(v)||.
. . |  ∃m:ℕ||v||.
. . |  ||concat(firstn(m;v))||≤n
. . |  & n-||concat(firstn(m;v))||<||v[m]||
. . |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . 6. n : ℕ||u @ concat(v)||
. . 7. ¬n<||u||
. . 8. m : ℕ||v||
. . 9. ||concat(firstn(m;v))||≤n-||u||
. . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . 11. concat(v)[(n-||u||)] = v[m][(n-||u||-||concat(firstn(m;v))||)]
. . ⊢  ||concat(firstn(m+1;[u / v]))||≤n
. . |  & n-||concat(firstn(m+1;[u / v]))||<||[u / v][(m+1)]||
. . |  & (u @ concat(v))[n]
. . |    =
. . |    [u / v][(m+1)][(n-||concat(firstn(m+1;[u / v]))||)]
. . | by Subst (firstn(m+1;[u / v]) ∼ [u / firstn(m;v)]) 0
. . |
. . |\
. . | .....equality .....
. . | 1. T : Type
. . | 2. (T List) List
. . | 3. u : T List
. . | 4. v : (T List) List
. . | 5. ∀n:ℕ||concat(v)||.
. . | |  ∃m:ℕ||v||.
. . | |  ||concat(firstn(m;v))||≤n
. . | |  & n-||concat(firstn(m;v))||<||v[m]||
. . | |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . | 6. n : ℕ||u @ concat(v)||
. . | 7. ¬n<||u||
. . | 8. m : ℕ||v||
. . | 9. ||concat(firstn(m;v))||≤n-||u||
. . | 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
```

```
. . | 11. concat(v)[(n-||u||)]
. . | |   =
. . | |   v[m][(n-||u||-||concat(firstn(m;v))||)]
. . | ⊢  firstn(m+1;[u / v]) ~ [u / firstn(m;v)]
. . | | by RW (AddrC [1] (RecUnfoldC ‘firstn‘)) 0 THEN Reduce 0
. . | |    THEN
. . | |    SplitOnConclITE
. . | |
. . | |\
. . | | .....truecase .....
. . | | 1. T : Type
. . | | 2. (T List) List
. . | | 3. u : T List
. . | | 4. v : (T List) List
. . | | 5. ∀n:ℕ||concat(v)||.
. . | |    ∃m:ℕ||v||.
. . | |    ||concat(firstn(m;v))||≤n
. . | |    & n-||concat(firstn(m;v))||<||v[m]||
. . | |    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . | | 6. n : ℕ||u @ concat(v)||
. . | | 7. ¬n<||u||
. . | | 8. m : ℕ||v||
. . | | 9. ||concat(firstn(m;v))||≤n-||u||
. . | | 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . | | 11. concat(v)[(n-||u||)]
. . | |     =
. . | |     v[m][(n-||u||-||concat(firstn(m;v))||)]
. . | | 12. 0<m+1
. . | | ⊢  [u / firstn(m+1-1;v)] ~ [u / firstn(m;v)]
. . | |   by Repeat Analyze
. . | |
. . | .\
. . | . .....falsecase .....
. . | . 1. T : Type
. . | . 2. (T List) List
. . | . 3. u : T List
. . | . 4. v : (T List) List
. . | . 5. ∀n:ℕ||concat(v)||.
. . | .    ∃m:ℕ||v||.
. . | .    ||concat(firstn(m;v))||≤n
. . | .    & n-||concat(firstn(m;v))||<||v[m]||
. . | .    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . | . 6. n : ℕ||u @ concat(v)||
. . | . 7. ¬n<||u||
. . | . 8. m : ℕ||v||
. . | . 9. ||concat(firstn(m;v))||≤n-||u||
. . | . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . | . 11. concat(v)[(n-||u||)]
. . | .     =
. . | .     v[m][(n-||u||-||concat(firstn(m;v))||)]
. . | . 12. m+1≤0
. . | . ⊢  nil ~ [u / firstn(m;v)]  by Auto
. . |
. . .\
. . . 1. T : Type
```

