```
empty_lang_dec  THM

⊢  ∀Alph,St:Type, Auto:Automata(Alph;St).
|  Fin(Alph) & Fin(St) ⇒ Dec(∃l:Alph*. LangOf(Auto)(l))  by UnivCD
|
1. Alph : Type
2. St : Type
3. Auto : Automata(Alph;St)
4. Fin(Alph) & Fin(St)
⊢  Dec(∃l:Alph*. LangOf(Auto)(l))  by Analyze 4
|
4. Fin(Alph)
5. Fin(St)
⊢   by Unfold `finite` 5 THEN Analyze 5
|
5. n : ℕ
6. ∃f:(ℕn→St). Bij(ℕn; St; f)
⊢
| by Assert
|    ((∃l:Alph*. LangOf(Auto)(l))
|     ⟺
|     (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l)))
|
|\
| .....assertion .....
| ⊢  (∃l:Alph*. LangOf(Auto)(l))
| |  ⟺
| |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))
| | by Analyze 0
| |
| |\
| | ⊢  (∃l:Alph*. LangOf(Auto)(l)) ⇒
| | |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))
| | | by Unfold `auto_lang` 0 THEN Reduce 0
| | |
| | ⊢  (∃l:Alph*. Auto(l)↓) ⇒
| | |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. Auto(l)↓)
| | | by Unfold `accept_list` 0
| | |
| | ⊢  (∃l:Alph*. FinalState(Auto)(Result(Auto)l)) ⇒
| | |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }.
| | |   FinalState(Auto)(Result(Auto)l))
| | | by RWH
| | |    (LemmaC
| | |     Thm* ∀Auto:Automata(Alph;St), l:Alph*.
| | |          (Result(Auto)l) = (Action(Auto):l←InitialState(Auto))
| | |     {auto_maction})
| | |    0
| | |
| | ⊢  (∃l:Alph*.
| | |   FinalState(Auto)((Action(Auto):l←InitialState(Auto))))
| | |  ⇒
| | |
| | |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }.
| | |   FinalState(Auto)((Action(Auto):l←InitialState(Auto))))
```

```
| | | by Analyze 0
| | |
| | 7. ∃l:Alph*. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | ⊢  ∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }.
| | |  FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | by Analyze 7
| | |
| | 7. l : Alph*
| | 8. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | ⊢
| | | by Inst
| | |    Thm* ∀n:ℕ+, Alph:Type, S:ActionSet(Alph), s,f:S.car.
| | |         #(S.car)=n  ⇒
| | |         (∃l:Alph*. (S:l←s) = f) ⇒
| | |         (∃l:Alph*. ||l||≤n & (S:l←s) = f)
| | |    {pump_thm_cor}
| | |    [n
| | |    ;Alph
| | |    ;Action(Auto)
| | |    ;InitialState(Auto)
| | |    ;(Action(Auto):l←InitialState(Auto))]
| | |
| | |\
| | | .....set predicate .....
| | | ⊢  0<n
| | | | by RWH
| | | |    (LemmaC
| | | |     Thm* (∃f:(A→B). Bij(A; B; f)) ⟺ (A ∼ B)
| | | |     {bij_iff_1_1_corr})
| | | |    6
| | | |
| | | 6. ℕn ∼ St
| | | 7. l : Alph*
| | | 8. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | ⊢   by Unfold ‘one_one_corr‘ 6
| | | |
| | | 6. ∃f:(ℕn→St), g:(St→ℕn). InvFuns(ℕn; St; f; g)
| | | 7. l : Alph*
| | | 8. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | ⊢   by Analyze 6 THEN Analyze 7
| | | |
| | | 6. f : ℕn→St
| | | 7. g : St→ℕn
| | | 8. InvFuns(ℕn; St; f; g)
| | | 9. l : Alph*
| | | 10. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | ⊢   by Assert (g(InitialState(Auto)) ∈ ℕn)

| | |
| | |\
| | | .....antecedent .....
