Goals

- To know the area of NLP and their applications
- To understand the problem associated to natural language processing and the syntactic and semantic analysis levels
- Basic knowledge of the programming of DCG analyzers

Goals and areas of NLP

- The use of language to communicate among people it is one of the most important capacities of the human being
- The **goal** of Natural Language Processing (NLP) is to build computational systems that **understand** and/or **generate** human languages in any of its forms

Goals and areas of NLP

To achieve this goal we need:

- To know how people **generates** correct and understandable expressions
- To know how people **understands** expressions
- To be able to **formalize** the knowledge and the processes needed to be implemented as a computational process that is tractable

Multidisciplinarity

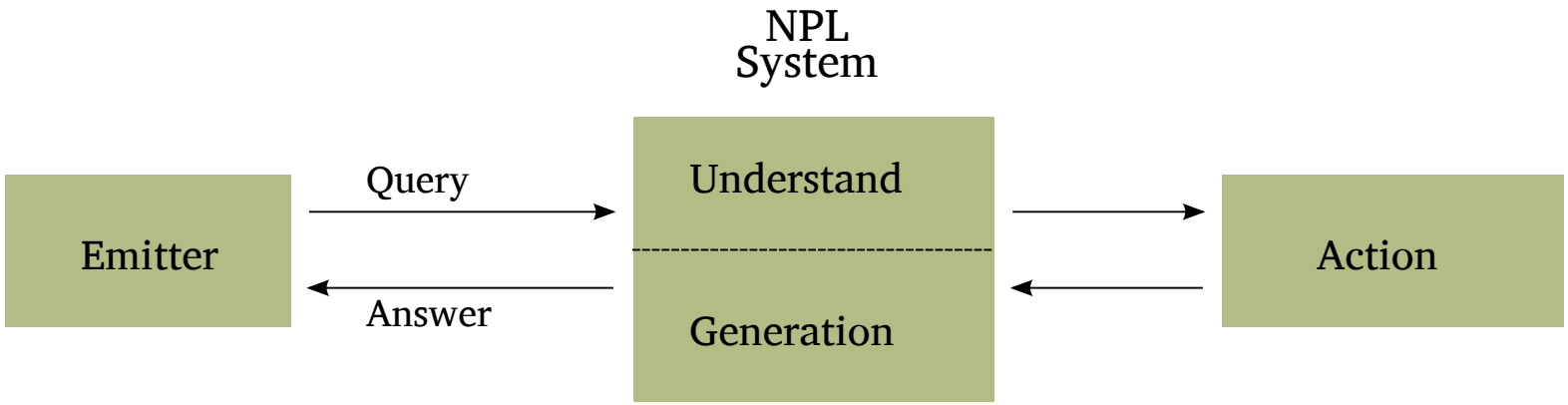
Areas associated to NLP:

- Linguistics
  - Computational linguistics
- Formal languages theory
  - Compilers
- Artificial Intelligence
  - Knowledge representation
  - Machine learning
  - Reasoning

Understanding/Generation

- Two basic operations:

<Natural Language Interfaces>

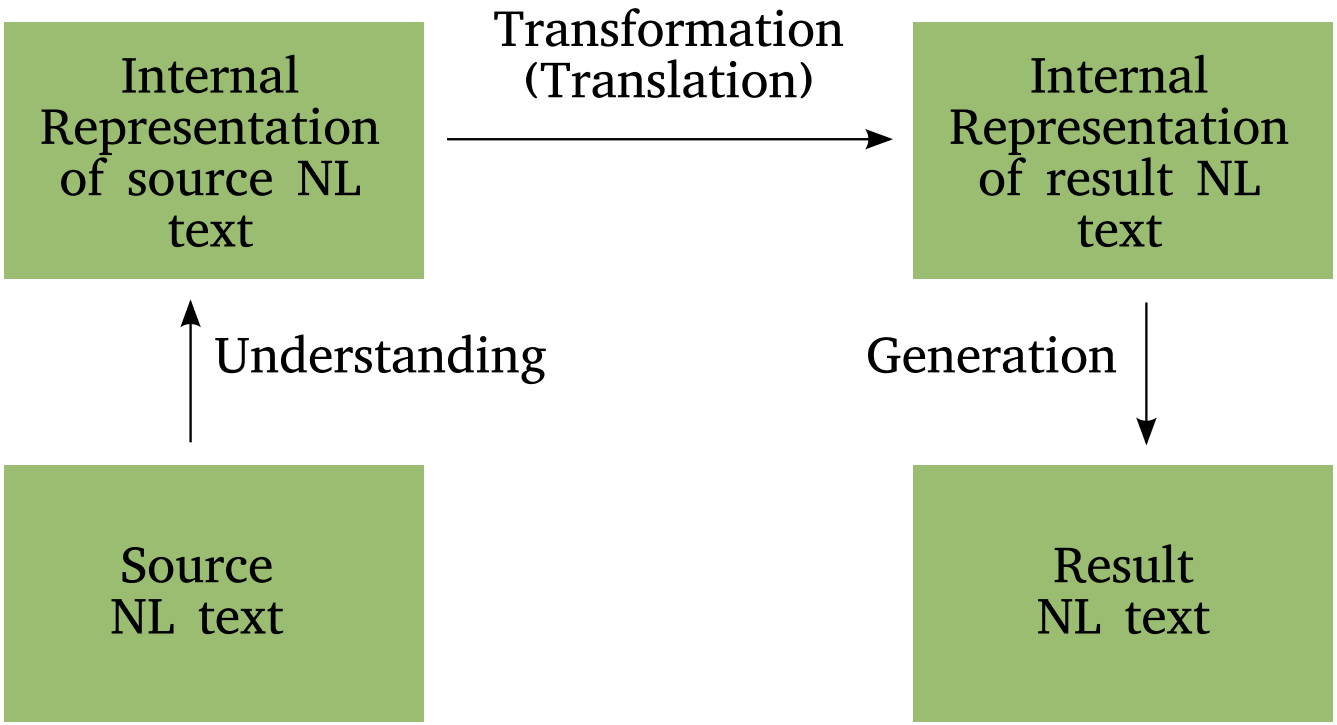


- The understanding of the query and the generation of the answer can be **oral**: speech recognition/synthesis

Understanding/Generation

- Machine Translation:

<Machine Translation: Transfer Model>



- Translation can also be done from speech

What is to understand?

"Understand something means to transform from one representation to another, so this second representation corresponds to a set of actions that can be performed and the transformation guarantees that for each element to understand the correct action is done"

(Rich, 1991)

Understanding Natural Language

- To obtain the meaning of a text in order to perform the adequate actions
  - Query of a database
  - Actuators
  - Summarize a text
  - Translate a text

How to understand Natural Language?

- Understanding needs
  - Extract the individual meaning of the words
  - Extract the meaning of the relations among words
  - Refer the literal meaning to the actuation context of the system: Metaphors, rhetoric, irony, intonation
- Tools
  - Analysis of the components of language at different levels

Information for the analysis

- **lexical**, **syntactical**, **semantical** information
- Example:

  "Of course, you are talking about the urban restructuring of Barcelona"

- It Has to be detected:
  - Individual words with meaning and connectives: Barcelona, restructuring, of, the, urban, talking
  - Gather information to know their role on the sentence and figure out possible meanings:
    - Morphosintactic category: name, proper name, composed name, verb, article, etc.
  - Information about the relation among meanings to determine the global meanings
    - Syntactic role: subject, direct complement, etc.

Levels of analysis

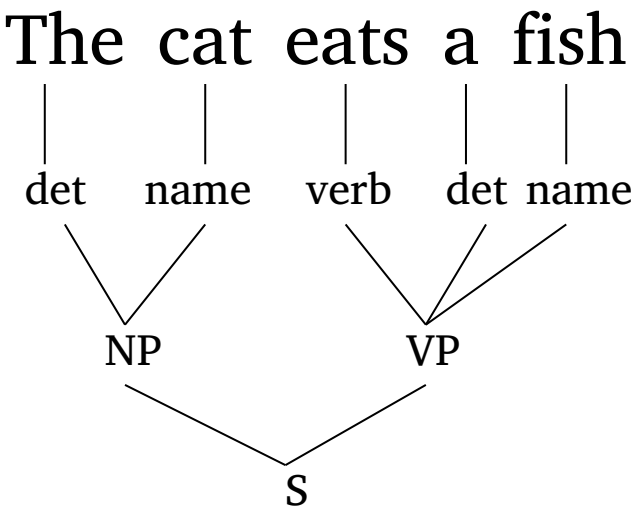
- Phonological
  - Treatment of sounds to found units of expression
- Textual
  - Segmentation of the text on tractable units (paragraphs, sentences, etc.)
  - Localization (identification) of words and lexical units
- Morphological
  - Formation of words considering derivation, composition, etc.
  - The morphological analysis gathers information to help in finding the roots and affixes (prefixes, suffixes, infixes) of the words
  - The word as a composition of morphemes
- Lexical
  - Considering a word as the meaning unit of the text
  - Obtaining lexical-semantical information (ontologies, semantic dictionaries)

Levels of analysis (2)

- Syntactical (1): Detection of syntactically correct structures

The cat eats a fish
The cat eat a fish

- Syntactical (2): Extraction and representation of syntactically correct structures



Levels of analysis (3)

- Logical
  - Extraction of the literal meaning of the sentence
  - Representation of the meaning using Predicate Calculus, frames, semantic networks, etc.
  - In the case of Predicate Calculus: representation using variables, predicates, functions, constants, logical connectives, quantifiers, etc.

$$\text{“The cat eats a fish”} \equiv \exists x \exists y Cat(x) \wedge Fish(y) \wedge Eats(x, y)$$

Levels of analysis (4)

- Semantical
  - Interpretation of the logical form: Relation among the logical entities (constants, variables, terms, etc.) and the real world (or its representation): objects of the domain
  - Ex:
    - A cat is a feline,
    - fish is edible
    - The actor of eat must be a living being,
    - ...

Levels of analysis (5)

- Pragmatical
  - Interpretation in a context (using implicit information)
  - ex: **The plane detected the school** (of fishes)
- Illocutionary
  - Detection of the intent of the speaker
  - Ex: "There are dirty dishes in the sink"
    - Is it a neutral declarative sentence?
    - Is it an invitation to perform an action? ("wash them!")
    - Is it a complaint? ("You always forget to wash the dishes and I have to do it")

Problems of natural language (1)

- Lexical ambiguity
  - *The **bear** is climbing the tree*
  - **bear** could be a noun or a verb (POS tagging)
  - *The fisherman went to the **bank**.*
  - The slope of land adjoining a body of water? A business establishment in which money is kept? (WSD)
- Syntactical ambiguity
  - "The man went to the house with a boat"
  - "The chicken is ready to eat" (PP-attachment)

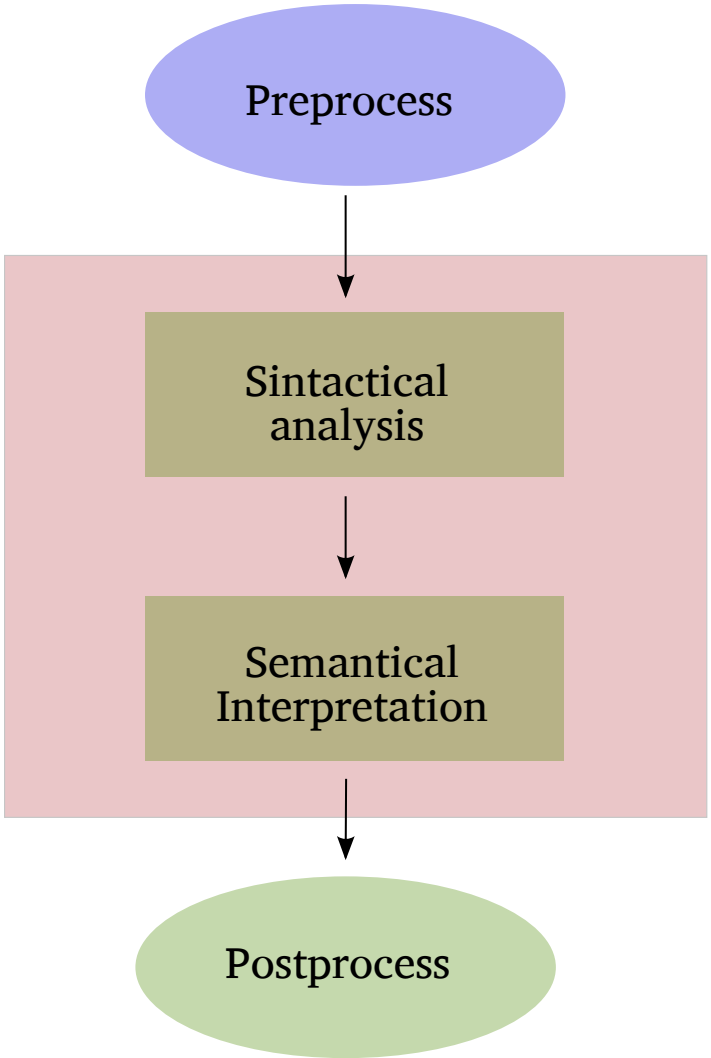
Problems of natural language (2)

- Semantic ambiguity
  - "She gave a cake to the children"
  - 1 for all?, 1 for each? (scope of quatifiers)
  - "The green ideas sleep furiously" (Chomsky)
- References, ellipsis (pragmatic level)
  - "She gave her a book" (...) "She didn't like it".
- Illocution: problems to find out the intent
  - "There are dirty dishes in the sink" (wash them right now!)

Analysis strategies

- The resolution to some of these ambiguities needs of the collaboration **among the levels of analysis**(an analyzer for each level?): more knowledge about the context
- Cooperation among analyzers:
  - Stratified: Sequential / waterfall
  - Global (parallel)

Natural language processing



What is the capital of France?

Preprocess

- Segmentation
- Location of units (words)
- Stemming, morphological analysis
- Morphosyntactic disambiguation (POS tagging)
- Semantic tagging
- Semantic disambiguation (WSD)
- Detection and classification of named entities (Named Entity Recognition, NER)

Textual analysis

- Detection of tractable units: paragraphs and sentences
  - Simple methodologies,
    - Based on punctuation marks: ".", "?", "!", "...", etc.
  - Problems: acronyms, initials, etc.
- Methodologies based on machine learning (classification)
  - Contextual information is used

Lexical analysis: goals

- Detect words (units of meaning)
  - Needs to be able to recognize and fragment adequately the words: "The/ man/ checked in/ his/ luggage"
- Gather information and apply knowledge that reduces the cost of the next analysis processes
  - Associate grammatical categories
  - Associate semantical information to the lexical units (using ontologies, dictionaries, etc.)
  - Recognize and classify names and entities

Problems of lexical analysis (1)

- **Correspondence among orthographical and grammatical word**
  - Domain knowledge is needed to detect cases like:
    1 orthographic word, many grammatical words (contractions in english), 2 orthographic word, 1 grammatical words (phrasal verbs in english)
- **Homonimy**
  - Same word and different grammatical categories: *bear* (verb to bear), *bear* (name) $\longrightarrow$ Syntactical analysis
- **Polysemy**
  - Same word and category, different meanings: *bank*

Problems of lexical analysis (2)

- **Acronyms**
  - "When a PCB is generated it can inserted on a FIFO queue"
  - "The cell's DNA sample was identified by PRC, a process approved by the official UBI"
- **Abbreviations**
  - "Dr. Smith talk about Nat. Lang. Proc."
- **Formulas and measures**
  - "add two mg. of DM-oxane and put it in a PVC vial"
  - "Given that x=y*2 + k, where k is a constant"
- **Volume of information**

Domain Knowledge

- Lexicons are the main source
  - "Lexical dictionaries"
  - They collect information that helps to recognize and categorize the place in the text of words
- We need to decide the kind of information that the dictionary needs:
  - Syntactical category
    - Determinant, proposition, proper nouns, substantive, verb, etc.
    - Granularity problem (verb -> transitive/intransitive)
  - Syntactical properties of agreement
    - gender (masculine/feminine)
    - number (singular/plural)
    - person (first, second...)
    - case (accusative,dative..)

Domain Knowledge

- Other syntactical properties:
  - Possible kinds of complements of the verb
  - Prepositions that follow a word
- Semantical categories
- Morphological information
  - Derivation: prefixes/infixes/suffixes

Morphological analysis

- Simple methods: list of forms with morphological information and their lemmas (formaries)
- Morphological analyzers:
  - Morpheme dictionaries: Dictionaries of roots, of suffixes, prefixes, etc.
  - Morphotactics: rules of morpheme combination
  - Phonological variations: changes combining the morphemes
- Types of analyzers
  - 1 level: FSA
  - 2 levels: FST
  - More than two levels: sequence of FSTs

Result of the preprocess (lexical/morphological)

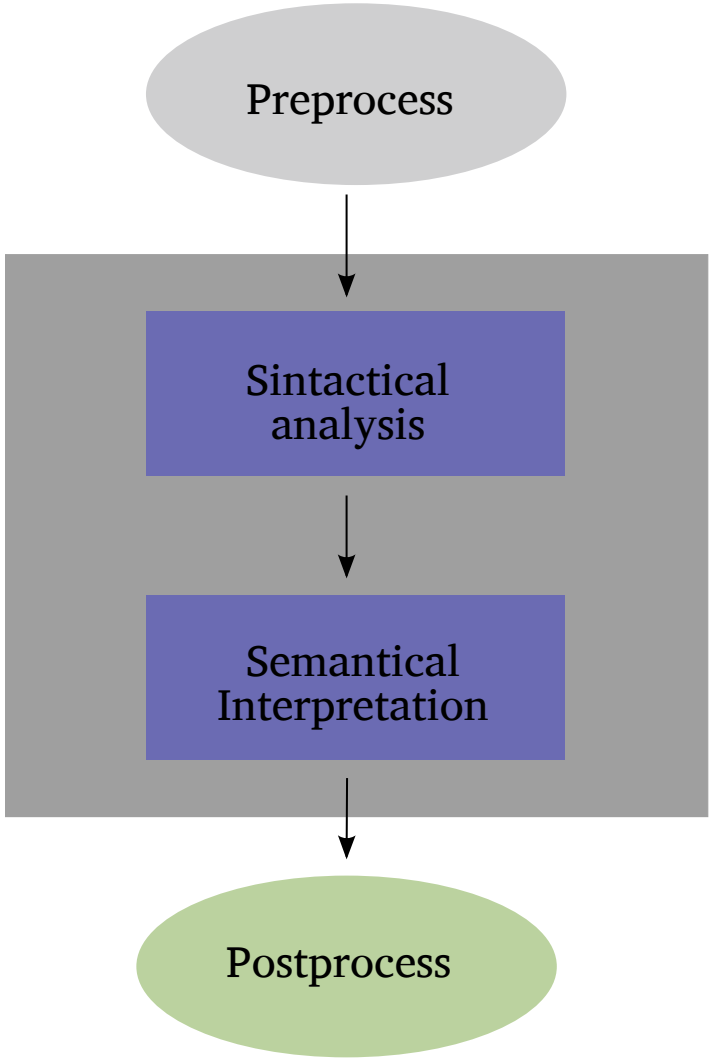
What is the capital of France? Results of the morphological analysis

What	PRON00034		
is	VERB02604760		
The	DET0001		
capital	NOUN13354420	NOUN08518747	ADJ02342778
of	PREP00017		
France	NOUN08929922		

Results of POS tagging

What	PRON00034
is	VERB02604760
The	DET0001
capital	NOUN08518747
of	PREP00017
France	NOUN08929922

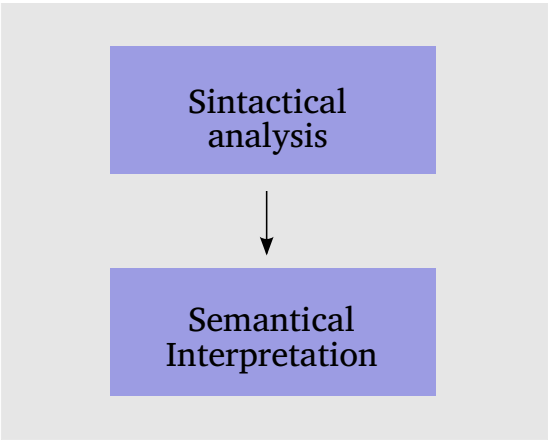
Syntactical analysis



What is the capital of France?

```
(sentence
 (interrogative_sentence
   (interrogative_pronoun (what))
   (verb (is)
     (sj (the capital of France)))
   )
 )
)
```

Types of collaboration



- Without syntax
- Without semantics
- Sequential process (1):
  syntax | semantics
- Sequential process (2):
  {syntax + semantic filter} | semantics
- Parallel process:
  {syntax, semantics}

Syntactical analysis (1)

- Goals
  - Determine if the sentence (textual unit) is syntactically correct
  - Create a syntactical structure with information that is needed by the semantical analysis and other analysis

Syntactical analysis (2)

- Alphabet (vocabulary) Σ
- Concatenation operation
- Σ^* set of all the strings with symbols from Σ (free monoid)
- Language $L \subseteq \Sigma^*$
- Given a string from Σ^* $w_1^n = w_1, ...w_n \;\; w_i \in \Sigma$, determine if $w_1^n \in L$

Defining the membership of a word

- Grammar
  - $G \Rightarrow L(G)$
  - $w_1^n \in L(G)$?
- Language model
  - $P(w_1^n)$
  - if $P(w_1^n) > 0 \Rightarrow w_1^n \in L$
- Corpus (sentences, patterns) that define the correct sentences
  - Syntactic dictionary
  - Rules of composition
- Rules of well formation
  - filters, negative grammars, ...

