```
nd_auto_eq_auto  THM

⊢  ∀NDA:NDA(Alph;St).
|  Fin(St) ⇒
|  (∀x,y:St, a:Alph. Dec(δNDA(x,a,y))) ⇒
|  (∃S:Type. Fin(S) & (∃DA:Automata(Alph;S). L(NDA) = LangOf(DA)))
| by UnivCD
|
1. Alph : Type
2. St : Type
3. NDA : NDA(Alph;St)
4. Fin(St)
5. ∀x,y:St, a:Alph. Dec(δNDA(x,a,y))
⊢  ∃S:Type. Fin(S) & (∃DA:Automata(Alph;S). L(NDA) = LangOf(DA))
| by Witness St→𝔹
|
⊢  Fin(St→𝔹) & (∃DA:Automata(Alph;St→𝔹). L(NDA) = LangOf(DA))
| by Analyze 0
|
|\
| ⊢  Fin(St→𝔹)
| | by BackThru
| |    Thm* Fin(S) & Fin(T) ⇒ Fin(S→T)    {fun_preserves_fin}
| |
| ⊢  Fin(𝔹)  by BackThru Thm* Fin(𝔹)    {bool_is_finite}
|
.\
. ⊢  ∃DA:Automata(Alph;St→𝔹). L(NDA) = LangOf(DA)
. | by Inst Thm* Fin(T) ⇒ (∀x,y:T. Dec(x = y))    {fin_is_decid}
. |    [St]
. |
. 6. ∀x,y:St. Dec(x = y)
. ⊢   by Witness6 I(NDA)
. |
. 6. ∀y:St. Dec(I(NDA) = y)
. ⊢
. | by Inst
. |    Thm* ∀f:(T→Prop).
. |         (∀t:T. Dec(f(t))) ⇒ (∃g:(T→𝔹). ∀t:T. f(t) ⟺ g(t))
. |    {decid_is_comp}
. |    [St;λy.I(NDA) = y]
. |    THENA
. |    Reduce 0
. |
. 7. ∃g:(St→𝔹). ∀t:St. (λy.I(NDA) = y)(t) ⟺ g(t)
. ⊢   by Reduce 7
. |
. 7. ∃g:(St→𝔹). ∀t:St. I(NDA) = t ⟺ g(t)
. ⊢   by Analyze 7
. |
. 7. g : St→𝔹
. 8. ∀t:St. I(NDA) = t ⟺ g(t)
. ⊢   by Assert (∀f:(St→𝔹). Dec(∃s:St. f(s) & F(NDA)(s)))
. |
. |\
```

```
. | .....assertion .....
. | ⊢  ∀f:(St→𝔹). Dec(∃s:St. f(s) & F(NDA)(s))  by Analyze 0
. | |
. | 9. f : St→𝔹
. | ⊢  Dec(∃s:St. f(s) & F(NDA)(s))
. | | by Inst
. | |    Thm* ∀R:(T→Prop).
. | |         Fin(T) & (∀t:T. Dec(R(t))) ⇒ Dec(∃t:T. R(t))
. | |    {auto2_lemma_6}
. | |    [St;λs.f(s) & F(NDA)(s)]
. | |
. | |\
. | | .....antecedent .....
. | | ⊢  Fin(St) & (∀t:St. Dec((λs.f(s) & F(NDA)(s))(t)))  by Auto
. | | |
. | | .....decidable? .....
. | | 10. t : St
. | | ⊢  Dec((λs.f(s) & F(NDA)(s))(t))  by Reduce 0
. | | |
. | | ⊢  Dec(f(t) & F(NDA)(t))  by ProveDecidable
. | |
. | .\
. | . 10. Dec(∃t:St. (λs.f(s) & F(NDA)(s))(t))
. | . ⊢   by Reduce 10
. |
. .\
. . 9. ∀f:(St→𝔹). Dec(∃s:St. f(s) & F(NDA)(s))
. . ⊢
. . | by Inst
. . |    Thm* ∀f:(T→Prop).
. . |         (∀t:T. Dec(f(t))) ⇒ (∃g:(T→𝔹). ∀t:T. f(t) ⟺ g(t))
. . |    {decid_is_comp}
. . |    [St→𝔹;λf.∃s:St. f(s) & F(NDA)(s)]
. . |
. . |\
. . | .....antecedent .....
