```
once__realizes  THM

⊢  ∀a,i:Id.
|  λloc.⊕(once(loc;a;i))
|  realizes es.∃e@i.kind(e) = locl(a)
|              & ∀e@i.∀e'@i.kind(e) = locl(a)
|                           ⇒
|                kind(e') = locl(a) ⇒ e = e'
| by RealizerSetup
|    THEN
|    PreInitRule1 i a "done" 𝔹 Unit false₂ (λ₂x,v. ¬x)
|
1. a : Id
2. i : Id
3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
4. D' : Dsys
5. λloc.⊕(once(loc;a;i)) ⊆ D'
6. w : World
7. p : FairFifo
8. PossibleWorld(D';w)
9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
10. vartype(i;"done") ⊆r 𝔹
11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
12. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
14. e : E
15. loc(e) = i
16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
⊢  ∃e@i.kind(e) = locl(a)
|  & ∀e@i.∀e'@i.kind(e) = locl(a) ⇒ kind(e') = locl(a) ⇒ e = e'
| by Assert
|    ∀e@i.¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
|
|\
| .....assertion .....
| 1. a : Id
| 2. i : Id
| 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
| 4. D' : Dsys
| 5. λloc.⊕(once(loc;a;i)) ⊆ D'
| 6. w : World
| 7. p : FairFifo
| 8. PossibleWorld(D';w)
| 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
| 10. vartype(i;"done") ⊆r 𝔹
| 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
| 12. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
| 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
| 14. e : E
| 15. loc(e) = i
| 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
| ⊢  ∀e@i.¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
|   by FrameRule
|
.\
```

```
. 1. a : Id
. 2. i : Id
. 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. 4. D' : Dsys
. 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. 6. w : World
. 7. p : FairFifo
. 8. PossibleWorld(D';w)
. 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. 10. vartype(i;"done") ⊆r 𝔹
. 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. 12. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. 14. e : E
. 15. loc(e) = i
. 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. 17. ∀e@i.¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
. ⊢  ∃e@i.kind(e) = locl(a)
. |  & ∀e@i.∀e'@i.kind(e) = locl(a) ⇒ kind(e') = locl(a) ⇒ e = e'
. | by Analyze 0
. |
. |\
. | 1. a : Id
. | 2. i : Id
. | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | 4. D' : Dsys
. | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | 6. w : World
. | 7. p : FairFifo
. | 8. PossibleWorld(D';w)
. | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. | 10. vartype(i;"done") ⊆r 𝔹
. | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | 12. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | 14. e : E
. | 15. loc(e) = i
. | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. | 17. ∀e@i.¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
. | ⊢  ∃e@i.kind(e) = locl(a)  by Analyze -2
. | |
. | |\
. | | 1. a : Id
. | | 2. i : Id
. | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | | 4. D' : Dsys
. | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | | 6. w : World
. | | 7. p : FairFifo
. | | 8. PossibleWorld(D';w)
. | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. | | 10. vartype(i;"done") ⊆r 𝔹
. | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | | 12. ∀e:E.
. | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
```

```
. | | 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | | 14. e : E
. | | 15. loc(e) = i
. | | 16. kind(e) = locl(a)
. | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | |          ⇒
. | |     kind(e) = locl(a)
. | | ⊢  ∃e@i.kind(e) = locl(a)
. | |   by Unfold ‘existse-at‘ 0 THEN InstConcl [e]
. | |
. | .\
. | . 1. a : Id
. | . 2. i : Id
. | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . 4. D' : Dsys
. | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . 6. w : World
. | . 7. p : FairFifo
. | . 8. PossibleWorld(D';w)
. | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. | . 10. vartype(i;"done") ⊆r 𝔹
. | . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . 12. ∀e:E.
. | . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . 14. e : E
. | . 15. loc(e) = i
. | . 16. ∀v:Unit. ¬¬("done" after e)
. | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . |        ⇒
. | . |   kind(e) = locl(a)
. | . ⊢  ∃e@i.kind(e) = locl(a)  by InstHyp [•] -2
. | . |
. | . 1. a : Id
. | . 2. i : Id
. | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . 4. D' : Dsys
. | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . 6. w : World
. | . 7. p : FairFifo
. | . 8. PossibleWorld(D';w)
. | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. | . 10. vartype(i;"done") ⊆r 𝔹
. | . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . 12. ∀e:E.
. | . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . 14. e : E
. | . 15. loc(e) = i
. | . 16. ∀v:Unit. ¬¬("done" after e)
. | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . |        ⇒
. | . |   kind(e) = locl(a)
. | . 18. ¬¬("done" after e)
. | . ⊢  ∃e@i.kind(e) = locl(a)
```

```
. | . | by Inst Thm: change_DASH_since_DASH_init
. | . |    [ES(w);"done";i;𝔹;false₂;e]
. | . |    THENA
. | . |    (Auto THEN TrySubsume)
. | . |
. | . |\
. | . | .....antecedent .....
. | . | 1. a : Id
. | . | 2. i : Id
. | . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . | 4. D' : Dsys
. | . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . | 6. w : World
. | . | 7. p : FairFifo
. | . | 8. PossibleWorld(D';w)
. | . | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . | |  ∈ Dsys
. | . | 10. vartype(i;"done") ⊆r 𝔹
. | . | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . | 12. ∀e:E.
. | . | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . | 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . | 14. e : E
. | . | 15. loc(e) = i
. | . | 16. ∀v:Unit. ¬¬("done" after e)
. | . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . | |        ⇒
. | . | |   kind(e) = locl(a)
. | . | 18. ¬¬("done" after e)
. | . | ⊢  ¬("done" after e) = false₂
. | . | | by MoveToConcl -1 THEN AutoBoolCase ("done" after e)
. | . | |
. | . | |\
. | . | | .....wf .....
. | . | | 1. a : Id
. | . | | 2. i : Id
. | . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . | | 4. D' : Dsys
. | . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . | | 6. w : World
. | . | | 7. p : FairFifo
. | . | | 8. PossibleWorld(D';w)
. | . | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . | | |  ∈ Dsys
. | . | | 10. vartype(i;"done") ⊆r 𝔹
. | . | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . | | 12. ∀e:E.
. | . | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . | | 13. ∀e:E.
. | . | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . | | 14. e : E
. | . | | 15. loc(e) = i
. | . | | 16. ∀v:Unit. ¬¬("done" after e)
. | . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . | | |        ⇒
```

```
. | . | | |   kind(e) = locl(a)
. | . | | ⊢  ("done" after e) ∈ 𝔹  by DoSubsume THEN TrySubsume
. | . | | |
. | . | | .....subtype_rel .....
. | . | | 1. a : Id
. | . | | 2. i : Id
. | . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . | | 4. D' : Dsys
. | . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . | | 6. w : World
. | . | | 7. p : FairFifo
. | . | | 8. PossibleWorld(D';w)
. | . | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . | |    ∈ Dsys
. | . | | 10. vartype(i;"done") ⊆r 𝔹
. | . | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . | | 12. ∀e:E.
. | . | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . | | 13. ∀e:E.
. | . | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . | | 14. e : E
. | . | | 15. loc(e) = i
. | . | | 16. ∀v:Unit. ¬¬("done" after e)
. | . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . | |          ⇒
. | . | |     kind(e) = locl(a)
. | . | | ⊢  vartype(loc(e);"done") ⊆r 𝔹
. | . | |   by HypSubst' -3 0 THEN TrySubsume
. | . | |
. | . | |\
. | . | | 1. a : Id
. | . | | 2. i : Id
. | . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . | | 4. D' : Dsys
. | . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . | | 6. w : World
. | . | | 7. p : FairFifo
. | . | | 8. PossibleWorld(D';w)
. | . | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . | | |  ∈ Dsys
. | . | | 10. vartype(i;"done") ⊆r 𝔹
. | . | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . | | 12. ∀e:E.
. | . | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . | | 13. ∀e:E.
. | . | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . | | 14. e : E
. | . | | 15. loc(e) = i
. | . | | 16. ∀v:Unit. ¬¬("done" after e)
. | . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . | | |        ⇒
. | . | | |   kind(e) = locl(a)
. | . | | 18. ("done" after e)
. | . | | ⊢  ¬¬True ⇒ ¬true₂ = false₂  by Auto
. | . | | |
```

```
. | . | | 1. a : Id
. | . | | 2. i : Id
. | . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . | | 4. D' : Dsys
. | . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . | | 6. w : World
. | . | | 7. p : FairFifo
. | . | | 8. PossibleWorld(D';w)
. | . | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . | |    ∈ Dsys
. | . | | 10. vartype(i;"done") ⊆r 𝔹
. | . | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . | | 12. ∀e:E.
. | . | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . | | 13. ∀e:E.
. | . | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . | | 14. e : E
. | . | | 15. loc(e) = i
. | . | | 16. ∀v:Unit. ¬¬("done" after e)
. | . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . | |          ⇒
. | . | |     kind(e) = locl(a)
. | . | | 18. ("done" after e)
. | . | | 19. ¬¬True
. | . | | ⊢  ¬true₂ = false₂  by Obvious
. | . | |
. | . | .\
. | . | . 1. a : Id
. | . | . 2. i : Id
. | . | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . | . 4. D' : Dsys
. | . | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . | . 6. w : World
. | . | . 7. p : FairFifo
. | . | . 8. PossibleWorld(D';w)
. | . | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . | .    ∈ Dsys
. | . | . 10. vartype(i;"done") ⊆r 𝔹
. | . | . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . | . 12. ∀e:E.
. | . | .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . | . 13. ∀e:E.