The usual method: Grammar

- Constituents grammars
  - Derivation tree
- Dependency grammars
  - Dependency schemes
- Case grammars
  - Actant models $\Rightarrow$ Semantic networks

Syntagmatic Structure Grammars

- $< V, \Sigma, P, S >$
  - V: Non Terminal Vocabulary (Set of variables)
  - Σ: Terminal vocabulary (alphabet)
  - P: Set of productions
  - S: Initial variable

$$\Sigma \cap V = \varnothing$$
$$\Sigma \cup V = \text{Vocabulary}$$
$$S \in V$$

Types of grammars - Hierarchy of Chomsky (1)

Type 0 **Unrestricted grammars**

- The elements of P are rewriting rules of the form

$$u \rightarrow w, \quad w, u \in (V \cup \Sigma)^*$$

- Corresponds to the recursively enumerable languages
- Recognized by Turing machines

Type 1 **Context-sensitive Grammars**

- The length of the rule is limited

$$u \rightarrow w, \quad w, u \in (V \cup \Sigma)^* and |u| \leq |v|$$

- Corresponds to Context sensitive languages
- Recognized by Linear Bounded Automaton

Types of grammars - Hierarchy of Chomsky (2)

Type 2 **Context-free Grammars (CFG)**

- The elements of P are rewriting rules restricted to:

$$A \to w, \quad A \in V, w \in (V \cup \Sigma)^*$$

- Corresponds to context free languages
- Recognized by non non-deterministic push down automaton

Type 3 **Regular Grammars (RG)**

- The elements of P are rewriting rules of the form:

$$A \to a$$
$$A \to aB, \qquad A, B \in V, a \in \Sigma$$

- Corresponds to regular languages
- Recognized by finite state automaton

Gramaticalitity

- A sentence w (a word from Σ^*) belongs to the language generated by the grammar:

$$w \in L(G)) \Longleftrightarrow s \rightarrow_G^* w$$

- In other words, the grammar G can generate the word w using the productions from S.

Building the grammar

- Define the terminals labels (tagset, Σ)
- Define the non terminal labels (V)
- Grammar rules (P)
  - Manual development
  - Automatic development: Grammatical inference (induction)
  - Semi automatic development

Grammars for Natural Language Processing

- At least context free grammars
- Is NL a context free language?
- Enough? Usually NO
- Solution
  - CFG + {procedural context additions}
  - Logical and unification grammars
  - Grammars enhanced with statistical information
  - Lexicalized grammars

Example of context free grammar

```
(1) Sentence -> NG,VG
(2) NG   -> det,n
(3) NG   -> n
(4) VG   -> iv
(5) VG   -> tv, NG
(6) det -> the | a
(7) n    -> cat | fish
(8) tv   -> eat | ...
(9) iv   -> eat | ...
```

CFG + {Procedural Context Additions}

```
sentence -> question | command | ...
command  -> v,ng {imperative(1), command(1)}
ng       -> ngbase, [ngmods] | pn {agreement(1,2)}
ngbase   -> [det], n, [adjs] {agreement(1,2,3)}
adjs     -> adj, [adjs]
ngmods   -> ngmod, [ngmods]
ngmod    -> ps | ...
ps       -> prep, ng
pn       -> barcelona | valencia | ...
n        -> ticket | euromed | ...
v        -> give | buy | ...
det      -> a | the | ...
```

Factors that intervene in the syntactical analysis process

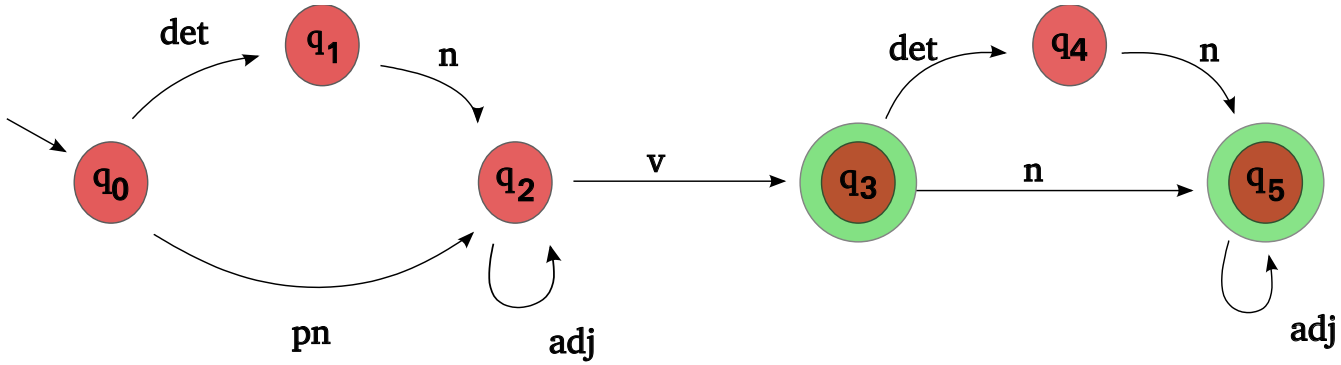
- Grammar expressiveness (FSA, CFG, CFG+procedures)
- Coverage (lexicon, types of sentences)
- Sources of knowledge (linguistic knowledge)
- Analysis strategy (left to right, right to left, ...)
- Analysis direction (direction of writing, opposite direction of writing)
- Order applying the rules of the grammar (determinist, non determinist, preferences, ...)
- Treatment of ambiguity (use of semantic information)

CFG analyzers and extensions

- Simplifications of CFG
  - $CFG \Rightarrow RG$: Finite automata techniques: FSA
  - $CFG \Rightarrow DCFG$: Determinist analyzers: LL, LR
- Extensions of Finite State Automat
  - $TN \Rightarrow RTN \Rightarrow ATN$ (Woods, 1970)
- Well Formed String Tables (WFST) , Charts
- Tabular methodologies: CKY, Earley (1970)
- Grammars of sentence structure: LSP (N. Sager, 1981), Diagram (A. Robinson, 1981)

Transition networks (TN)

- Finite automata
- States associated to parts of the sentence
- Transitions: Labels referring to morphosyntactic categories
- Non determinism



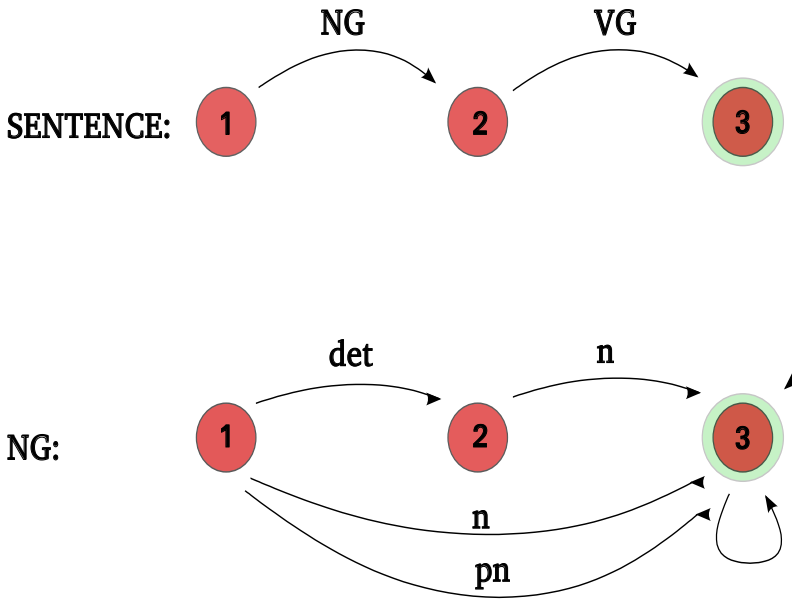
TN: limitations

- Limited to regular languages
- It is not really an analyzer: It **just recognizes**
- No-determinism $\Rightarrow$ backtracking $\Rightarrow$ Inefficient
- No distinction between grammar and analyzer
  - grammar $\Rightarrow$ description of the syntactic model
  - analyzer (parser) $\Rightarrow$ control

Recurrent Transition Networks (RTN)

- Collection of labeled transition networks (TN)
  - Edges labeled with categories $\Rightarrow$ as in TN
    - Terminal labels
  - Edges labeled with identifiers of TN
    - Non terminal labels
    - The end states of the TN return to the destination state of the transition that has called the TN
- RTN are weakly equivalent to CFG

RTN: examples



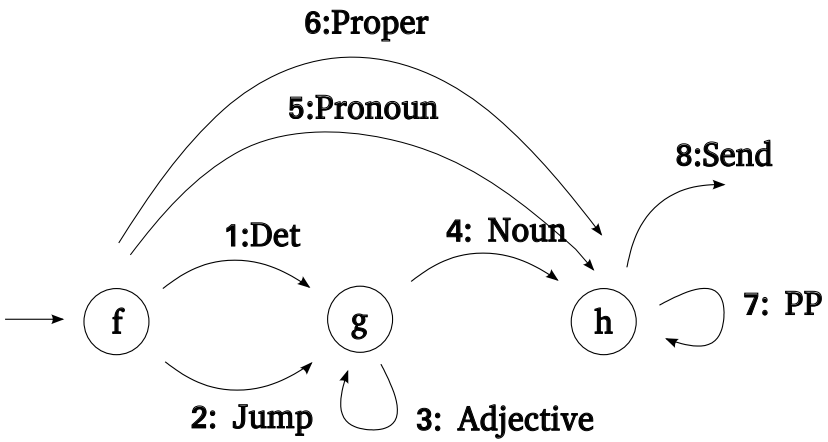
RTN: limitations

- The transitions only depend on categories (local/limited expressiveness)
  - Context free language
- Recognize but does not analyze
- Inefficient because of backtracking

Augmented Transition Networks (ATN) (Woods, 1970)

- ATN = RTN with operations on the edges and use of variables
- Operations
  - Conditions: Filter transitions among states
  - Actions: Build output structures and transform the recognizer in an analyzer
  - Initializations
- Allow to express contextual constraints

ATN Example (Winograd, 1983)



Feature Dimensions: Number: Singular, Plural: default -empty-

Initializations, Conditions and Actions:

NP-1: $_f$Determiner$_g$

A: Set Number to the Number of *

NP-4: $_g$Noun$_h$

C: Number is empty or Number is the Number of *

A: Set Number to the Number of *

NP-5: $_f$Pronoun$_h$

A: Set Number to the Number of *

NP-6: $_f$Proper$_h$

A: Set Number to the Number of *

ATN: limitations

- Adequate for top-down analysis but difficult for bottom-up analysis or hybrid analysis
- Operation redundancy because of backtracking
  - Inefficient
- Notational expressiveness problems:
  - Grammar is mixed with actions

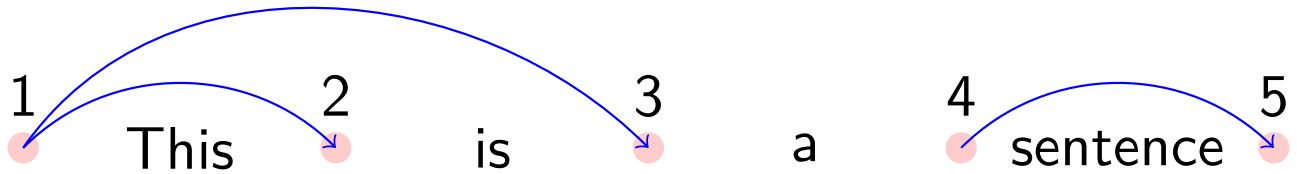
Charts (Kay, 1973, 1980)

- **Chart** = Directed graph that is built dynamically and incrementally while the analysis is performed
- The **nodes** correspond to the beginning, the end of the sentence and the separations among the words (N+1 nodes)

- The redundancies during the analysis are reduced (Less Backtracking) by memorizing partially built structures
- **Problems:** space, construction time, only stores well formed components

Charts (elements)

- The edges appear dynamically
- An edge from position i to j ($j \geq i$) includes all the words that are between position i and j.
- Edges can be
  - **active** = goals or hypothesis to complete
  - **inactive** = components totally analyzed



Charts: syntax (1)

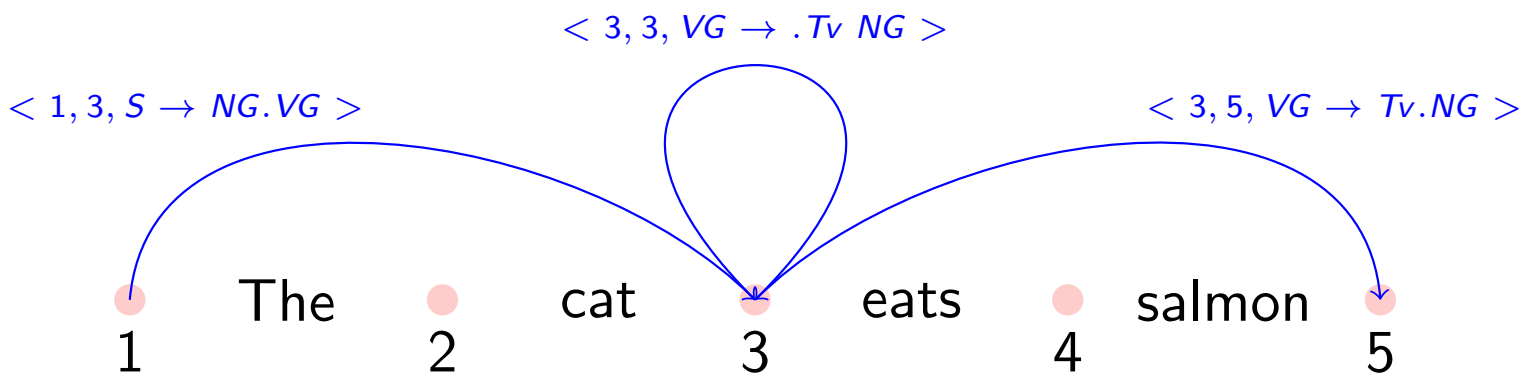
- **Dotted rule:** rule of the grammar that has a dot inside the right side of the rule
- For example, from the rule $A \rightarrow BCD$ can be derived the following dotted rules:

$$
\begin{array}{ll}
A \rightarrow .BCD & \text{(corresponding to an active edge)} \\
A \rightarrow B.CD & \text{(corresponding to an active edge)} \\
A \rightarrow BC.D & \text{(corresponding to an active edge)} \\
A \rightarrow BCD. & \text{(corresponding to an inactive edge)}
\end{array}
$$

Charts: syntax (2)

- **Edge** of a chart $< i, j, X \rightarrow a.b >$

i, j :	Source and destination nodes
$X \rightarrow ab$	rule of the grammar
$X \rightarrow a.b$	Dotted Rule



Basic Combination Rule

The **basic combination rule** and the **ascendant** or **descendant** rules (or both at the same time) give the analysis method

Active edge: $< i, j, A \rightarrow a.Bb >$

Inactive edge: $< j, k, B \rightarrow g. >$

$$\Downarrow$$

Result: $< i, k, A \longrightarrow aB.b >$

Ascendant Strategy

Basic Rule

Each time that an inactive edge is added to the Chart $< i, j, A \rightarrow a. >$, a new active edge $< i, i, B \rightarrow .Ab >$ has to be added to its leftmost vertice for each rules $B \rightarrow Ab$ in the grammar

- **Initialization:** add all the inactive edges corresponding to the lexical categories (terminals).
  Ex: $< 1, 2, Det \rightarrow the. >$

Descendant Strategy

Basic Rule

Each time that an active edge is added to the Chart $< i, j, A \rightarrow a.Bb >$, for each rule $B \rightarrow b$ of the grammar an active edge $< j, j, B \rightarrow .b >$ has to be added to its rightmost vertice

- **Initialization:** the same than before but adding the active edge corresponding to the rule that recognizes a sentence.
  Ex: $< 1, 1, S \rightarrow .NG\ VG >$

Unification formalisms. Logic Grammars

- Unification based formalisms $\subset$ Logic Grammars
- Usual implementation language: Prolog Characteristics
  - **Unification** as basic composition mechanism among constituents
  - **Syntagmatic approach** as basic way of grammatical description

Notation

- Assertions (facts):
  `man (X)` $\leftarrow$
- Conditions (rules) (`Consequent` $\leftarrow$ `antecedent`)
  `mortal (X)` $\leftarrow$ `home (X)`
- Negations (Existential queries)
  $\leftarrow$ `immortal(X)`
- Contradictions
  $\square$

Grammatical analysis as automatic theorem proving

- The definition of the grammar and the lexicon as Horn clauses allow to apply resolution and reasoning by refutation as analysis mechanism

(1) sentence (X,Y)	← nomg(X,Z), verg(Z,Y)	**Grammar**
(2) nomg(X,Y)	← det(X,Z), name(Z,Y)	**Rules**
(3) verg(X,Y)	← ver(X,Y)	
(4) det(X,Y)	← the(X,Y)	**Lexicon**
(5) name(X,Y)	← dog(X,Y)	
(6) ver(X,Y)	← barks(X,Y)	

Example (1)

(7) the(1,2) ←
(8) dog(2,3) ←
(9) barks(3,4) ←

- THEOREM to prove `sentence (1,4)`
- Reasoning by refutation we have to negate...
  ← **sentence(1,4)**
- ...and by resolution prove
  □ (a contradiction)

Example (2)

`sentence(1,4)` $\leftarrow$

(R1) (X=1, Y = 4) by unification

$\leftarrow$ `nomg (1,Z), verg(Z,4)`

(R2) and (R4) applied to nomg(1,Z) and det(1,U)

$\leftarrow$ `det(1,U), name(U,Z), verg(Z,4)`

$\leftarrow$ `the(1,2), name(U,Z), ver(gZ,4)`

(R7) (U = 2)

$\leftarrow$ `name(2,Z), verg(Z,4)`

(R5) and (R8) (Z = 3)

$\leftarrow$ `dog(2,3), verg(3,4)`

$\leftarrow$ `verg(3,4)`

(R3) and (R6) and(R9)

$\leftarrow$ `ver(3,4)`

$\leftarrow$ `barks(3,4)`

$\square$

Example (3)


```
sentence(1,4) ←
← nomg (1,Z), verg(Z,4)
← det(1,U),name(U,Z), verg(Z,4)
← the(1,2), name(U,Z), ver(gZ,4)
← name(2,Z), verg(Z,4)
← dog(2,3), verg(3,4)
← verg(3,4)
← ver(3,4)
← barks(3,4)
□
```

Direct interpretation in Prolog !!