. . | ⊢  ∀t:(St→𝔹). Dec((λf.∃s:St. f(s) & F(NDA)(s))(t))
. . |   by Reduce 0
. . |
. . .\
. . . 10. ∃g:((St→𝔹)→𝔹).
. . . |   ∀t:(St→𝔹). (λf.∃s:St. f(s) & F(NDA)(s))(t) ⟺ g(t)
. . . ⊢   by Analyze 10
. . . |
. . . 10. g1 : (St→𝔹)→𝔹
. . . 11. ∀t:(St→𝔹). (λf.∃s:St. f(s) & F(NDA)(s))(t) ⟺ g1(t)
. . . ⊢   by Reduce 11

. . . |
. . . 11. ∀t:(St→𝔹). (∃s:St. t(s) & F(NDA)(s)) ⟺ g1(t)
. . . ⊢
. . . | by Assert
. . . |    (∀f:(St→𝔹), a:Alph, y:St. Dec(∃s:St. f(s) & δNDA(s,a,y)))
. . . |
. . . |\
. . . | .....assertion .....
```

```
. . . | ⊢  ∀f:(St→𝔹), a:Alph, y:St. Dec(∃s:St. f(s) & δNDA(s,a,y))
. . . | | by Auto
. . . | |
. . . | .....decidable? .....
. . . | 12. f : St→𝔹
. . . | 13. a : Alph
. . . | 14. y : St
. . . | ⊢  Dec(∃s:St. f(s) & δNDA(s,a,y))
. . . | | by Inst
. . . | |    Thm* ∀R:(T→Prop).
. . . | |         Fin(T) & (∀t:T. Dec(R(t))) ⇒ Dec(∃t:T. R(t))
. . . | |    {auto2_lemma_6}
. . . | |    [St;λs.f(s) & δNDA(s,a,y)]
. . . | |
. . . | |\
. . . | | .....antecedent .....
. . . | | ⊢  Fin(St) & (∀t:St. Dec((λs.f(s) & δNDA(s,a,y))(t)))
. . . | | | by Auto
. . . | | |
. . . | | .....decidable? .....
. . . | | 15. t : St
. . . | | ⊢  Dec((λs.f(s) & δNDA(s,a,y))(t))  by Reduce 0
. . . | |
. . . | .\
. . . | . 15. Dec(∃t:St. (λs.f(s) & δNDA(s,a,y))(t))
. . . | . ⊢   by Reduce 15
. . . |
. . . .\
. . . . 12. ∀f:(St→𝔹), a:Alph, y:St. Dec(∃s:St. f(s) & δNDA(s,a,y))
. . . . ⊢
. . . . | by Inst
. . . . |    Thm* ∀f:(T→Prop).
. . . . |         (∀t:T. Dec(f(t))) ⇒ (∃g:(T→𝔹). ∀t:T. f(t) ⟺ g(t))
. . . . |    {decid_is_comp}
. . . . |    [(St→𝔹)×Alph×St
. . . . |    ;λt.∃s:St. 1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t)))]
. . . . |
. . . . |\
. . . . | .....antecedent .....
. . . . | ⊢  ∀t:((St→𝔹)×Alph×St).
. . . . | |  Dec((λt.∃s:St.
. . . . | |       1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t))))
. . . . | |      (t))  by Analyze 0
. . . . | |
. . . . | 13. t : (St→𝔹)×Alph×St
. . . . | ⊢  Dec((λt.∃s:St.
. . . . | |       1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t))))
. . . . | |      (t))  by InstHyp [1of(t);1of(2of(t));2of(2of(t))] 12
. . . . | |
. . . . | 14. Dec(∃s:St. 1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t))))
. . . . | ⊢   by Reduce 0
. . . . |
. . . . .\
. . . . . 13. ∃g:((St→𝔹)×Alph×St→𝔹).
. . . . . |   ∀t:((St→𝔹)×Alph×St).
```

```
. . . . . |   (λt.∃s:St. 1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t))))
. . . . . |   (t)
. . . . . |   ⟺
. . . . . |   g(t)
. . . . . ⊢   by Analyze 13
. . . . . |
. . . . . 13. g2 : (St→𝔹)×Alph×St→𝔹
. . . . . 14. ∀t:((St→𝔹)×Alph×St).
