```
s-pre-rule1  THM

⊢  ∀i,a,x:Id, A,T:Type, P:(A→T→Prop).
|  @i: ma-single-pre1(x;A;a;T;x,v.P(x,v)) ∈ Dsys
|  & (∀D:Dsys.
|     @i: ma-single-pre1(x;A;a;T;x,v.P(x,v)) ⊆ D
|     ⇒
|     D
|     realizes es.(vartype(i;x) ⊆r A)
|                 & (∀e:E.
|                    loc(e) = i
|                    ⇒
|                    kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
|                 & (∀e:E.
|                    loc(e) = i
|                    ⇒
|                    (kind(e) = locl(a) ⇒ P((x when e),val(e)))
|                    & (∃e':E.
|                       (e <loc e') ∨ e = e'
|                       & kind(e') = locl(a)
|                         ∨ (∀v:T. ¬P((x after e'),v)))))
| by Unfold `ma-single-pre1` 0 THEN UnivCD
|    THEN
|    Inst Thm: s_DASH_pre_DASH_rule [i;a;T;x : A;λs,v. P(s(x),v)]
|    THENA
|    Try (Complete Auto)
|    THEN
|    Try (Unfold `s-decl` 0)
|
|\
| .....wf .....
| 1. Id
| 2. Id
| 3. x : Id
| 4. A : Type
| 5. T : Type
| 6. P : A→T→Prop
| ⊢  (λs,v. P(s(x),v)) ∈ State(x : A)→T→Prop
|   by Repeat
|      (Unfolds [`ma-state`;`fpf-single`;`fpf-cap`;`fpf-ap`;`fpf-dom`]
|       0
|       THEN
|       Reduce 0)
|      THEN
|      DoSubsume
|      THEN
|      RWO
|      Thm* ∀d:EqDecider(A), i:A. (eqof(d)(i,i)) ~ true₂
|      {eqof_eq_btrue}
|      0
|      THEN
|      Reduce 0
|      THEN
|      Analyze 0
|
```

```
.\
. 1. i : Id
. 2. a : Id
. 3. x : Id
. 4. A : Type
. 5. T : Type
. 6. P : A→T→Prop
. 7. @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λs,v. P(s(x),v) s v)
. |  ∈ Dsys
. |  & (∀D:Dsys.
. |     @i: (with ds: x : A
. |      action a:T
. |      precondition a(v) is
. |      λs,v. P(s(x),v) s v) ⊆ D
. |     ⇒
. |     D
. |     realizes es.(∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |                 & (∀e:E.
. |                    loc(e) = i
. |                    ⇒
. |                    kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |                 & (∀e:E.
. |                    loc(e) = i
. |                    ⇒
. |                    (kind(e) = locl(a)
. |                     ⇒
. |                     (λs,v. P(s(x),v))((λz.(z when e)),val(e)))
. |                    & (∃e':E.
. |                       (e <loc e') ∨ e = e'
. |                       & kind(e') = locl(a)
. |                         ∨ (∀v:T.
. |                            ¬(λs,v. P(s(x),v))
. |                             ((λz.(z after e'))
. |                             ,v)))))
. ⊢  @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λ2s,v. P(s(x),v) s v)
. |  ∈ Dsys
. |  & (∀D:Dsys.
. |     @i: (with ds: x : A
. |      action a:T
. |      precondition a(v) is
. |      λ2s,v. P(s(x),v) s v) ⊆ D
. |     ⇒
. |     D
. |     realizes es.(vartype(i;x) ⊆r A)
. |                 & (∀e:E.
. |                    loc(e) = i
. |                    ⇒
. |                    kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |                 & (∀e:E.
```

```
. |                     loc(e) = i
. |                     ⇒
. |                     (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |                     & (∃e':E.
. |                        (e <loc e') ∨ e = e'
. |                        & kind(e') = locl(a)
. |                          ∨ (∀v:T. ¬P((x after e'),v)))))
. | by Analyze -1 THEN Analyze 0 THEN Try Trivial THEN All Reduce
. |    THEN
. |    ParallelOp -2
. |    THEN
. |    RepeatFor 7 (ParallelOp -1)
. |    THEN
. |    ExRepD
. |    THEN
. |    Analyze 0
. |    THEN
. |    Try Trivial
. |
. 1. i : Id
. 2. a : Id
. 3. x : Id
. 4. A : Type
. 5. T : Type
. 6. P : A→T→Prop
. 7. @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λs,v. P(s(x),v) s v)
. |   ∈ Dsys
. 8. ∀D:Dsys.
