

# Compositional Semantics for Relative Clauses in Lexicalized Tree Adjoining Grammars

Chung-hye Han  
Simon Fraser University  
chunghye@sfu.ca

TAG+ 6, May 2002 — Venice, Italy

# Elementary Objects in LTAG and Implications for Compositional Semantics

- In LTAG, the elementary objects are lexicalized trees that encapsulate all syntactic/semantic arguments of the anchor.
- The meaning of a sentence in LTAG is computed as a function of meaning of elementary trees put together to derive the sentence.
- Semantics in LTAG can be defined to operate on bigger objects than in a phrase-structure based approach in which the meaning of a sentence is a function of meaning of each node in the tree.

## Compositional Semantics in LTAG on Derivation Trees

- Compositional semantics in TAG should be done on the derivation tree, not on the derived tree.
- This gives a direct correspondence between semantic composition and the syntactic operations of substitution and adjunction.
- Allows incremental and monotonic semantic interpretation for partial derivations.

# Flat Semantics

- Two ways of doing compositional semantics on the derivation tree:  
(i) Synchronous TAG (Abeillé 1994); (ii) Flat Semantics (Joshi and Vijay-Shanker 1999, Joshi and Kallmeyer 2000).
- I pursue the flat semantics approach to propose a compositional semantics for relative clauses.
- The main operation for semantic composition:  
Conjunction of the semantics associated with each elementary tree;  
Unification of variables contributed by each semantic representation.
- Why relative clauses?  
Semantics for the relative pronoun; Relation between the head noun and the relative pronoun; Predicate/argument relation within the relative clause.

# Outline

- Illustration of TAG compositional semantics with attributive adjectives
- Relative clause with simple WHNP
- Relative clause with complex WHNP
- Proposal for an analysis of relative clauses
- Generalization of the analysis:
  - Recursive pied-piping structure
  - Adjunct relative clause
- Conclusion and future work

# Attributive Adjective

(1) a difficult problem

$\begin{array}{c} N \\   \\ \text{problem} \end{array}$	$\frac{q_1: \text{problem}}{\text{arg: } -}$	$\begin{array}{c} N \\ / \quad \backslash \\ A \quad N^* \\   \\ \text{difficult} \end{array}$	$\frac{l_1: \lambda x. \text{difficult}(x) \wedge p_1(x)}{\text{arg: } p_1}$
---	--	--	--

$\alpha$ problem  
|  
 $\beta$ difficult

$q_1: \text{problem}$   
 $l_1: \lambda x. \text{difficult}(x) \wedge \text{problem}(x)$

## Relative Clause with a Simple WHNP

(2) a problem which is difficult

<p style="text-align: center;">NP   N   problem</p> <hr/> <p>q<sub>1</sub>: problem arg: –</p>	<p style="text-align: center;">NP / \ NP* S / \ WHNP<sub>i</sub>↓ S / \ NP VP   / \ t<sub>i</sub> is_difficult</p> <hr/> <p>I<sub>1</sub>: λx. p<sub>1</sub>(x) ∧ is_difficult(x) ∧ p<sub>2</sub>(x) arg: p<sub>1</sub>, p<sub>2</sub></p>	<p style="text-align: center;">NP   N   which</p> <hr/> <p>q<sub>2</sub>: which arg: –</p>
--	--	--