. | . | .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . | . 14. e : E
. | . | . 15. loc(e) = i
. | . | . 16. ∀v:Unit. ¬¬("done" after e)
. | . | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . | .          ⇒
. | . | .     kind(e) = locl(a)
. | . | . 18. ¬("done" after e)
. | . | . ⊢  ¬¬False ⇒ ¬false₂ = false₂
. | . | .   by Auto THEN Analyze -1 THEN Analyze 0
. | . |
. | . .\
. | . . 1. a : Id
```

```
. | . . 2. i : Id
. | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . . 4. D' : Dsys
. | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . . 6. w : World
. | . . 7. p : FairFifo
. | . . 8. PossibleWorld(D';w)
. | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . . |  ∈ Dsys
. | . . 10. vartype(i;"done") ⊆r 𝔹
. | . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . . 12. ∀e:E.
. | . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . . 14. e : E
. | . . 15. loc(e) = i
. | . . 16. ∀v:Unit. ¬¬("done" after e)
. | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . . |        ⇒
. | . . |   kind(e) = locl(a)
. | . . 18. ¬¬("done" after e)
. | . . 19. ∃ev:E. ev ≤ e  & ¬("done" after ev) = ("done" when ev)
. | . . ⊢  ∃e@i.kind(e) = locl(a)
. | . . | by ExRepD THEN Unfold `existse-at` 0 THEN InstConcl [ev]
. | . . |
. | . . |\
. | . . | 1. a : Id
. | . . | 2. i : Id
. | . . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . . | 4. D' : Dsys
. | . . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . . | 6. w : World
. | . . | 7. p : FairFifo
. | . . | 8. PossibleWorld(D';w)
. | . . | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . . |    ∈ Dsys
. | . . | 10. vartype(i;"done") ⊆r 𝔹
. | . . | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . . | 12. ∀e:E.
. | . . |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . . | 13. ∀e:E.
. | . . |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . . | 14. e : E
. | . . | 15. loc(e) = i
. | . . | 16. ∀v:Unit. ¬¬("done" after e)
. | . . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . . |          ⇒
. | . . |     kind(e) = locl(a)
. | . . | 18. ¬¬("done" after e)
. | . . | 19. ev : E
. | . . | 20. ev ≤ e
. | . . | 21. ¬("done" after ev) = ("done" when ev)
. | . . | ⊢  loc(ev) = i
. | . . |   by AllHyps
. | . . |      (λh.
```

```
. | . . |       Try (Fold ‘es-le‘ h)
. | . . |       THEN
. | . . |       FwdThru
. | . . |       Thm* ∀es:ES, e,e':E. e ≤ e'  ⇒ loc(e) = loc(e') ∈ Id
. | . . |       {es_DASH_le_DASH_loc}
. | . . |       [h])
. | . . |
. | . . .\
. | . . . 1. a : Id
. | . . . 2. i : Id
. | . . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . . . 4. D' : Dsys
. | . . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . . . 6. w : World
. | . . . 7. p : FairFifo
. | . . . 8. PossibleWorld(D';w)
. | . . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . . . |  ∈ Dsys
. | . . . 10. vartype(i;"done") ⊆r 𝔹
. | . . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . . . 12. ∀e:E.
. | . . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . . . 13. ∀e:E.
. | . . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . . . 14. e : E
. | . . . 15. loc(e) = i
. | . . . 16. ∀v:Unit. ¬¬("done" after e)
. | . . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. | . . . |        ⇒
. | . . . |   kind(e) = locl(a)
. | . . . 18. ¬¬("done" after e)
. | . . . 19. ev : E
. | . . . 20. ev ≤ e
. | . . . 21. ¬("done" after ev) = ("done" when ev)
. | . . . ⊢  kind(ev) = locl(a)
. | . . . | by AllHyps (λh.Unfold ‘alle-at‘ h THEN BackThru h)
. | . . . |
. | . . . 1. a : Id
. | . . . 2. i : Id
. | . . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. | . . . 4. D' : Dsys
. | . . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. | . . . 6. w : World
. | . . . 7. p : FairFifo
. | . . . 8. PossibleWorld(D';w)
. | . . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. | . . .    ∈ Dsys
. | . . . 10. vartype(i;"done") ⊆r 𝔹
. | . . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. | . . . 12. ∀e:E.
. | . . .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. | . . . 13. ∀e:E.
. | . . .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. | . . . 14. e : E
. | . . . 15. loc(e) = i
```

```
. | . . . 16. ∀v:Unit. ¬¬("done" after e)
. | . . . 17. ∀e:E.
. | . . .     loc(e) = i
. | . . .     ⇒
. | . . .     ¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
. | . . . 18. ¬¬("done" after e)
. | . . . 19. ev : E
. | . . . 20. ev ≤ e
. | . . . 21. ¬("done" after ev) = ("done" when ev)
. | . . . ⊢  loc(ev) = i
. | . . .   by AllHyps
. | . . .      (λh.
. | . . .       Try (Fold ‘es-le‘ h)
. | . . .       THEN
. | . . .       FwdThru
. | . . .       Thm* ∀es:ES, e,e':E. e ≤ e'  ⇒ loc(e) = loc(e') ∈ Id
. | . . .       {es_DASH_le_DASH_loc}
. | . . .       [h])
. |
. .\
. . 1. a : Id
. . 2. i : Id
. . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . 4. D' : Dsys
. . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . 6. w : World
. . 7. p : FairFifo
. . 8. PossibleWorld(D';w)
. . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. . 10. vartype(i;"done") ⊆r 𝔹
. . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . 12. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . 14. e : E
. . 15. loc(e) = i
. . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . 17. ∀e@i.¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
. . ⊢  ∀e@i.∀e'@i.kind(e) = locl(a) ⇒ kind(e') = locl(a) ⇒ e = e'
. . | by Unfold ‘alle-at‘ 0
. . |
. . 1. a : Id
. . 2. i : Id
. . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . 4. D' : Dsys
. . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . 6. w : World
. . 7. p : FairFifo
. . 8. PossibleWorld(D';w)
. . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. . 10. vartype(i;"done") ⊆r 𝔹
. . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . 12. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . 14. e : E
. . 15. loc(e) = i
```

```
. . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . 17. ∀e@i.¬("done" after e) = ("done" when e) ⇒ kind(e) = locl(a)
. . ⊢  ∀e:E.
. .  |  loc(e) = i
. .  |  ⇒
. .  |  (∀e':E.
. .  |   loc(e') = i
. .  |   ⇒
. .  |   kind(e) = locl(a) ⇒ kind(e') = locl(a) ⇒ e = e')
. .  | by Assert
. .  |    (∀e':E.
. .  |     loc(e') = i
. .  |     ⇒
. .  |     (∀e:E.
. .  |      (e <loc e') ⇒ kind(e) = locl(a) ⇒ ¬kind(e') = locl(a)))
. .  |
. .  |\
. .  | .....assertion .....
. .  | 1. a : Id
. .  | 2. i : Id
. .  | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. .  | 4. D' : Dsys
. .  | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. .  | 6. w : World
. .  | 7. p : FairFifo
. .  | 8. PossibleWorld(D';w)
. .  | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. .  | 10. vartype(i;"done") ⊆r 𝔹
. .  | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. .  | 12. ∀e:E.
. .  |  |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. .  | 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. .  | 14. e : E
. .  | 15. loc(e) = i
. .  | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. .  | 17. ∀e@i.¬("done" after e) = ("done" when e)
. .  |  |        ⇒
. .  |  |   kind(e) = locl(a)
. .  | ⊢  ∀e':E.
. .  |  |  loc(e') = i
. .  |  |  ⇒
. .  |  |  (∀e:E.
. .  |  |   (e <loc e') ⇒ kind(e) = locl(a) ⇒ ¬kind(e') = locl(a))
. .  |  | by Assert
. .  |  |    (∀e:E.
. .  |  |     loc(e) = i
. .  |  |     ⇒
. .  |  |     kind(e) = locl(a)
. .  |  |     ⇒
. .  |  |     (∀e':E. (e <loc e') ⇒ ("done" when e')))
. .  |  |
. .  |  |\
. .  |  |  .....assertion .....
. .  |  |  1. a : Id
. .  |  |  2. i : Id
```

```
. . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | 4. D' : Dsys
. . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | 6. w : World
. . | | 7. p : FairFifo
. . | | 8. PossibleWorld(D';w)
. . | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | |  ∈ Dsys
. . | | 10. vartype(i;"done") ⊆r 𝔹
. . | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | 12. ∀e:E.
. . | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | 14. e : E
. . | | 15. loc(e) = i
. . | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | |        ⇒
. . | | |   kind(e) = locl(a)
. . | | ⊢  ∀e:E.
. . | | |  loc(e) = i
. . | | |  ⇒
. . | | |  kind(e) = locl(a)
. . | | |  ⇒
. . | | |  (∀e':E. (e <loc e') ⇒ ("done" when e'))
. . | | | by RepeatFor 3 (Analyze 0 THENA (Auto THEN TrySubsume))
. . | | |
. . | | 1. a : Id
. . | | 2. i : Id
. . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | 4. D' : Dsys
. . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | 6. w : World
. . | | 7. p : FairFifo
. . | | 8. PossibleWorld(D';w)
. . | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | |  ∈ Dsys
. . | | 10. vartype(i;"done") ⊆r 𝔹
. . | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | 12. ∀e:E.
. . | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | 14. e : E
. . | | 15. loc(e) = i
. . | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | |        ⇒
. . | | |   kind(e) = locl(a)
. . | | 18. e1 : E
. . | | 19. loc(e1) = i
. . | | 20. kind(e1) = locl(a)
. . | | ⊢  ∀e':E. (e1 <loc e') ⇒ ("done" when e')  by LocLessInd

. . | | |
. . | | |\
. . | | | .....subterm .....
```

```
. . | | | 1. a : Id
. . | | | 2. i : Id
. . | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | 4. D' : Dsys
. . | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | 6. w : World
. . | | | 7. p : FairFifo
. . | | | 8. PossibleWorld(D';w)
. . | | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | |    ∈ Dsys
. . | | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | 12. ∀e:E.
. . | | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | 13. ∀e:E.
. . | | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | 14. e : E
. . | | | 15. loc(e) = i
. . | | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | |          ⇒
. . | | |     kind(e) = locl(a)
. . | | | 18. e1 : E
. . | | | 19. loc(e1) = i
. . | | | 20. kind(e1) = locl(a)
. . | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | 22. e' : E
. . | | | 23. (e1 <loc e')
. . | | | ⊢  ("done" when e') ∈ 𝔹
. . | | |   by DoSubsume THEN TrySubsume THEN Subst (loc(e') = i) 0
. . | | |      THEN
. . | | |      TrySubsume
. . | | |      THEN
. . | | |      Unfold `es-locl` -1
. . | | |
. . | | |\
. . | | | 1. a : Id
. . | | | 2. i : Id
. . | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | 4. D' : Dsys
. . | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | 6. w : World
. . | | | 7. p : FairFifo
. . | | | 8. PossibleWorld(D';w)
. . | | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | | |  ∈ Dsys
. . | | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | 12. ∀e:E.
. . | | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | 13. ∀e:E.
. . | | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | 14. e : E
. . | | | 15. loc(e) = i
. . | | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
```

```
. . | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | |        ⇒
. . | | | |   kind(e) = locl(a)
. . | | | 18. e1 : E
. . | | | 19. loc(e1) = i
. . | | | 20. kind(e1) = locl(a)
. . | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | 22. j : E
. . | | | 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | ⊢  (e1 <loc j) ⇒ ("done" when j)  by Analyze 0
. . | | | |
. . | | | 1. a : Id
. . | | | 2. i : Id
. . | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | 4. D' : Dsys
. . | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | 6. w : World
. . | | | 7. p : FairFifo
. . | | | 8. PossibleWorld(D';w)
. . | | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | | |  ∈ Dsys
. . | | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | 12. ∀e:E.
. . | | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | 13. ∀e:E.
. . | | | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | 14. e : E
. . | | | 15. loc(e) = i
. . | | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | |        ⇒
. . | | | |   kind(e) = locl(a)
. . | | | 18. e1 : E
. . | | | 19. loc(e1) = i
. . | | | 20. kind(e1) = locl(a)
. . | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | 22. j : E
. . | | | 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | 24. (e1 <loc j)
. . | | | ⊢  ("done" when j)
. . | | | | by AssertBY ("done" ∈ Id) TrySubsume

. . | | | |    THEN
. . | | | |    InstESAxiom ES(w) [j;"done"] 6
. . | | | |    THENA
. . | | | |    (Auto THEN TrySubsume)
. . | | | |
. . | | | |\
. . | | | | .....antecedent .....
. . | | | | 1. a : Id
. . | | | | 2. i : Id
. . | | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | | 4. D' : Dsys
. . | | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | | 6. w : World
```

```
. . | | | | 7. p : FairFifo
. . | | | | 8. PossibleWorld(D';w)
. . | | | | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | | |    ∈ Dsys
. . | | | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | | 11. ∀e:E.
. . | | | |     loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | | 12. ∀e:E.
. . | | | |     loc(e) = i
. . | | | |     ⇒
. . | | | |     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | | 13. ∀e:E.
. . | | | |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | | 14. e : E
. . | | | | 15. loc(e) = i
. . | | | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | |          ⇒
. . | | | |     kind(e) = locl(a)
. . | | | | 18. e1 : E
. . | | | | 19. loc(e1) = i
. . | | | | 20. kind(e1) = locl(a)
. . | | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | | 22. j : E
. . | | | | 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | | 24. (e1 <loc j)
. . | | | | 25. "done" ∈ Id
. . | | | | 26. ∀e:E.
. . | | | |     ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | | | | ⊢  ¬first(j)
. . | | | |   by RW0
. . | | | |      Thm* ∀the_es:ES, e,e':E.
. . | | | |           (e <loc e')
. . | | | |           ⟺
. . | | | |           ¬first(e')
. . | | | |           & e = pred(e') ∈ E ∨ (e <loc pred(e'))
. . | | | |      {es_DASH_locl_DASH_iff}
. . | | | |      -3
. . | | | |
. . | | | .\
. . | | | . 1. a : Id
. . | | | . 2. i : Id
. . | | | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . 4. D' : Dsys
. . | | | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . 6. w : World
. . | | | . 7. p : FairFifo
. . | | | . 8. PossibleWorld(D';w)
. . | | | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | | . |  ∈ Dsys
. . | | | . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . 11. ∀e:E.
. . | | | . |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . 12. ∀e:E.
. . | | | . |   loc(e) = i
```

```
. . | | | . |   ⇒
. . | | | . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . 13. ∀e:E.
. . | | | . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . 14. e : E
. . | | | . 15. loc(e) = i
. . | | | . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . |       ⇒
. . | | | . |   kind(e) = locl(a)
. . | | | . 18. e1 : E
. . | | | . 19. loc(e1) = i
. . | | | . 20. kind(e1) = locl(a)
. . | | | . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . 22. j : E
. . | | | . 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . 24. (e1 <loc j)
. . | | | . 25. "done" ∈ Id
. . | | | . 26. ("done" when j) = ("done" after pred(j))
. . | | | . ⊢  ("done" when j)  by HypSubst -1 0
. . | | | . |
. . | | | . |\
. . | | | . | 1. a : Id
. . | | | . | 2. i : Id
. . | | | . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | 4. D' : Dsys
. . | | | . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | 6. w : World
. . | | | . | 7. p : FairFifo
. . | | | . | 8. PossibleWorld(D';w)
. . | | | . | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;...;x,v.¬x)
. . | | | . | |  ∈ Dsys
. . | | | . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | 11. ∀e:E.
. . | | | . | |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . | 12. ∀e:E.
. . | | | . | |   loc(e) = i
. . | | | . | |   ⇒
. . | | | . | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | 13. ∀e:E.
. . | | | . | |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | 14. e : E
. . | | | . | 15. loc(e) = i
. . | | | . | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | |       ⇒
. . | | | . | |   kind(e) = locl(a)
. . | | | . | 18. e1 : E
. . | | | . | 19. loc(e1) = i
. . | | | . | 20. kind(e1) = locl(a)
. . | | | . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | 22. j : E
. . | | | . | 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | 24. (e1 <loc j)
. . | | | . | 25. "done" ∈ Id
```

```
. . | | | . | 26. ("done" when j) = ("done" after pred(j))
. . | | | . | ⊢  ("done" after pred(j))
. . | | | . | | by RWO
. . | | | . | |    Thm* ∀the_es:ES, e,e':E.
. . | | | . | |         (e <loc e')
. . | | | . | |         ⟺
. . | | | . | |         ¬first(e')
. . | | | . | |         & e = pred(e') ∈ E ∨ (e <loc pred(e'))
. . | | | . | |    {es_DASH_locl_DASH_iff}
. . | | | . | |    -3
. . | | | . | |    THEN
. . | | | . | |    Analyze -3
. . | | | . | |
. . | | | . | |\
. . | | | . | | 1. a : Id
. . | | | . | | 2. i : Id
. . | | | . | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | 4. D' : Dsys
. . | | | . | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | 6. w : World
. . | | | . | | 7. p : FairFifo
. . | | | . | | 8. PossibleWorld(D';w)
. . | | | . | | 9. @i: ma-single-pre-init1("done";𝔹;...;a;Unit;x,v.¬x)
. . | | | . | | |  ∈ Dsys
. . | | | . | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | | 11. ∀e:E.
. . | | | . | | |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . | | 12. ∀e:E.
. . | | | . | | |   loc(e) = i
. . | | | . | | |   ⇒
. . | | | . | | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | 13. ∀e:E.
. . | | | . | | |   loc(e) = i
. . | | | . | | |   ⇒
. . | | | . | | |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | | 14. e : E
. . | | | . | | 15. loc(e) = i
. . | | | . | | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | |        ⇒
. . | | | . | | |   kind(e) = locl(a)
. . | | | . | | 18. e1 : E
. . | | | . | | 19. loc(e1) = i
. . | | | . | | 20. kind(e1) = locl(a)
. . | | | . | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | 22. j : E
. . | | | . | | 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | 24. ¬first(j)
. . | | | . | | 25. e1 = pred(j)
. . | | | . | | 26. "done" ∈ Id
. . | | | . | | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | ⊢  ("done" after pred(j))  by StrongRevHypSubst -3 0
. . | | | . | | |
. . | | | . | | |\
. . | | | . | | | 1. a : Id
```

```
. . | | | . | | | 2. i : Id
. . | | | . | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | | 4. D' : Dsys
. . | | | . | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | | 6. w : World
. . | | | . | | | 7. p : FairFifo
. . | | | . | | | 8. PossibleWorld(D';w)
. . | | | . | | | 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | | |  | ∈ Dsys
. . | | | . | | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | | | 11. ∀e:E.
. . | | | . | | |  |  loc(e) = i
. . | | | . | | |  |  ⇒
. . | | | . | | |  |  first(e) ⇒ ("done" when e) = false₂
. . | | | . | | | 12. ∀e:E.
. . | | | . | | |  |  loc(e) = i
. . | | | . | | |  |  ⇒
. . | | | . | | |  |  kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | | 13. ∀e:E.
. . | | | . | | |  |  loc(e) = i
. . | | | . | | |  |  ⇒
. . | | | . | | |  |  kind(e) = locl(a) ⇒ ¬("done" when e)

. . | | | . | | | 14. e : E
. . | | | . | | | 15. loc(e) = i
. . | | | . | | | 16. kind(e) = locl(a)
. . | | | . | | |  |  ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | |  |       ⇒
. . | | | . | | |  |  kind(e) = locl(a)
. . | | | . | | | 18. e1 : E
. . | | | . | | | 19. loc(e1) = i
. . | | | . | | | 20. kind(e1) = locl(a)
. . | | | . | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | | 22. j : E
. . | | | . | | | 23. ∀k:E.
