```
effect-rule1  THM

⊢  ∀i,x,y:Id, k:Knd, T,A,B:Type, f:(A→B→T→A).
|  ¬y = x
|  ⇒
|  @i: ma-single-effect1(x;A;y;B;k;T;f) ∈ Dsys
|  & (∀D:Dsys.
|     @i: ma-single-effect1(x;A;y;B;k;T;f) ⊆ D
|     ⇒
|     D
|     realizes es.(vartype(i;x) ⊆r A) & (vartype(i;y) ⊆r B)
|                 & (∀e:E.
|                    loc(e) = i ⇒ kind(e) = k ⇒ (valtype(e) ⊆r T))
|                 & (∀e:E.
|                    loc(e) = i
|                    ⇒
|                    kind(e) = k
|                    ⇒
|                    (x after e) = f((x when e),(y when e),val(e))))
| by UnivCD THEN Unfold ‘ma-single-effect1‘ 0
|    THEN
|    AssertBY (x : A ⊕ y : B ∈ x:Id fp-> Type) Auto
|    THEN
|    AssertBY (k : T ∈ k:Knd fp-> Type) Auto
|    THEN
|    Inst Thm: s_DASH_effect_DASH_rule
|    [i;x;k;x : A ⊕ y : B;k : T;λs,v. f(s(x),s(y),v)]
|
|\
| .....subterm .....
| 1. Id
| 2. x : Id
| 3. y : Id
| 4. k : Knd
| 5. T : Type
| 6. A : Type
| 7. B : Type
| 8. f : A→B→T→A
| 9. ¬y = x
| 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| 11. k : T ∈ k:Knd fp-> Type
| 12. s : State(x : A ⊕ y : B)
| 13. v : ma-valtype(k : T; k)
| ⊢  f(s(x),s(y),v) ∈ x : A ⊕ y : B(x)?Void
| | by Assert (x : A ⊕ y : B(x)?Void = A)
| |
| |\
| | .....assertion .....
| | 1. Id
| | 2. x : Id
| | 3. y : Id
| | 4. k : Knd
| | 5. T : Type
| | 6. A : Type
| | 7. B : Type
```

```
| | 8. A→B→T→A
| | 9. ¬y = x
| | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| | 11. k : T ∈ k:Knd fp-> Type
| | 12. State(x : A ⊕ y : B)
| | 13. ma-valtype(k : T; k)
| | ⊢  x : A ⊕ y : B(x)?Void = A
| | | by RWO
| | |    Thm* ∀eq:EqDecider(A), f,g:a:A fp-> Top, x:A, z:Top.
| | |         f ⊕ g(x)?z ∼ if x ∈ dom(f)→ f(x)?z else g(x)?z fi
| | |    {fpf_DASH_join_DASH_cap_DASH_sq}
| | |    0
| | |    THEN
| | |    SplitOnConclITE
| | |    THEN
| | |    Reduce 0
| | |
| | .....falsecase .....
| | 1. Id
| | 2. x : Id
| | 3. y : Id
| | 4. k : Knd
| | 5. T : Type
| | 6. A : Type
| | 7. B : Type
| | 8. A→B→T→A
| | 9. ¬y = x
| | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| | 11. k : T ∈ k:Knd fp-> Type
| | 12. State(x : A ⊕ y : B)
| | 13. ma-valtype(k : T; k)
| | 14. ¬x ∈ dom(x : A)
| | ⊢  y : B(x)?Void = A
| |   by Analyze -1
| |      THEN
| |      RWO
| |      Thm* ∀eq:EqDecider(A), x,y:A, v:Top. x ∈ dom(y : v) ⟺ x = y
| |      {fpf_DASH_single_DASH_dom}
| |      0
| |
| .\
| . 1. Id
| . 2. x : Id
| . 3. y : Id
| . 4. k : Knd
| . 5. T : Type
| . 6. A : Type
| . 7. B : Type
| . 8. f : A→B→T→A
| . 9. ¬y = x
| . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . 11. k : T ∈ k:Knd fp-> Type
| . 12. s : State(x : A ⊕ y : B)
| . 13. v : ma-valtype(k : T; k)
| . 14. x : A ⊕ y : B(x)?Void = A
```

```
| . ⊢  f(s(x),s(y),v) ∈ x : A ⊕ y : B(x)?Void
| . | by Assert (x : A ⊕ y : B(x)?Top = A)
| . |
| . |\
| . | .....assertion .....