## Relative Clause with a Simple WHNP

(2) a problem which is difficult

$\alpha$ problem  
|  
 $\beta$ is\_difficult  
|  
 $\alpha$ which

$q_1$ : problem

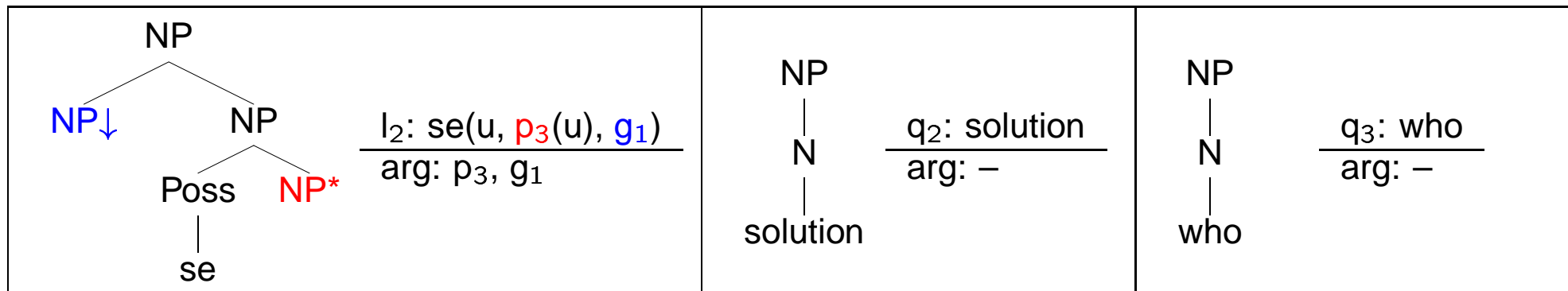
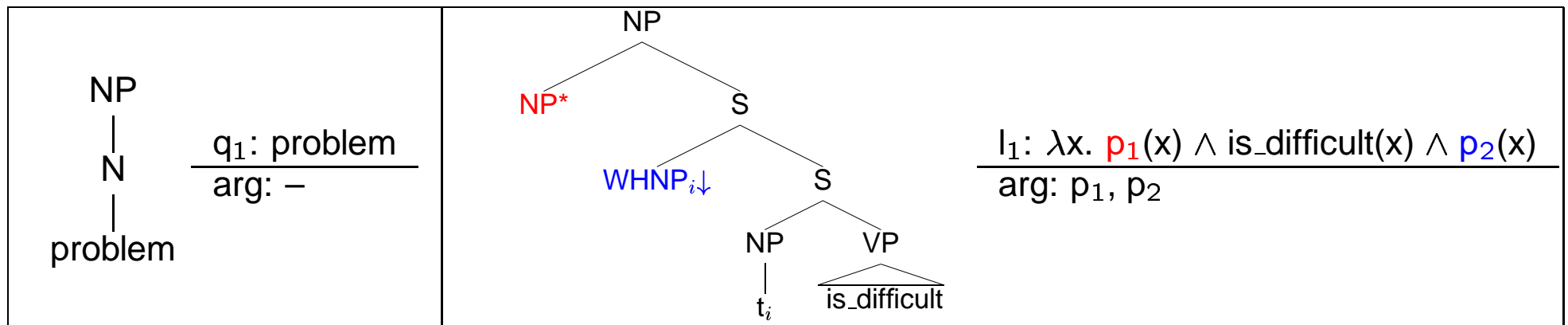
$q_2$ : which

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{is\_difficult}(x) \wedge \text{which}(x)$



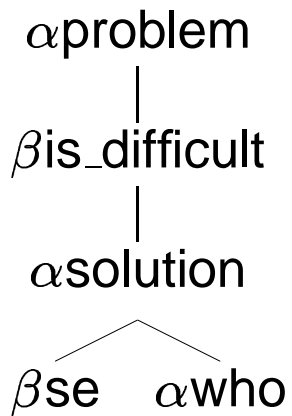
# Relative Clause with a Complex WHNP

(3) a problem whose solution is difficult



## Relative Clause with a Complex WHNP

(3) a problem whose solution is difficult



q<sub>3</sub>: who

q<sub>2</sub>: solution

l<sub>2</sub>: se(u, solution(u), who)

q<sub>1</sub>: problem

l<sub>1</sub>:  $\lambda x. \text{problem}(x) \wedge \text{is\_difficult}(x) \wedge \text{who}(x)$

## Necessary Components for a Proper Analysis of Relative Clauses

- A way to ensure that the head of WHNP is an argument of the predicate of the relative clause.

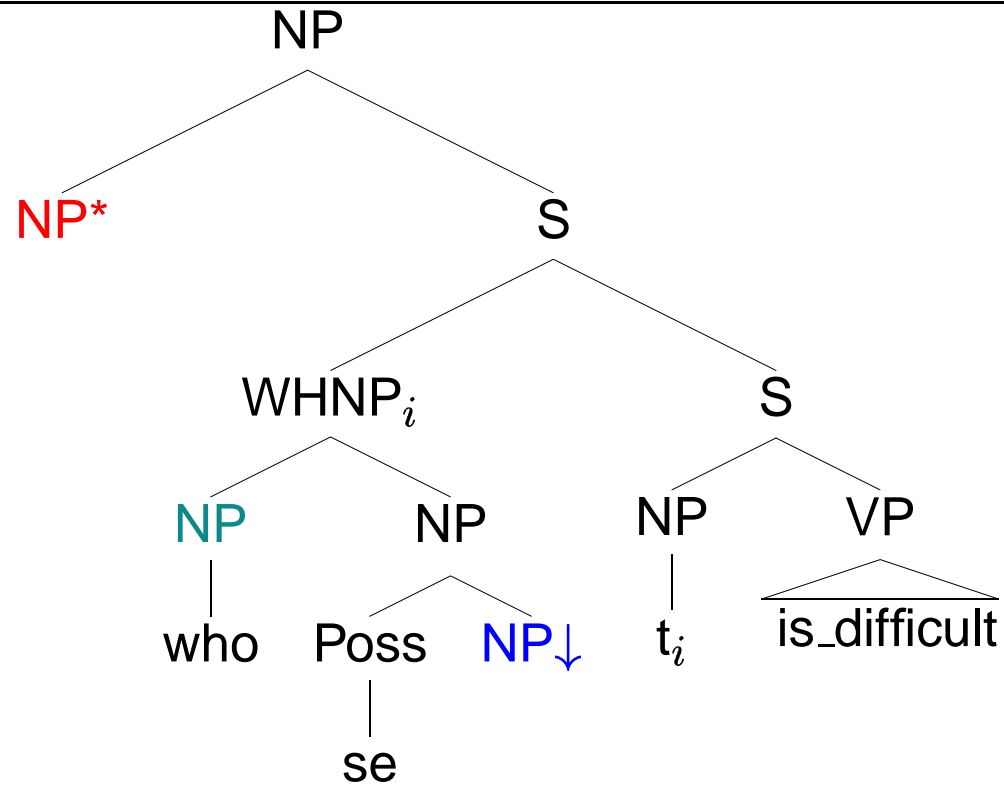
In *a problem whose solution is difficult*, what is difficult is *solution*, not *problem*.