```
. . . 2. (T List) List
. . . 3. u : T List
. . . 4. v : (T List) List
. . . 5. ∀n:ℕ||concat(v)||.
. . . |  ∃m:ℕ||v||.
. . . |  ||concat(firstn(m;v))||≤n
. . . |  & n-||concat(firstn(m;v))||<||v[m]||
. . . |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . 6. n : ℕ||u @ concat(v)||
. . . 7. ¬n<||u||
. . . 8. m : ℕ||v||
. . . 9. ||concat(firstn(m;v))||≤n-||u||
. . . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . 11. concat(v)[(n-||u||)]
. . . |   =
. . . |   v[m][(n-||u||-||concat(firstn(m;v))||)]
. . . ⊢  ||concat([u / firstn(m;v)])||≤n
. . . |  & n-||concat([u / firstn(m;v)])||<||[u / v][(m+1)]||
. . . |  & (u @ concat(v))[n]
. . . |    =
. . . |    [u / v][(m+1)][(n-||concat([u / firstn(m;v)])||)]
. . . | by Unfold ‘concat‘ 0 THEN Reduce 0 THEN Fold ‘concat‘ 0
. . . |
. . . 1. T : Type
. . . 2. (T List) List
. . . 3. u : T List
. . . 4. v : (T List) List
. . . 5. ∀n:ℕ||concat(v)||.
. . . |  ∃m:ℕ||v||.
. . . |  ||concat(firstn(m;v))||≤n
. . . |  & n-||concat(firstn(m;v))||<||v[m]||
. . . |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . 6. n : ℕ||u @ concat(v)||
. . . 7. ¬n<||u||
. . . 8. m : ℕ||v||
. . . 9. ||concat(firstn(m;v))||≤n-||u||
. . . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . 11. concat(v)[(n-||u||)]
. . . |   =
. . . |   v[m][(n-||u||-||concat(firstn(m;v))||)]
. . . ⊢  ||u @ concat(firstn(m;v))||≤n
. . . |  & n-||u @ concat(firstn(m;v))||<||[u / v][(m+1)]||
. . . |  & (u @ concat(v))[n]
. . . |    =
. . . |    [u / v][(m+1)][(n-||u @ concat(firstn(m;v))||)]  by Auto
. . . |
. . . |\
. . . | 1. T : Type
. . . | 2. (T List) List
. . . | 3. u : T List
. . . | 4. v : (T List) List
. . . | 5. ∀n:ℕ||concat(v)||.
. . . | |  ∃m:ℕ||v||.
. . . | |  ||concat(firstn(m;v))||≤n
. . . | |  & n-||concat(firstn(m;v))||<||v[m]||
```

```
. . . | |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . | 6. n : ℕ||u @ concat(v)||
. . . | 7. ¬n<||u||
. . . | 8. m : ℕ||v||
. . . | 9. ||concat(firstn(m;v))||≤n-||u||
. . . | 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . | 11. concat(v)[(n-||u||)]
. . . | |    =
. . . | |    v[m][(n-||u||-||concat(firstn(m;v))||)]
. . . | ⊢  n-||u @ concat(firstn(m;v))||<||[u / v][(m+1)]||
. . . | | by Subst ([u / v][(m+1)] ∼ v[m]) 0
. . . | |
. . . | |\
. . . | | .....equality .....
. . . | | 1. T : Type
. . . | | 2. (T List) List
. . . | | 3. u : T List
. . . | | 4. v : (T List) List
. . . | | 5. ∀n:ℕ||concat(v)||.
. . . | |    ∃m:ℕ||v||.
. . . | |    ||concat(firstn(m;v))||≤n
. . . | |    & n-||concat(firstn(m;v))||<||v[m]||
. . . | |    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . | | 6. n : ℕ||u @ concat(v)||
. . . | | 7. ¬n<||u||
. . . | | 8. m : ℕ||v||
. . . | | 9. ||concat(firstn(m;v))||≤n-||u||
. . . | | 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . | | 11. concat(v)[(n-||u||)]
. . . | |     =
. . . | |     v[m][(n-||u||-||concat(firstn(m;v))||)]
. . . | | ⊢  [u / v][(m+1)] ∼ v[m]
. . . | |   by RWO
. . . | |      Thm* ∀x:T, l:T List, i:ℕ||l||. [x / l][(i+1)] ∼ l[i]
. . . | |      {select_cons_tl_sq}
. . . | |      0
. . . | |
. . . | .\
. . . | . 1. T : Type
. . . | . 2. (T List) List
. . . | . 3. u : T List
. . . | . 4. v : (T List) List
. . . | . 5. ∀n:ℕ||concat(v)||.
. . . | .    ∃m:ℕ||v||.
. . . | .    ||concat(firstn(m;v))||≤n
. . . | .    & n-||concat(firstn(m;v))||<||v[m]||
. . . | .    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . | . 6. n : ℕ||u @ concat(v)||
. . . | . 7. ¬n<||u||
. . . | . 8. m : ℕ||v||
. . . | . 9. ||concat(firstn(m;v))||≤n-||u||
. . . | . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . | . 11. concat(v)[(n-||u||)]
. . . | .     =
. . . | .     v[m][(n-||u||-||concat(firstn(m;v))||)]
```