. |  @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λs,v. P(s(x),v) s v) ⊆ D
. |  ⇒
. |  D
. |  realizes es.(∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |               & (∀e:E.
. |                  loc(e) = i
. |                  ⇒
. |                  kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |               & (∀e:E.
. |                  loc(e) = i
. |                  ⇒
. |                  (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |                  & (∃e':E.
. |                     (e <loc e') ∨ e = e'
. |                     & kind(e') = locl(a)
. |                       ∨ (∀v:T. ¬P((x after e'),v))))
. 9. @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λ2s,v. P(s(x),v) s v)
. |  ∈ Dsys
```

```
. 10. D : Dsys
. 11. @i: (with ds: x : A
. |    action a:T
. |    precondition a(v) is
. |    λ2s,v. P(s(x),v) s v) ⊆ D
. 12. ∀D':Dsys.
. |   D ⊆ D'
. |   ⇒
. |   (∀w:World, p:FairFifo.
. |    PossibleWorld(D';w)
. |    ⇒
. |    (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |    & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |    & (∀e:E.
. |       loc(e) = i
. |       ⇒
. |       (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |       & (∃e':E.
. |          (e <loc e') ∨ e = e'
. |          & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v)))))
. 13. D' : Dsys
. 14. D ⊆ D'
. 15. ∀w:World, p:FairFifo.
. |   PossibleWorld(D';w)
. |   ⇒
. |   (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |   & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |   & (∀e:E.
. |      loc(e) = i
. |      ⇒
. |      (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |      & (∃e':E.
. |         (e <loc e') ∨ e = e'
. |         & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v))))
. 16. w : World
. 17. ∀p:FairFifo.
. |   PossibleWorld(D';w)
. |   ⇒
. |   (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |   & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |   & (∀e:E.
. |      loc(e) = i
. |      ⇒
. |      (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |      & (∃e':E.
. |         (e <loc e') ∨ e = e'
. |         & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v))))
. 18. p : FairFifo
. 19. PossibleWorld(D';w)
. 20. ∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top
. 21. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T)
. 22. ∀e:E.
. |   loc(e) = i
. |   ⇒
. |   (kind(e) = locl(a) ⇒ P((x when e),val(e)))
```

```
. |   & (∃e':E.
. |      (e <loc e') ∨ e = e'
. |      & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v)))
. ⊢  (vartype(i;x) ⊆r A)
. |  & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. | by Analyze 0 THEN Try Trivial
. |
. 1. i : Id
. 2. a : Id
. 3. x : Id
. 4. A : Type
. 5. T : Type
. 6. P : A→T→Prop
. 7. @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λs,v. P(s(x),v) s v)
. |  ∈ Dsys
. 8. ∀D:Dsys.
. |  @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λs,v. P(s(x),v) s v) ⊆ D
. |  ⇒
. |  D
. |  realizes es.(∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |              & (∀e:E.
. |                 loc(e) = i
. |                 ⇒
. |                 kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |              & (∀e:E.
. |                 loc(e) = i
. |                 ⇒
. |                 (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |                 & (∃e':E.
. |                    (e <loc e') ∨ e = e'
. |                    & kind(e') = locl(a)
. |                      ∨ (∀v:T. ¬P((x after e'),v))))
. 9. @i: (with ds: x : A
. |   action a:T
. |   precondition a(v) is
. |   λ2s,v. P(s(x),v) s v)
. |  ∈ Dsys
. 10. D : Dsys
. 11. @i: (with ds: x : A
. |    action a:T
. |    precondition a(v) is
. |    λ2s,v. P(s(x),v) s v) ⊆ D
. 12. ∀D':Dsys.
. |   D ⊆ D'
. |   ⇒
. |   (∀w:World, p:FairFifo.
. |    PossibleWorld(D';w)
. |    ⇒
. |    (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
```

```
. |    & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |    & (∀e:E.
. |       loc(e) = i
. |       ⇒
. |       (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |       & (∃e':E.
. |          (e <loc e') ∨ e = e'
. |          & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v)))))
. 13. D' : Dsys
. 14. D ⊆ D'
. 15. ∀w:World, p:FairFifo.
. |   PossibleWorld(D';w)
. |   ⇒
. |   (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |   & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |   & (∀e:E.
. |      loc(e) = i
. |      ⇒
. |      (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |      & (∃e':E.
. |         (e <loc e') ∨ e = e'
. |         & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v))))
. 16. w : World
. 17. ∀p:FairFifo.