- A way to ensure that the head noun and the relative pronoun are predicating over the same variable.

The connection between *who* and *problem* must be ensured.

# Expanded Relative Clause Tree

(3) a problem whose solution is difficult

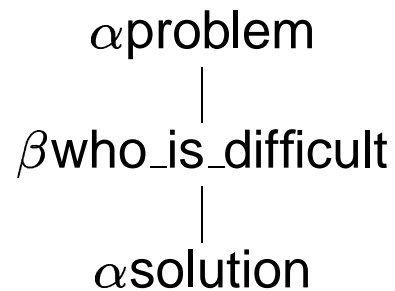


$I_1: \lambda x. p_1(x) \wedge se(u, p_2(u), g_1(who(x)) \wedge is\_difficult(u)$

arg:  $p_1, p_2, g_1$

## Expanded Relative Clause Tree

(3) a problem whose solution is difficult



$q_1$ : problem

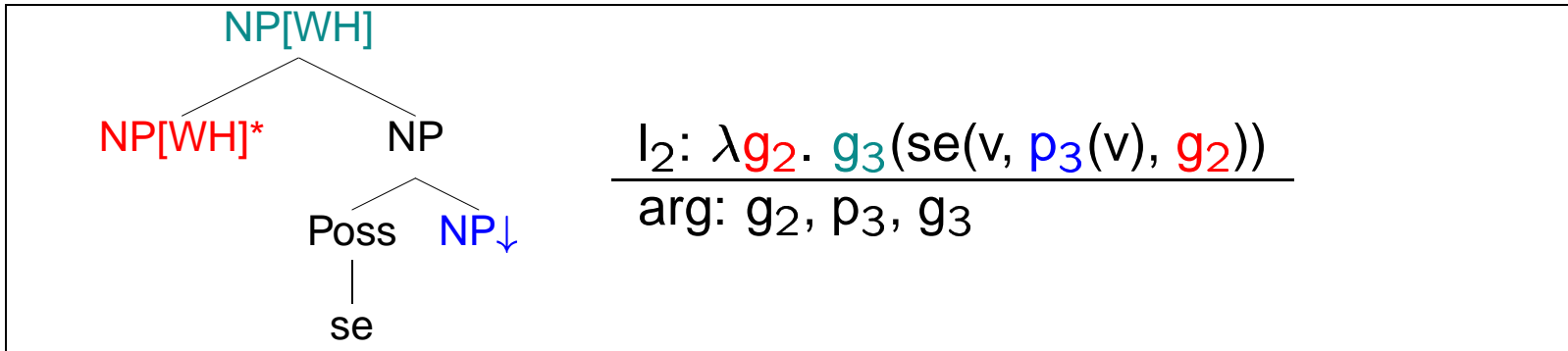
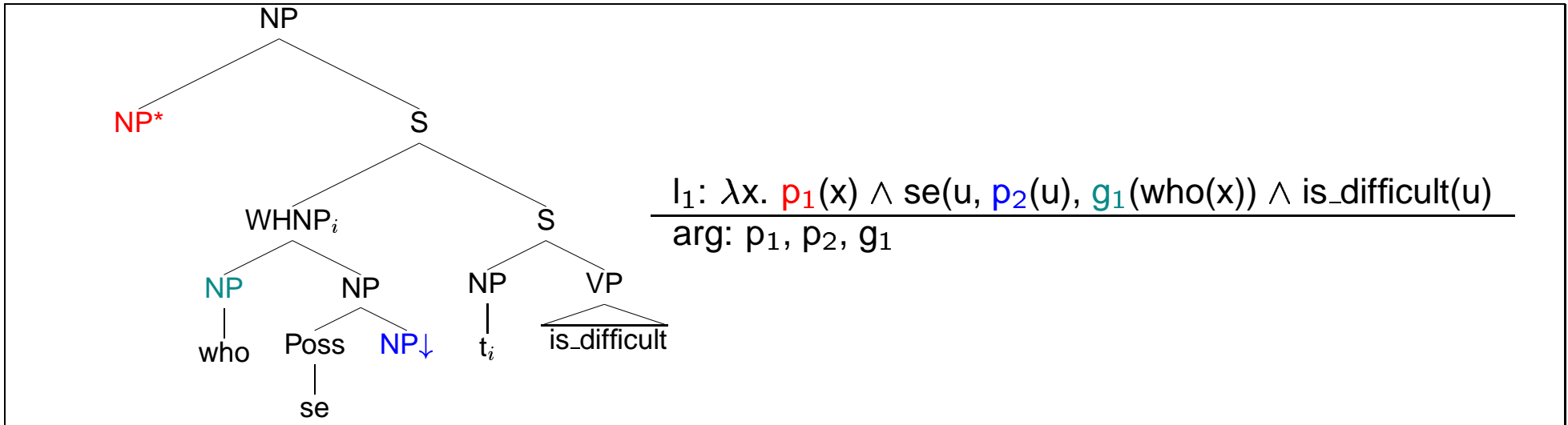
$q_2$ : solution