. . | | | . | | |  |  (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | | 24. ¬first(j)
. . | | | . | | | 25. e1 = pred(j)
. . | | | . | | | 26. "done" ∈ Id
. . | | | . | | | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | | ⊢  ("done" after e1)
. . | | | . | | |  | by EffectRule0 i "done" locl(a) Unit 𝔹
. . | | | . | | |  |    (λx,v. true₂)
. . | | | . | | |  |
. . | | | . | | | 1. a : Id
. . | | | . | | | 2. i : Id
. . | | | . | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | | 4. D' : Dsys
. . | | | . | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | | 6. w : World
. . | | | . | | | 7. p : FairFifo
. . | | | . | | | 8. PossibleWorld(D';w)
. . | | | . | | | 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | | |  | ∈ Dsys
. . | | | . | | | 10. vartype(i;"done") ⊆r 𝔹
```

```
. . | | | . | | | 11. ∀e:E.
. . | | | . | | | |   loc(e) = i
. . | | | . | | | |   ⇒
. . | | | . | | | |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | | | 12. ∀e:E.
. . | | | . | | | |   loc(e) = i
. . | | | . | | | |   ⇒
. . | | | . | | | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | | 13. ∀e:E.
. . | | | . | | | |   loc(e) = i
. . | | | . | | | |   ⇒
. . | | | . | | | |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | | | 14. e : E
. . | | | . | | | 15. loc(e) = i
. . | | | . | | | 16. kind(e) = locl(a)
. . | | | . | | | |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | | |        ⇒
. . | | | . | | | |   kind(e) = locl(a)
. . | | | . | | | 18. e1 : E
. . | | | . | | | 19. loc(e1) = i
. . | | | . | | | 20. kind(e1) = locl(a)
. . | | | . | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | | 22. j : E
. . | | | . | | | 23. ∀k:E.
. . | | | . | | | |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | | 24. ¬first(j)
. . | | | . | | | 25. e1 = pred(j)
. . | | | . | | | 26. "done" ∈ Id
. . | | | . | | | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | | 28. @i: ma-single-effect0("done";𝔹;locl(a);Unit;...)
. . | | | . | | | |   ∈ Dsys
. . | | | . | | | 29. vartype(i;"done") ⊆r 𝔹
. . | | | . | | | 30. ∀e:E.
. . | | | . | | | |   loc(e) = i
. . | | | . | | | |   ⇒
. . | | | . | | | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | | 31. ∀e:E.
. . | | | . | | | |   loc(e) = i
. . | | | . | | | |   ⇒
. . | | | . | | | |   kind(e) = locl(a) ⇒ ("done" after e) = true₂
. . | | | . | | | ⊢  ("done" after e1)  by InstHyp [e1] -1
. . | | | . | | | |
. . | | | . | | | 1. a : Id
. . | | | . | | | 2. i : Id
. . | | | . | | | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | | 4. D' : Dsys
. . | | | . | | | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | | 6. w : World
. . | | | . | | | 7. p : FairFifo
. . | | | . | | | 8. PossibleWorld(D';w)
. . | | | . | | | 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | | |    ∈ Dsys
. . | | | . | | | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | | | 11. ∀e:E.
```

```
. . | | | . | | |     loc(e) = i
. . | | | . | | |     ⇒
. . | | | . | | |     first(e) ⇒ ("done" when e) = false₂
. . | | | . | | | 12. ∀e:E.
. . | | | . | | |     loc(e) = i
. . | | | . | | |     ⇒
. . | | | . | | |     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | | 13. ∀e:E.
. . | | | . | | |     loc(e) = i
. . | | | . | | |     ⇒
. . | | | . | | |     kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | | | 14. e : E

. . | | | . | | | 15. loc(e) = i
. . | | | . | | | 16. kind(e) = locl(a)
. . | | | . | | |     ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | |          ⇒
. . | | | . | | |     kind(e) = locl(a)
. . | | | . | | | 18. e1 : E
. . | | | . | | | 19. loc(e1) = i
. . | | | . | | | 20. kind(e1) = locl(a)
. . | | | . | | | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | | 22. j : E
. . | | | . | | | 23. ∀k:E.
. . | | | . | | |     (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | | 24. ¬first(j)
. . | | | . | | | 25. e1 = pred(j)
. . | | | . | | | 26. "done" ∈ Id
. . | | | . | | | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | | 28. @i: ma-single-effect0("done";𝔹;locl(a);Unit;...)
. . | | | . | | |     ∈ Dsys
. . | | | . | | | 29. vartype(i;"done") ⊆r 𝔹
. . | | | . | | | 30. ∀e:E.
. . | | | . | | |     loc(e) = i
. . | | | . | | |     ⇒
. . | | | . | | |     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | | 31. ∀e:E.
. . | | | . | | |     loc(e) = i
. . | | | . | | |     ⇒
. . | | | . | | |     kind(e) = locl(a) ⇒ ("done" after e) = true₂
. . | | | . | | | 32. ("done" after e1) = true₂
. . | | | . | | | ⊢  ("done" after e1)
. . | | | . | | |   by HypSubst -1 0 THEN Reduce 0
. . | | | . | | |
. . | | | . | | .\
. . | | | . | | . .....subterm .....
. . | | | . | | . 1. a : Id
. . | | | . | | . 2. i : Id
. . | | | . | | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | . 4. D' : Dsys
. . | | | . | | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | . 6. w : World
. . | | | . | | . 7. p : FairFifo
. . | | | . | | . 8. PossibleWorld(D';w)
. . | | | . | | . 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
```

```
. . | | | . | | . |  ∈ Dsys
. . | | | . | | . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | | . 11. ∀e:E.
. . | | | . | | . |   loc(e) = i
. . | | | . | | . |   ⇒
. . | | | . | | . |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | | . 12. ∀e:E.
. . | | | . | | . |   loc(e) = i
. . | | | . | | . |   ⇒
. . | | | . | | . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | . 13. ∀e:E.
. . | | | . | | . |   loc(e) = i
. . | | | . | | . |   ⇒
. . | | | . | | . |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | | . 14. e : E
. . | | | . | | . 15. loc(e) = i
. . | | | . | | . 16. kind(e) = locl(a)
. . | | | . | | . |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | . |        ⇒
. . | | | . | | . |   kind(e) = locl(a)
. . | | | . | | . 18. e1 : E
. . | | | . | | . 19. loc(e1) = i
. . | | | . | | . 20. kind(e1) = locl(a)
. . | | | . | | . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | . 22. j : E
. . | | | . | | . 23. ∀k:E.
. . | | | . | | . |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | . 24. ¬first(j)
. . | | | . | | . 25. e1 = pred(j)
. . | | | . | | . 26. "done" ∈ Id
. . | | | . | | . 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | . 28. z : E
. . | | | . | | . 29. z = e1
. . | | | . | | . ⊢  ("done" after z) ∈ 𝔹  by DoSubsume
. . | | | . | | . |
. . | | | . | | . .....subtype_rel .....
. . | | | . | | . 1. a : Id
. . | | | . | | . 2. i : Id
. . | | | . | | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | . 4. D' : Dsys
. . | | | . | | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | . 6. w : World
. . | | | . | | . 7. p : FairFifo
. . | | | . | | . 8. PossibleWorld(D';w)
. . | | | . | | . 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | | . |  ∈ Dsys
. . | | | . | | . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | | . 11. ∀e:E.
. . | | | . | | . |   loc(e) = i
. . | | | . | | . |   ⇒
. . | | | . | | . |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | | . 12. ∀e:E.
. . | | | . | | . |   loc(e) = i
. . | | | . | | . |   ⇒
```

```
. . | | | . | | . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | . 13. ∀e:E.
. . | | | . | | . |   loc(e) = i
. . | | | . | | . |   ⇒
. . | | | . | | . |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | | . 14. e : E
. . | | | . | | . 15. loc(e) = i
. . | | | . | | . 16. kind(e) = locl(a)
. . | | | . | | . |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | . |       ⇒
. . | | | . | | . |   kind(e) = locl(a)
. . | | | . | | . 18. e1 : E
. . | | | . | | . 19. loc(e1) = i
. . | | | . | | . 20. kind(e1) = locl(a)
. . | | | . | | . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | . 22. j : E
. . | | | . | | . 23. ∀k:E.
. . | | | . | | . |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | . 24. ¬first(j)
. . | | | . | | . 25. e1 = pred(j)
. . | | | . | | . 26. "done" ∈ Id
. . | | | . | | . 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | . 28. z : E
. . | | | . | | . 29. z = e1
. . | | | . | | . ⊢  vartype(loc(z);"done") ⊆r 𝔹
. . | | | . | | . | by Subst (loc(z) = i) 0 THEN TrySubsume
. . | | | . | | . |
. . | | | . | | . .....equality .....
. . | | | . | | . 1. a : Id
. . | | | . | | . 2. i : Id
. . | | | . | | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | | . 4. D' : Dsys
. . | | | . | | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | | . 6. w : World
. . | | | . | | . 7. p : FairFifo
. . | | | . | | . 8. PossibleWorld(D';w)
. . | | | . | | . 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | | .    ∈ Dsys
. . | | | . | | . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | | . 11. ∀e:E.
. . | | | . | | .     loc(e) = i
. . | | | . | | .     ⇒
. . | | | . | | .     first(e) ⇒ ("done" when e) = false2
. . | | | . | | . 12. ∀e:E.
. . | | | . | | .     loc(e) = i
. . | | | . | | .     ⇒
. . | | | . | | .     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | | . 13. ∀e:E.