. . . . . |   (λt.∃s:St. 1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t))))
. . . . . |   (t)
. . . . . |   ⟺
. . . . . |   g2(t)
. . . . . ⊢   by Reduce 14
. . . . . |
. . . . . 14. ∀t:((St→𝔹)×Alph×St).
. . . . . |   (∃s:St. 1of(t)(s) & δNDA(s,1of(2of(t)),2of(2of(t))))
. . . . . |   ⟺
. . . . . |   g2(t)
. . . . . ⊢
. . . . . | by Assert
. . . . . |    (∃g0:((St→𝔹)→Alph→St→𝔹).
. . . . . |     ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . |     (∃s:St. ds(s) & δNDA(s,a,y)) ⟺ g0(ds,a,y))
. . . . . |
. . . . . |\
. . . . . | .....assertion .....
. . . . . | ⊢  ∃g0:((St→𝔹)→Alph→St→𝔹).
. . . . . | |  ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . | |  (∃s:St. ds(s) & δNDA(s,a,y)) ⟺ g0(ds,a,y)
. . . . . | | by Witness λds,a,y. g2(<ds,a,y>)
. . . . . | |
. . . . . | ⊢  ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . | |  (∃s:St. ds(s) & δNDA(s,a,y))
. . . . . | |  ⟺
. . . . . | |  (λds,a,y. g2(<ds,a,y>))(ds,a,y)  by Reduce 0
. . . . . | |
. . . . . | ⊢  ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . | |  (∃s:St. ds(s) & δNDA(s,a,y)) ⟺ g2(<ds,a,y>)
. . . . . | | by RWH (RevHypC 14) 0
. . . . . | |
. . . . . | ⊢  ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . |    (∃s:St. ds(s) & δNDA(s,a,y))
. . . . . |    ⟺
. . . . . |    (∃s:St.
. . . . . |     1of(<ds,a,y>)(s)
. . . . . |     & δNDA(s,1of(2of(<ds,a,y>)),2of(2of(<ds,a,y>))))
. . . . . |   by Reduce 0
. . . . . |
. . . . . .\
. . . . . . 15. ∃g0:((St→𝔹)→Alph→St→𝔹).
. . . . . . |   ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . . |   (∃s:St. ds(s) & δNDA(s,a,y)) ⟺ g0(ds,a,y)
. . . . . . ⊢   by Analyze 15
. . . . . . |
. . . . . . 15. g0 : (St→𝔹)→Alph→St→𝔹
```

```
. . . . . 16. ∀ds:(St→𝔹), a:Alph, y:St.
. . . . . |   (∃s:St. ds(s) & δNDA(s,a,y)) ⟺ g0(ds,a,y)
. . . . . ⊢   by Witness <g0,g,g1>
. . . . . |
. . . . . |\
. . . . . | .....wf .....
. . . . . | ⊢  <g0,g,g1> ∈ Automata(Alph;St→𝔹)
. . . . . |   by Unfold `automata` 0
. . . . . |
. . . . . |\
. . . . . | ⊢  L(NDA) = LangOf(<g0,g,g1>)  by Unfold `lang_eq` 0
. . . . . | |
. . . . . | ⊢  ∀l:Alph*. L(NDA)(l) ⟺ LangOf(<g0,g,g1>)(l)
. . . . . | | by Analyze 0
. . . . . | |
. . . . . | 17. l : Alph*
. . . . . | ⊢  L(NDA)(l) ⟺ LangOf(<g0,g,g1>)(l)
. . . . . | | by Assert (<g0,g,g1> ∈ Automata(Alph;St→𝔹))
. . . . . | | |    THENA
. . . . . | | |    Unfold `automata` 0
. . . . . | | |
. . . . . | 18. <g0,g,g1> ∈ Automata(Alph;St→𝔹)
. . . . . | ⊢
. . . . . | | by Assert
. . . . . | | |    (∀m:Alph*, q:St.
. . . . . | | |     NDA(m)⊢q ⟺ (Result(<g0,g,g1>)m)(q))
. . . . . | | |
. . . . . | | |\
. . . . . | | | .....assertion .....
