card_split_prod_intseg_family_decbl THM

```
⊢  (∀x:A. Dec(P(x)))
|  ⇒
|  ((x:A→B(x)) ∼ ((x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))))
| by UnivCD
|
1. A : Type
2. B : A→Type
3. P : A→Prop
4. ∀x:A. Dec(P(x))
⊢  (x:A→B(x)) ∼ ((x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x)))
| by Witness: λh.<h,h>
|
⊢  ∃g:((x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))→(x:A→B(x))).
|  InvFuns(x:A→B(x);(x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))
|         ;λh.<h,h>;g)
| by get choice decision function from ∀x:A. Dec(P(x))
|
4. d : A→ℕ2
5. ∀x:A. if d(x)=₂0→ P(x) else ¬P(x) fi
⊢  ∃g:((x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))→(x:A→B(x))).
|  InvFuns(x:A→B(x);(x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))
|         ;λh.<h,h>;g)
| by Witness:
|      λfg,x. if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi
|      with type (x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))→
|                (x:A→B(x))
|
|\
| .....assertion .....
| ⊢  (λfg,x. if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi)
| |  ∈ (x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))→(x:A→B(x))
| | by 2 Times Analyze
| |
| 6. fg : (x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))
| 7. x : A
| ⊢  if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi ∈ B(x)  by Witness5: x
| |
| 8. if d(x)=₂0→ P(x) else ¬P(x) fi
| ⊢  if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi ∈ B(x)
|   by SplitITE Hyp:-1 THEN SplitITE Concl
|
.\
. 6. (λfg,x. if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi)
. |  ∈ (x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))→(x:A→B(x))
. ⊢  InvFuns(x:A→B(x);(x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))
. |         ;λh.<h,h>;λfg,x. if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi)
. | by Analyze THEN Reduce 0
. |
. |\
. | ⊢  ∀x:(x:A→B(x)). (λx@0.if d(x@0)=₂0→ x(x@0) else x(x@0) fi) = x
. | | by New:h Analyze THEN Reduce 0
. | |
. | 7. h : x:A→B(x)
```

```
. | ⊢  (λx@0.if d(x@0)=₂0→ h(x@0) else h(x@0) fi) = h
. | | by FunExtensionality THEN Reduce 0
. | |
. | 8. x : A
. | ⊢  if d(x)=₂0→ h(x) else h(x) fi = h(x)  by SplitITE 0
. |
. .\
. . ⊢  ∀y:((x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))).
. . |  <λx.if d(x)=₂0→ (y.1)(x) else (y.2)(x) fi
. . |  ,λx.if d(x)=₂0→ (y.1)(x) else (y.2)(x) fi>
. . |  =
. . |  y
. . | by Analyze THEN New:[f | g] Analyze-1
. . |    THEN
. . |    Guarding <<*>,<*>> (Reduce 0)
. . |
. . 7. f : x:A st P(x)→B(x)
. . 8. g : x:A st ¬P(x)→B(x)
. . ⊢  <λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi
. . |  ,λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi>
. . |  =
. . |  <f,g>
. . | by (λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi)
. . |    ∈ x:A→B(x)
. . |    Asserted
. . |
. . |\
. . | .....assertion .....
. . | ⊢  (λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi)
. . | |  ∈ x:A→B(x)  by ApFun: Hyp:6 to: <f,g>
. . | |
. . | .....fun wf .....
. . | 9. fg : (x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))
. . | ⊢  (λx.if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi) ∈ x:A→B(x)
. . | | by Analyze
. . | |
. . | 10. x : A
. . | ⊢  if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi ∈ B(x)
. . | | by Witness5: x
. . | |
. . | 11. if d(x)=₂0→ P(x) else ¬P(x) fi
. . | ⊢  if d(x)=₂0→ (fg.1)(x) else (fg.2)(x) fi ∈ B(x)
. . |   by SplitITE Hyp:-1 THEN SplitITE 0
. . |
. . .\
. . . 9. (λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi)
. . . |  ∈ x:A→B(x)

. . . ⊢  <λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi
. . . |  ,λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi>
. . . |  =
. . . |  <f,g>
. . . |  ∈ (x:A st P(x)→B(x))×(x:A st ¬P(x)→B(x))  by Analyze
. . . |
. . . |\
. . . | ⊢  (λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi)
```

```
. . . | |  =
. . . | |  f
. . . | |  ∈ x:A st P(x)→B(x)
. . . | | by FunExtensionality THENW Inclusion 9 THEN Reduce Concl
. . . | |
. . . | 10. x : {x:A| P(x) }
. . . | ⊢  if d(x)=₂0→ f(x) else g(x) fi = f(x)
. . . | | by Analyze-1 THEN Witness5: x
. . . | |
. . . | 10. x : A
. . . | 11. P(x)
. . . | 12. if d(x)=₂0→ P(x) else ¬P(x) fi
. . . | ⊢  if d(x)=₂0→ f(x) else g(x) fi = f(x)  by SplitITE Hyp:-1
. . . | |
. . . | .....truecase .....
. . . | 12. P(x)
. . . | 13. d(x) = 0
. . . | ⊢  if d(x)=₂0→ f(x) else g(x) fi = f(x)  by SplitITE Concl
. . . |
. . . .\
. . . . ⊢  (λx.if d(x)=₂0→ (<f,g>.1)(x) else (<f,g>.2)(x) fi)
. . . .    =
. . . .    g
. . . .    ∈ x:A st ¬P(x)→B(x)  by Like the other case.
END
```