. . | | | . | | .     loc(e) = i
. . | | | . | | .     ⇒
. . | | | . | | .     kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | | . 14. e : E
. . | | | . | | . 15. loc(e) = i
. . | | | . | | . 16. kind(e) = locl(a)
```

```
. . | | | . | | .     ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | | .          ⇒
. . | | | . | | .     kind(e) = locl(a)
. . | | | . | | . 18. e1 : E
. . | | | . | | . 19. loc(e1) = i
. . | | | . | | . 20. kind(e1) = locl(a)
. . | | | . | | . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | | . 22. j : E
. . | | | . | | . 23. ∀k:E.
. . | | | . | | .     (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | | . 24. ¬first(j)
. . | | | . | | . 25. e1 = pred(j)
. . | | | . | | . 26. "done" ∈ Id
. . | | | . | | . 27. ("done" when j) = ("done" after pred(j))
. . | | | . | | . 28. z : E
. . | | | . | | . 29. z = e1
. . | | | . | | . ⊢  loc(z) = i  by HypSubst -1 0
. . | | | . | |
. . | | | . | .\
. . | | | . | . 1. a : Id
. . | | | . | . 2. i : Id
. . | | | . | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . 4. D' : Dsys
. . | | | . | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . 6. w : World
. . | | | . | . 7. p : FairFifo
. . | | | . | . 8. PossibleWorld(D';w)
. . | | | . | . 9. @i: ma-single-pre-init1("done";𝔹;...;a;Unit;x,v.¬x)
. . | | | . | . |  ∈ Dsys
. . | | | . | . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . 11. ∀e:E.
. . | | | . | . |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . | . 12. ∀e:E.
. . | | | . | . |   loc(e) = i
. . | | | . | . |   ⇒
. . | | | . | . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . 13. ∀e:E.
. . | | | . | . |   loc(e) = i
. . | | | . | . |   ⇒
. . | | | . | . |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . 14. e : E
. . | | | . | . 15. loc(e) = i
. . | | | . | . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | . |        ⇒
. . | | | . | . |   kind(e) = locl(a)
. . | | | . | . 18. e1 : E
. . | | | . | . 19. loc(e1) = i
. . | | | . | . 20. kind(e1) = locl(a)
. . | | | . | . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . 22. j : E
. . | | | . | . 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . 24. ¬first(j)
. . | | | . | . 25. (e1 <loc pred(j))
```

```
. . | | | . | . 26. "done" ∈ Id
. . | | | . | . 27. ("done" when j) = ("done" after pred(j))
. . | | | . | . ⊢  ("done" after pred(j))
. . | | | . | . | by Assert ("done" when pred(j))
. . | | | . | . |
. . | | | . | . |\
. . | | | . | . | .....assertion .....
. . | | | . | . | 1. a : Id
. . | | | . | . | 2. i : Id
. . | | | . | . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . | 4. D' : Dsys
. . | | | . | . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . | 6. w : World
. . | | | . | . | 7. p : FairFifo
. . | | | . | . | 8. PossibleWorld(D';w)
. . | | | . | . | 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | . |    ∈ Dsys
. . | | | . | . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . | 11. ∀e:E.
. . | | | . | . |     loc(e) = i
. . | | | . | . |     ⇒
. . | | | . | . |     first(e) ⇒ ("done" when e) = false₂
. . | | | . | . | 12. ∀e:E.
. . | | | . | . |     loc(e) = i
. . | | | . | . |     ⇒
. . | | | . | . |     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . | 13. ∀e:E.
. . | | | . | . |     loc(e) = i
. . | | | . | . |     ⇒
. . | | | . | . |     kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . | 14. e : E
. . | | | . | . | 15. loc(e) = i
. . | | | . | . | 16. kind(e) = locl(a)
. . | | | . | . |     ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | . |          ⇒
. . | | | . | . |     kind(e) = locl(a)
. . | | | . | . | 18. e1 : E
. . | | | . | . | 19. loc(e1) = i
. . | | | . | . | 20. kind(e1) = locl(a)
. . | | | . | . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . | 22. j : E
. . | | | . | . | 23. ∀k:E.
. . | | | . | . |     (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . | 24. ¬first(j)
. . | | | . | . | 25. (e1 <loc pred(j))
. . | | | . | . | 26. "done" ∈ Id
. . | | | . | . | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | . | ⊢  ("done" when pred(j))
. . | | | . | . |   by BackThruSomeHyp
. . | | | . | . |      THEN
. . | | | . | . |      BackThru
. . | | | . | . |      Thm* ∀the_es:ES, j:E.
. . | | | . | . |           ¬first(j) ⇒ (pred(j) <loc j)
. . | | | . | . |      {es_DASH_pred_DASH_locl}
```

```
. . | | | . | . |
. . | | | . | . .\
. . | | | . | . . 1. a : Id
. . | | | . | . . 2. i : Id
. . | | | . | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . . 4. D' : Dsys
. . | | | . | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . . 6. w : World
. . | | | . | . . 7. p : FairFifo
. . | | | . | . . 8. PossibleWorld(D';w)
. . | | | . | . . 9. @i: ma-single-pre-init1(...;𝔹;...;a;Unit;x,v.¬x)
. . | | | . | . . |  ∈ Dsys
. . | | | . | . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . 11. ∀e:E.
. . | | | . | . . |   loc(e) = i
. . | | | . | . . |   ⇒
. . | | | . | . . |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . 12. ∀e:E.
. . | | | . | . . |   loc(e) = i
. . | | | . | . . |   ⇒
. . | | | . | . . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . 13. ∀e:E.
. . | | | . | . . |   loc(e) = i
. . | | | . | . . |   ⇒
. . | | | . | . . |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . 14. e : E
. . | | | . | . . 15. loc(e) = i
. . | | | . | . . 16. kind(e) = locl(a)
. . | | | . | . . |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | . . |        ⇒
. . | | | . | . . |   kind(e) = locl(a)
. . | | | . | . . 18. e1 : E
. . | | | . | . . 19. loc(e1) = i
. . | | | . | . . 20. kind(e1) = locl(a)
. . | | | . | . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . 22. j : E
. . | | | . | . . 23. ∀k:E.
. . | | | . | . . |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . 24. ¬first(j)
. . | | | . | . . 25. (e1 <loc pred(j))
. . | | | . | . . 26. "done" ∈ Id
. . | | | . | . . 27. ("done" when j) = ("done" after pred(j))
. . | | | . | . . 28. ("done" when pred(j))
. . | | | . | . . ⊢  ("done" after pred(j))
. . | | | . | . . | by AutoBoolCase ("done" after pred(j))
. . | | | . | . . |
. . | | | . | . . |\
. . | | | . | . . | .....wf .....
. . | | | . | . . | 1. a : Id
. . | | | . | . . | 2. i : Id
. . | | | . | . . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . . | 4. D' : Dsys
. . | | | . | . . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . . | 6. w : World
```

```
. . | | | . | . . | 7. p : FairFifo
. . | | | . | . . | 8. PossibleWorld(D';w)
. . | | | . | . . | 9. @i: ma-single-pre-init1(...;𝔹;...;a;...;x,v.¬x)
. . | | | . | . . | |  ∈ Dsys
. . | | | . | . . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . | 11. ∀e:E.
. . | | | . | . . | |   loc(e) = i
. . | | | . | . . | |   ⇒
. . | | | . | . . | |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . | 12. ∀e:E.
. . | | | . | . . | |   loc(e) = i
. . | | | . | . . | |   ⇒
. . | | | . | . . | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . | 13. ∀e:E.
. . | | | . | . . | |   loc(e) = i
. . | | | . | . . | |   ⇒
. . | | | . | . . | |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . | 14. e : E
. . | | | . | . . | 15. loc(e) = i
. . | | | . | . . | 16. kind(e) = locl(a)
. . | | | . | . . | |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | . . | |       ⇒
. . | | | . | . . | |   kind(e) = locl(a)
. . | | | . | . . | 18. e1 : E
. . | | | . | . . | 19. loc(e1) = i
. . | | | . | . . | 20. kind(e1) = locl(a)
. . | | | . | . . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . | 22. j : E
. . | | | . | . . | 23. ∀k:E.
. . | | | . | . . | |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . | 24. ¬first(j)
. . | | | . | . . | 25. (e1 <loc pred(j))
. . | | | . | . . | 26. "done" ∈ Id
. . | | | . | . . | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | . . | 28. ("done" when pred(j))
. . | | | . | . . | ⊢  ("done" after pred(j)) ∈ 𝔹  by DoSubsume
. . | | | . | . . | |
. . | | | . | . . | .....subtype_rel .....
. . | | | . | . . | 1. a : Id
. . | | | . | . . | 2. i : Id
. . | | | . | . . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . . | 4. D' : Dsys
. . | | | . | . . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . . | 6. w : World
. . | | | . | . . | 7. p : FairFifo
. . | | | . | . . | 8. PossibleWorld(D';w)
. . | | | . | . . | 9. @i: ma-single-pre-init1(...;𝔹;...;a;...;x,v.¬x)
. . | | | . | . . |    ∈ Dsys
. . | | | . | . . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . | 11. ∀e:E.
. . | | | . | . . |     loc(e) = i
. . | | | . | . . |     ⇒
. . | | | . | . . |     first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . | 12. ∀e:E.
```

```
. . | | | . | . . |     loc(e) = i
. . | | | . | . . |     ⇒
. . | | | . | . . |     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . | 13. ∀e:E.
. . | | | . | . . |     loc(e) = i
. . | | | . | . . |     ⇒
. . | | | . | . . |     kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . | 14. e : E
. . | | | . | . . | 15. loc(e) = i
. . | | | . | . . | 16. kind(e) = locl(a)
. . | | | . | . . |     ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | . . |          ⇒
. . | | | . | . . |     kind(e) = locl(a)
. . | | | . | . . | 18. e1 : E
. . | | | . | . . | 19. loc(e1) = i
. . | | | . | . . | 20. kind(e1) = locl(a)
. . | | | . | . . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . | 22. j : E
. . | | | . | . . | 23. ∀k:E.