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{se}(u, \text{solution}(u), g_1(\text{who}(x))) \wedge \text{is\_difficult}(u)$

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{se}(u, \text{solution}(u), \text{who}(x)) \wedge \text{is\_difficult}(u)$

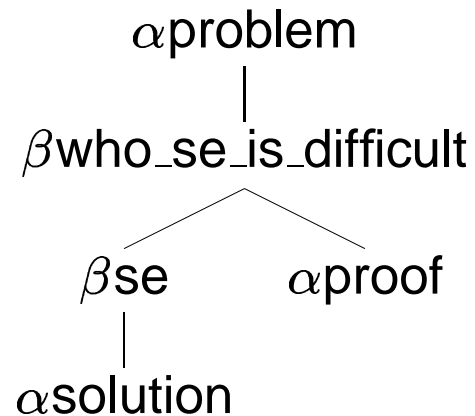
# Recursive Genitive Structure

(4) a problem whose solution's proof is difficult



# Recursive Genitive Structure

(4) a problem whose solution's proof is difficult



$q_2$ : solution

$l_2$ :  $\lambda g_2. g_3(\text{se}(v, \text{solution}(v), g_2))$

$l_2$ :  $\lambda g_2. \text{se}(v, \text{solution}(v), g_2)$

$q_3$ : proof

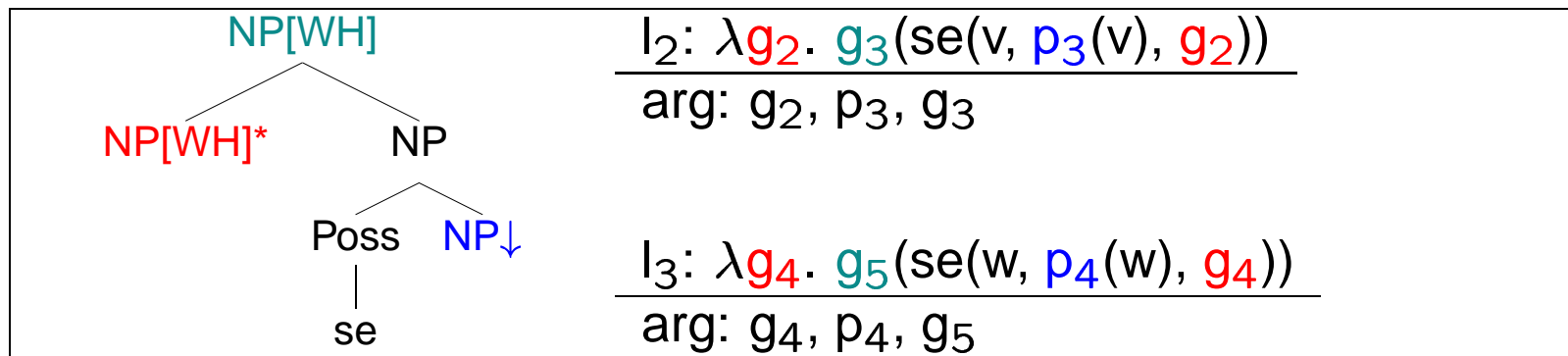
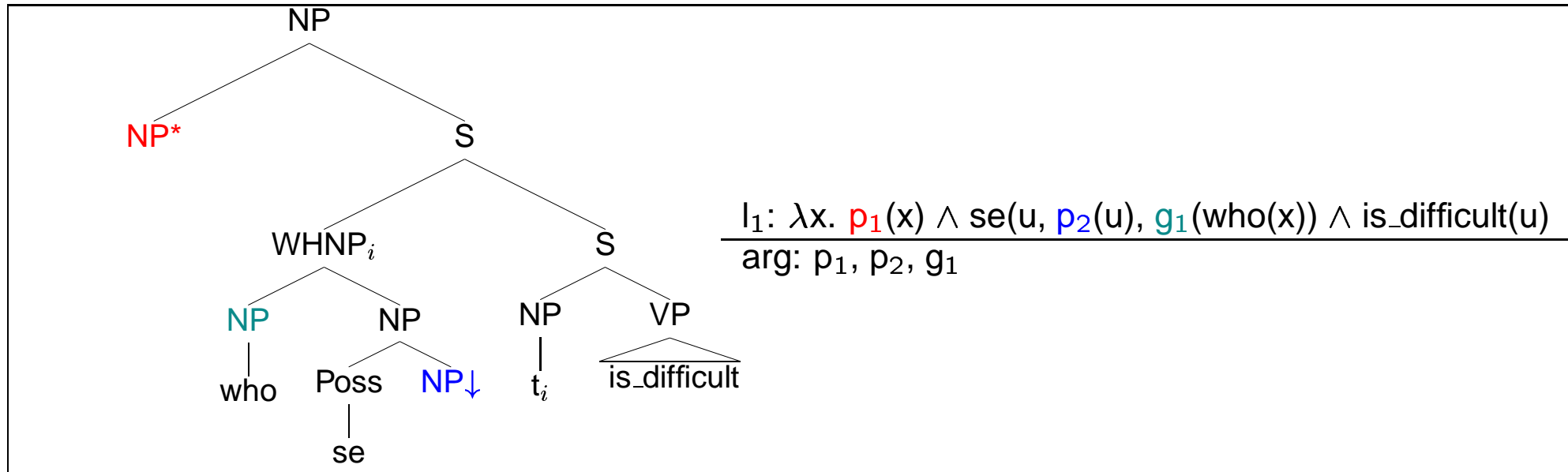
$q_1$ : problem