Unification analyzers

- Logic formalisms
  - Expressiveness and treatment
  - Definite Clause Grammars (DCG)
- Prolog as analyzer
- Unification
  - Term representation
  - Unification algorithm

Definite Clause Grammars (DCG)

- The definite clause grammars allow to develop logic grammars as PROLOG programs
- PROLOG is a rule language that uses backward reasoning as resolution method
- A special syntax is defined to hide the treatment of sentences and to differentiate the elements of the grammar from the procedures used to augment the context free grammar
- The rules use variables to pass on information and to check the constraints that the grammar needs

Definite Clause Grammars (Syntax)

- A grammar rule has the following syntax
  `left --> right1, right2, right3, ..., rightN`
- Each symbol of the grammar can have variables, they are used for different purposes (pass or get information from other rules, build a result, ...)
- The symbols from the input are consumed by using the operator square brackets and giving a list of variables and/or constants that have to be unified with the input
  `aaa --> [W], bbb`
- PROLOG code can be included using braces
  `aaa(W) --> [W], bbb(W), {number(W)}`
- To execute a DCG the main symbol of the grammar is invoked with two parameters, a list with the words in the sentence and an empty list
  `sentence([the,cat,eats,a,fish],[])`

Definite Clause Grammars (Example)

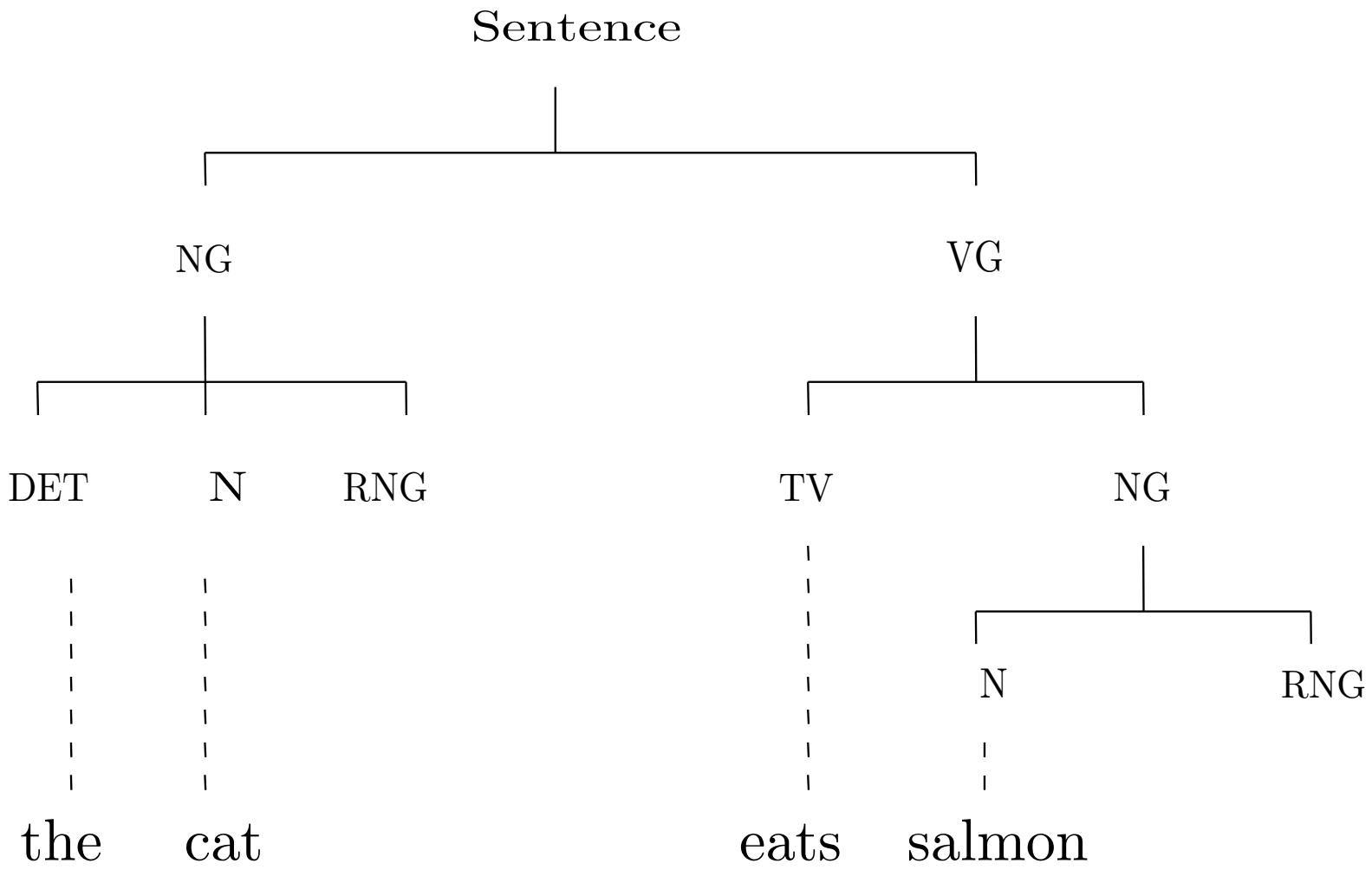
```
analisis(X,Y):- asercion(X,Y).

asercion --> sn, verb,compl.
compl --> [].
compl --> prep,sn.
compl--> sn.
sn--> npr.
sn--> det,n.

verb--> [W],{verbo(W)}.
npr--> [W],{npropio(W)}.
n--> [W],{nombre(W)}.
det--> [W],{determ(W)}.
prep--> [W],{prepo(W)}.
```

```
npropio(clara).
npropio(maria).
npropio(barcelona).
nombre(hombre).
nombre(profesor).
nombre(libro).
determ(un).
determ(el).
verbo(esta).
verbo(rie).
verbo(piensa).
verbo(habla).
verbo(lee).
prepo(en).
prepo(con).
prepo(de).
```

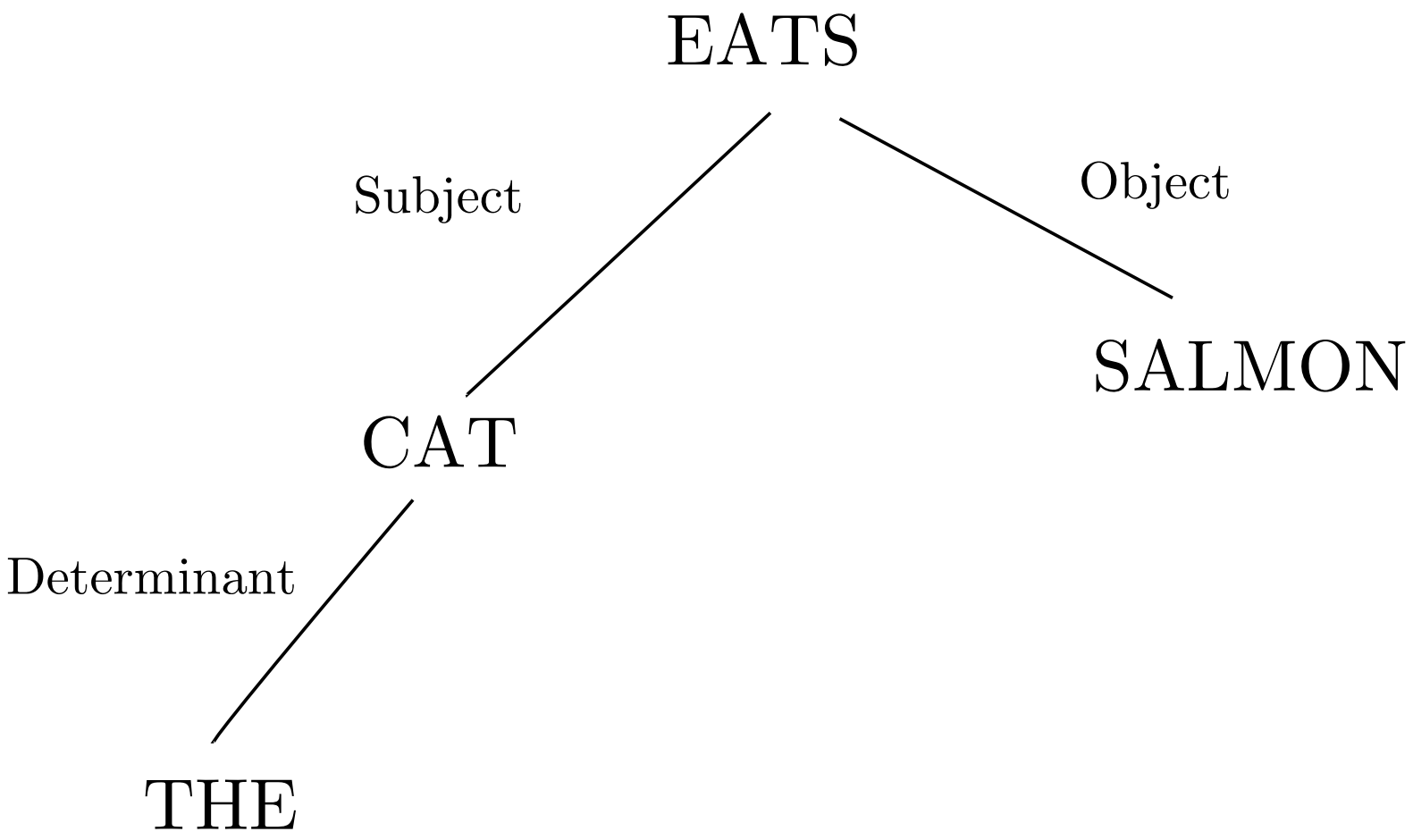
Result of the syntactical analysis

- Tree of analysis
  - Structure of components
- Structure of dependencies
- Actant model
  - Semantic network
- Logical form

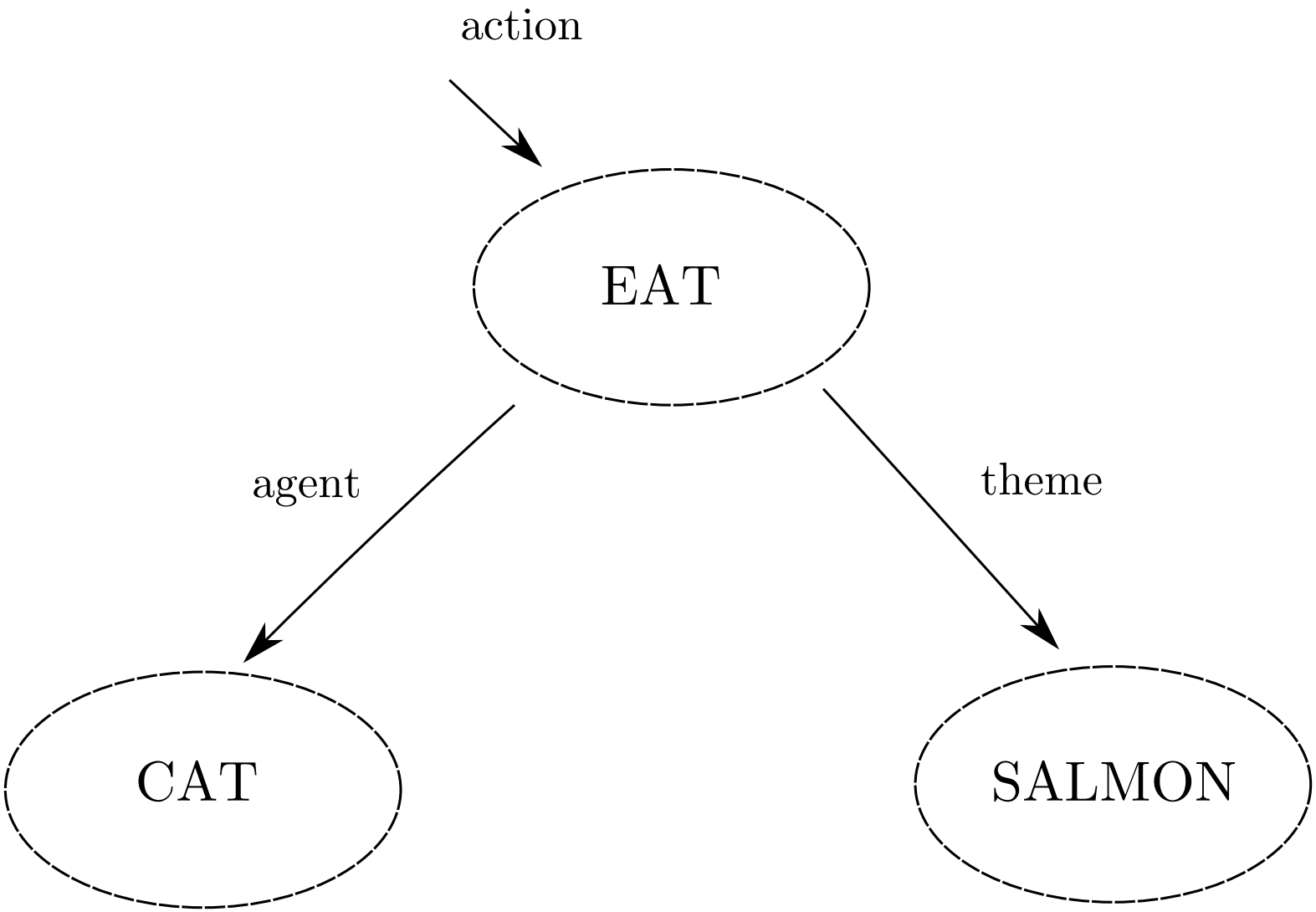
Structure of components



Structure of dependencies



Actant model

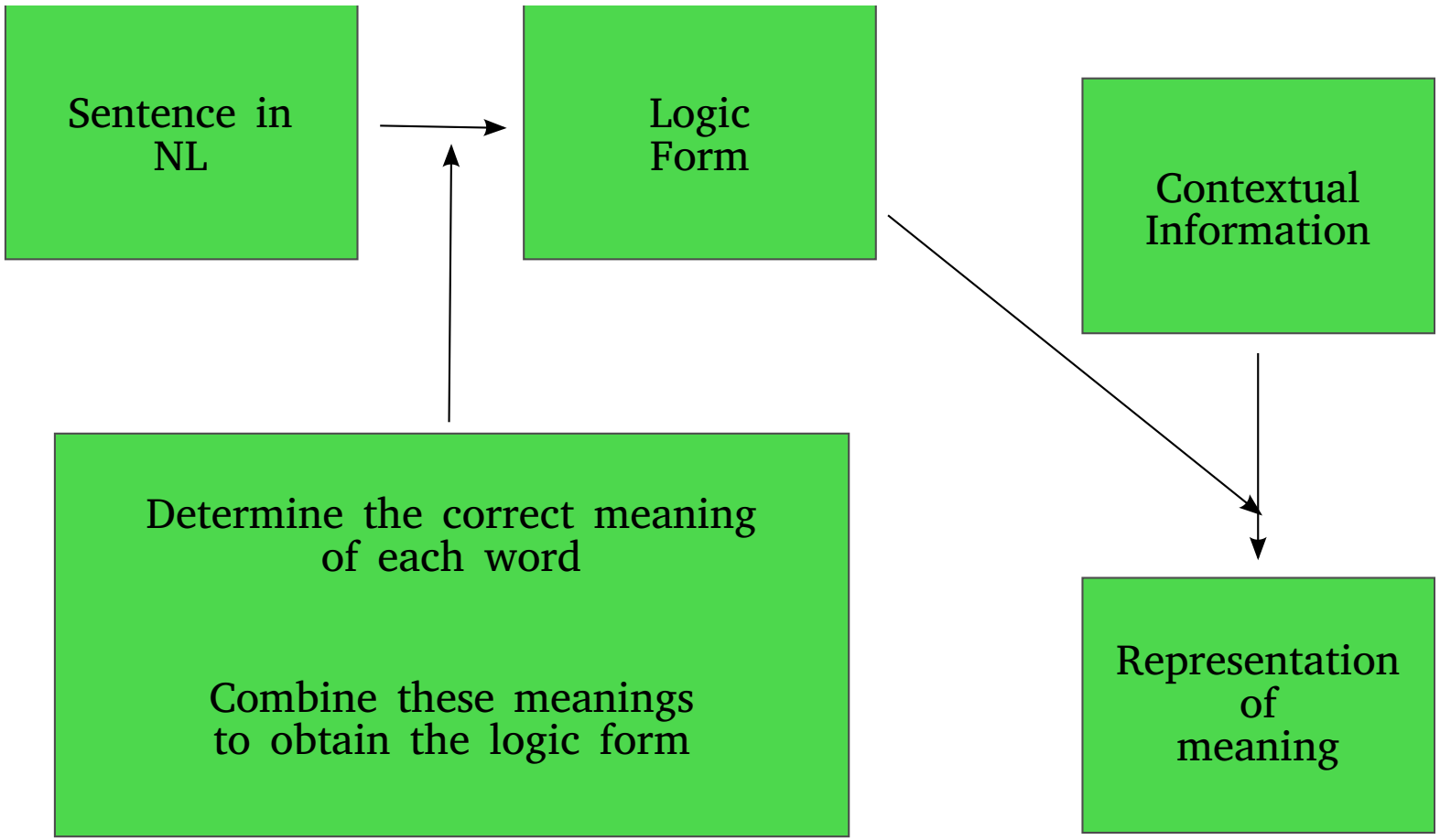


Logic Form

$$\exists x(Cat(x) \wedge \exists y(Salmon(y) \wedge Eat(x, y)))$$

Semantics

- Semantics studies the meaning of sentences
- Semantic interpretation (SI) is the process of extraction of this meaning



Properties of the Semantic Model

- Compositive semantics: The representation of the semantic of an object has to be obtained from the semantic representation of its components:

$$\text{SI} = f(\text{SI}(\text{Syntactical Components}))$$

- Based on a theory
- Definition of a Semantic Representation System
- Syntax-Semantics interface
- SI has to be robust to ambiguity
- SI has to be able to treat complex things like: *Quantification*, *modality*, *negation*

Two problems

- Meaning representation
- Semantic interpretation

Representation of meaning (1)

input:

Who leads the PSOE?