| . | 1. Id
| . | 2. x : Id
| . | 3. y : Id
| . | 4. k : Knd
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. A→B→T→A
| . | 9. ¬y = x
| . | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . | 11. k : T ∈ k:Knd fp-> Type
| . | 12. State(x : A ⊕ y : B)
| . | 13. ma-valtype(k : T; k)
| . | 14. x : A ⊕ y : B(x)?Void = A
| . | ⊢  x : A ⊕ y : B(x)?Top = A
| . | | by RWO
| . | |    Thm* ∀eq:EqDecider(A), f,g:a:A fp-> Top, x:A, z:Top.
| . | |         f ⊕ g(x)?z ∼ if x ∈ dom(f)→ f(x)?z else g(x)?z fi
| . | |    {fpf_DASH_join_DASH_cap_DASH_sq}
| . | |    0
| . | |    THEN
| . | |    SplitOnConclITE
| . | |    THEN
| . | |    Reduce 0
| . | |
| . | .....falsecase .....
| . | 1. Id
| . | 2. x : Id
| . | 3. y : Id
| . | 4. k : Knd
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. A→B→T→A
| . | 9. ¬y = x
| . | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . | 11. k : T ∈ k:Knd fp-> Type
| . | 12. State(x : A ⊕ y : B)
| . | 13. ma-valtype(k : T; k)
| . | 14. x : A ⊕ y : B(x)?Void = A
| . | 15. ¬x ∈ dom(x : A)
| . | ⊢  y : B(x)?Top = A
| . |   by Analyze -1
| . |      THEN
| . |      RWO
| . |      Thm* ∀eq:EqDecider(A), x,y:A, v:Top.
| . |           x ∈ dom(y : v) ⟺ x = y
| . |      {fpf_DASH_single_DASH_dom}
| . |      0
| . |
```

```
| . .\
| . . 1. Id
| . . 2. x : Id
| . . 3. y : Id
| . . 4. k : Knd
| . . 5. T : Type
| . . 6. A : Type
| . . 7. B : Type
| . . 8. f : A→B→T→A
| . . 9. ¬y = x
| . . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . 11. k : T ∈ k:Knd fp-> Type
| . . 12. s : State(x : A ⊕ y : B)
| . . 13. v : ma-valtype(k : T; k)
| . . 14. x : A ⊕ y : B(x)?Void = A
| . . 15. x : A ⊕ y : B(x)?Top = A
| . . ⊢  f(s(x),s(y),v) ∈ x : A ⊕ y : B(x)?Void
| . . | by GenConcl (s(x) = z) THEN Try (Complete Auto)
| . . |
| . . |\
| . . | .....wf .....
| . . | 1. Id
| . . | 2. x : Id
| . . | 3. y : Id
| . . | 4. k : Knd
| . . | 5. T : Type
| . . | 6. A : Type
| . . | 7. B : Type
| . . | 8. A→B→T→A
| . . | 9. ¬y = x
| . . | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . | 11. k : T ∈ k:Knd fp-> Type
| . . | 12. s : State(x : A ⊕ y : B)
| . . | 13. ma-valtype(k : T; k)
| . . | 14. x : A ⊕ y : B(x)?Void = A
| . . | 15. x : A ⊕ y : B(x)?Top = A
| . . | ⊢  s(x) ∈ A  by AllHyps (Unfold ‘ma-state‘)
| . . |
| . . .\
| . . . 1. Id
| . . . 2. x : Id
| . . . 3. y : Id
| . . . 4. k : Knd
| . . . 5. T : Type
| . . . 6. A : Type
| . . . 7. B : Type
| . . . 8. f : A→B→T→A
| . . . 9. ¬y = x
| . . . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . 11. k : T ∈ k:Knd fp-> Type
| . . . 12. s : State(x : A ⊕ y : B)
| . . . 13. v : ma-valtype(k : T; k)
| . . . 14. x : A ⊕ y : B(x)?Void = A
| . . . 15. x : A ⊕ y : B(x)?Top = A
| . . . 16. z : A
```

```
| . . . 17. s(x) = z
| . . . ⊢  f(z,s(y),v) ∈ x : A ⊕ y : B(x)?Void  by Assert (v ∈ T)
| . . . |
| . . . |\
| . . . | .....assertion .....
| . . . | 1. Id
| . . . | 2. x : Id
| . . . | 3. y : Id
| . . . | 4. k : Knd
| . . . | 5. T : Type
| . . . | 6. A : Type
| . . . | 7. B : Type
| . . . | 8. A→B→T→A
| . . . | 9. ¬y = x
| . . . | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . | 11. k : T ∈ k:Knd fp-> Type
| . . . | 12. s : State(x : A ⊕ y : B)
| . . . | 13. v : ma-valtype(k : T; k)
| . . . | 14. x : A ⊕ y : B(x)?Void = A
| . . . | 15. x : A ⊕ y : B(x)?Top = A
| . . . | 16. z : A
| . . . | 17. s(x) = z
| . . . | ⊢  v ∈ T  by DoSubsume THEN SubtypeRelEq
| . . . | |
| . . . | .....equality .....