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{se}(u, \text{proof}(u), \lambda g_2. \text{se}(v, \text{solution}(v), g_2)(\text{who}(x)))$   
 $\wedge \text{is\_difficult}(u)$

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{se}(u, \text{proof}(u), \text{se}(v, \text{solution}(v), (\text{who}(x)))) \wedge \text{is\_difficult}(u)$

# More Recursive Genitive Structure

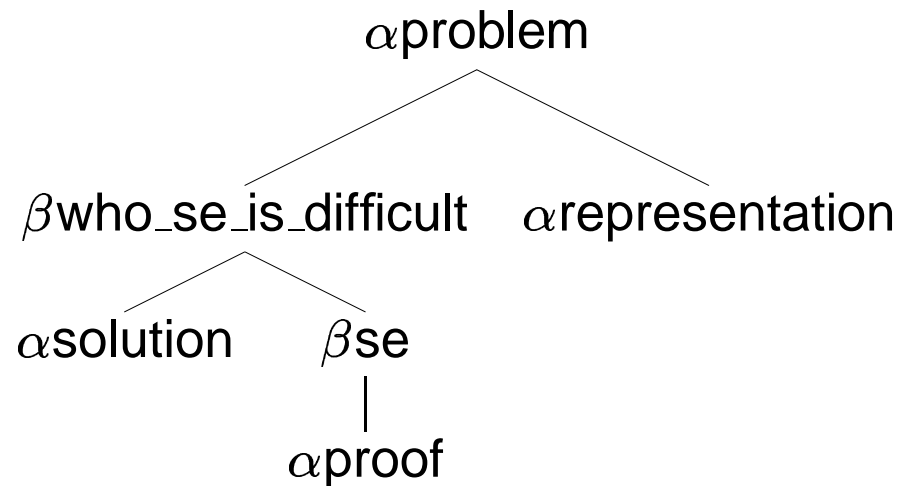
(5) a problem whose solution's proof's representation is difficult