Logic form:

```
(question
    (referent (X))
        (X instance (X, person)
            (el1 (Y instancia(Y, political_party) name(Y, "PSOE"))
                (Z instance(Z, lead)
                present(Z)
                value_prop(Z, agent, X)
                value_prop(Z, pacient, Y)))))
```

Representation of meaning (2)

- This formula has four different informations:
  - Logical structure
  - Conceptual content (semantic)
  - Acts of speech
  - Pragmatic annotations
- The formalism should have enough expressive capacity to guarantee the description of all four kinds of information

Representation of meaning (3)

- Representation formalisms:
  - Predicate calculus (Modal logic, Temporal Logic, Description Logic, ...)
    - Usually in clausal form
  - Other logic formalisms
  - Semantic networks
  - Frames

Representations based on Logic

- A vocabulary of **predicates** with their arity (number of arguments and sometimes their types).
- A vocabulary of **constants** and **variables**.
- A set of logic **connectives**.
- A vocabulary of **functions** with their arity.
- A set of **quantifiers** that will affect the predicates that have to (or can) be quantified.

Semantic interpretation

- Interaction with the syntactical analysis:
  - Sequential process (1): syntax | semantics
  - Sequential process (2): {syntax + semantic filter} | semantics
  - Parallel process: {syntax, semantics}
- Methods for obtaining the composition
  - Composition function
  - SI activation

Example of composition function

- Interpretation by lambda evaluations:

$$(lambda(x)(...)) = (\lambda(x)(...))$$

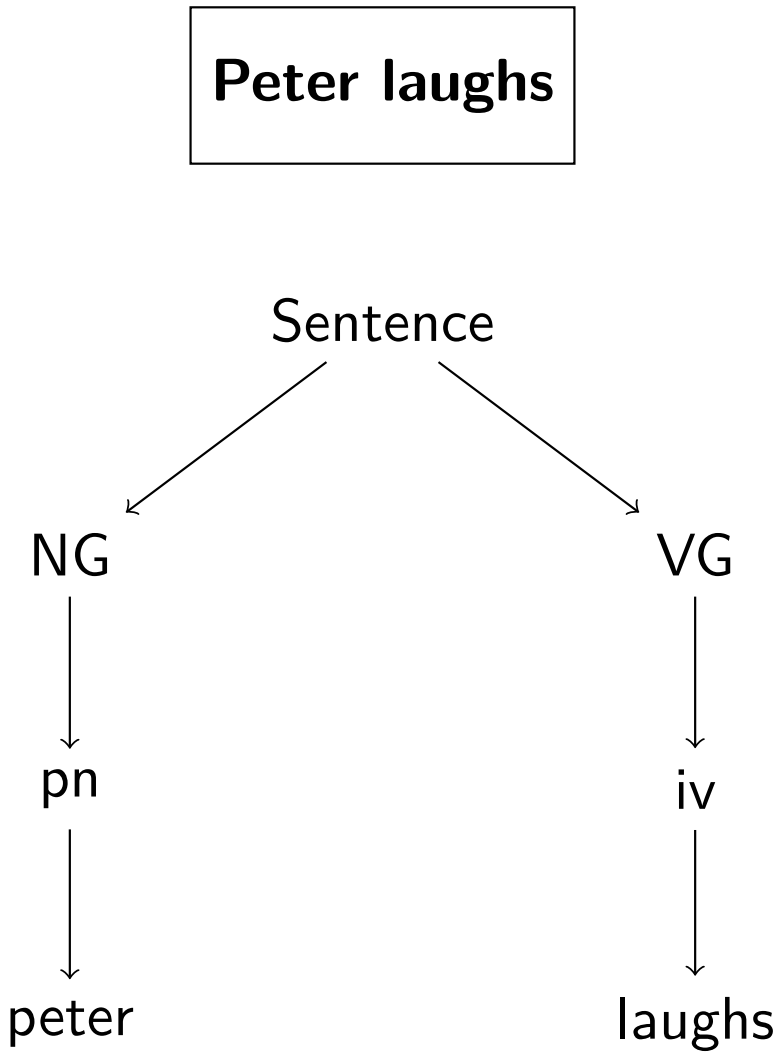
Grammar

Sentence	$\leftarrow$ NG VG (2 1)
NG	$\leftarrow$ pn (1)
VG	$\leftarrow$ iv (1)
VG	$\leftarrow$ tv NG (1 2)

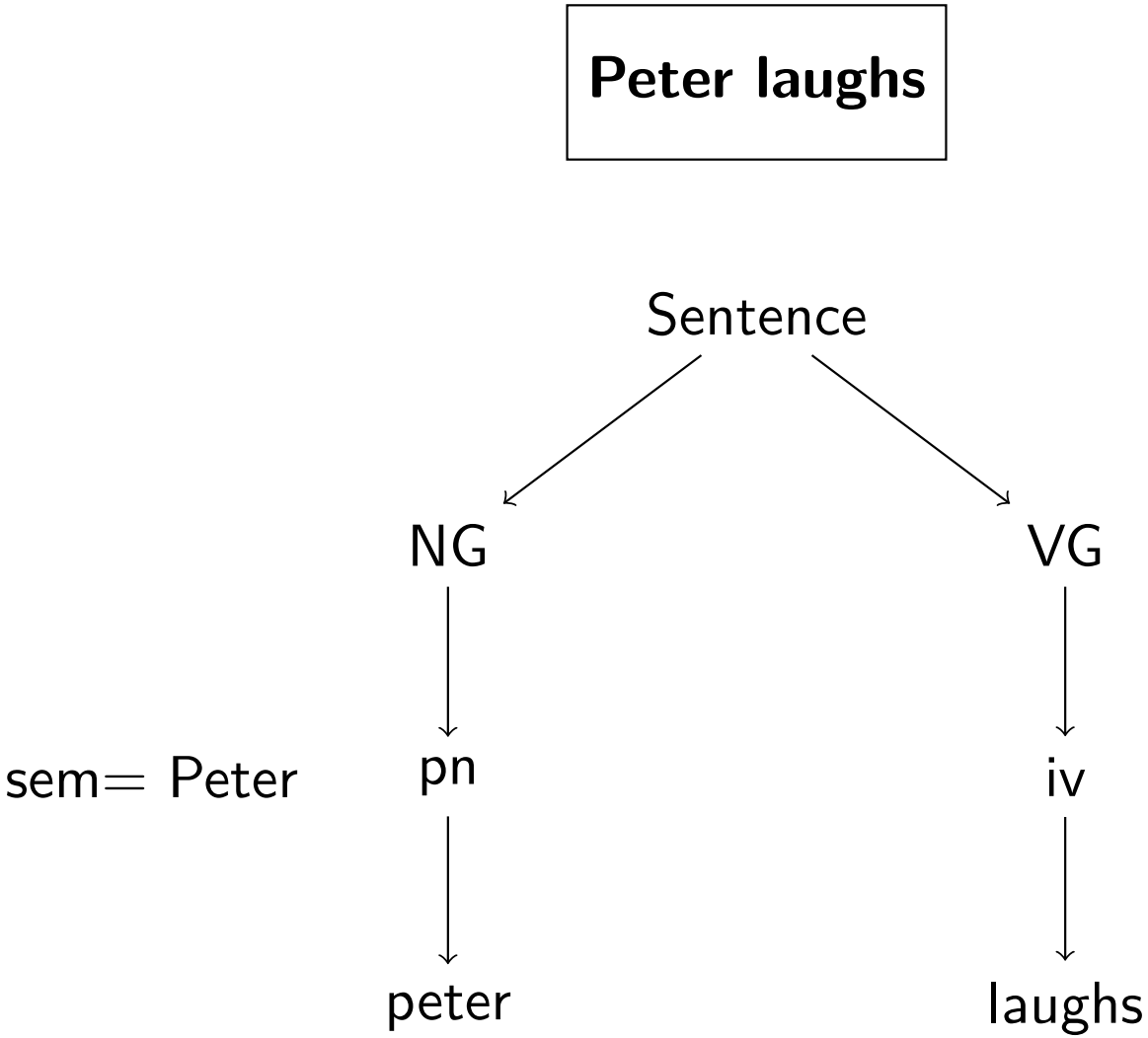
Lexicon

Peter	$\leftarrow$ np, peter
Mary	$\leftarrow$ np, mary
laughs	$\leftarrow$ vi, (λ(x) (laughs(x)))
loves	$\leftarrow$ vt, ((λ(x) (λ(y), loves(y, x))))