| . . . | 1. Id
| . . . | 2. x : Id
| . . . | 3. y : Id
| . . . | 4. k : Knd
| . . . | 5. T : Type
| . . . | 6. A : Type
| . . . | 7. B : Type
| . . . | 8. A→B→T→A
| . . . | 9. ¬y = x
| . . . | 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . | 11. k : T ∈ k:Knd fp-> Type
| . . . | 12. s : State(x : A ⊕ y : B)
| . . . | 13. ma-valtype(k : T; k)
| . . . | 14. x : A ⊕ y : B(x)?Void = A
| . . . | 15. x : A ⊕ y : B(x)?Top = A
| . . . | 16. z : A
| . . . | 17. s(x) = z
| . . . | ⊢  ma-valtype(k : T; k) = T
| . . . |   by Repeat
| . . . |      (Unfolds [‘ma-valtype‘;‘kindcase‘;‘islocal‘;‘actof‘] 0
| . . . |       THEN
| . . . |       Reduce 0)
| . . . |
| . . . .\
| . . . . .....subterm .....
| . . . . 1. Id
| . . . . 2. x : Id
| . . . . 3. y : Id
| . . . . 4. k : Knd
| . . . . 5. T : Type
```

```
| . . . . 6. A : Type
| . . . . 7. B : Type
| . . . . 8. A→B→T→A
| . . . . 9. ¬y = x
| . . . . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . . 11. k : T ∈ k:Knd fp-> Type
| . . . . 12. s : State(x : A ⊕ y : B)
| . . . . 13. v : ma-valtype(k : T; k)
| . . . . 14. x : A ⊕ y : B(x)?Void = A
| . . . . 15. x : A ⊕ y : B(x)?Top = A
| . . . . 16. z : A
| . . . . 17. s(x) = z
| . . . . 18. v ∈ T
| . . . . ⊢  s(y) ∈ B
| . . . . | by GenConcl (s(y) = w) THEN Try (Complete Auto)
| . . . . |
| . . . . .....wf .....
| . . . . 1. Id
| . . . . 2. x : Id
| . . . . 3. y : Id
| . . . . 4. k : Knd
| . . . . 5. T : Type
| . . . . 6. A : Type
| . . . . 7. B : Type
| . . . . 8. A→B→T→A
| . . . . 9. ¬y = x
| . . . . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . . 11. k : T ∈ k:Knd fp-> Type
| . . . . 12. s : State(x : A ⊕ y : B)
| . . . . 13. v : ma-valtype(k : T; k)
| . . . . 14. x : A ⊕ y : B(x)?Void = A
| . . . . 15. x : A ⊕ y : B(x)?Top = A
| . . . . 16. z : A
| . . . . 17. s(x) = z
| . . . . 18. v ∈ T
| . . . . ⊢  s(y) ∈ B  by AllHyps (Unfold ‘ma-state‘) THEN DoSubsume
| . . . . |
| . . . . .....subtype_rel .....
| . . . . 1. Id
| . . . . 2. x : Id
| . . . . 3. y : Id
| . . . . 4. k : Knd
| . . . . 5. T : Type
| . . . . 6. A : Type
| . . . . 7. B : Type
| . . . . 8. A→B→T→A
| . . . . 9. ¬y = x
| . . . . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . . 11. k : T ∈ k:Knd fp-> Type
| . . . . 12. s : x@0:Id→x : A ⊕ y : B(x@0)?Top
| . . . . 13. v : ma-valtype(k : T; k)
| . . . . 14. x : A ⊕ y : B(x)?Void = A
| . . . . 15. x : A ⊕ y : B(x)?Top = A
| . . . . 16. z : A
| . . . . 17. s(x) = z
```

```
| . . . . 18. v ∈ T
| . . . . ⊢  x : A ⊕ y : B(y)?Top ⊆r B
| . . . . | by RWO
| . . . . |    Thm* ∀eq:EqDecider(A), f,g:a:A fp-> Top, x:A, z:Top.
| . . . . |         f ⊕ g(x)?z ∼ if x ∈ dom(f)→ f(x)?z else g(x)?z fi
| . . . . |    {fpf_DASH_join_DASH_cap_DASH_sq}
| . . . . |    0
| . . . . |    THEN
| . . . . |    SplitOnConclITE
| . . . . |    THEN
| . . . . |    Reduce 0
| . . . . |
| . . . . .....truecase .....