```
. . . | . ⊢  n-||u @ concat(firstn(m;v))||<||v[m]||
. . . | .   by Inst
. . . | .     Thm* ∀as,bs:T List, i:{||as||..(||as||+||bs||)⁻}.
. . . | .          (as @ bs)[i] = bs[(i-||as||)]
. . . | .     {select_append_back}
. . . | .     [T;u;concat(v);n]
. . . | .     THEN
. . . | .     Subst'
. . . | .     (n-||u @ concat(firstn(m;v))||
. . . | .      =
. . . | .      n-||u||-||concat(firstn(m;v))||)
. . . | .     0
. . . |
. . . .\
. . . . 1. T : Type
. . . . 2. (T List) List
. . . . 3. u : T List
. . . . 4. v : (T List) List
. . . . 5. ∀n:ℕ||concat(v)||.
. . . . |  ∃m:ℕ||v||.
. . . . |  ||concat(firstn(m;v))||≤n
. . . . |  & n-||concat(firstn(m;v))||<||v[m]||
. . . . |  & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . . 6. n : ℕ||u @ concat(v)||
. . . . 7. ¬n<||u||
. . . . 8. m : ℕ||v||
. . . . 9. ||concat(firstn(m;v))||≤n-||u||
. . . . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . . 11. concat(v)[(n-||u||)]
. . . . |   =
. . . . |   v[m][(n-||u||-||concat(firstn(m;v))||)]
. . . . ⊢  (u @ concat(v))[n]
. . . . |  =
. . . . |  [u / v][(m+1)][(n-||u @ concat(firstn(m;v))||)]
. . . . | by Subst ([u / v][(m+1)] ∼ v[m]) 0
. . . . |
. . . . |\
. . . . | .....equality .....
. . . . | 1. T : Type
. . . . | 2. (T List) List
. . . . | 3. u : T List
. . . . | 4. v : (T List) List
. . . . | 5. ∀n:ℕ||concat(v)||.
. . . . |    ∃m:ℕ||v||.
. . . . |    ||concat(firstn(m;v))||≤n
. . . . |    & n-||concat(firstn(m;v))||<||v[m]||
. . . . |    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . . | 6. n : ℕ||u @ concat(v)||
. . . . | 7. ¬n<||u||
. . . . | 8. m : ℕ||v||
. . . . | 9. ||concat(firstn(m;v))||≤n-||u||
. . . . | 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . . | 11. concat(v)[(n-||u||)]
. . . . |     =
. . . . |     v[m][(n-||u||-||concat(firstn(m;v))||)]
```

```
. . . . | ⊢  [u / v][(m+1)] ~ v[m]
. . . . |   by RWO
. . . . |      Thm* ∀x:T, l:T List, i:ℕ||l||. [x / l][(i+1)] ~ l[i]
. . . . |      {select_cons_tl_sq}
. . . . |      0
. . . . |
. . . . .\
. . . . . 1. T : Type
. . . . . 2. (T List) List
. . . . . 3. u : T List
. . . . . 4. v : (T List) List
. . . . . 5. ∀n:ℕ||concat(v)||.
. . . . .    ∃m:ℕ||v||.
. . . . .    ||concat(firstn(m;v))||≤n
. . . . .    & n-||concat(firstn(m;v))||<||v[m]||
. . . . .    & concat(v)[n] = v[m][(n-||concat(firstn(m;v))||)]
. . . . . 6. n : ℕ||u @ concat(v)||
. . . . . 7. ¬n<||u||
. . . . . 8. m : ℕ||v||
. . . . . 9. ||concat(firstn(m;v))||≤n-||u||
. . . . . 10. n-||u||-||concat(firstn(m;v))||<||v[m]||
. . . . . 11. concat(v)[(n-||u||)]
. . . . .     =
. . . . .     v[m][(n-||u||-||concat(firstn(m;v))||)]
. . . . . ⊢  (u @ concat(v))[n]
. . . . .    =
. . . . .    v[m][(n-||u @ concat(firstn(m;v))||)]
. . . . .   by Inst
. . . . .      Thm* ∀as,bs:T List, i:{||as||..(||as||+||bs||)⁻}.
. . . . .           (as @ bs)[i] = bs[(i-||as||)]
. . . . .      {select_append_back}
. . . . .      [T;u;concat(v);n]
. . . . .      THEN
. . . . .      Subst'
. . . . .      (n-||u @ concat(firstn(m;v))||
. . . . .       =
. . . . .       n-||u||-||concat(firstn(m;v))||)
. . . . .      0
END
```