```
theory ALC  imports Main begin

typedecl i  type_synonym τ = "(i ⇒ bool)" type_synonym σ = "(i ⇒ i ⇒ bool)"

abbreviation bot ::"τ"   ("⊥")              where "⊥    ≡ λx. False"
abbreviation top ::"τ"   ("⊤")              where "⊤    ≡ λx. True"
abbreviation neg         ("∼")              where "∼A   ≡ λx. ¬A(x)"
abbreviation disj        (infixr "⊔" 40)    where "A⊔B  ≡ λx. A(x) ∨ B(x)"
abbreviation conj        (infixr "⊓" 41)    where "A⊓B  ≡ λx. A(x) ∧ B(x)"
abbreviation exi_r       ("∃")              where "∃r A  ≡ λx. ∃y. r x y ∧ A(y)"
abbreviation all_r       ("∀")              where "∀r A  ≡ λx. ∀y. r x y ⟶ A(y)"

abbreviation sub (infixr "⊑" 39)            where "A⊑B  ≡ ∀x. A(x) ⟶ B(x)"
abbreviation eq  (infixr "≜" 38)            where "A≜B  ≡ A ⊑ B ∧ B ⊑ A"

(* Einfaches Beispiele für etwas Meta-Theorie *)
lemma "A⊓B ≜ ∼(∼A⊔∼B)" by metis
lemma "∃r C ≜ ∼(∀r (∼C))" by metis  (* sledgehammer [remote_leo2] *)
lemma "A⊓B ≜ A⊔B" nitpick oops
```


100% Auto update Update Detach

```
Nitpicking formula...
Nitpick found a counterexample for card 'a = 2:

  Free variables:
    A = (λx. _)(a₁ := False, a₂ := False)
    B = (λx. _)(a₁ := False, a₂ := True)
```