| . . . . 1. Id
| . . . . 2. x : Id
| . . . . 3. y : Id
| . . . . 4. k : Knd
| . . . . 5. T : Type
| . . . . 6. A : Type
| . . . . 7. B : Type
| . . . . 8. A→B→T→A
| . . . . 9. ¬y = x
| . . . . 10. x : A ⊕ y : B ∈ x:Id fp-> Type
| . . . . 11. k : T ∈ k:Knd fp-> Type
| . . . . 12. s : x@0:Id→x : A ⊕ y : B(x@0)?Top
| . . . . 13. v : ma-valtype(k : T; k)
| . . . . 14. x : A ⊕ y : B(x)?Void = A
| . . . . 15. x : A ⊕ y : B(x)?Top = A
| . . . . 16. z : A
| . . . . 17. s(x) = z
| . . . . 18. v ∈ T
| . . . . 19. y ∈ dom(x : A)
| . . . . ⊢  x : A(y)?Top ⊆r B
| . . . .   by RWO
| . . . .      Thm* ∀eq:EqDecider(A), x,y:A, v:Top.
| . . . .           x ∈ dom(y : v) ⟺ x = y
| . . . .      {fpf_DASH_single_DASH_dom}
| . . . .      -1
|
.\
. 1. i : Id
. 2. x : Id
. 3. y : Id
. 4. k : Knd
. 5. T : Type
. 6. A : Type
. 7. B : Type
. 8. f : A→B→T→A
. 9. ¬y = x
. 10. x : A ⊕ y : B ∈ x:Id fp-> Type
. 11. k : T ∈ k:Knd fp-> Type
. 12. @i: ma-single-effect(x : A ⊕ y : B;
.                          k : T;
.                          k;
.                          x;
```

```
.                          (λs,v. f(s(x),s(y),v)))
.     ∈ Dsys
.     & (∀D:Dsys.
.        @i: ma-single-effect(x : A ⊕ y : B;
.                             k : T;
.                             k;
.                             x;
.                             (λs,v. f(s(x),s(y),v))) ⊆ D
.        ⇒
.        D
.        realizes es.(∀x@0:Id.
.                     vartype(i;x@0) ⊆r x : A ⊕ y : B(x@0)?Top)
.                    & (∀e:E.
.                       loc(e) = i
.                       ⇒
.                       (valtype(e) ⊆r ma-valtype(k : T; kind(e))))
.                    & (∀e:E.
.                       loc(e) = i
.                       ⇒
.                       kind(e) = k
.                       ⇒
.                       (x after e)
.                       =
.                       (λs,v. f(s(x),s(y),v))
.                       ((λz.(z when e))
.                       ,val(e))))
. ⊢  @i: ma-single-effect(x : A ⊕ y : B;
.                         k : T;
.                         k;
.                         x;
.                         (λs,v. f(s(x),s(y),v)))
.    ∈ Dsys
.    & (∀D:Dsys.
.       @i: ma-single-effect(x : A ⊕ y : B;
.                            k : T;
.                            k;
.                            x;
.                            (λs,v. f(s(x),s(y),v))) ⊆ D
.       ⇒
.       D
.       realizes es.(vartype(i;x) ⊆r A) & (vartype(i;y) ⊆r B)
.                   & (∀e:E.
.                      loc(e) = i ⇒ kind(e) = k ⇒ (valtype(e) ⊆r T))
.                   & (∀e:E.
.                      loc(e) = i
.                      ⇒
.                      kind(e) = k
.                      ⇒
.                      (x after e) = f((x when e),(y when e),val(e))))
.   by ExRepD THEN BetterSplitAndConcl THEN Try Trivial
.      THEN
.      All Reduce
.      THEN
.      ParallelOp -2
.      THEN
```

```
.       RepeatFor 7 (ParallelOp -1)
.       THEN
.       Unfold ‘ma-valtype‘ -1
.       THEN
.       Reduce -1
.       THEN
.       ExRepD
.       THEN
.       BetterSplitAndConcl
.       THEN
.       Try Trivial
.       THEN
.       AllHyps
.       (λh.
.        InstHyp [x] h THENA Complete Auto THEN ReduceFpf -1
.        THEN
.        Complete Auto)
.       THEN
.       AllHyps
.       (λh.
.        InstHyp [y] h THENA Complete Auto THEN ReduceFpf -1
.        THEN
.        SplitOnHypITE -1
.        THEN
.        AllHyps (λh.Analyze h THEN Complete Auto)
.        THEN
.        Trivial)
.       THEN
.       AllHyps
.       (λh.
.        ParallelOp h THEN ThinTrivial THEN WeakSubstFor kind(e) -1
.        THEN
.        Reduce -1
.        THEN
.        Complete Auto)
.       THEN
.       AllHyps (λh.ReduceFpf h THEN Trivial)
END
```