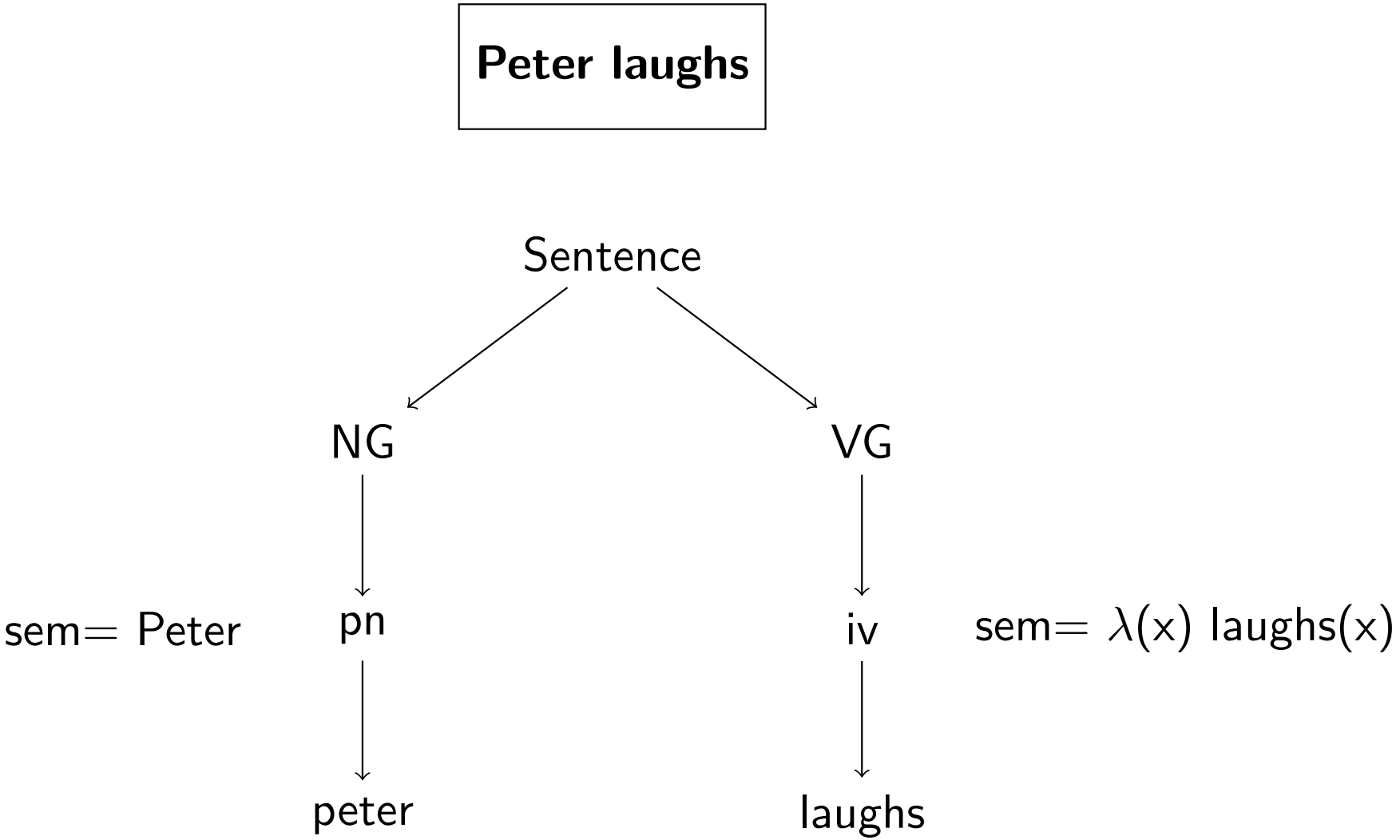
Composition function: example 1



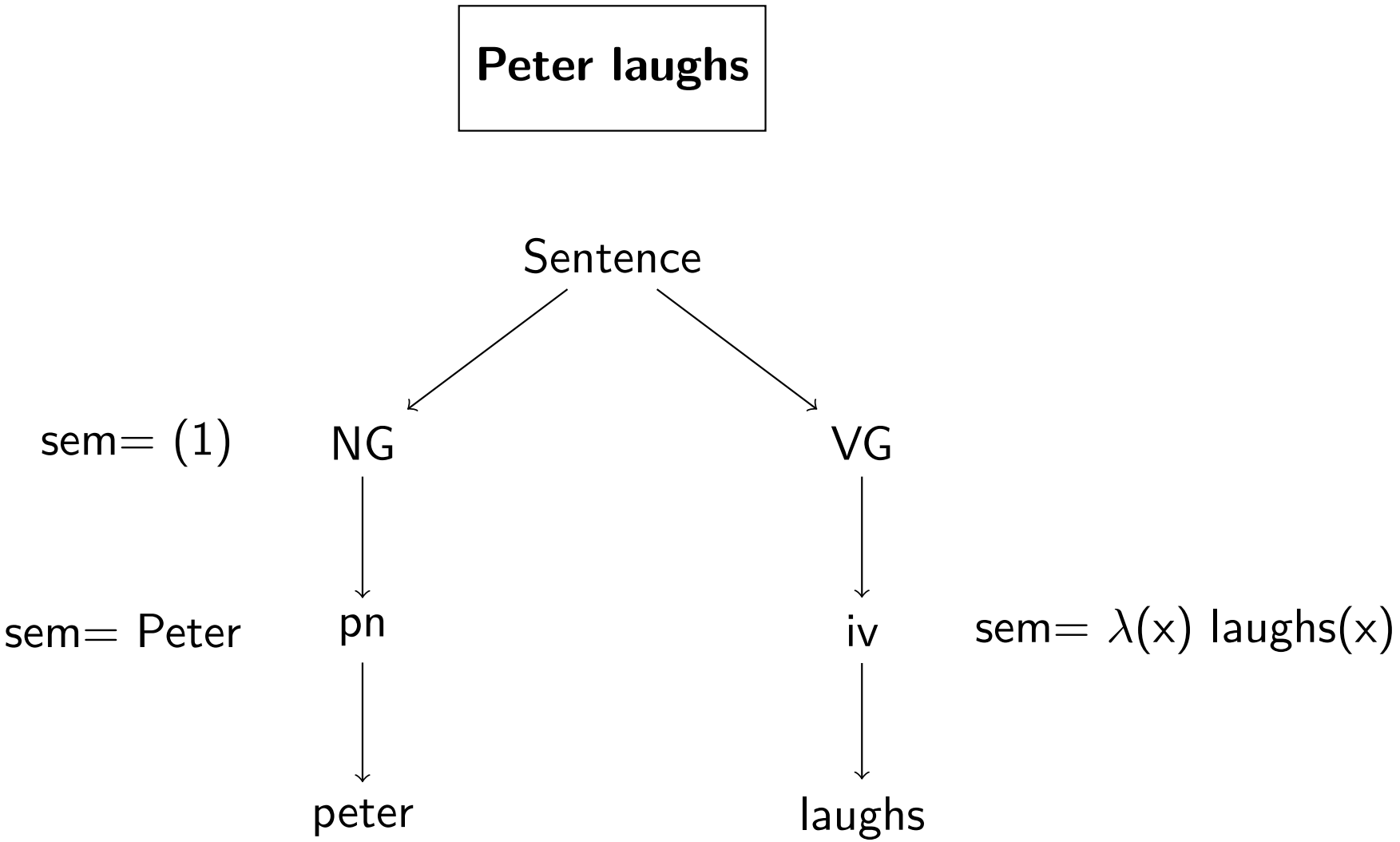
Composition function: example 1



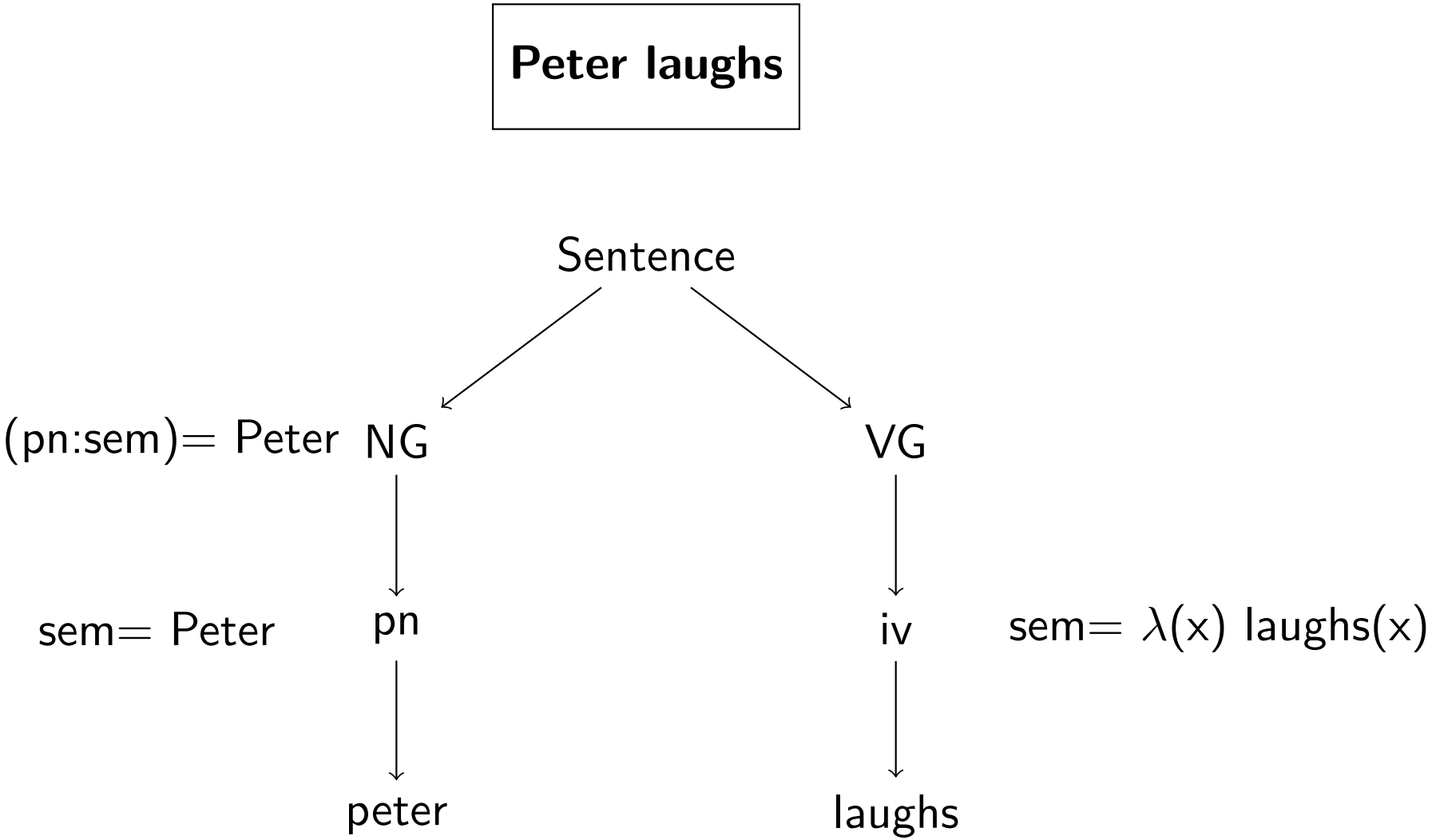
Composition function: example 1



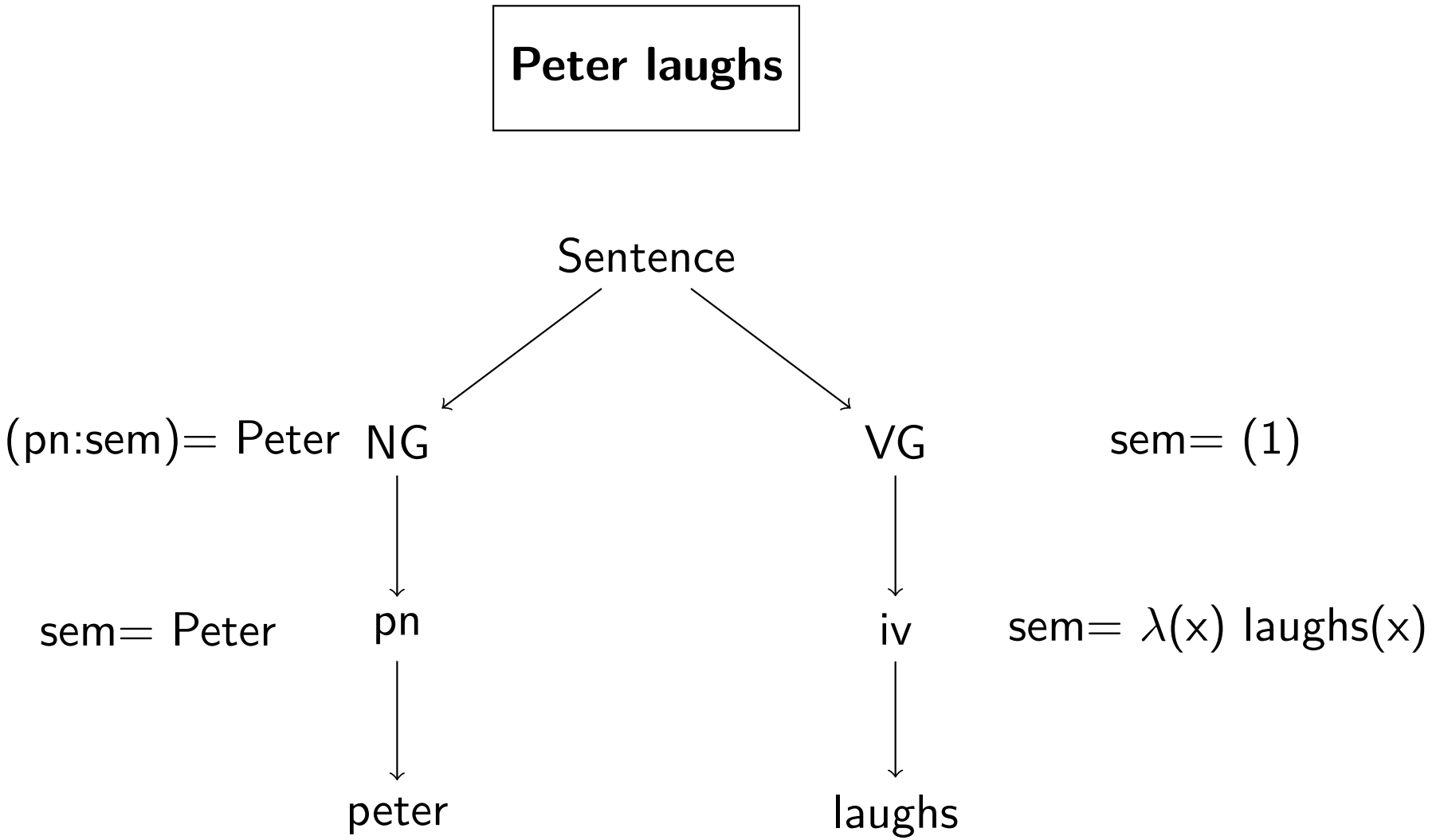
Composition function: example 1



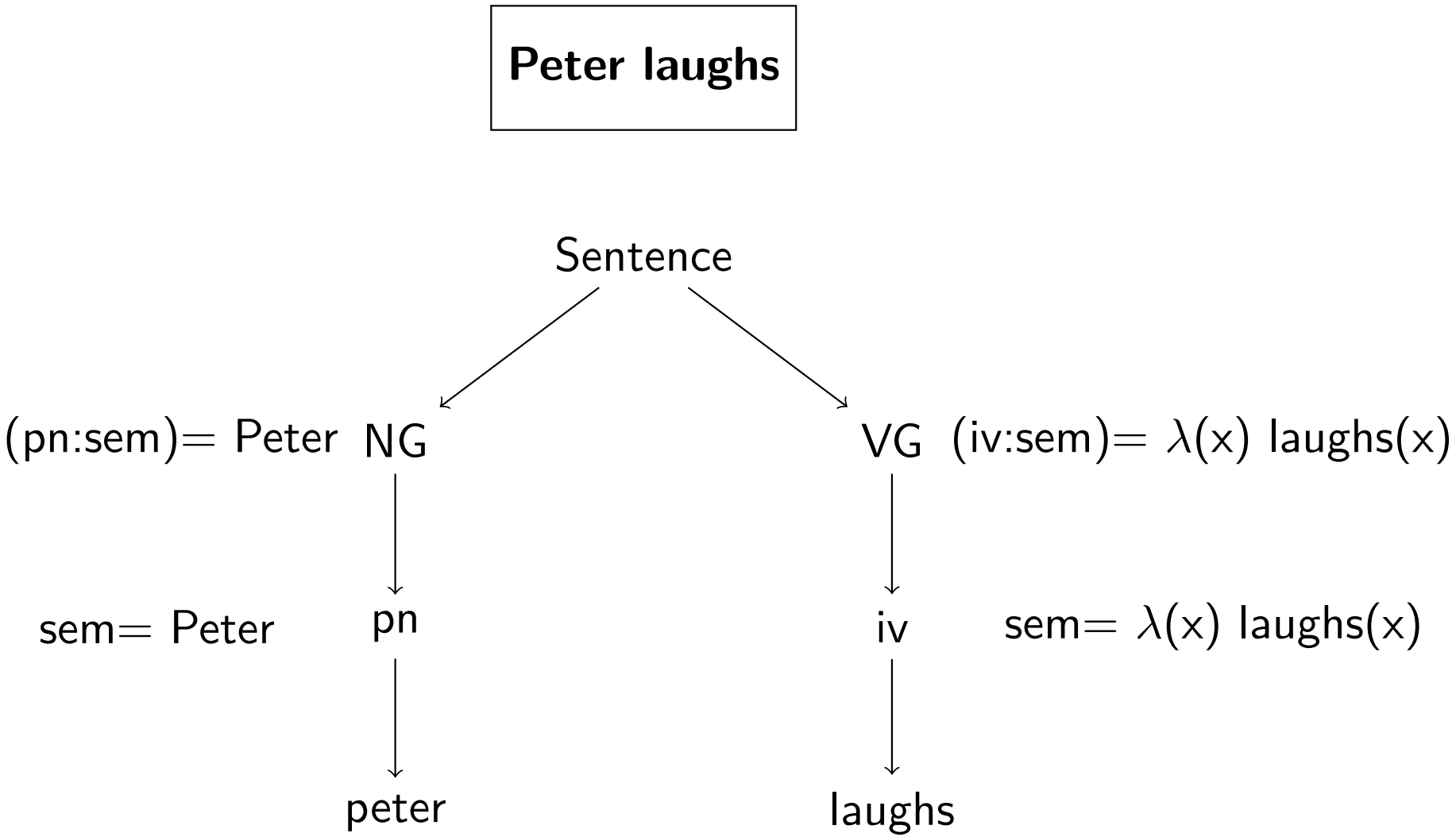
Composition function: example 1



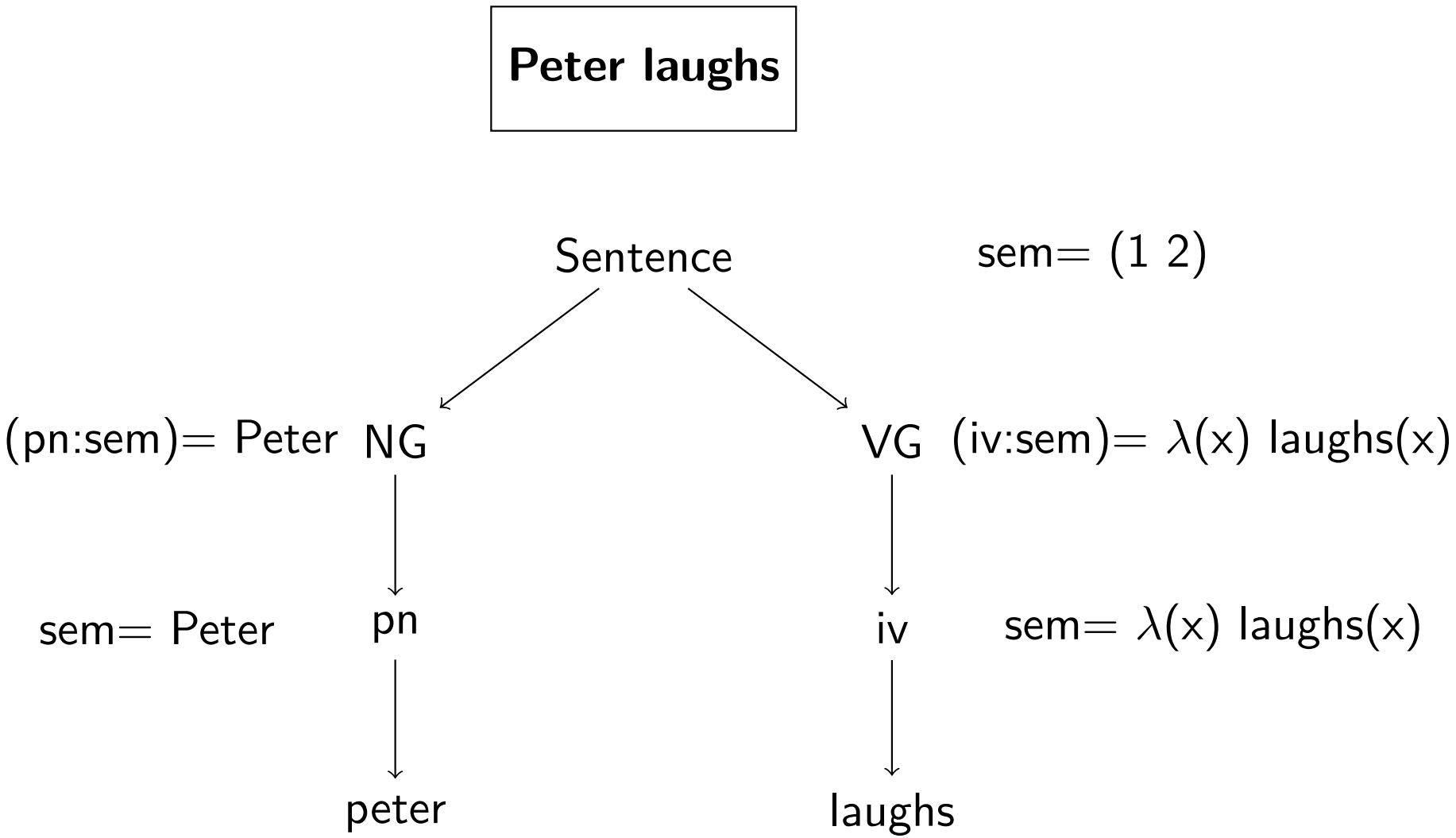