. . | | | . | . . |     (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . | 24. ¬first(j)
. . | | | . | . . | 25. (e1 <loc pred(j))
. . | | | . | . . | 26. "done" ∈ Id
. . | | | . | . . | 27. ("done" when j) = ("done" after pred(j))
. . | | | . | . . | 28. ("done" when pred(j))
. . | | | . | . . | ⊢  vartype(loc(pred(j));"done") ⊆r 𝔹

. . | | | . | . . |   by Unfold ‘es-locl‘ -4
. . | | | . | . . |      THEN
. . | | | . | . . |      Subst (loc(pred(j)) = i) 0
. . | | | . | . . |
. . | | | . | . .\
. . | | | . | . . 1. a : Id
. . | | | . | . . 2. i : Id
. . | | | . | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . . 4. D' : Dsys
. . | | | . | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . . 6. w : World
. . | | | . | . . 7. p : FairFifo
. . | | | . | . . 8. PossibleWorld(D';w)
. . | | | . | . . 9. @i: ma-single-pre-init1(...;𝔹;...;a;...;x,v.¬x)
. . | | | . | . . |  ∈ Dsys
. . | | | . | . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . 11. ∀e:E.
. . | | | . | . . |   loc(e) = i
. . | | | . | . . |   ⇒
. . | | | . | . . |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . 12. ∀e:E.
. . | | | . | . . |   loc(e) = i
. . | | | . | . . |   ⇒
. . | | | . | . . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . 13. ∀e:E.
. . | | | . | . . |   loc(e) = i
. . | | | . | . . |   ⇒
. . | | | . | . . |   kind(e) = locl(a) ⇒ ¬("done" when e)
```

```
. . | | | . | . . . 14. e : E
. . | | | . | . . . 15. loc(e) = i
. . | | | . | . . . 16. kind(e) = locl(a)
. . | | | . | . . . |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . | . . . |        ⇒
. . | | | . | . . . |   kind(e) = locl(a)
. . | | | . | . . . 18. e1 : E
. . | | | . | . . . 19. loc(e1) = i
. . | | | . | . . . 20. kind(e1) = locl(a)
. . | | | . | . . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . . 22. j : E
. . | | | . | . . . 23. ∀k:E.
. . | | | . | . . . |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . . 24. ¬first(j)
. . | | | . | . . . 25. (e1 <loc pred(j))
. . | | | . | . . . 26. "done" ∈ Id
. . | | | . | . . . 27. ("done" when j) = ("done" after pred(j))
. . | | | . | . . . 28. ("done" when pred(j))
. . | | | . | . . . 29. ¬("done" after pred(j))
. . | | | . | . . . ⊢  False
. . | | | . | . . . | by Unfold ‘es-locl‘ -5
. . | | | . | . . . |    THEN
. . | | | . | . . . |    AllHyps
. . | | | . | . . . |    (λh.
. . | | | . | . . . |     Unfold ‘alle-at‘ h
. . | | | . | . . . |     THEN
. . | | | . | . . . |     InstHyp [pred(j)] h)
. . | | | . | . . . |
. . | | | . | . . . |\
. . | | | . | . . . | 1. a : Id
. . | | | . | . . . | 2. i : Id
. . | | | . | . . . | 3. (λloc.⨁(once(loc;a;i))) ∈ Dsys
. . | | | . | . . . | 4. D' : Dsys
. . | | | . | . . . | 5. λloc.⨁(once(loc;a;i)) ⊆ D'
. . | | | . | . . . | 6. w : World
. . | | | . | . . . | 7. p : FairFifo
. . | | | . | . . . | 8. PossibleWorld(D';w)
. . | | | . | . . . | 9. @i: ...
. . | | | . | . . . | |  ∈ Dsys
. . | | | . | . . . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . . | 11. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . . | 12. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . . | 13. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . . | 14. e : E
. . | | | . | . . . | 15. loc(e) = i
```

```
. . | | | . | . . . | 16. kind(e) = locl(a)
. . | | | . | . . . | |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . . | 17. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   ¬("done" after e) = ("done" when e)
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   kind(e) = locl(a)
. . | | | . | . . . | 18. e1 : E
. . | | | . | . . . | 19. loc(e1) = i
. . | | | . | . . . | 20. kind(e1) = locl(a)
. . | | | . | . . . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . . | 22. j : E
. . | | | . | . . . | 23. ∀k:E.
. . | | | . | . . . | |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . . | 24. ¬first(j)
. . | | | . | . . . | 25. loc(e1) = loc(pred(j))

. . | | | . | . . . | 26. (e1 < pred(j))
. . | | | . | . . . | 27. "done" ∈ Id
. . | | | . | . . . | 28. ("done" when j) = ("done" after pred(j))
. . | | | . | . . . | 29. ("done" when pred(j))
. . | | | . | . . . | 30. ¬("done" after pred(j))
. . | | | . | . . . | ⊢  ¬("done" after pred(j))
. . | | | . | . . . | |   =
. . | | | . | . . . | |   ("done" when pred(j))  by Id
. . | | | . | . . . | |
. . | | | . | . . . | 1. a : Id
. . | | | . | . . . | 2. i : Id
. . | | | . | . . . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . . . | 4. D' : Dsys
. . | | | . | . . . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . . . | 6. w : World
. . | | | . | . . . | 7. p : FairFifo
. . | | | . | . . . | 8. PossibleWorld(D';w)
. . | | | . | . . . | 9. @i: ...
. . | | | . | . . . | |  ∈ Dsys
. . | | | . | . . . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . . | 11. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . . | 12. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . . | 13. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . . | 14. e : E
. . | | | . | . . . | 15. loc(e) = i
. . | | | . | . . . | 16. kind(e) = locl(a)
. . | | | . | . . . | |   ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . . | 17. ∀e:E.
. . | | | . | . . . | |   loc(e) = i
```

```
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   ¬("done" after e) = ("done" when e)
. . | | | . | . . . | |   ⇒
. . | | | . | . . . | |   kind(e) = locl(a)
. . | | | . | . . . | 18. e1 : E
. . | | | . | . . . | 19. loc(e1) = i
. . | | | . | . . . | 20. kind(e1) = locl(a)
. . | | | . | . . . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . . | 22. j : E
. . | | | . | . . . | 23. ∀k:E.
. . | | | . | . . . | |   (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . . | 24. ¬first(j)
. . | | | . | . . . | 25. loc(e1) = loc(pred(j))
. . | | | . | . . . | 26. (e1 < pred(j))
. . | | | . | . . . | 27. "done" ∈ Id
. . | | | . | . . . | 28. ("done" when j) = ("done" after pred(j))
. . | | | . | . . . | 29. ("done" when pred(j))
. . | | | . | . . . | 30. ¬("done" after pred(j))
. . | | | . | . . . | ⊢  ¬("done" after pred(j))
. . | | | . | . . . | |   =
. . | | | . | . . . | |   ("done" when pred(j))
. . | | | . | . . . | | by MoveToConcl -1 THEN MoveToConcl -1
. . | | | . | . . . | |    THEN
. . | | | . | . . . | |    AssertBY (("done" after pred(j)) ∈ 𝔹)
. . | | | . | . . . | |    (DoSubsume THEN Subst (loc(pred(j)) = i) 0)
. . | | | . | . . . | |    THEN
. . | | | . | . . . | |    AssertBY (("done" when pred(j)) ∈ 𝔹)
. . | | | . | . . . | |    (DoSubsume THEN Subst (loc(pred(j)) = i) 0)
. . | | | . | . . . | |    THEN
. . | | | . | . . . | |    AutoBoolCase ("done" after pred(j))
. . | | | . | . . . | |    THEN
. . | | | . | . . . | |    AutoBoolCase ("done" when pred(j))
. . | | | . | . . . | |
. . | | | . | . . . | 1. a : Id
. . | | | . | . . . | 2. i : Id
. . | | | . | . . . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . | . . . | 4. D' : Dsys
. . | | | . | . . . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . | . . . | 6. w : World
. . | | | . | . . . | 7. p : FairFifo
. . | | | . | . . . | 8. PossibleWorld(D';w)
. . | | | . | . . . | 9. @i: ...
. . | | | . | . . . |    ∈ Dsys
. . | | | . | . . . | 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . . | 11. ∀e:E.
. . | | | . | . . . |     loc(e) = i
. . | | | . | . . . |     ⇒
. . | | | . | . . . |     first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . . | 12. ∀e:E.
. . | | | . | . . . |     loc(e) = i
. . | | | . | . . . |     ⇒
. . | | | . | . . . |     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . . | 13. ∀e:E.
. . | | | . | . . . |     loc(e) = i
. . | | | . | . . . |     ⇒
```

```
. . | | | . | . . . |     kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . . | 14. e : E
. . | | | . | . . . | 15. loc(e) = i
. . | | | . | . . . | 16. kind(e) = locl(a)
. . | | | . | . . . |     ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . . | 17. ∀e:E.
. . | | | . | . . . |     loc(e) = i
. . | | | . | . . . |     ⇒
. . | | | . | . . . |     ¬("done" after e) = ("done" when e)
. . | | | . | . . . |     ⇒
. . | | | . | . . . |     kind(e) = locl(a)
. . | | | . | . . . | 18. e1 : E
. . | | | . | . . . | 19. loc(e1) = i
. . | | | . | . . . | 20. kind(e1) = locl(a)
. . | | | . | . . . | 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . . | 22. j : E
. . | | | . | . . . | 23. ∀k:E.