| | | ⊢  #(Action(Auto).car)=n   by Unfold ‘card‘ 0 THEN Reduce 0
| | | |
| | | ⊢  St ∼ ℕn  by Unfold ‘one_one_corr‘ 0
| | | |
| | | ⊢  ∃f:(St→ℕn), g:(ℕn→St). InvFuns(St; ℕn; f; g)
```

```
| | | | by RWH
| | | |    (LemmaC
| | | |     Thm* (∃f:(A→B). Bij(A; B; f)) ⟺ (A ~ B)
| | | |     {bij_iff_1_1_corr})
| | | |    6
| | | |
| | | 6. ℕn ~ St
| | | 7. l : Alph*
| | | 8. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | ⊢   by Unfold `one_one_corr` 6
| | | |
| | | 6. ∃f:(ℕn→St), g:(St→ℕn). InvFuns(ℕn; St; f; g)
| | | 7. l : Alph*
| | | 8. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | ⊢   by Analyze 6 THEN Analyze 7
| | | |
| | | 6. f : ℕn→St
| | | 7. g : St→ℕn
| | | 8. InvFuns(ℕn; St; f; g)
| | | 9. l : Alph*
| | | 10. FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | | ⊢   by Witness g THEN Witness f
| | | |
| | | ⊢  InvFuns(St; ℕn; g; f)
| | |   by BackThru
| | |      Thm* ∀f:(A→B), g:(B→A).
| | |           InvFuns(A; B; f; g) ⇒ InvFuns(B; A; g; f)
| | |      {inv_funs_sym}
| | |
| | |\
| | | .....antecedent .....
| | | ⊢  ∃l@0:Alph*.
| | |    (Action(Auto):l@0←InitialState(Auto))
| | |    =
| | |    (Action(Auto):l←InitialState(Auto))  by Witness l
| | |
| | .\
| | . 9. ∃l@0:Alph*.
| | . |  ||l@0||≤n
| | . |  & (Action(Auto):l@0←InitialState(Auto))
| | . |    =
| | . |    (Action(Auto):l←InitialState(Auto))
| | . ⊢   by Analyze 9
| | . |
| | . 9. l@0 : Alph*
| | . 10. ||l@0||≤n
| | . |   & (Action(Auto):l@0←InitialState(Auto))
| | . |     =
| | . |     (Action(Auto):l←InitialState(Auto))
| | . ⊢   by Witness ||l@0||
| | . |
| | . ⊢  ∃l:{l:(Alph*)| ||l|| = ||l@0|| ∈ ℤ }.
| | . |  FinalState(Auto)((Action(Auto):l←InitialState(Auto)))
| | . | by Witness l@0
| | . |
```

```
| | . ⊢  FinalState(Auto)((Action(Auto):l@0←InitialState(Auto)))
| | . | by Analyze 10
| | . |
| | . 10. ||l@0||≤n
| | . 11. (Action(Auto):l@0←InitialState(Auto))
| | .     =
| | .     (Action(Auto):l←InitialState(Auto))
| | . ⊢   by RWH (HypC 11) 0
| |
| .\
| . ⊢  (∃l:Alph*. LangOf(Auto)(l)) ⇐ (∃k:ℕ(n+1), l:{l:(Alph*)
| . |                                                  | ||l|| = k }.
