

```

┆ ∀[T:Type]. ∀[P:T → ℙ]. ∀[C:ℙ].
|   ((C ∨ (¬C)) ⇒ (∃x:T. True) ⇒ (C ⇒ (∃x:T. (P x))) ⇒ (∃x:T. (C ⇒ (P x))))
|
BY RepeatFor 3 ((UD THENA Auto))
|
[1]. T: Type
[2]. P: T → ℙ
[3]. C: ℙ
┆ (C ∨ (¬C)) ⇒ (∃x:T. True) ⇒ (C ⇒ (∃x:T. (P x))) ⇒ (∃x:T. (C ⇒ (P x)))
|
BY RepeatFor 3 ((D 0 THENA Auto))
|
4. C ∨ (¬C)
5. ∃x:T. True
6. C ⇒ (∃x:T. (P x))
┆ ∃x:T. (C ⇒ (P x))
|
BY D 4
| \
| 4. C
| ┆ ∃x:T. (C ⇒ (P x))
| |
1 BY D 6
| | \
| | ┆ C
| | |
1 2 BY NthHyp 4
| \
| 6. ∃x:T. (P x)
| ┆ ∃x:T. (C ⇒ (P x))
| |
1 BY D 6
| |
| 6. x: T
| 7. P x
| ┆ ∃x:T. (C ⇒ (P x))
| |
1 BY (InstConcl [x]. THENA Auto)
| |
| ┆ C ⇒ (P x)
| |
1 BY (D 0 THENA Auto)
| |
| 8. C
| ┆ P x
| |
1 BY NthHyp 7
| \
| 4. ¬C
| ┆ ∃x:T. (C ⇒ (P x))
| |
BY D 5
| |
| 5. x: T
| 6. True
| 7. C ⇒ (∃x:T. (P x))

```

```

⊢ ∃x:T. (C ⇒ (P x))
|
BY (InstConcl ['x']· THENA Auto)
|
⊢ C ⇒ (P x)
|
BY (D 0 THENA Auto)
|
8. C
⊢ P x
|
BY (Unfold 'not' 4 THEN D 4)
| \
| 4. x: T
| 5. True
| 6. C ⇒ (∃x:T. (P x))
| 7. C
| ⊢ C
| |
1 BY NthHyp 7
| \
| 4. x: T
| 5. True
| 6. C ⇒ (∃x:T. (P x))
| 7. C
| 8. False
| ⊢ P x
|
BY FalseHD 8

```

Extract:

```

λf,g,h. case f of
  inl(c) => let x,p = h c in <x, λc1.p>
| inr(nc) => let x,true = g in <x, λc2.any (nc c2)>

```

```

where f : C ∨ (¬C)
      g : ∃x:T. True
      h : C ⇒ (∃x:T. (P x))
      c : C
      p : P x
      c1 : C
      nc : ¬C ≡ (C ⇒ False)
      c2 : C

```