. . . . . | | | ⊢  ∀m:Alph*, q:St. NDA(m)⊢q ⟺ (Result(<g0,g,g1>)m)(q)
. . . . . | | | | by Analyze 0
. . . . . | | | |
. . . . . | | | 19. m : Alph*
. . . . . | | | ⊢  ∀q:St. NDA(m)⊢q ⟺ (Result(<g0,g,g1>)m)(q)
. . . . . | | | | by ListInd 19
. . . . . | | | |
. . . . . | | | |\
. . . . . | | | | ⊢  ∀q:St. NDA(nil)⊢q ⟺ (Result(<g0,g,g1>)nil)(q)
. . . . . | | | | | by RecUnfold `nd_compute_list` 0
. . . . . | | | | |    THEN
. . . . . | | | | |    RecUnfold `compute_list` 0
. . . . . | | | | |    THEN
. . . . . | | | | |    Reduce 0
. . . . . | | | | |
. . . . . | | | | ⊢  ∀q:St. q = I(NDA) ⟺ g(q)  by RWH (RevHypC 8) 0
. . . . . | | | |
. . . . . | | | .\
. . . . . | | | . 20. u : Alph
. . . . . | | | . 21. v : Alph*
. . . . . | | | . 22. ∀q:St. NDA(v)⊢q ⟺ (Result(<g0,g,g1>)v)(q)
. . . . . | | | . ⊢  ∀q:St. NDA(u.v)⊢q ⟺ (Result(<g0,g,g1>)u.v)(q)
. . . . . | | | . | by RecUnfold `nd_compute_list` 0
. . . . . | | | . |    THEN
. . . . . | | | . |    RecUnfold `compute_list` 0
. . . . . | | | . |    THEN
```

```
. . . . . | | . |    Reduce 0
. . . . . | | . |
. . . . . | | . ⊢  ∀q:St.
. . . . . | | . |  (∃t:St. NDA(v)⊢t & δNDA(t,u,q))
. . . . . | | . |  ⟺
. . . . . | | . |  g0((Result(<g0,g,g1>)v),u,q)
. . . . . | | . | by RWH (RevHypC 16) 0
. . . . . | | . |
. . . . . | | . ⊢  ∀q:St.
. . . . . | | .    (∃t:St. NDA(v)⊢t & δNDA(t,u,q))
. . . . . | | .    ⟺
. . . . . | | .    (∃s:St. (Result(<g0,g,g1>)v)(s) & δNDA(s,u,q))
. . . . . | | .   by RWH (RevHypC 22) 0
. . . . . | |
. . . . . | .\
. . . . . | . 19. ∀m:Alph*, q:St.
. . . . . | . |   NDA(m)⊢q ⟺ (Result(<g0,g,g1>)m)(q)
. . . . . | . ⊢
. . . . . | . | by Unfold ‘nd_auto_lang‘ 0 THEN Unfold ‘auto_lang‘ 0
. . . . . | . |    THEN
. . . . . | . |    Reduce 0
. . . . . | . |
. . . . . | . ⊢  NDA(l)↓ ⟺ <g0,g,g1>(l)↓
. . . . . | . | by Unfold ‘accept_list‘ 0
. . . . . | . |    THEN
. . . . . | . |    Unfold ‘nd_accept_list‘ 0
. . . . . | . |    THEN

. . . . . | . |    Reduce 0
. . . . . | . |
. . . . . | . ⊢  (∃q:St. NDA(l)⊢q & F(NDA)(q))
. . . . . | . |  ⟺
. . . . . | . |  g1(Result(<g0,g,g1>)l)  by RWH (RevHypC 11) 0
. . . . . | . |
. . . . . | . ⊢  (∃q:St. NDA(l)⊢q & F(NDA)(q))
. . . . . | .    ⟺
. . . . . | .    (∃s:St. (Result(<g0,g,g1>)l)(s) & F(NDA)(s))
. . . . . | .   by RWH (RevHypC 19) 0
. . . . . |
. . . . . .\
. . . . . . .....wf .....
. . . . . . 17. DA : Automata(Alph;St→𝔹)
. . . . . . ⊢  L(NDA) = LangOf(DA) ∈ Prop  by Auto
END
```