```
mn_23_refinment  THM

⊢  ∀L:LangOver(Alph), R:(Alph*→Alph*→Prop).
|  (EquivRel x,y:Alph*. x R y) ⇒
|  (∃g:((x,y:Alph*//(x R y))→𝔹). ∀l:Alph*. L(l) ⟺ g(l))
|  & (∀x,y,z:Alph*. (x R y) ⇒ ((z @ x) R (z @ y)))
|  ⇒
|  (∀x,y:Alph*. (x R y) ⇒ (x L-induced Equiv y))  by GenExRepD
|
|\
| 1. Alph : Type
| 2. L : LangOver(Alph)
| 3. R : Alph*→Alph*→Prop
| 4. EquivRel x,y:Alph*. x R y
| 5. g : (x,y:Alph*//(x R y))→𝔹
| 6. ∀l:Alph*. L(l) ⟺ g(l)
| 7. ∀x,y,z:Alph*. (x R y) ⇒ ((z @ x) R (z @ y))
| 8. x : Alph*
| 9. y : Alph*
| 10. x R y
| ⊢  x L-induced Equiv y  by (Unfold ‘lang_rel‘ 0) THEN (Reduce 0)
| |
| ⊢  ∀z:Alph*. L(z @ x) ⟺ L(z @ y)  by Analyze 0
| |
| 11. z : Alph*
| ⊢  L(z @ x) ⟺ L(z @ y)  by InstHyp [x;y;z] 7
| |
| 12. (z @ x) R (z @ y)
| ⊢   by Assert ((z @ x) = (z @ y))
| |
| |\
| | .....assertion .....
| | ⊢  (z @ x) = (z @ y)  by Analyze
| |
| .\
| . 13. (z @ x) = (z @ y)
| . ⊢   by ApplyFunToHypEquands g 13
| . |
| . 14. g(z @ x) = g(z @ y)
| . ⊢   by RW0 "6" 0
| . |
| . |\
| . | ⊢  L ∈ Alph*→Prop  by Unfold ‘languages‘ 2
| . |
| . |\
| . | ⊢  L ∈ Alph*→Prop  by Unfold ‘languages‘ 2
| . |
| . .\
| . . ⊢  g(z @ x) ⟺ g(z @ y)  by HypSubst 14 0
|
.\
. 1. Alph : Type
. 2. L : LangOver(Alph)
. 3. R : Alph*→Alph*→Prop
. 4. EquivRel x,y:Alph*. x R y
```

```
. 5. (x,y:Alph*//(x R y))→𝔹
. 6. Alph*
. ⊢  L ∈ Alph*→Prop  by Unfold ‘languages‘ 2
END
```