. |   PossibleWorld(D';w)
. |   ⇒
. |   (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
. |   & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
. |   & (∀e:E.
. |      loc(e) = i
. |      ⇒
. |      (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |      & (∃e':E.
. |         (e <loc e') ∨ e = e'
. |         & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v))))
. 18. p : FairFifo
. 19. PossibleWorld(D';w)
. 20. ∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top
. 21. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T)
. 22. ∀e:E.
. |   loc(e) = i
. |   ⇒
. |   (kind(e) = locl(a) ⇒ P((x when e),val(e)))
. |   & (∃e':E.
. |      (e <loc e') ∨ e = e'
. |      & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v)))
. ⊢  vartype(i;x) ⊆r A  by InstHyp [x] -3
. |
. 1. i : Id
. 2. a : Id
. 3. x : Id
. 4. A : Type
. 5. T : Type
. 6. P : A→T→Prop
. 7. @i: (with ds: x : A
```

```
.     action a:T
.     precondition a(v) is
.     λs,v. P(s(x),v) s v)
.    ∈ Dsys
. 8. ∀D:Dsys.
.    @i: (with ds: x : A
.     action a:T
.     precondition a(v) is
.     λs,v. P(s(x),v) s v) ⊆ D
.    ⇒
.    D
.    realizes es.(∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
.                & (∀e:E.
.                   loc(e) = i
.                   ⇒
.                   kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
.                & (∀e:E.
.                   loc(e) = i
.                   ⇒
.                   (kind(e) = locl(a) ⇒ P((x when e),val(e)))
.                   & (∃e':E.
.                      (e <loc e') ∨ e = e'
.                      & kind(e') = locl(a)
.                        ∨ (∀v:T. ¬P((x after e'),v))))
. 9. @i: (with ds: x : A
.     action a:T
.     precondition a(v) is
.     λ2s,v. P(s(x),v) s v)
.    ∈ Dsys
. 10. D : Dsys
. 11. @i: (with ds: x : A
.      action a:T
.      precondition a(v) is
.      λ2s,v. P(s(x),v) s v) ⊆ D
. 12. ∀D':Dsys.
.     D ⊆ D'
.     ⇒
.     (∀w:World, p:FairFifo.
.      PossibleWorld(D';w)
.      ⇒
.      (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
.      & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
.      & (∀e:E.
.         loc(e) = i
.         ⇒
.         (kind(e) = locl(a) ⇒ P((x when e),val(e)))
.         & (∃e':E.
.            (e <loc e') ∨ e = e'
.            & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v)))))
. 13. D' : Dsys
. 14. D ⊆ D'
. 15. ∀w:World, p:FairFifo.
.     PossibleWorld(D';w)
.     ⇒
.     (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
```

```
.     & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
.     & (∀e:E.
.        loc(e) = i
.        ⇒
.        (kind(e) = locl(a) ⇒ P((x when e),val(e)))
.        & (∃e':E.
.           (e <loc e') ∨ e = e'
.           & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v))))
. 16. w : World
. 17. ∀p:FairFifo.
.     PossibleWorld(D';w)
.     ⇒
.     (∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top)
.     & (∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T))
.     & (∀e:E.
.        loc(e) = i
.        ⇒

.        (kind(e) = locl(a) ⇒ P((x when e),val(e)))
.        & (∃e':E.
.           (e <loc e') ∨ e = e'
.           & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v))))
. 18. p : FairFifo
. 19. PossibleWorld(D';w)
. 20. ∀x@0:Id. vartype(i;x@0) ⊆r x : A(x@0)?Top
. 21. ∀e:E. loc(e) = i ⇒ kind(e) = locl(a) ⇒ (valtype(e) ⊆r T)
. 22. ∀e:E.
.     loc(e) = i
.     ⇒
.     (kind(e) = locl(a) ⇒ P((x when e),val(e)))
.     & (∃e':E.
.        (e <loc e') ∨ e = e'
.        & kind(e') = locl(a) ∨ (∀v:T. ¬P((x after e'),v)))
. 23. vartype(i;x) ⊆r x : A(x)?Top
. ⊢  vartype(i;x) ⊆r A
.   by Repeat
.      (Unfolds [`fpf-single`;`fpf-cap`;`fpf-ap`;`fpf-dom`] -1
.       THEN
.       Reduce -1)
.      THEN
.      RWO
.      Thm* ∀d:EqDecider(A), i:A. (eqof(d)(i,i)) ~ true2
.      {eqof_eq_btrue}
.      -1
.      THEN
.      Reduce -1
END
```