Composition function: example 1



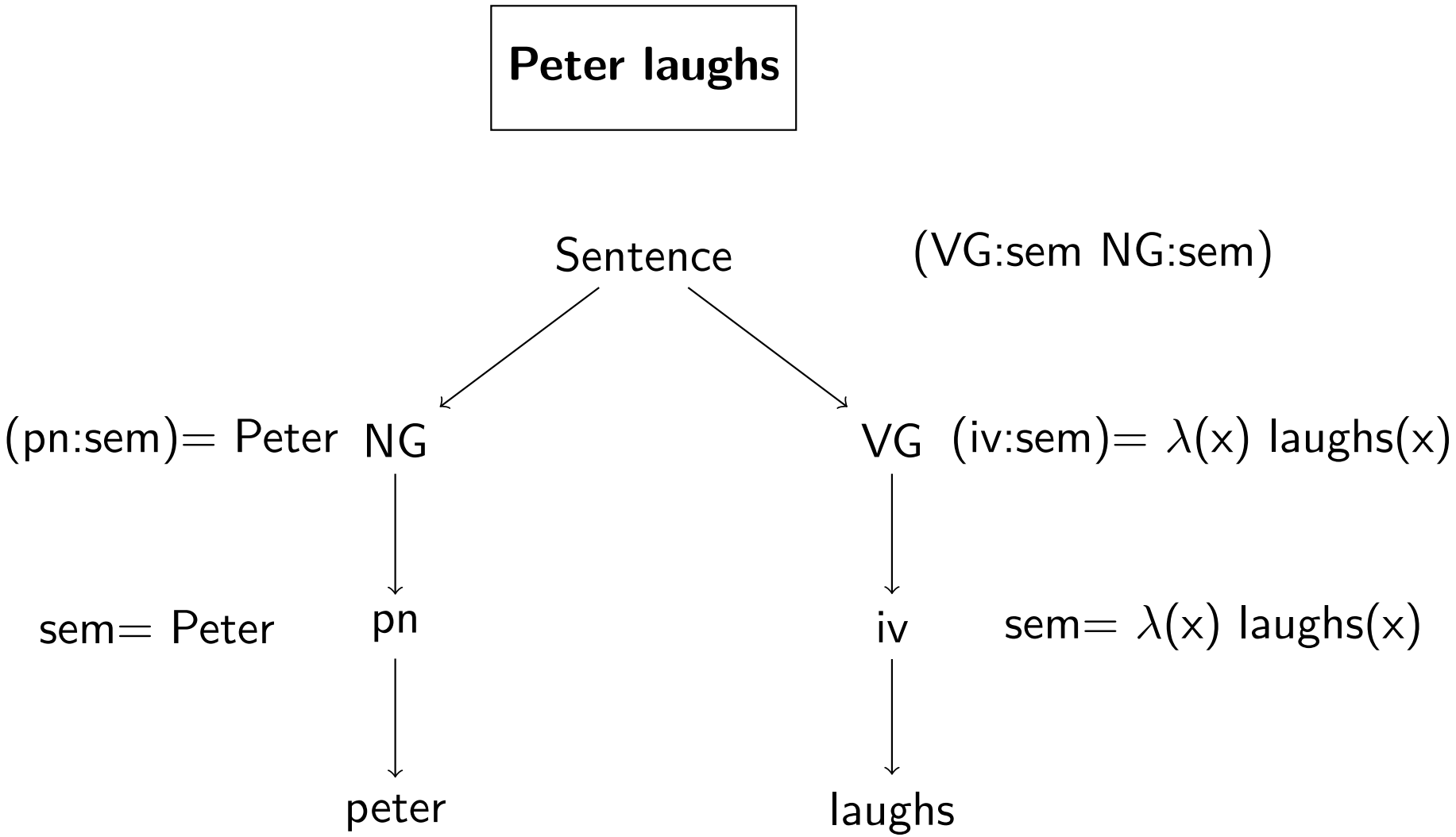
Composition function: example 1



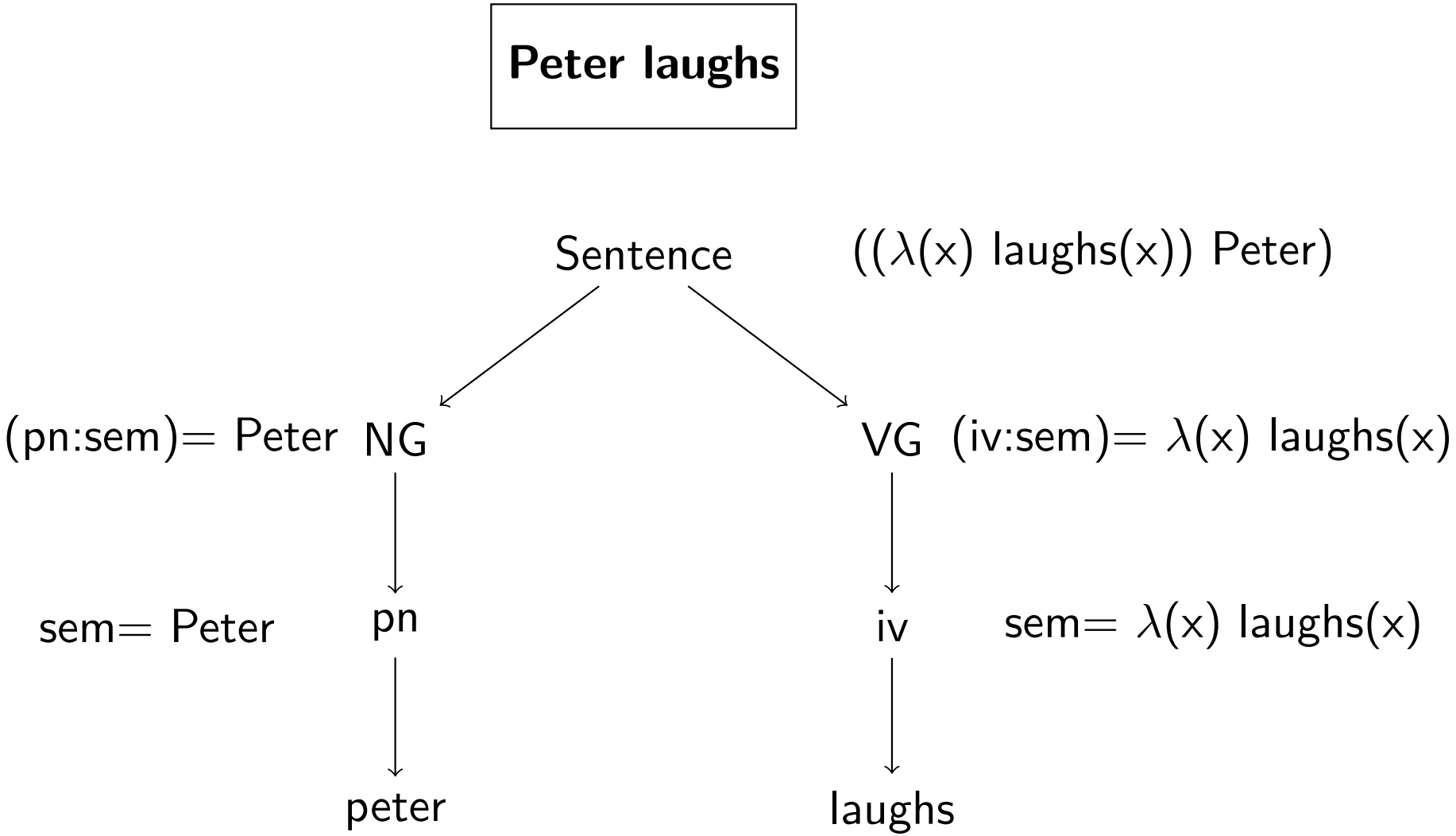
Composition function: example 1



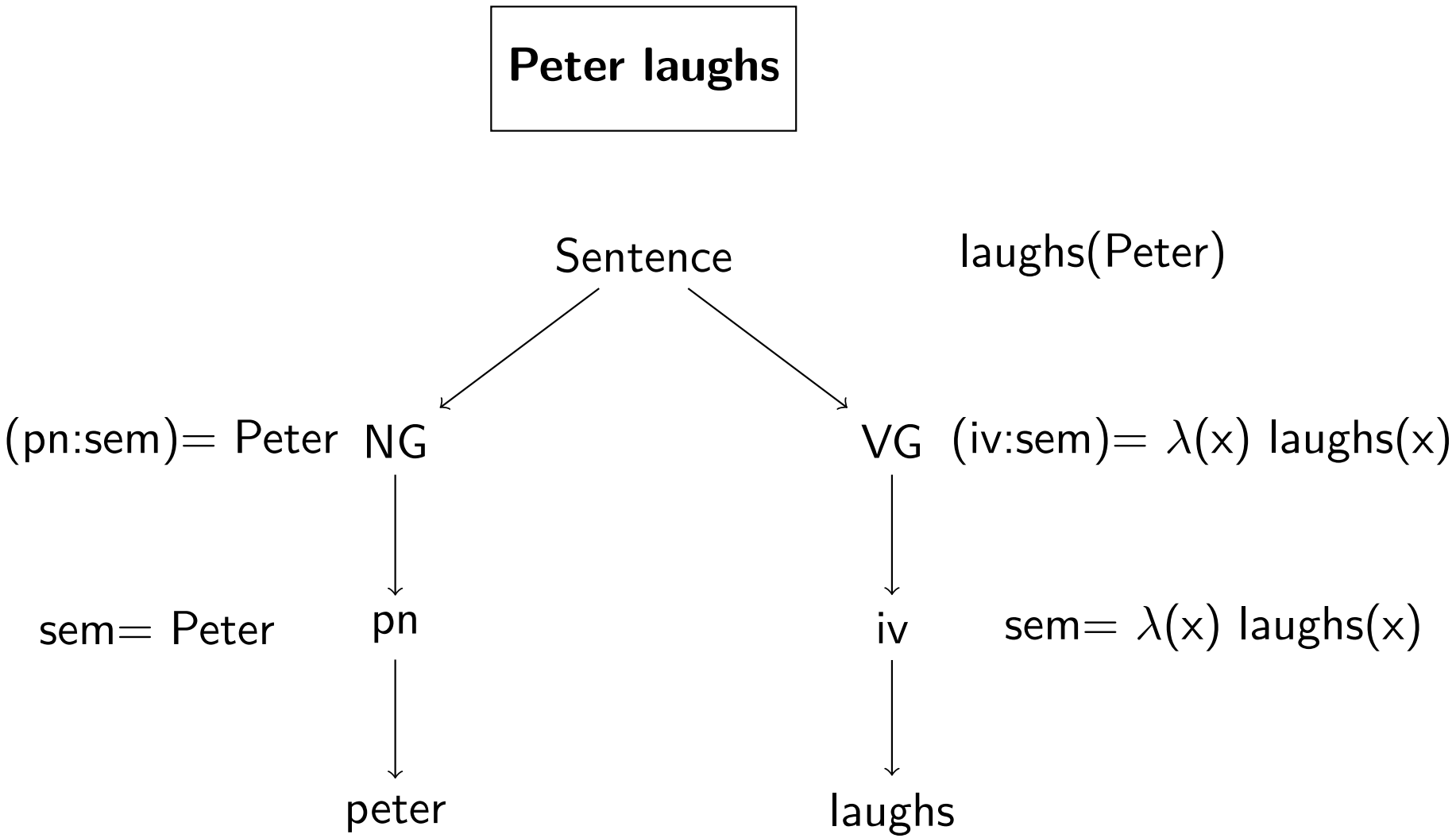
Composition function: example 1



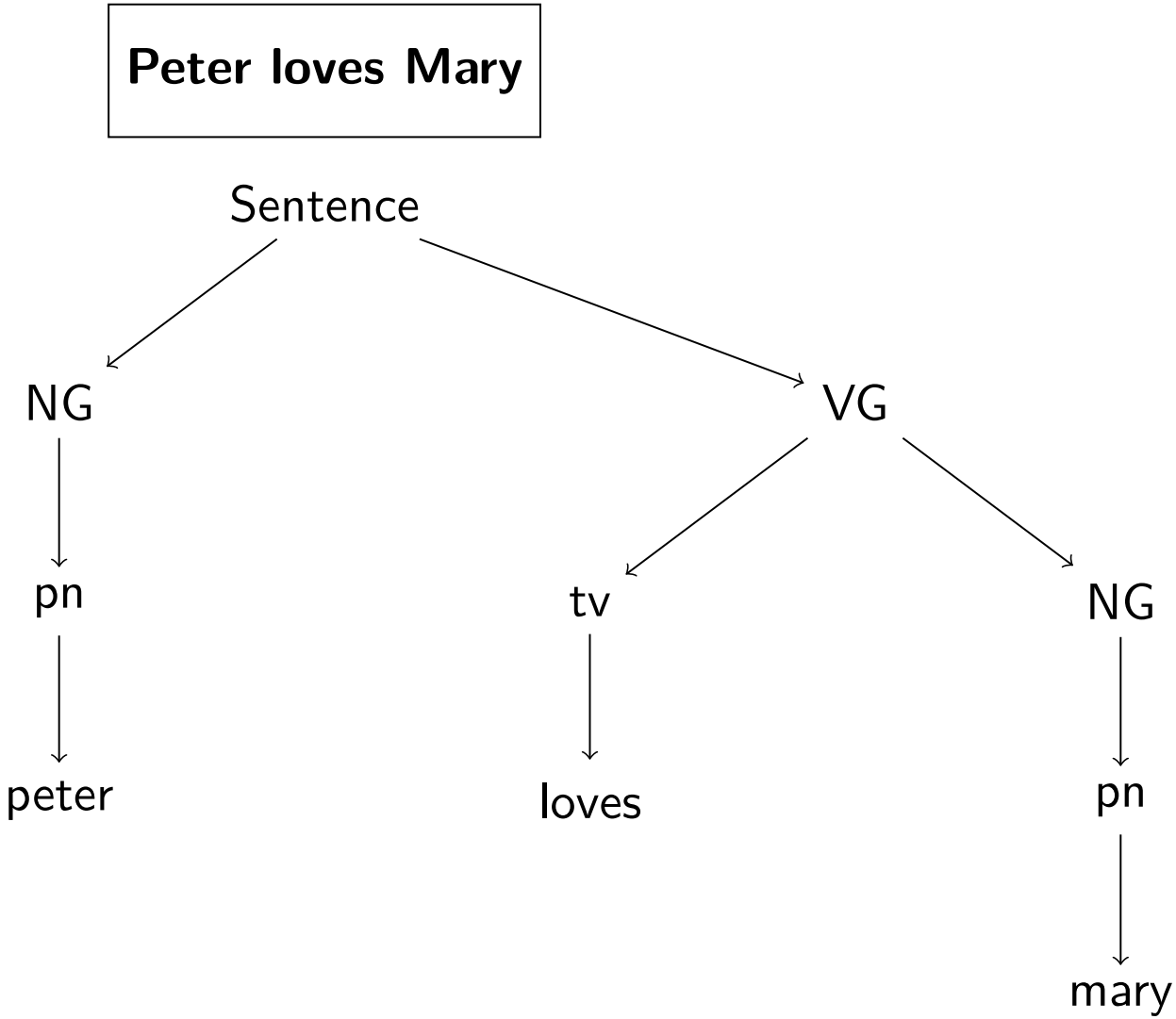
Composition function: example 1



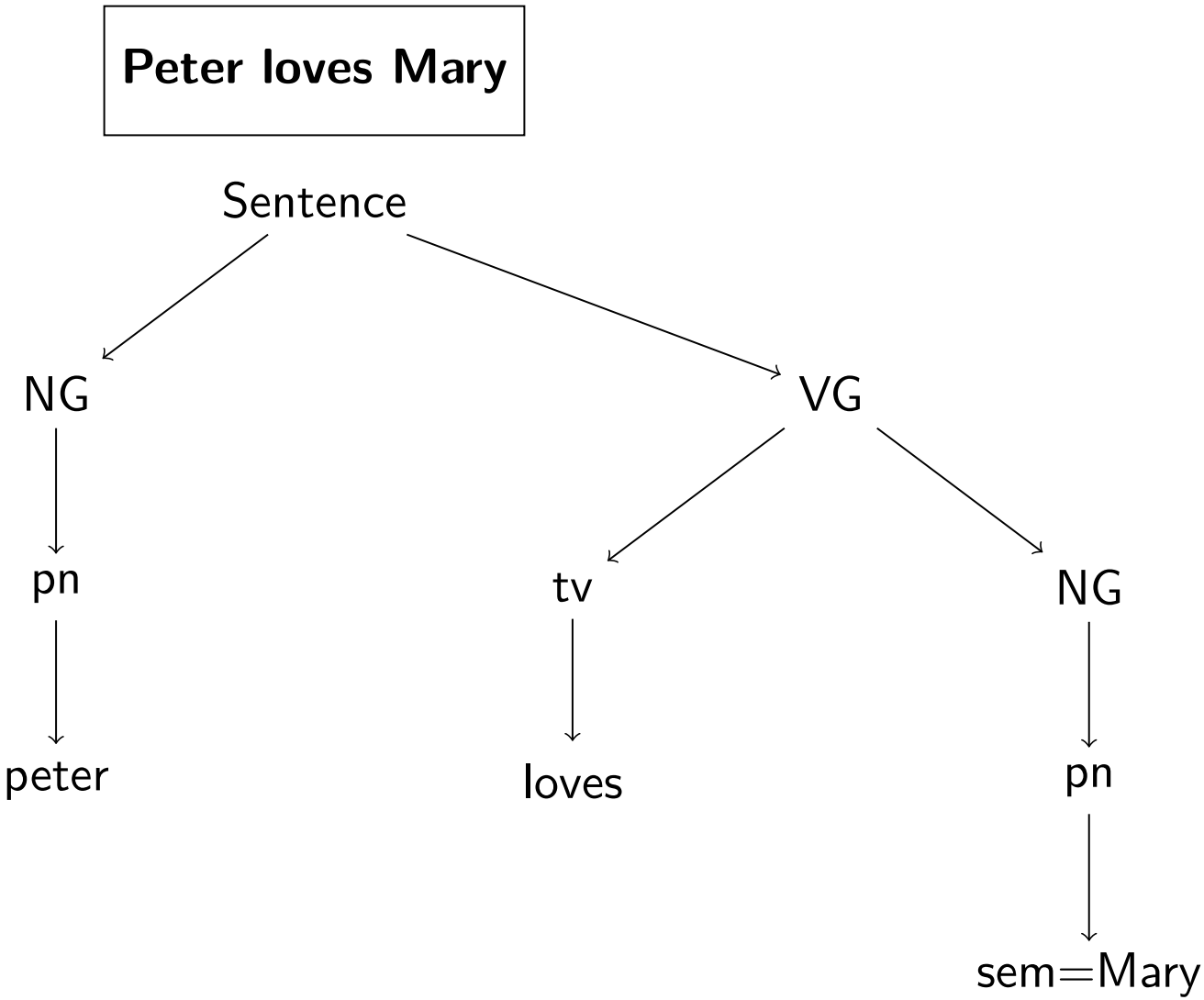
Composition function: example 1



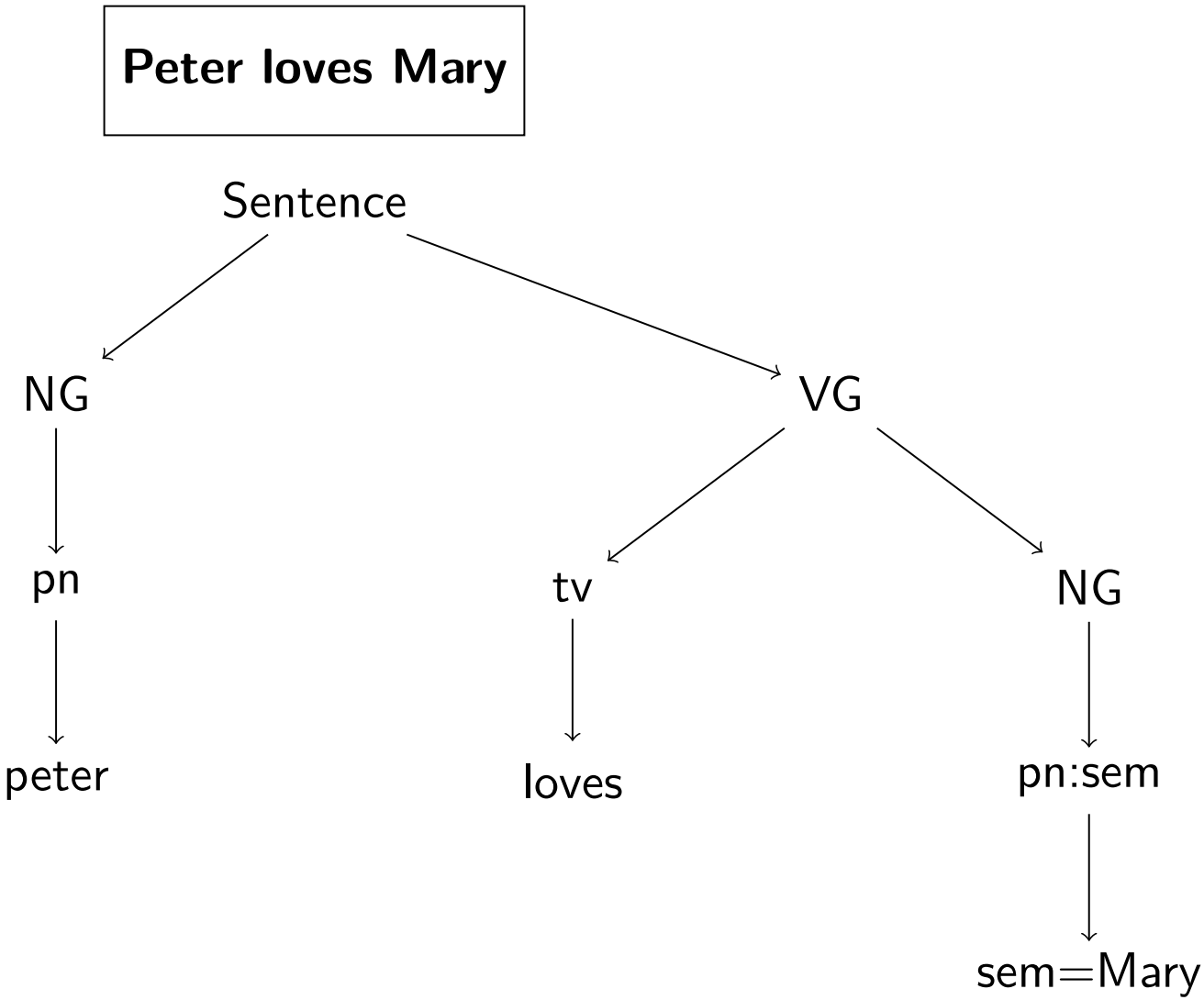