| . |  LangOf(Auto)(l))  by Analyze 0
| . |
| . |\
| . | 7. ∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l)
| . | ⊢  ∃l:Alph*. LangOf(Auto)(l)  by Analyze 7
| . | |
| . | 7. k : ℕ(n+1)
| . | 8. ∃l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l)
| . | ⊢   by Analyze 8
| . | |
| . | 8. l : {l:(Alph*)| ||l|| = k }
| . | 9. LangOf(Auto)(l)
| . | ⊢   by Witness l
| . | |
| . | 10. Alph*
| . | ⊢  LangOf(Auto) ∈ Alph*→Prop  by Fold ‘languages‘ 0
| . |
| . .\
| . . 7. k : ℕ(n+1)
| . . 8. {l:(Alph*)| ||l|| = k }
| . . ⊢  LangOf(Auto) ∈ Alph*→Prop  by Fold ‘languages‘ 0
|
.\
. 7. (∃l:Alph*. LangOf(Auto)(l))
. |  ⟺
. |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))
. ⊢   by RWH (HypC 7) 0 THENA (Auto THEN Fold ‘languages‘ 0)
. |
. ⊢  Dec(∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))
. | by Inst
. |    Thm* ∀R:(T→Prop).
. |         Fin(T) & (∀t:T. Dec(R(t))) ⇒ Dec(∃t:T. R(t))
. |    {auto2_lemma_6}
. |    [ℕ(n+1);λk.∃l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l)]
. |    THEN
. |    Try (Fold ‘languages‘ 0)
. |
. |\
. | 7. (∃l:Alph*. LangOf(Auto)(l)) ⇒
. |    (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))
. | 8. (∃l:Alph*. LangOf(Auto)(l)) ⇐ (∃k:ℕ(n+1), l:{l:(Alph*)
. |                                                  | ||l|| = k }.
. |    LangOf(Auto)(l))
```

```
. | ⊢  Fin(ℕ(n+1))
. |   by BackThru Thm* ∀n:ℕ. Fin(ℕn)    {nsub_is_finite}
. |
. |\
. | .....decidable? .....
. | 7. (∃l:Alph*. LangOf(Auto)(l)) ⇒
. | |  (∃k:ℕ(n+1), l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))
. | 8. (∃l:Alph*. LangOf(Auto)(l)) ⇐ (∃k:ℕ(n+1), l:{l:(Alph*)
. | |                                                | ||l|| = k }.
. | |  LangOf(Auto)(l))
. | 9. t : ℕ(n+1)
. | ⊢  Dec((λk.∃l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))(t))
. | | by Reduce 0
. | |
. | ⊢  Dec(∃l:{l:(Alph*)| ||l|| = t }. LangOf(Auto)(l))
. | | by Inst
. | |    Thm* ∀R:(T→Prop).
. | |         Fin(T) & (∀t:T. Dec(R(t))) ⇒ Dec(∃t:T. R(t))
. | |    {auto2_lemma_6}
. | |    [{l:(Alph*)| ||l|| = t };λl.LangOf(Auto)(l)]
. | |
. | |\
. | | 10. {l:(Alph*)| ||l|| = t }
. | | ⊢  LangOf(Auto) ∈ Alph*→Prop  by Fold ‘languages‘ 0
. | |
. | |\
. | | ⊢  Fin({l:(Alph*)| ||l|| = t })
. | |   by BackThru
. | |      Thm* ∀n:ℕ. Fin(Alph) ⇒ Fin({l:(Alph*)| ||l|| = n })
. | |      {auto2_lemma_5}
. | |
. | |\
. | | .....decidable? .....
. | | 10. t1 : {l:(Alph*)| ||l|| = t }
. | | ⊢  Dec((λl.LangOf(Auto)(l))(t1))  by Reduce 0
. | | |
. | | ⊢  Dec(LangOf(Auto)(t1))
. | |   by Unfold ‘auto_lang‘ 0 THEN Reduce 0 THEN ProveDecidable
. | |
. | .\
. | . 10. Dec(∃t:{l:(Alph*)| ||l|| = t }. (λl.LangOf(Auto)(l))(t))
. | . ⊢   by Reduce 10
. |
. .\
. . 8. Dec(∃t:ℕ(n+1).
. .    (λk.∃l:{l:(Alph*)| ||l|| = k }. LangOf(Auto)(l))(t))
. . ⊢   by Reduce 8
END
```