Table 1: NLP transform formal language example one

Proposition	Language Representation
If A is B, and if B is C, then if A is C. Premise: $A \to B, B \to C$. Conclusion: $A \to C$. (A -> B) -> (B -> C) -> (A->C)	NLP First-order Logic Tigrazul Typical Logic Tigrazul Proof

```
|-A:Type;
|-B:Type;
|-C:Type;
|-test1:(A->B) -> (B->C) -> (A->C)
        := f1:(A->B)
        |-> f2:(B->C)
        |-> (x:A
                |-> (x |>f1) |> f2);
```

Table 2: NLP transform formal language example two

Proposition	Language Representation
If A is valid, then A is valid or B is valid. Premise: A. Conclusion: $A \vee B$. A -> B |> A |> OR	NLP First-order Logic Tigrazul Typical Logic Tigrazul Proof

```
|-A:Type;
|-B:Type;
|-test2:A -> B |> A |> OR
        := a:A
        -- f0: B |> A |> OR
        -- f0: C:Type -> (A->C) ->(B->C) -> C
        |-> C:Type
        |-> f1:(A->C)
        |-> f2:(B->C)
        |-> a |> f1;
```

Table 3: NLP transform formal language example three

Proposition	Language Representation
The commutative law of logic and Premise: $A \wedge B$. Conclusion: $B \wedge A$. (B |> A |> AND) -> (A |> B |> AND)	NLP First-order Logic Tigrazul Typical Logic Tigrazul Proof

```
|-A:Type;
|-B:Type;
|-test3:(B |> A |> AND) -> (A |> B |> AND)
        -- f1:(B |> A |> AND)
        := f1:(C1:Type |-> (A->B->C1)->C1)
        -- f0:(A |> B |> AND)
        -- f0:(C:Type |-> (B->A->C)->C)
        |-> C:Type
        |-> f2:(B->A->C)
        |->(a:A|-> b:B|-> (a |> b|> f2) ) |>
C |> f1 ;
```

Table 4: NLP transform formal language example four

Proposition	Language Representation
If A or B happens, then either A does not happen or B happens Premise: $A \vee B$. Conclusion: $\neg A \rightarrow B$. (B |> A |> OR) -> ((A |> NOT) -> B)	NLP First-order Logic Tigrazul Typical Logic Tigrazul Proof

```
|-A:Type;
|-B:Type;
|-test4:(B |> A |> OR) -> ((A |> NOT) -> B)
        -- f1:(B |> A |> OR)
        := f1:(C1:Type |-> (A->C1)->(B->C1)->C1)
        -- f0:((A |> NOT) -> B)
        -- f0:((A->FALSE) -> B)
        |-> f2:(A->FALSE)
        |-> (z:B |-> z) |> (z:A |->B |> z |> f2) |> B |> f1;
```

Table 5: NLP transform formal language example five

Proposition	Language Representation
If there is no x that makes A(x), then A(x) does not hold for any x Premise:$\neg\exists_x A(x)$. Conclusion: $\forall_x \neg A(x)$. ((P |> S |> EXIST) -> FALSE) -> (Q |> S |> FORALL)	NLP First-order Logic Tigrazul Typical Logic Tigrazul Proof

```
|-S:Type;
|-P:S->Type;
|-Q:(s:S)|->(s|>P -> FALSE);
|-test5:((P |> S |> EXIST)->FALSE) ->
(Q |> S |> FORALL)
        := f1:((P |> S |> EXIST)->FALSE)
        -- f1: ((a:Type |->((x:S|->((x |> P)->a))->a))->FALSE)
        -- f2: (Q |> S |> FORALL)
        -- f2: (x:S |-> x |> Q)
        -- f2: (x:S |-> x |> P -> FALSE)
        |-> y1:S |-> v:(y1 |> P)
        |-> (a:Type|->w:(x:S|->((x|>P)->a)) |-> v |> y1 |> w ) |> f1;
```