Composition function: example 2



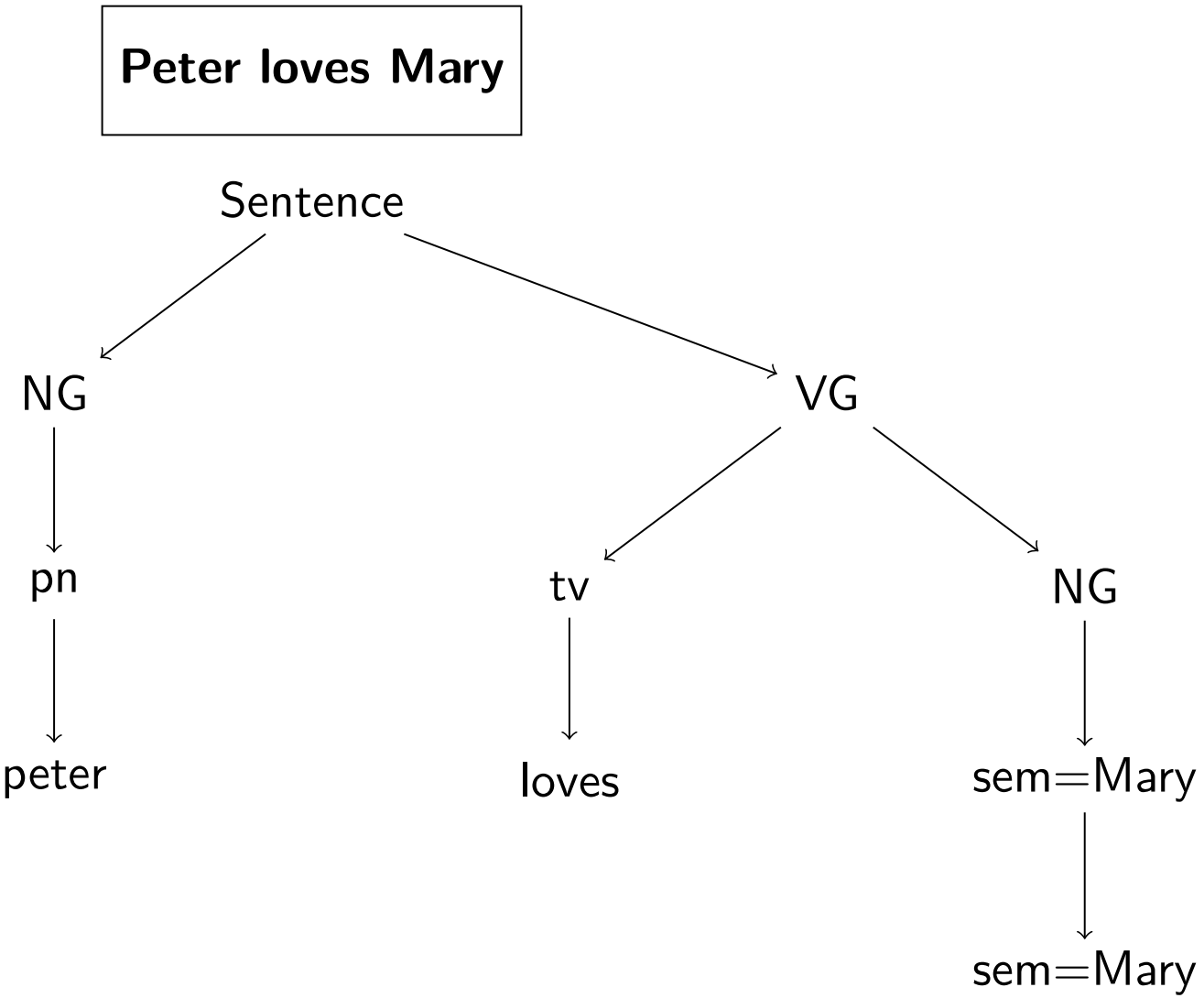
Composition function: example 2



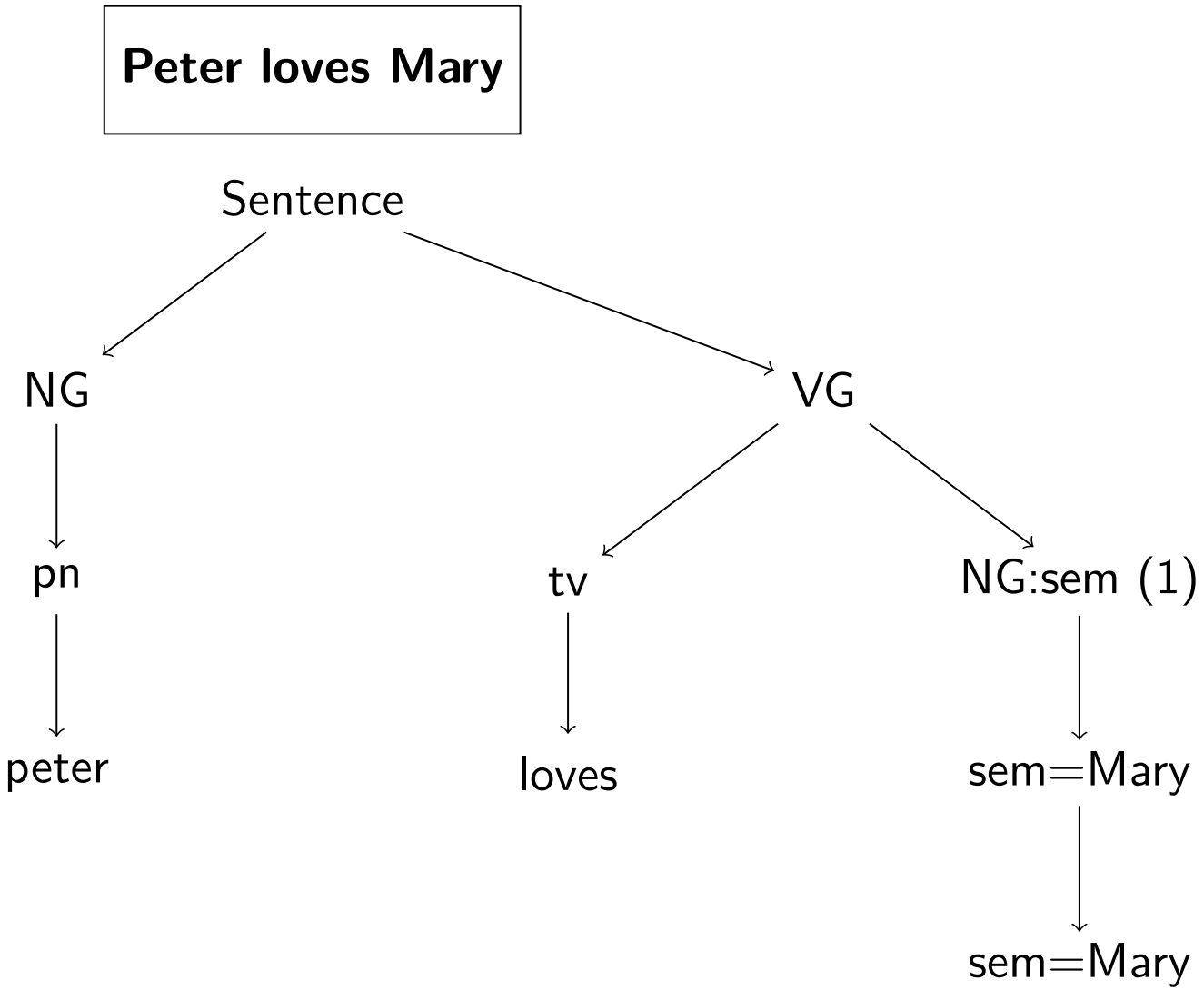
Composition function: example 2



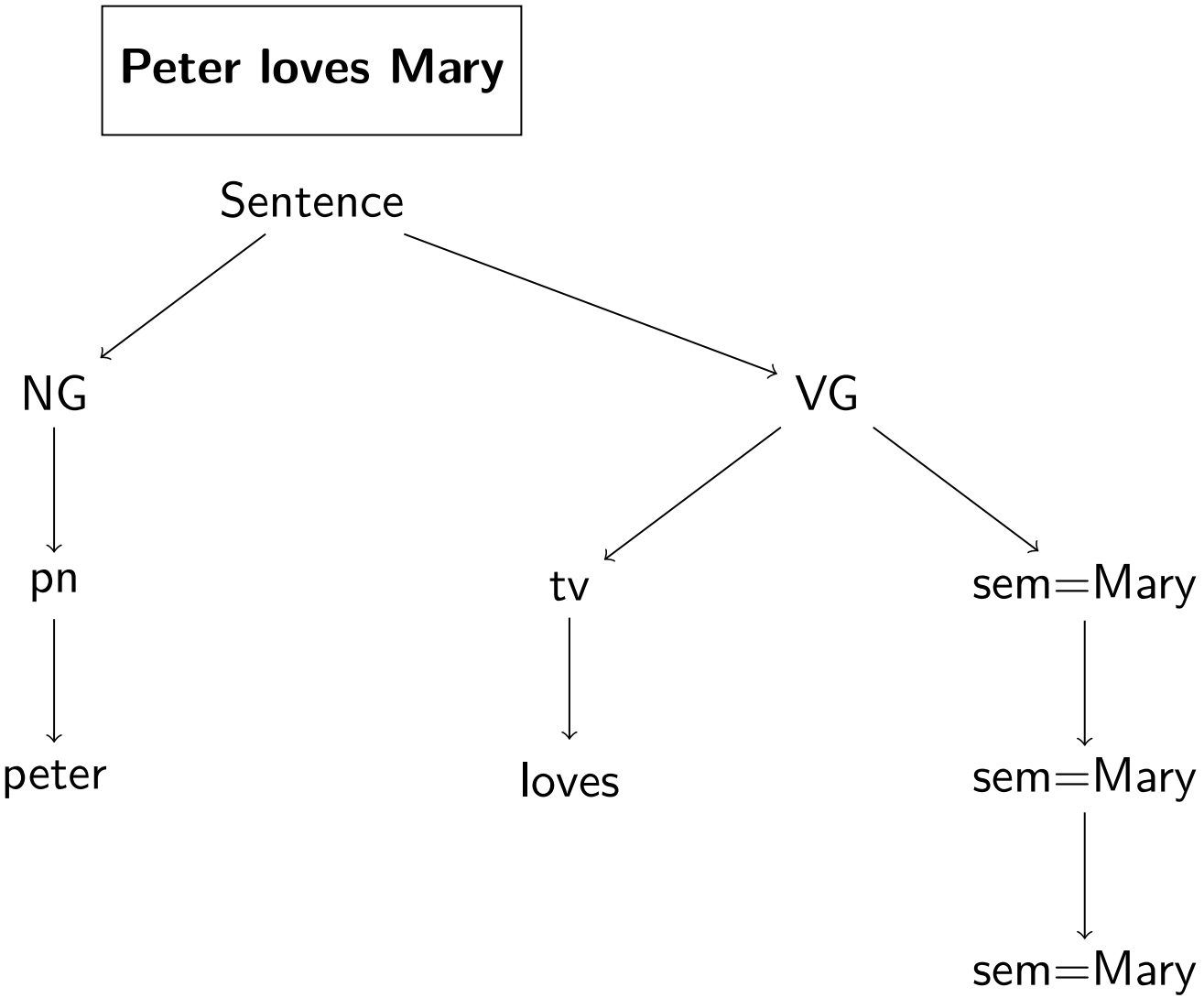
Composition function: example 2



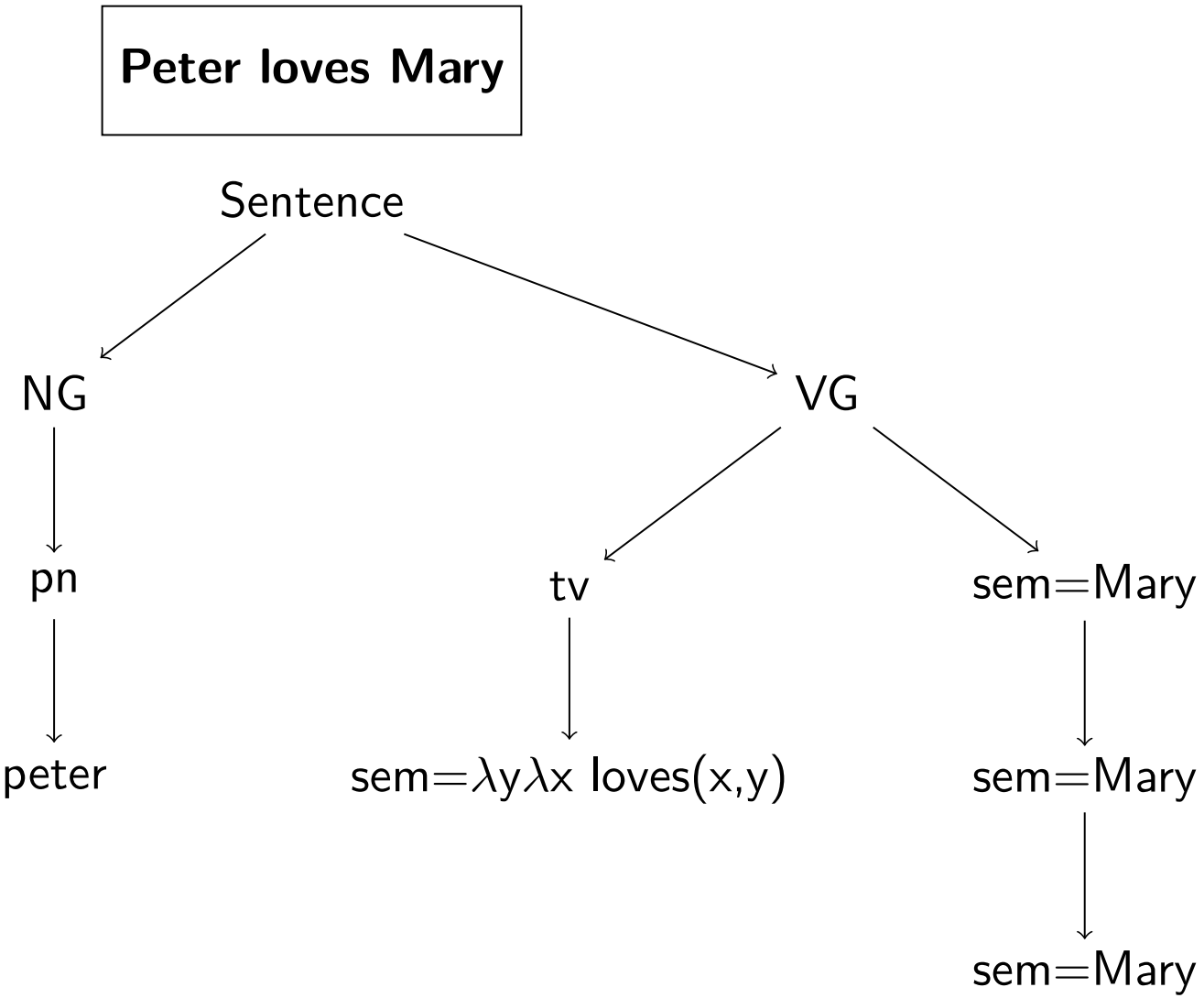
Composition function: example 2



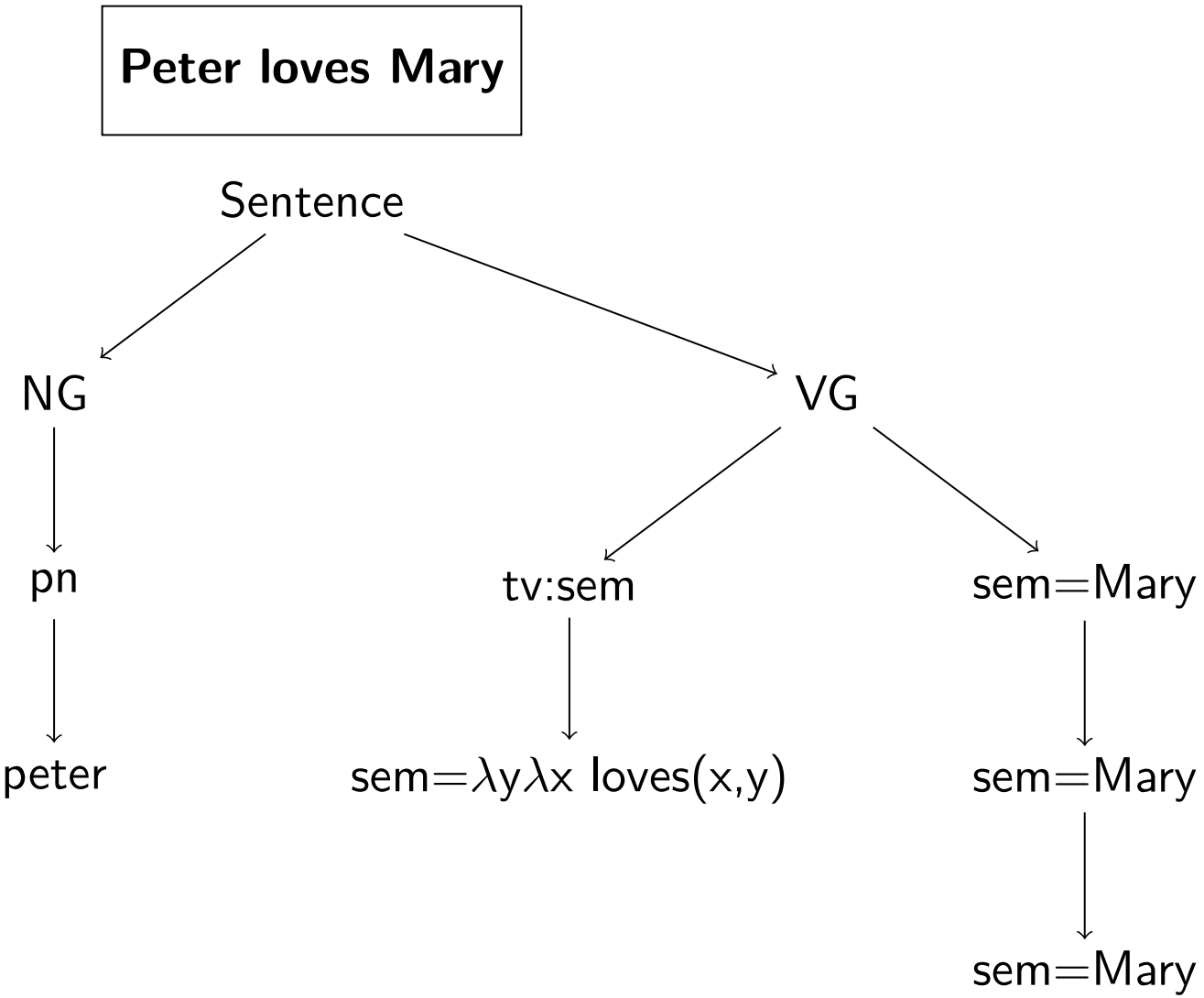
Composition function: example 2