## More Recursive Genitive Structure

(5) a problem whose solution's proof's representation is difficult



$l_3: \lambda g_4. se(w, proof(w), g_4)$

$l_2: \lambda g_2. g_3(se(v, solution(v), g_2))$

$l_2: \lambda g_2. \lambda g_4. se(w, proof(w), g_4)(se(v, solution(v), g_2))$

$l_2: \lambda g_2. se(w, proof(w), se(v, solution(v), g_2))$

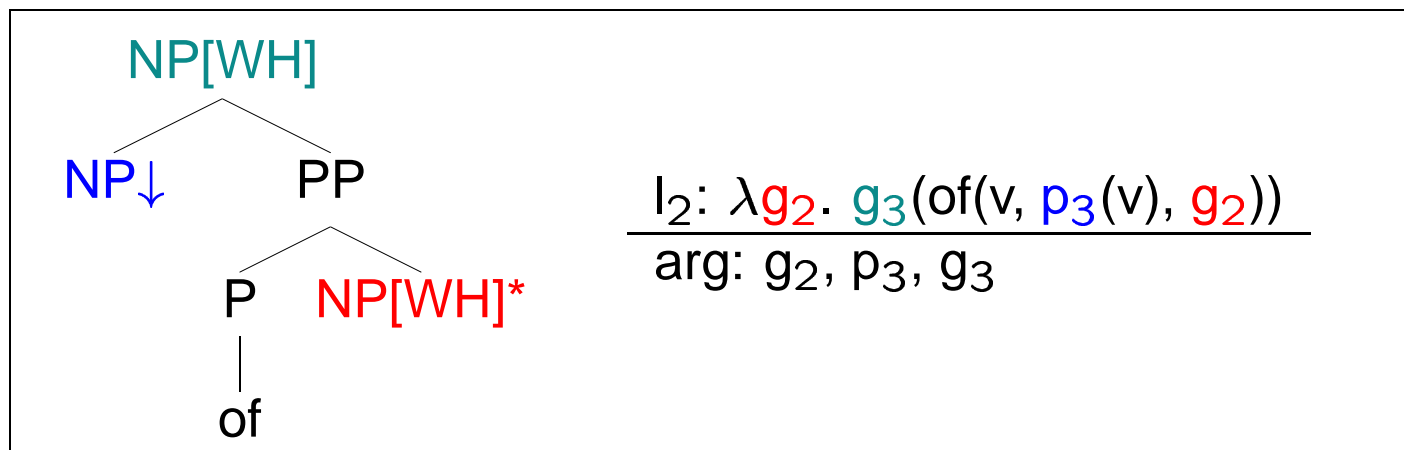
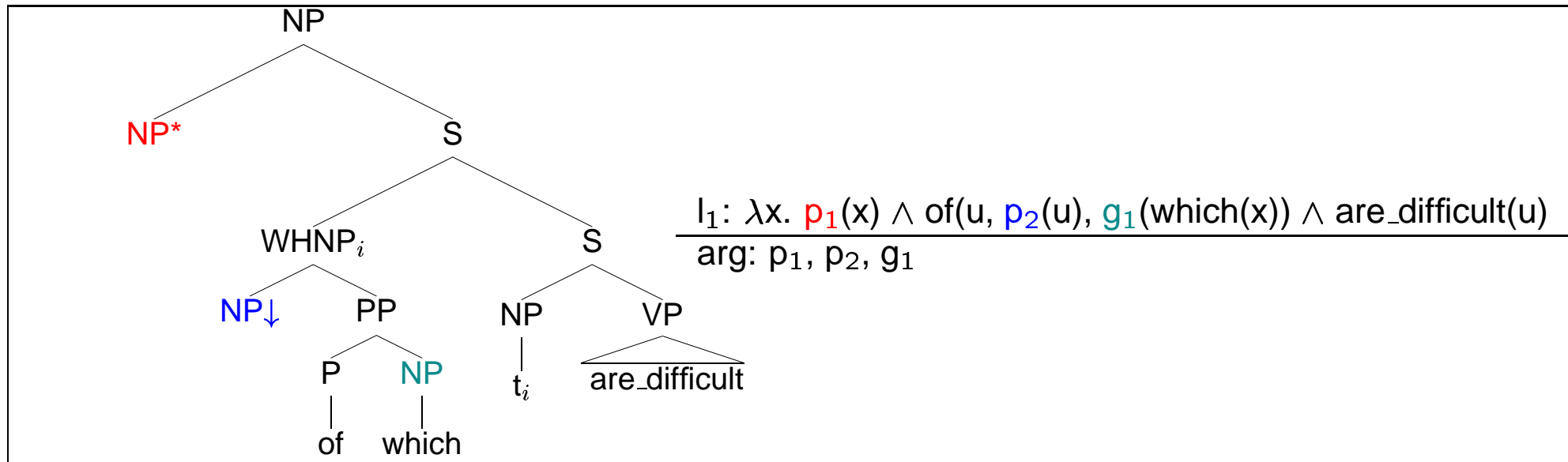
$l_1: \lambda x. problem(x) \wedge se(u, representation(u), g_1(who(x))) \wedge is\_difficult(u)$

$l_1: \lambda x. problem(x) \wedge se(u, representation(u), \lambda g_2. se(w, proof(w), se(v, solution(v), g_2)(who(x)))) \wedge is\_difficult(u)$

$l_1: \lambda x. problem(x) \wedge se(u, representation(u), se(w, proof(w), se(v, solution(v), who(x)))) \wedge is\_difficult(u)$

