```
ma-single-effect1_wf  THM

⊢  ∀x,y:Id, k:Knd, A,B,T:Type, f:(A→B→T→A).
|  ¬y = x ⇒ ma-single-effect1(x;A;y;B;k;T;f) ∈ MsgA
| by Auto THEN Unfold `ma-single-effect1` 0 THEN Analyze
|    THEN
|    Try (Complete Auto)
|
.....subterm .....
1. x : Id
2. y : Id
3. k : Knd
4. A : Type
5. B : Type
6. T : Type
7. f : A→B→T→A
8. ¬y = x
⊢  (λs,v. f(s(x),s(y),v))
|  ∈ State(x : A ⊕ y : B)→ma-valtype(k : T; k)→x : A ⊕ y : B(x)?Void
| by Analyze THENL [Analyze THENL [Id;Auto];Auto]
|
.....subterm .....
1. x : Id
2. y : Id
3. k : Knd
4. A : Type
5. B : Type
6. T : Type
7. f : A→B→T→A
8. ¬y = x
9. s : State(x : A ⊕ y : B)
10. 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. x : Id
| 2. y : Id
| 3. k : Knd
| 4. A : Type
| 5. B : Type
| 6. T : Type
| 7. A→B→T→A
| 8. ¬y = x
| 9. State(x : A ⊕ y : B)
| 10. 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. x : Id
| 2. y : Id
| 3. k : Knd
| 4. A : Type
| 5. B : Type
| 6. T : Type
| 7. A→B→T→A
| 8. ¬y = x
| 9. State(x : A ⊕ y : B)
| 10. ma-valtype(k : T; k)
| 11. ¬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. x : Id
. 2. y : Id
. 3. k : Knd
. 4. A : Type
. 5. B : Type
. 6. T : Type
. 7. f : A→B→T→A
. 8. ¬y = x
. 9. s : State(x : A ⊕ y : B)
. 10. v : ma-valtype(k : T; k)
. 11. 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. x : Id
. | 2. y : Id
. | 3. k : Knd
. | 4. A : Type
. | 5. B : Type
. | 6. T : Type
. | 7. A→B→T→A
. | 8. ¬y = x
. | 9. State(x : A ⊕ y : B)
. | 10. ma-valtype(k : T; k)
. | 11. 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. x : Id
. | 2. y : Id
. | 3. k : Knd
. | 4. A : Type
. | 5. B : Type
. | 6. T : Type
. | 7. A→B→T→A
. | 8. ¬y = x
. | 9. State(x : A ⊕ y : B)
. | 10. ma-valtype(k : T; k)
. | 11. x : A ⊕ y : B(x)?Void = A
. | 12. ¬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. x : Id
. . 2. y : Id
. . 3. k : Knd
. . 4. A : Type
. . 5. B : Type
. . 6. T : Type
. . 7. f : A→B→T→A
. . 8. ¬y = x
. . 9. s : State(x : A ⊕ y : B)
. . 10. v : ma-valtype(k : T; k)
. . 11. x : A ⊕ y : B(x)?Void = A
. . 12. 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. x : Id
. . | 2. y : Id
. . | 3. k : Knd
. . | 4. A : Type
. . | 5. B : Type
. . | 6. T : Type
. . | 7. A→B→T→A
. . | 8. ¬y = x
. . | 9. s : State(x : A ⊕ y : B)
```

```
. . | 10. ma-valtype(k : T; k)
. . | 11. x : A ⊕ y : B(x)?Void = A
. . | 12. x : A ⊕ y : B(x)?Top = A
. . | ⊢  s(x) ∈ A  by AllHyps (Unfold ‘ma-state‘)
. . |
. . .\
. . . 1. x : Id
. . . 2. y : Id
. . . 3. k : Knd
. . . 4. A : Type
. . . 5. B : Type
. . . 6. T : Type
. . . 7. f : A→B→T→A
. . . 8. ¬y = x
. . . 9. s : State(x : A ⊕ y : B)
. . . 10. v : ma-valtype(k : T; k)
. . . 11. x : A ⊕ y : B(x)?Void = A
. . . 12. x : A ⊕ y : B(x)?Top = A
. . . 13. z : A
. . . 14. s(x) = z
. . . ⊢  f(z,s(y),v) ∈ x : A ⊕ y : B(x)?Void  by Assert (v ∈ T)
. . . |
. . . |\
. . . | .....assertion .....
. . . | 1. x : Id
. . . | 2. y : Id
. . . | 3. k : Knd
. . . | 4. A : Type
. . . | 5. B : Type
. . . | 6. T : Type
. . . | 7. A→B→T→A
. . . | 8. ¬y = x
. . . | 9. s : State(x : A ⊕ y : B)
. . . | 10. v : ma-valtype(k : T; k)
. . . | 11. x : A ⊕ y : B(x)?Void = A
. . . | 12. x : A ⊕ y : B(x)?Top = A
. . . | 13. z : A
. . . | 14. s(x) = z
. . . | ⊢  v ∈ T  by DoSubsume THEN SubtypeRelEq
. . . | |
. . . | .....equality .....
. . . | 1. x : Id
. . . | 2. y : Id
. . . | 3. k : Knd
. . . | 4. A : Type
. . . | 5. B : Type
. . . | 6. T : Type
. . . | 7. A→B→T→A
. . . | 8. ¬y = x
. . . | 9. s : State(x : A ⊕ y : B)
. . . | 10. ma-valtype(k : T; k)
. . . | 11. x : A ⊕ y : B(x)?Void = A
. . . | 12. x : A ⊕ y : B(x)?Top = A
. . . | 13. z : A
. . . | 14. s(x) = z
```

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

```
. . . . 12. x : A ⊕ y : B(x)?Top = A
. . . . 13. z : A
. . . . 14. s(x) = z
. . . . 15. 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. x : Id
. . . . 2. y : Id
. . . . 3. k : Knd
. . . . 4. A : Type
. . . . 5. B : Type
. . . . 6. T : Type
. . . . 7. A⟶B⟶T⟶A
. . . . 8. ¬y = x
. . . . 9. s : x@0:Id⟶x : A ⊕ y : B(x@0)?Top
. . . . 10. v : ma-valtype(k : T; k)
. . . . 11. x : A ⊕ y : B(x)?Void = A

. . . . 12. x : A ⊕ y : B(x)?Top = A
. . . . 13. z : A
. . . . 14. s(x) = z
. . . . 15. v ∈ T
. . . . 16. 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
END
```