change-since-init THM

```
⊢  ∀es:ES, x,i:Id, T:Type, c:T.
|  (∀x,y:T. Dec(x = y))
|  ⇒
|  (vartype(i;x) ⊆r T)
|  ⇒
|  (∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c)
|  ⇒
|  (∀e':E.
|   loc(e') = i
|   ⇒
|   ¬(x after e') = c
|   ⇒
|   (∃ev:E. ev ≤ e'  & ¬(x after ev) = (x when ev)))
| by Repeat (Analyze 0 THENA Complete Auto)
|    THEN
|    AssertBY (∀e:E. loc(e) = i ⇒ (x when e) ∈ T)
|    (Auto THEN DoSubsume THEN HypSubst -1 0)
|    THEN
|    AssertBY (∀e:E. loc(e) = i ⇒ (x after e) ∈ T)
|    (Auto THEN DoSubsume THEN HypSubst -1 0)
|    THEN
|    Try BackThruSomeHyp
|
1. es : ES
2. x : Id
3. i : Id
4. T : Type
5. c : T
6. ∀x,y:T. Dec(x = y)
7. vartype(i;x) ⊆r T
8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
11. e' : E
12. loc(e') = i
13. ¬(x after e') = c
⊢  ∃ev:E. ev ≤ e'  & ¬(x after ev) = (x when ev)
| by Inst
|    Thm* ∀es:ES, e':E. ∃e:E. first(e) & e ≤ e'
|    {es_DASH_first_DASH_exists}
|    [es;e']
|    THEN
|    ExRepD
|
1. es : ES
2. x : Id
3. i : Id
4. T : Type
5. c : T
6. ∀x,y:T. Dec(x = y)
7. vartype(i;x) ⊆r T
8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
```

```
10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
11. e' : E
12. loc(e') = i
13. ¬(x after e') = c
14. e : E
15. first(e)
16. e ≤ e'
⊢  ∃ev:E. ev ≤ e'  & ¬(x after ev) = (x when ev)  by InstHyp [e] -7
|
|\
| .....antecedent .....
| 1. es : ES
| 2. x : Id
| 3. i : Id
| 4. T : Type
| 5. c : T
| 6. ∀x,y:T. Dec(x = y)
| 7. vartype(i;x) ⊆r T
| 8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
| 9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
| 10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
| 11. e' : E
| 12. loc(e') = i
| 13. ¬(x after e') = c
| 14. e : E
| 15. first(e)
| 16. e ≤ e'
| ⊢  loc(e) = i
|   by FwdThru
|      Thm* ∀es:ES, e,e':E. e ≤ e'  ⇒ loc(e) = loc(e') ∈ Id
|      {es_DASH_le_DASH_loc}
|      [-1]
|
.\
. 1. es : ES
. 2. x : Id
. 3. i : Id
. 4. T : Type
. 5. c : T
. 6. ∀x,y:T. Dec(x = y)
. 7. vartype(i;x) ⊆r T
. 8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
. 9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
. 10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
. 11. e' : E
. 12. loc(e') = i
. 13. ¬(x after e') = c
. 14. e : E
. 15. first(e)
. 16. e ≤ e'
. 17. (x when e) = c
. ⊢  ∃ev:E. ev ≤ e'  & ¬(x after ev) = (x when ev)
. | by Inst Thm: change_DASH_lemma [es;x;i;T;e';e]
. |
. |\
```

```
. | .....antecedent .....
. | 1. es : ES
. | 2. x : Id
. | 3. i : Id
. | 4. T : Type
. | 5. c : T
. | 6. ∀x,y:T. Dec(x = y)
. | 7. vartype(i;x) ⊆r T
. | 8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
. | 9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
. | 10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
. | 11. e' : E
. | 12. loc(e') = i
. | 13. ¬(x after e') = c
. | 14. e : E
. | 15. first(e)
. | 16. e ≤ e'
. | 17. (x when e) = c
. | ⊢  ¬(x after e') = (x when e)
. |   by HypSubst -1 0 THEN BackThruSomeHyp
. |
. .\
. . 1. es : ES
. . 2. x : Id
. . 3. i : Id
. . 4. T : Type
. . 5. c : T
. . 6. ∀x,y:T. Dec(x = y)
. . 7. vartype(i;x) ⊆r T
. . 8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
. . 9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
. . 10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
. . 11. e' : E
. . 12. loc(e') = i
. . 13. ¬(x after e') = c
. . 14. e : E
. . 15. first(e)
. . 16. e ≤ e'
. . 17. (x when e) = c
. . 18. ∃ev:E. e ≤ ev  & ev ≤ e'  & ¬(x after ev) = (x when ev)
. . ⊢  ∃ev:E. ev ≤ e'  & ¬(x after ev) = (x when ev)
. . | by ParallelOp -1
. . |    THEN
. . |    Try
. . |    (BackThruSomeHyp
. . |     THEN
. . |     FwdThru
. . |     Thm* ∀es:ES, e,e':E. e ≤ e'  ⇒ loc(e) = loc(e') ∈ Id
. . |     {es_DASH_le_DASH_loc}
. . |     [-1])
. . |
. . 1. es : ES
. . 2. x : Id
. . 3. i : Id
. . 4. T : Type
```

```
. . 5. c : T
. . 6. ∀x,y:T. Dec(x = y)
. . 7. vartype(i;x) ⊆r T
. . 8. ∀e:E. loc(e) = i ⇒ (x when e) ∈ T
. . 9. ∀e:E. loc(e) = i ⇒ (x after e) ∈ T
. . 10. ∀e:E. loc(e) = i ⇒ first(e) ⇒ (x when e) = c
. . 11. e' : E
. . 12. loc(e') = i
. . 13. ¬(x after e') = c
. . 14. e : E
. . 15. first(e)
. . 16. e ≤ e'
. . 17. (x when e) = c
. . 18. ev : E
. . 19. e ≤ ev  & ev ≤ e'  & ¬(x after ev) = (x when ev)
. . ⊢  ev ≤ e'  & ¬(x after ev) = (x when ev)  by Auto
END
```