

# An interesting idea and thinking for the question

Author: Duolei Wang

## 1. Core idea

### 1. Initial question

Firstly, we need to understand our question of the “Level 5 / 7 : Rewriting”.

Objects

P Q R S: Prop

Assumptions

h1 :  $R \leftrightarrow S$

h2 :  $\neg((P \rightarrow Q \vee \neg S) \wedge (S \vee P \rightarrow \neg Q) \rightarrow S \rightarrow Q) \leftrightarrow P \wedge Q \wedge \neg S$

Goal :

$\neg((P \rightarrow Q \vee \neg R) \wedge (R \vee P \rightarrow \neg Q) \rightarrow R \rightarrow Q) \leftrightarrow P \wedge Q \wedge \neg R$

### 2. The transformation of equivalent form

Denote the h2 as “hypothesis”:

hypothesis :  $\neg((P \rightarrow Q \vee \neg S) \wedge (S \vee P \rightarrow \neg Q) \rightarrow S \rightarrow Q) \leftrightarrow P \wedge Q \wedge \neg S$

For short, we use “LHS, RHS” to see it’s structure

hypothesis :  $\text{LHS}(P, Q, S) \leftrightarrow \text{RHS}(P, Q, S)$

$\text{LHS}(S) \leftrightarrow \text{RHS}(S)$

What we want is to prove

equation :  $\text{LHS}(R) \leftrightarrow \text{RHS}(R)$

Use the above notation, we have

Objects

P Q R S: Prop

Assumptions

equivalence :  $R \leftrightarrow S$

hypothesis :  $\text{LHS}(S) \leftrightarrow \text{RHS}(S)$

Goal :

equation :  $\text{LHS}(R) \leftrightarrow \text{RHS}(R)$

### 3. What we can do?

We can use the “apply” tactic to change our target, i.e.

|target : B

|h : A → B

apply F ⇒ |target\* : A

As a result, if we can prove the existence of some “F” in Lambda Term, such that

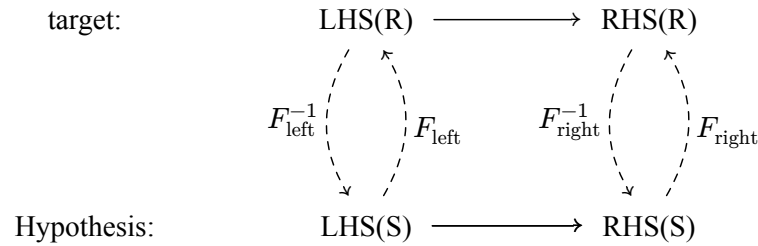
$F : \text{hypothesis} \rightarrow \text{equation},$

$(\text{LHS}(S) \leftrightarrow \text{RHS}(S)) \mapsto (\text{LHS}(R) \leftrightarrow \text{RHS}(R))$

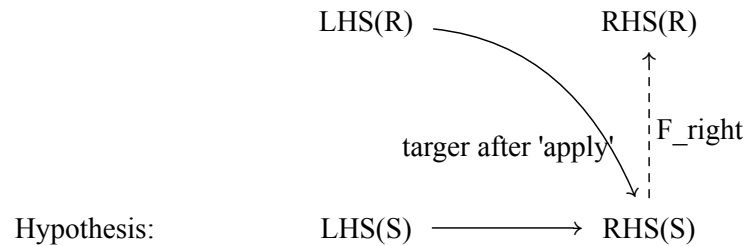
then, we can use “apply F” to get our new target  $\text{LHS}(S) \leftrightarrow \text{RHS}(S)$ , who is just the hypothesis we have. As a result, we can use “assumption” to kill it.

## 2. Build the “F” in L-term

I want to build and prove a theorem that for any Propositional Logic structure, we can substitute any term S with logical equivalent term R. But I found I can’t use the **pattern match** in the question. So, maybe it’s nearly impossible to prove the “substitution theorem” for general logic equation (At least I can’t make it at the moment). So, we can try to make it partially. I mean, after the “apply Iff.intro”, what we need is just one side of “F”. We denote the start with “\*”, it looks like



After the tactic “apply F\_right”, our new target will be  $\text{LHS}(R) \rightarrow \text{RHS}(S)$ :



As we can use lambda term with type annotation like  $\text{fun } x : P \wedge Q \wedge \neg S \Rightarrow \dots$ , I give a construction for the “F\_right”:

```
// apply F_right: P ∧ Q ∧ ¬S => P ∧ Q ∧ ¬R
apply (fun Right_S : P ∧ Q ∧ ¬S =>
  And.intro
    (And.left Right_S)
```

```

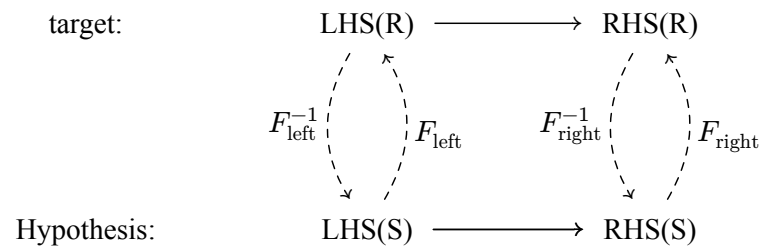
(And.intro
  (And.left $ And.right Right_S)
  (
    (fun hrs
      => fun hns
        => fun a
          => hns (mp a)) mp (And.right $ And.right Right_S)
  )
)

```

The key is to catch a term with type annotation, then use deconstructor to cut, rearrange the pieces. The next step is using “apply Hypothesis”; then we need to prove:

$$\text{target: LHS(R)} \rightarrow \text{LHS(S)}$$

which the “inverse” of part of “F” in previous graph:



And here, I believe it’s a little bit difficult to but such  $F_{\text{left}}^{-1}$  only in L-term. So I just write this document for fun.