. . | | | . | . . . |     (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . . | 24. ¬first(j)
. . | | | . | . . . | 25. loc(e1) = loc(pred(j))
. . | | | . | . . . | 26. (e1 < pred(j))
. . | | | . | . . . | 27. "done" ∈ Id
. . | | | . | . . . | 28. ("done" when j) = ("done" after pred(j))
. . | | | . | . . . | 29. ("done" after pred(j)) ∈ 𝔹
. . | | | . | . . . | 30. ("done" when pred(j)) ∈ 𝔹
. . | | | . | . . . | 31. ¬("done" after pred(j))
. . | | | . | . . . | 32. ("done" when pred(j))
. . | | | . | . . . | ⊢  True ⇒ ¬False ⇒ ¬false₂ = true₂
. . | | | . | . . . |   by Obvious
. . | | | . | . . . |
. . | | | . | . . . .\
. . | | | . | . . . . 1. a : Id
. . | | | . | . . . . 2. i : Id
. . | | | . | . . . . 3. (λloc.⨁(once(loc;a;i))) ∈ Dsys
. . | | | . | . . . . 4. D' : Dsys
. . | | | . | . . . . 5. λloc.⨁(once(loc;a;i)) ⊆ D'
. . | | | . | . . . . 6. w : World
. . | | | . | . . . . 7. p : FairFifo
. . | | | . | . . . . 8. PossibleWorld(D';w)
. . | | | . | . . . . 9. @i: ...
. . | | | . | . . . .    ∈ Dsys
. . | | | . | . . . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . | . . . . 11. ∀e:E.
. . | | | . | . . . .     loc(e) = i
. . | | | . | . . . .     ⇒
. . | | | . | . . . .     first(e) ⇒ ("done" when e) = false₂
. . | | | . | . . . . 12. ∀e:E.
. . | | | . | . . . .     loc(e) = i
. . | | | . | . . . .     ⇒
. . | | | . | . . . .     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . | . . . . 13. ∀e:E.
. . | | | . | . . . .     loc(e) = i
. . | | | . | . . . .     ⇒
. . | | | . | . . . .     kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . | . . . . 14. e : E
```

```
. . | | | . | . . . . 15. loc(e) = i
. . | | | . | . . . . 16. kind(e) = locl(a)
. . | | | . | . . . .     ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . | . . . . 17. ∀e:E.
. . | | | . | . . . .     loc(e) = i
. . | | | . | . . . .     ⇒
. . | | | . | . . . .     ¬("done" after e) = ("done" when e)
. . | | | . | . . . .     ⇒
. . | | | . | . . . .     kind(e) = locl(a)
. . | | | . | . . . . 18. e1 : E
. . | | | . | . . . . 19. loc(e1) = i
. . | | | . | . . . . 20. kind(e1) = locl(a)
. . | | | . | . . . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . | . . . . 22. j : E
. . | | | . | . . . . 23. ∀k:E.
. . | | | . | . . . .     (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . | . . . . 24. ¬first(j)
. . | | | . | . . . . 25. loc(e1) = loc(pred(j))
. . | | | . | . . . . 26. (e1 < pred(j))
. . | | | . | . . . . 27. "done" ∈ Id
. . | | | . | . . . . 28. ("done" when j) = ("done" after pred(j))
. . | | | . | . . . . 29. ("done" when pred(j))
. . | | | . | . . . . 30. ¬("done" after pred(j))
. . | | | . | . . . . 31. kind(pred(j)) = locl(a)
. . | | | . | . . . . ⊢  False
. . | | | . | . . . .   by AllHyps
. . | | | . | . . . .      (λh.
. . | | | . | . . . .       InstHyp [pred(j)] h THEN Thin -1
. . | | | . | . . . .       THEN
. . | | | . | . . . .       Analyze -1
. . | | | . | . . . .       THEN
. . | | | . | . . . .       Complete Auto)
. . | | | . |
. . | | | . .\
. . | | | . . .....subterm .....
. . | | | . . 1. a : Id
. . | | | . . 2. i : Id
. . | | | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . . 4. D' : Dsys
. . | | | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . . 6. w : World
. . | | | . . 7. p : FairFifo
. . | | | . . 8. PossibleWorld(D';w)
. . | | | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;...;x,v.¬x)
. . | | | . .  |  ∈ Dsys
. . | | | . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . . 11. ∀e:E.
. . | | | . .  |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . . 12. ∀e:E.
. . | | | . .  |   loc(e) = i
. . | | | . .  |   ⇒
. . | | | . .  |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . . 13. ∀e:E.
. . | | | . .  |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . . 14. e : E
```

```
. . | | | . . 15. loc(e) = i
. . | | | . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . . |        ⇒
. . | | | . . |   kind(e) = locl(a)
. . | | | . . 18. e1 : E
. . | | | . . 19. loc(e1) = i
. . | | | . . 20. kind(e1) = locl(a)
. . | | | . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . . 22. j : E
. . | | | . . 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . . 24. (e1 <loc j)
. . | | | . . 25. "done" ∈ Id
. . | | | . . 26. ("done" when j) = ("done" after pred(j))
. . | | | . . 27. z : vartype(loc(j);"done")
. . | | | . . ⊢  z ∈ 𝔹  by DoSubsume
. . | | | . . |
. . | | | . . .....subtype_rel .....
. . | | | . . 1. a : Id
. . | | | . . 2. i : Id
. . | | | . . 3. (λloc.⨁(once(loc;a;i))) ∈ Dsys
. . | | | . . 4. D' : Dsys
. . | | | . . 5. λloc.⨁(once(loc;a;i)) ⊆ D'
. . | | | . . 6. w : World
. . | | | . . 7. p : FairFifo
. . | | | . . 8. PossibleWorld(D';w)
. . | | | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;...;x,v.¬x)
. . | | | . . |  ∈ Dsys
. . | | | . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . . 11. ∀e:E.
. . | | | . . |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . . 12. ∀e:E.
. . | | | . . |   loc(e) = i
. . | | | . . |   ⇒
. . | | | . . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . . 13. ∀e:E.
. . | | | . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . . 14. e : E
. . | | | . . 15. loc(e) = i
. . | | | . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . . |        ⇒
. . | | | . . |   kind(e) = locl(a)
. . | | | . . 18. e1 : E
. . | | | . . 19. loc(e1) = i
. . | | | . . 20. kind(e1) = locl(a)
. . | | | . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . . 22. j : E
. . | | | . . 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . . 24. (e1 <loc j)
. . | | | . . 25. "done" ∈ Id
. . | | | . . 26. ("done" when j) = ("done" after pred(j))
. . | | | . . 27. vartype(loc(j);"done")
. . | | | . . ⊢  vartype(loc(j);"done") ⊆r 𝔹  by Subst (loc(j) = i) 0
. . | | | . . |
```

```
. . | | | . . .....equality .....
. . | | | . . 1. a : Id
. . | | | . . 2. i : Id
. . | | | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . . 4. D' : Dsys
. . | | | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . . 6. w : World
. . | | | . . 7. p : FairFifo
. . | | | . . 8. PossibleWorld(D';w)
. . | | | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;...;x,v.¬x)
. . | | | . . |  ∈ Dsys
. . | | | . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . . 11. ∀e:E.
. . | | | . . |   loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . . 12. ∀e:E.
. . | | | . . |   loc(e) = i
. . | | | . . |   ⇒
. . | | | . . |   kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . . 13. ∀e:E.
. . | | | . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . . 14. e : E
. . | | | . . 15. loc(e) = i
. . | | | . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . . |        ⇒
. . | | | . . |   kind(e) = locl(a)
. . | | | . . 18. e1 : E
. . | | | . . 19. loc(e1) = i
. . | | | . . 20. kind(e1) = locl(a)
. . | | | . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . . 22. j : E
. . | | | . . 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . . 24. (e1 <loc j)
. . | | | . . 25. "done" ∈ Id
. . | | | . . 26. ("done" when j) = ("done" after pred(j))
. . | | | . . 27. vartype(loc(j);"done")
. . | | | . . ⊢  loc(j) = i  by Assert (loc(e1) = loc(j))
. . | | | . . |
. . | | | . . .....assertion .....
. . | | | . . 1. a : Id
. . | | | . . 2. i : Id
. . | | | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | | . . 4. D' : Dsys
. . | | | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | | . . 6. w : World
. . | | | . . 7. p : FairFifo
. . | | | . . 8. PossibleWorld(D';w)
. . | | | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;...;x,v.¬x)
. . | | | . .    ∈ Dsys
. . | | | . . 10. vartype(i;"done") ⊆r 𝔹
. . | | | . . 11. ∀e:E.
. . | | | . .     loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | | . . 12. ∀e:E.
. . | | | . .     loc(e) = i
. . | | | . .     ⇒
```

```
. . | | | . .     kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | | . . 13. ∀e:E.
. . | | | . .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | | . . 14. e : E
. . | | | . . 15. loc(e) = i
. . | | | . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | | . .          ⇒
. . | | | . .     kind(e) = locl(a)
. . | | | . . 18. e1 : E
. . | | | . . 19. loc(e1) = i
. . | | | . . 20. kind(e1) = locl(a)
. . | | | . . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | | . . 22. j : E
. . | | | . . 23. ∀k:E. (k <loc j) ⇒ (e1 <loc k) ⇒ ("done" when k)
. . | | | . . 24. (e1 <loc j)
. . | | | . . 25. "done" ∈ Id
. . | | | . . 26. ("done" when j) = ("done" after pred(j))
. . | | | . . 27. vartype(loc(j);"done")
. . | | | . . ⊢  loc(e1) = loc(j)  by Unfold ‘es-locl‘ -4
. . | | |
. . | | .\
. . | | . .....subterm .....
. . | | . 1. a : Id
. . | | . 2. i : Id
. . | | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | | . 4. D' : Dsys
. . | | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | | . 6. w : World
. . | | . 7. p : FairFifo
. . | | . 8. PossibleWorld(D';w)
. . | | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | | .    ∈ Dsys
. . | | . 10. vartype(i;"done") ⊆r 𝔹
. . | | . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | | . 12. ∀e:E.
. . | | .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | | . 13. ∀e:E.