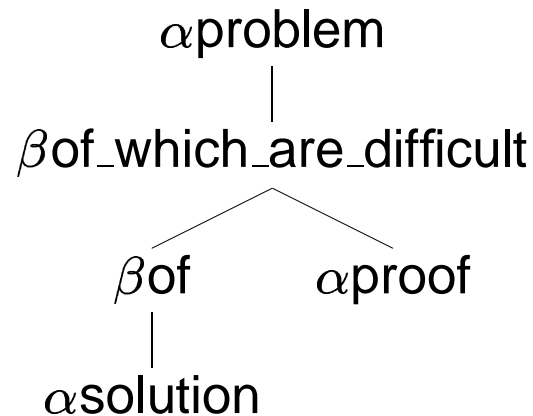
# Other Pied-Piping Structures

(6) a problem proofs of solutions of which are difficult



## Other Pied-Piping Structures

(6) a problem proofs of solutions of which are difficult



$q_2$ : solution

$l_2$ :  $\lambda g_2. \text{of}(v, \text{solution}(v), g_2)$

$q_3$ : proof

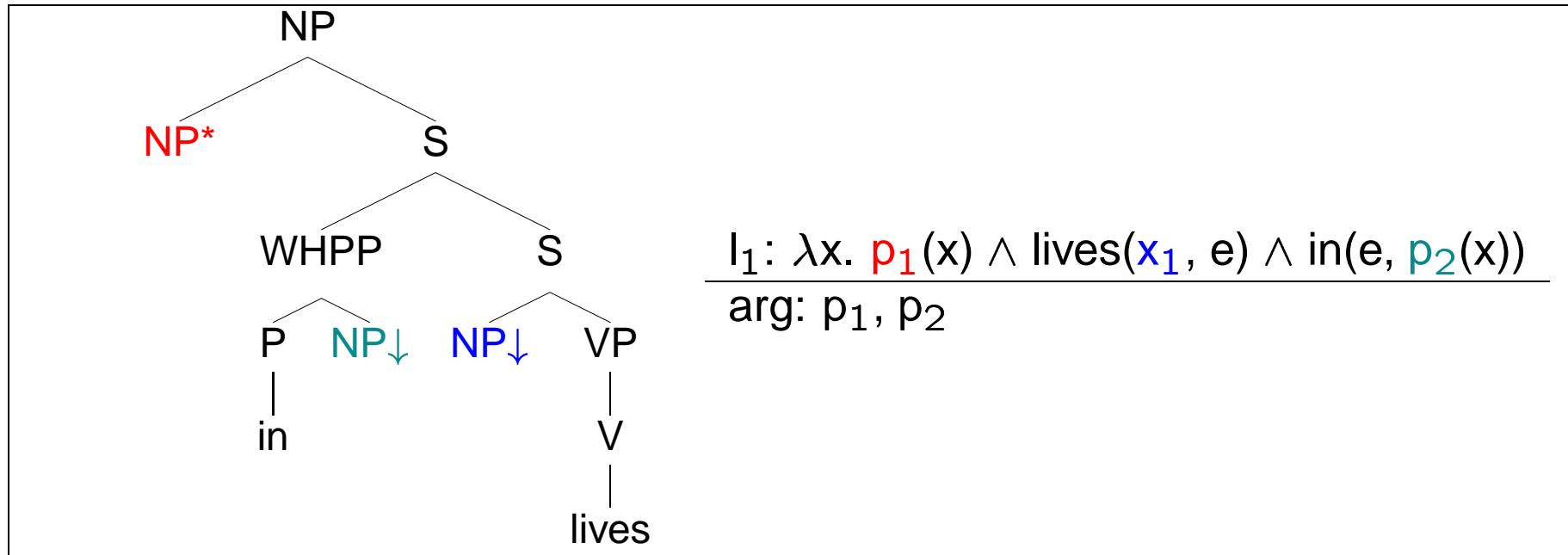
$q_1$ : problem

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{of}(u, \text{proof}(u), \lambda g_2. \text{of}(v, \text{solution}(v), g_2)(\text{which}(x))) \wedge \text{are\_difficult}(u)$

$l_1$ :  $\lambda x. \text{problem}(x) \wedge \text{of}(u, \text{proof}(u), \text{of}(v, \text{solution}(v), (\text{which}(x)))) \wedge \text{are\_difficult}(u)$

# Adjunct Relative Clauses

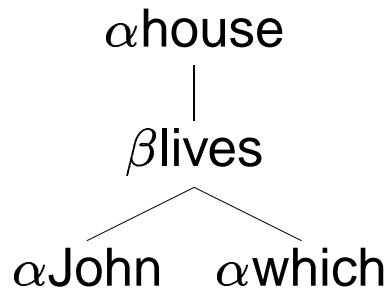
(7) the house in which John lives



<p>NP</p> <p> </p> <p>N</p> <p> </p> <p>house</p>	<p><math>\frac{q_1: \text{house}}{\text{arg: } -}</math></p>	<p>NP</p> <p> </p> <p>N</p> <p> </p> <p>which</p>	<p><math>\frac{q_2: \text{which}}{\text{arg: } -}</math></p>	<p>NP</p> <p> </p> <p>N</p> <p> </p> <p>John</p>	<p><math>\frac{l_2: \text{John}(y)}{\text{arg: } -}</math></p>
---	--	---	--	--	--

## Adjunct Relative Clauses

(7) the house in which John lives



$q_1$ : house

$q_2$ : which

$l_2$ : John(y)

$l_1$ :  $\lambda x. \text{house}(x) \wedge \text{lives}(y, e) \wedge \text{in}(e, \text{which}(x))$