. . | | .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | | . 14. e : E
. . | | . 15. loc(e) = i
. . | | . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | | .          ⇒
. . | | .     kind(e) = locl(a)
. . | | . 18. e1 : E
. . | | . 19. loc(e1) = i
. . | | . 20. kind(e1) = locl(a)
. . | | . 21. WellFnd{i}(E;x,y.(x <loc y))
. . | | . 22. j : E
. . | | . 23. k : E
. . | | . 24. (k <loc j)
. . | | . 25. (e1 <loc k)
. . | | . ⊢  ("done" when k) ∈ 𝔹
. . | | .   by DoSubsume THEN TrySubsume THEN Subst (loc(k) = i) 0
```

```
. . | | .      THEN
. . | | .      TrySubsume
. . | | .      THEN
. . | | .      Unfold ‘es-locl‘ -1
. . | |
. . | .\
. . | . 1. a : Id
. . | . 2. i : Id
. . | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | . 4. D' : Dsys
. . | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | . 6. w : World
. . | . 7. p : FairFifo
. . | . 8. PossibleWorld(D';w)
. . | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | . |  ∈ Dsys
. . | . 10. vartype(i;"done") ⊆r 𝔹
. . | . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | . 12. ∀e:E.
. . | . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | . 14. e : E
. . | . 15. loc(e) = i
. . | . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | . |        ⇒
. . | . |   kind(e) = locl(a)
. . | . 18. ∀e:E.
. . | . |   loc(e) = i
. . | . |   ⇒
. . | . |   kind(e) = locl(a)
. . | . |   ⇒
. . | . |   (∀e':E. (e <loc e') ⇒ ("done" when e'))
. . | . ⊢  ∀e':E.
. . | . |  loc(e') = i
. . | . |  ⇒
. . | . |  (∀e:E.
. . | . |   (e <loc e') ⇒ kind(e) = locl(a) ⇒ ¬kind(e') = locl(a))
. . | . | by Auto THEN TrySubsume
. . | . |
. . | . 1. a : Id
. . | . 2. i : Id
. . | . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | . 4. D' : Dsys
. . | . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | . 6. w : World
. . | . 7. p : FairFifo
. . | . 8. PossibleWorld(D';w)
. . | . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | . |  ∈ Dsys
. . | . 10. vartype(i;"done") ⊆r 𝔹
. . | . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | . 12. ∀e:E.
. . | . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
```

```
. . | . 14. e : E
. . | . 15. loc(e) = i
. . | . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | . |        ⇒
. . | . |   kind(e) = locl(a)
. . | . 18. ∀e:E.
. . | . |   loc(e) = i
. . | . |   ⇒
. . | . |   kind(e) = locl(a)
. . | . |   ⇒
. . | . |   (∀e':E. (e <loc e') ⇒ ("done" when e'))
. . | . 19. e' : E
. . | . 20. loc(e') = i
. . | . 21. e1 : E
. . | . 22. (e1 <loc e')
. . | . 23. kind(e1) = locl(a)
. . | . ⊢  ¬kind(e') = locl(a)  by AllHyps (λh.InstHyp [e1;e'] h)
. . | . |
. . | . |\
. . | . | .....antecedent .....
. . | . | 1. a : Id
. . | . | 2. i : Id
. . | . | 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | . | 4. D' : Dsys
. . | . | 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | . | 6. w : World
. . | . | 7. p : FairFifo
. . | . | 8. PossibleWorld(D';w)
. . | . | 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | . |    ∈ Dsys
. . | . | 10. vartype(i;"done") ⊆r 𝔹
. . | . | 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | . | 12. ∀e:E.
. . | . |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | . | 13. ∀e:E.
. . | . |     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | . | 14. e : E
. . | . | 15. loc(e) = i
. . | . | 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | . | 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | . |          ⇒
. . | . |     kind(e) = locl(a)
. . | . | 18. ∀e:E.
. . | . |     loc(e) = i
. . | . |     ⇒
. . | . |     kind(e) = locl(a)
. . | . |     ⇒
. . | . |     (∀e':E. (e <loc e') ⇒ ("done" when e'))
. . | . | 19. e' : E
. . | . | 20. loc(e') = i
. . | . | 21. e1 : E
. . | . | 22. (e1 <loc e')
. . | . | 23. kind(e1) = locl(a)
. . | . | ⊢  loc(e1) = i  by Unfold ‘es-locl‘ -2
```

```
. . | . |
. . | . .\
. . | . . 1. a : Id
. . | . . 2. i : Id
. . | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | . . 4. D' : Dsys
. . | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | . . 6. w : World
. . | . . 7. p : FairFifo
. . | . . 8. PossibleWorld(D';w)
. . | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | . . |  ∈ Dsys
. . | . . 10. vartype(i;"done") ⊆r 𝔹
. . | . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | . . 12. ∀e:E.
. . | . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | . . 13. ∀e:E.
. . | . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | . . 14. e : E
. . | . . 15. loc(e) = i
. . | . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | . . |        ⇒
. . | . . |   kind(e) = locl(a)
. . | . . 18. ∀e:E.
. . | . . |   loc(e) = i
. . | . . |   ⇒
. . | . . |   kind(e) = locl(a)
. . | . . |   ⇒
. . | . . |   (∀e':E. (e <loc e') ⇒ ("done" when e'))
. . | . . 19. e' : E
. . | . . 20. loc(e') = i
. . | . . 21. e1 : E
. . | . . 22. (e1 <loc e')
. . | . . 23. kind(e1) = locl(a)
. . | . . 24. ("done" when e')
. . | . . ⊢  ¬kind(e') = locl(a)  by Analyze 0
. . | . . |
. . | . . 1. a : Id
. . | . . 2. i : Id
. . | . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . | . . 4. D' : Dsys
. . | . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . | . . 6. w : World
. . | . . 7. p : FairFifo
. . | . . 8. PossibleWorld(D';w)
. . | . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x)
. . | . .    ∈ Dsys
. . | . . 10. vartype(i;"done") ⊆r 𝔹
. . | . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . | . . 12. ∀e:E.
. . | . .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . | . . 13. ∀e:E.
. . | . .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . | . . 14. e : E
```

```
. . | . . 15. loc(e) = i
. . | . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . | . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . | . .          ⇒
. . | . .     kind(e) = locl(a)
. . | . . 18. ∀e:E.
. . | . .     loc(e) = i
. . | . .     ⇒
. . | . .     kind(e) = locl(a)
. . | . .     ⇒
. . | . .     (∀e':E. (e <loc e') ⇒ ("done" when e'))
. . | . . 19. e' : E
. . | . . 20. loc(e') = i
. . | . . 21. e1 : E
. . | . . 22. (e1 <loc e')
. . | . . 23. kind(e1) = locl(a)
. . | . . 24. ("done" when e')
. . | . . 25. kind(e') = locl(a)
. . | . . ⊢  False  by AllHyps (λh.InstHyp [e'] h THEN Complete Auto)
. . |
. . .\
. . . 1. a : Id
. . . 2. i : Id
. . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . . 4. D' : Dsys
. . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . . 6. w : World
. . . 7. p : FairFifo
. . . 8. PossibleWorld(D';w)
. . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. . . 10. vartype(i;"done") ⊆r 𝔹
. . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . . 12. ∀e:E.
. . . |   loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . . 14. e : E
. . . 15. loc(e) = i
. . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . . |        ⇒
. . . |   kind(e) = locl(a)
. . . 18. ∀e':E.
. . . |   loc(e') = i
. . . |   ⇒
. . . |   (∀e:E.
. . . |    (e <loc e') ⇒ kind(e) = locl(a) ⇒ ¬kind(e') = locl(a))
. . . ⊢  ∀e:E.
. . . |  loc(e) = i
. . . |  ⇒
. . . |  (∀e':E.
. . . |   loc(e') = i
. . . |   ⇒
. . . |   kind(e) = locl(a) ⇒ kind(e') = locl(a) ⇒ e = e')
. . . | by Auto THEN TrySubsume
. . . |
```

```
. . . 1. a : Id
. . . 2. i : Id
. . . 3. (λloc.⊕(once(loc;a;i))) ∈ Dsys
. . . 4. D' : Dsys
. . . 5. λloc.⊕(once(loc;a;i)) ⊆ D'
. . . 6. w : World
. . . 7. p : FairFifo
. . . 8. PossibleWorld(D';w)
. . . 9. @i: ma-single-pre-init1("done";𝔹;false₂;a;Unit;x,v.¬x) ∈ Dsys
. . . 10. vartype(i;"done") ⊆r 𝔹
. . . 11. ∀e:E. loc(e) = i ⇒ first(e) ⇒ ("done" when e) = false₂
. . . 12. ∀e:E.

. . .     loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r Unit)
. . . 13. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ ¬("done" when e)
. . . 14. e : E
. . . 15. loc(e) = i
. . . 16. kind(e) = locl(a) ∨ (∀v:Unit. ¬¬("done" after e))
. . . 17. ∀e@i.¬("done" after e) = ("done" when e)
. . .           ⇒
. . .     kind(e) = locl(a)
. . . 18. ∀e':E.
. . .     loc(e') = i
. . .     ⇒
. . .     (∀e:E.
. . .      (e <loc e') ⇒ kind(e) = locl(a) ⇒ ¬kind(e') = locl(a))
. . . 19. e1 : E
. . . 20. loc(e1) = i
. . . 21. e' : E
. . . 22. loc(e') = i
. . . 23. kind(e1) = locl(a)
. . . 24. kind(e') = locl(a)
. . . ⊢  e1 = e'
. . .   by InstESAxiom ES(w) [e1;e'] 3 THEN Analyze -1 THEN Thin -1
. . .      THEN
. . .      Analyze -1
. . .      THEN
. . .      SplitOrHyps
. . .      THEN
. . .      AllHyps (λh.InstHyp [e1;e'] h THEN Complete Auto)
. . .      THEN
. . .      AllHyps (λh.InstHyp [e';e1] h THEN Complete Auto)
END
```