# Conclusion

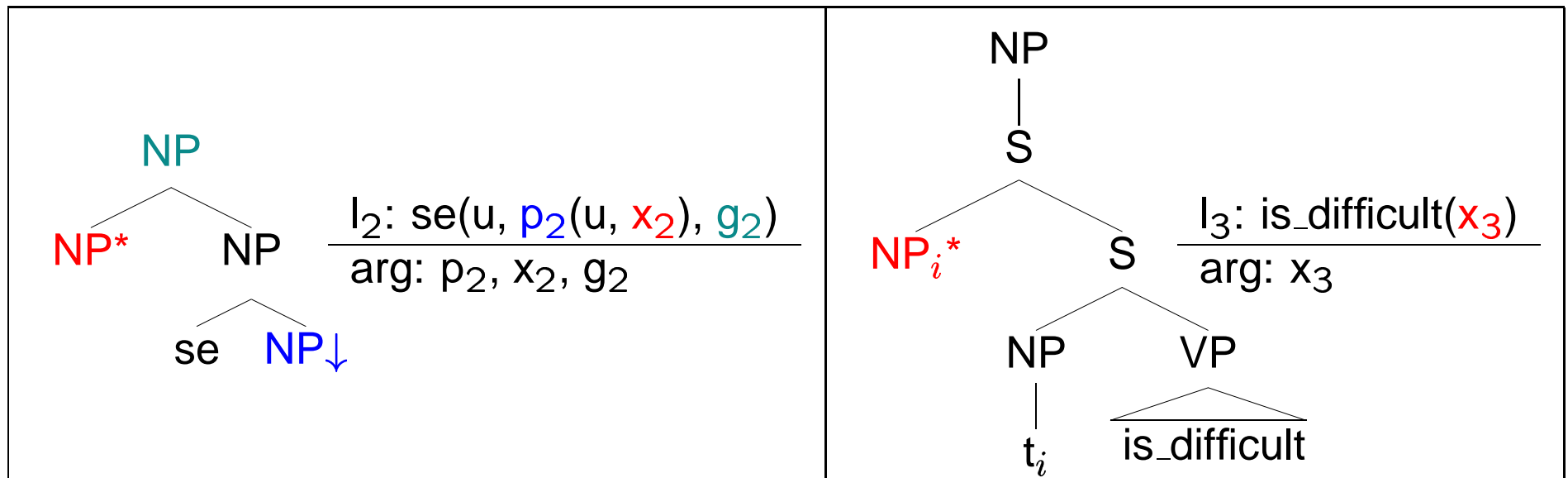
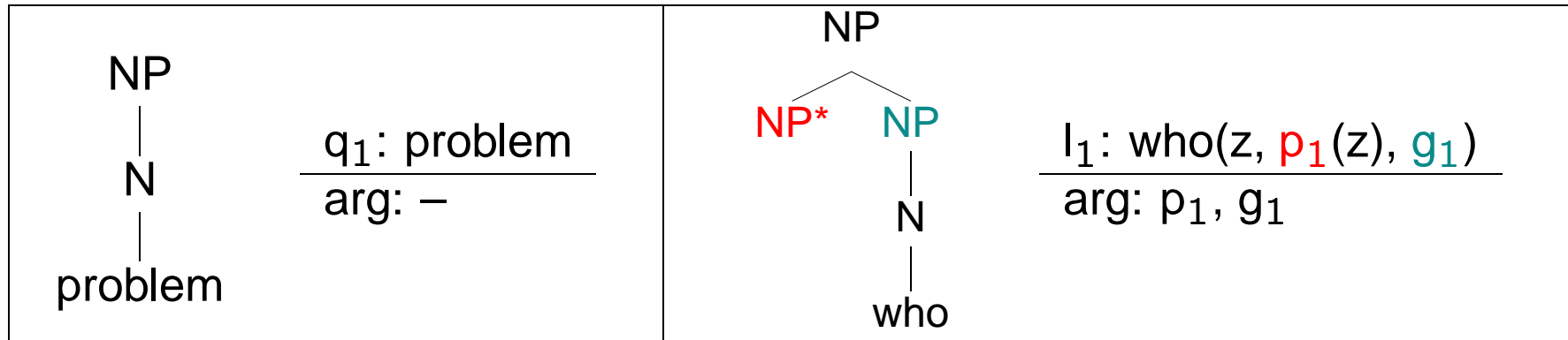
- Compositional semantics on the derivation tree is shown to be possible for complex constructions like relative clauses.
- I have proposed a flat semantics for relative clauses that keeps track of the variable for head of WHNP and the variable for relative pronoun separately.
- The analysis handles recursively embedded genitive WHNPs and other pied-piping structures.

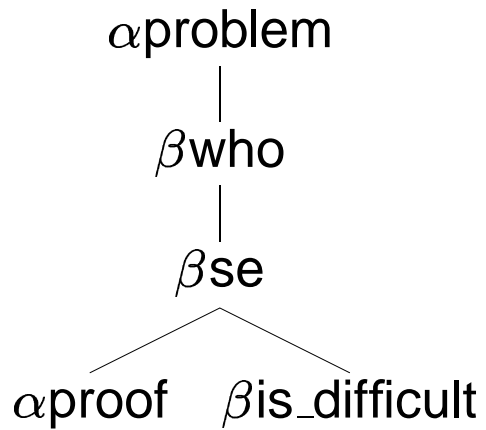
It also extends to adjunct relative clauses.

- Future work: extension to ‘picture NP’ cases like *a painting the picture of which I have a copy of*.

# Semantics Based on an Alternative Syntax of Relative Clauses

(8) a problem whose solution is difficult





$q_1$ : problem

$l_1$ : who(z, problem(z), se(u, p<sub>2</sub>(u,z), g<sub>2</sub>))

$l_1$ : who(z, problem(z), se(u, proof(u,z), g<sub>2</sub>))

$l_1$ : who(z, problem(z), se(u, proof(u,z), is\_difficult(u)))