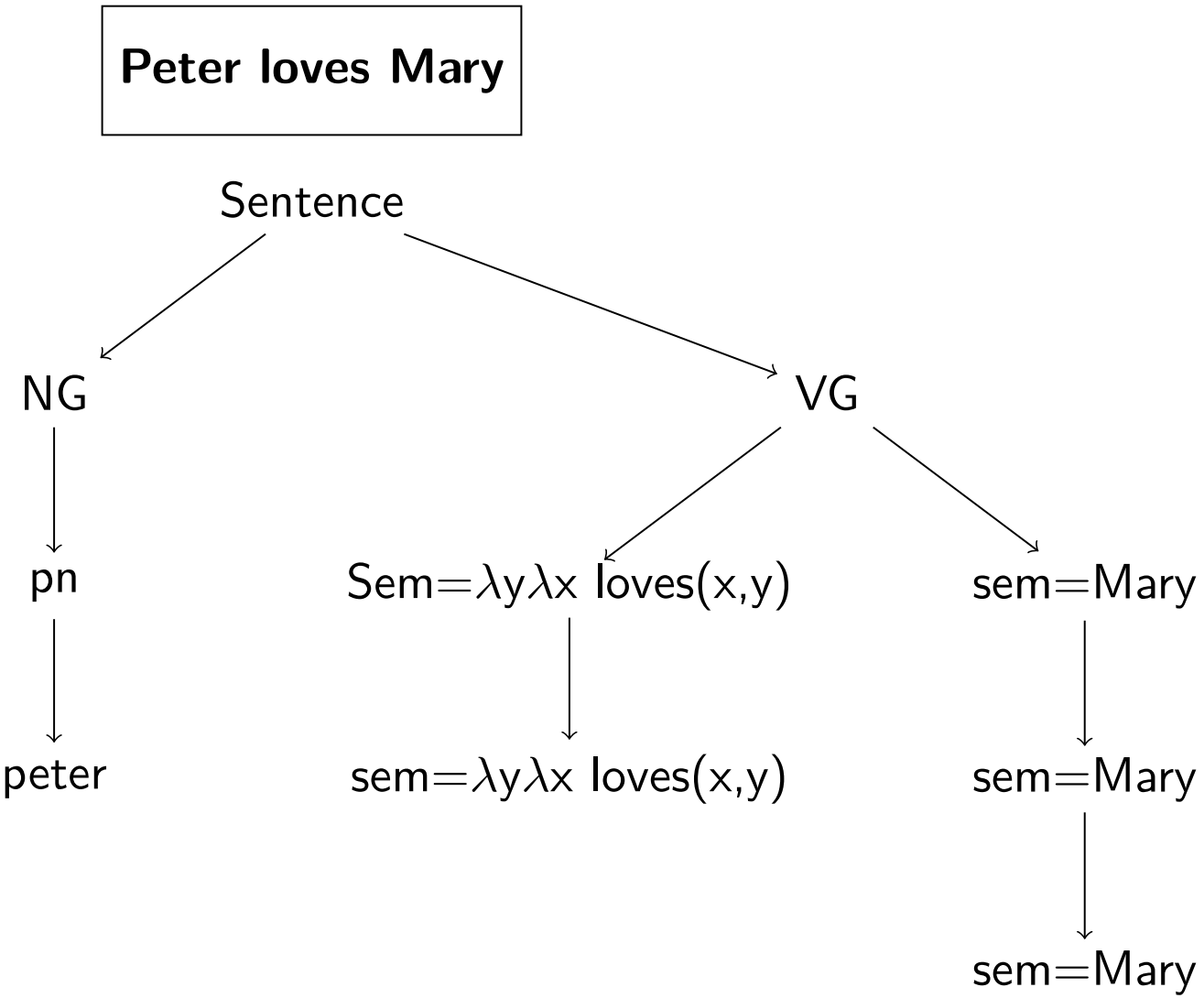
Composition function: example 2



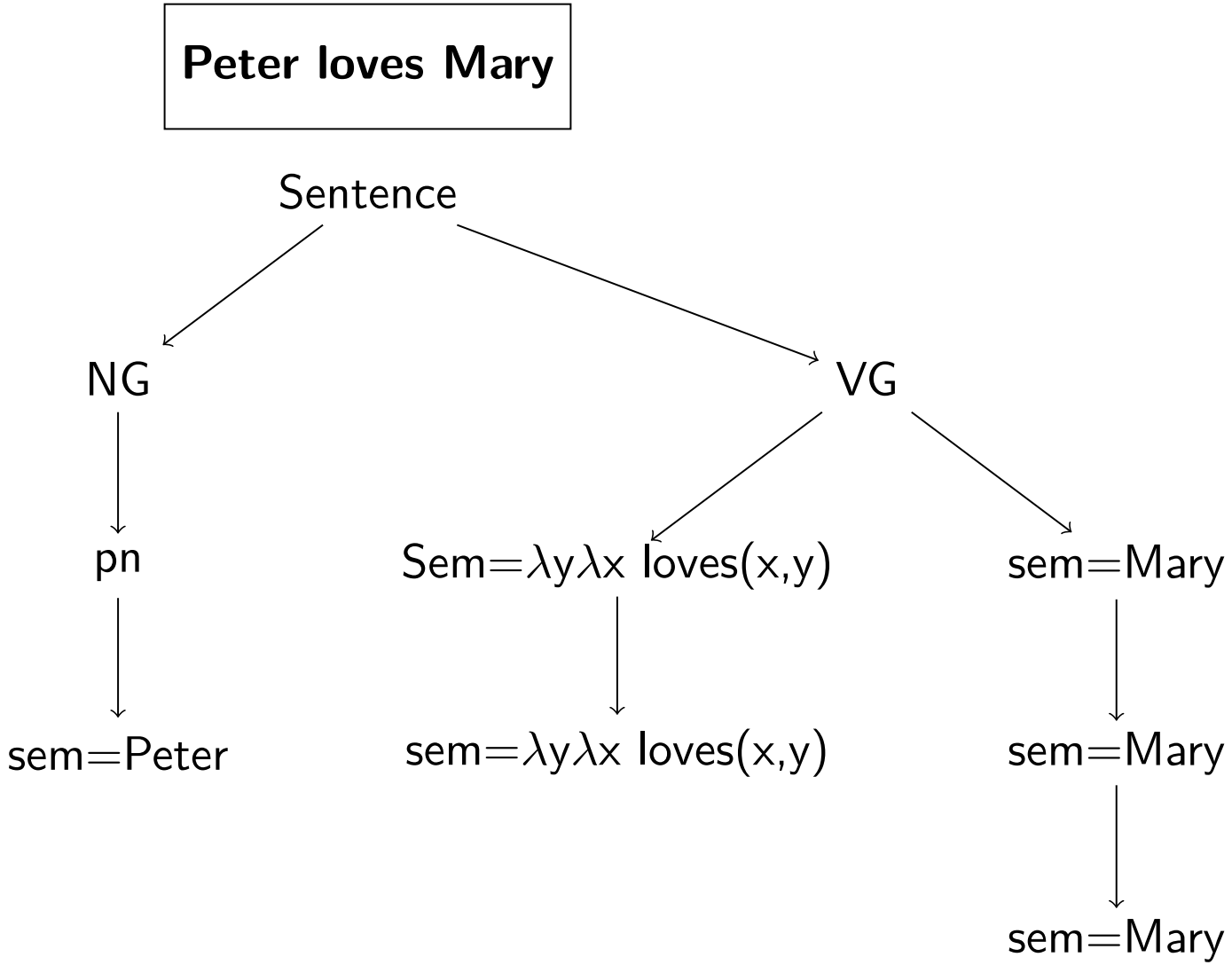
Composition function: example 2



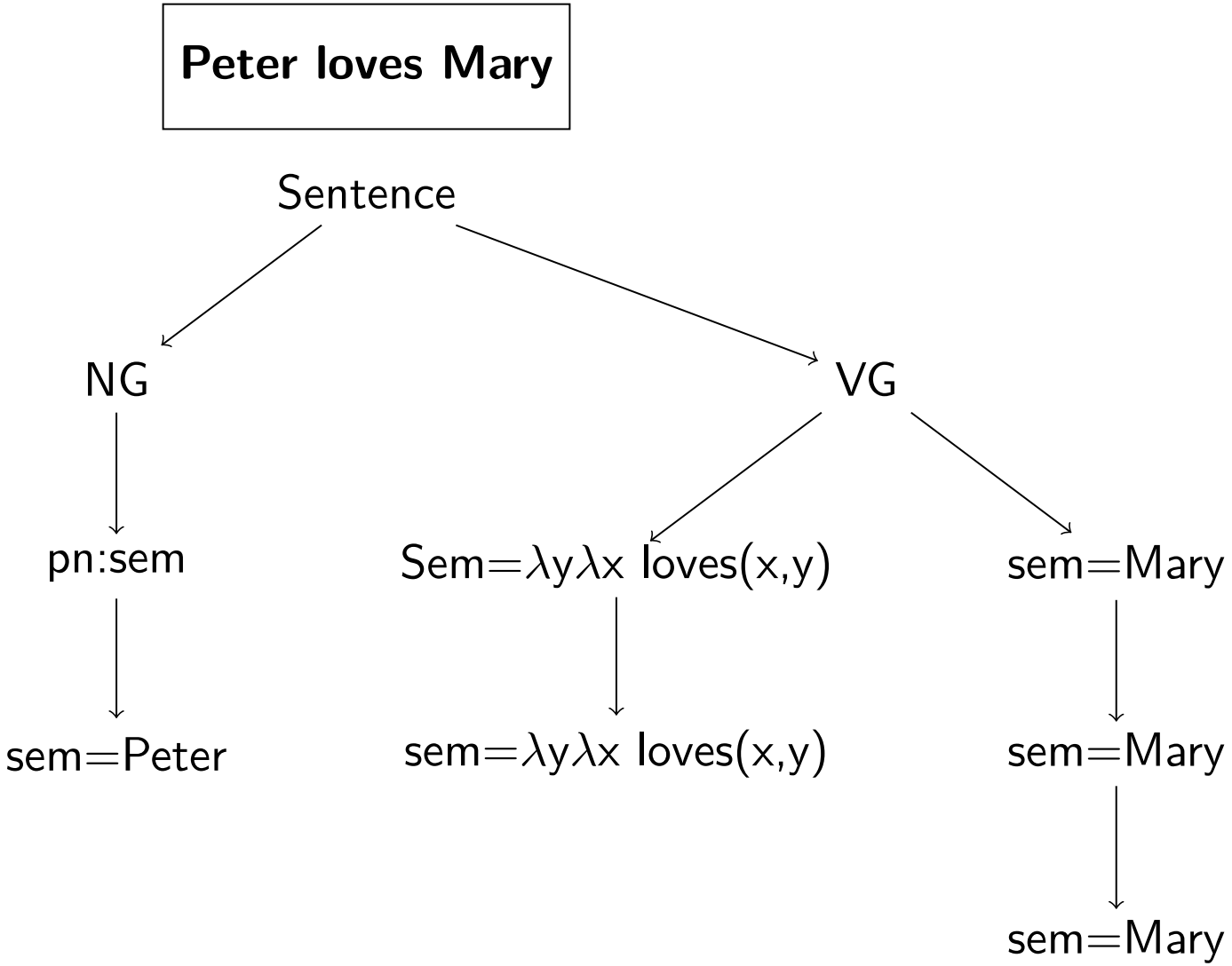
Composition function: example 2



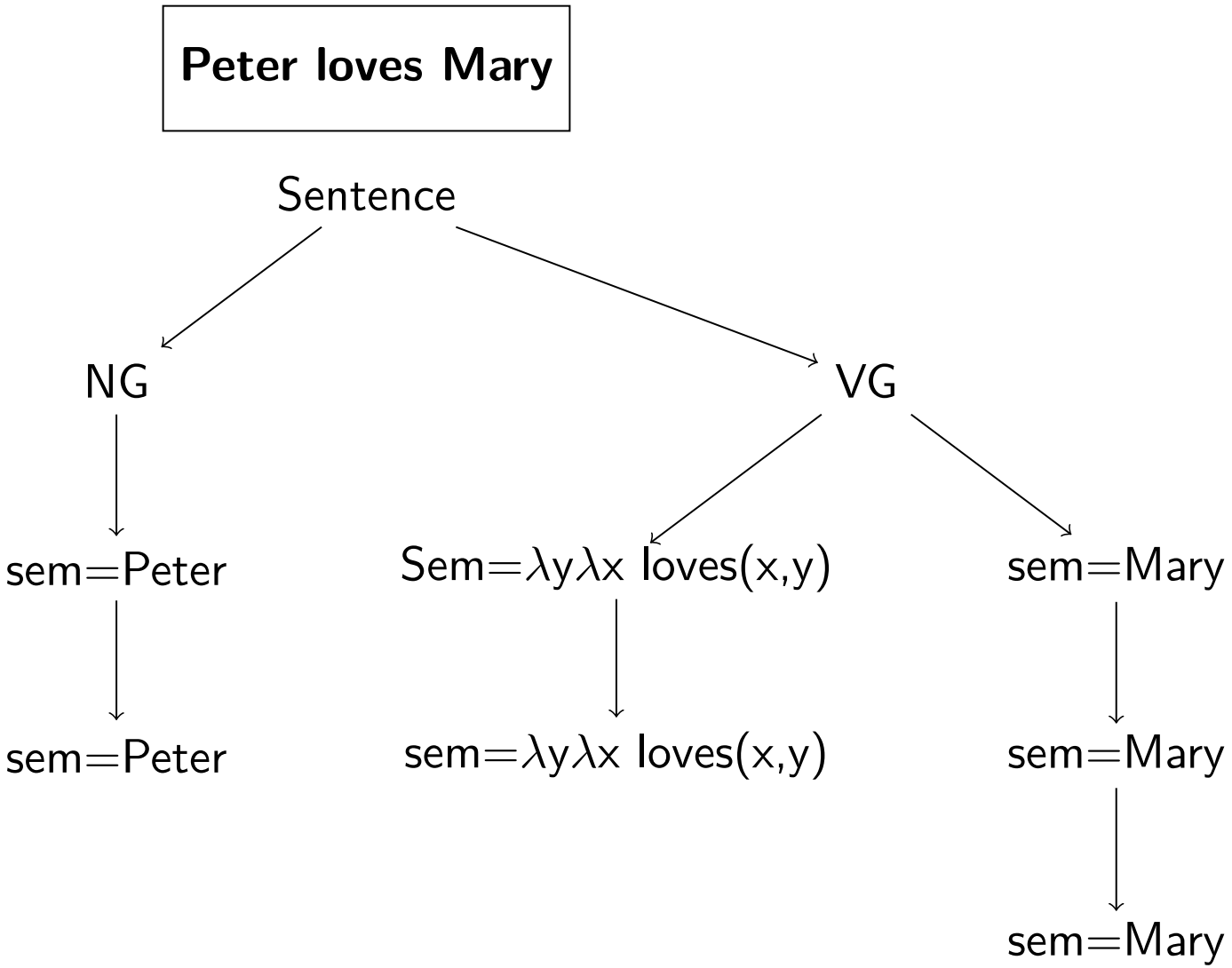
Composition function: example 2



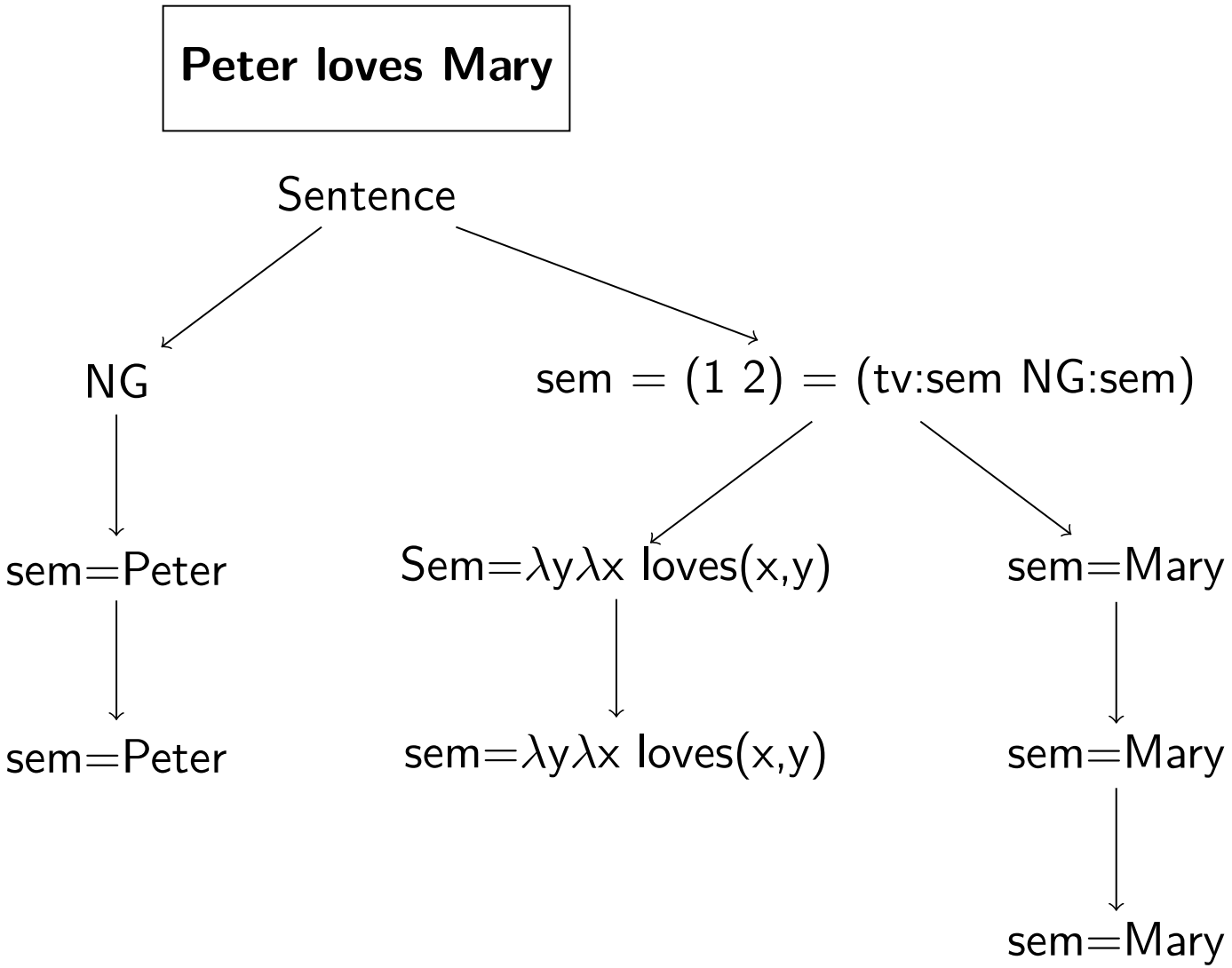
Composition function: example 2



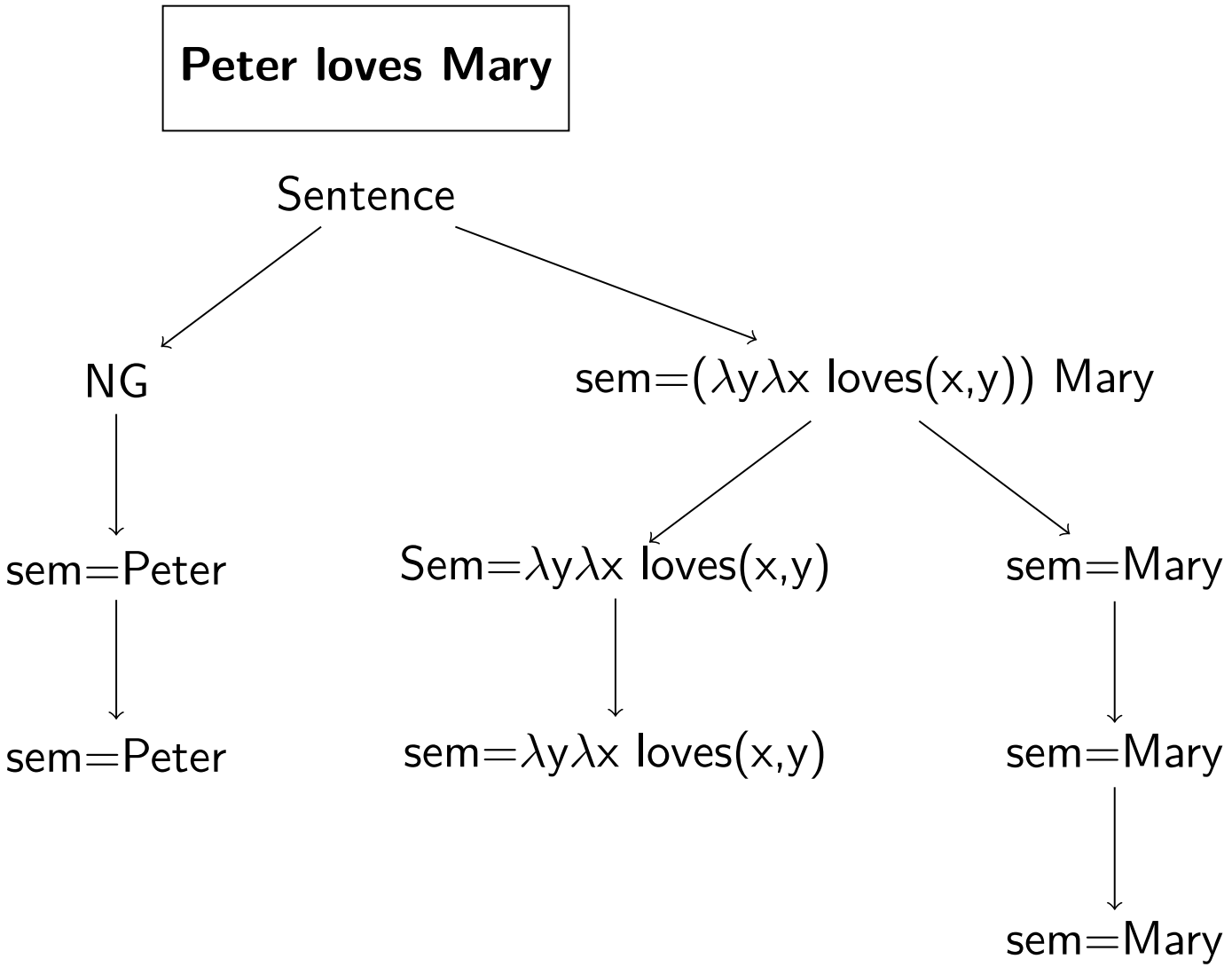
Composition function: example 2



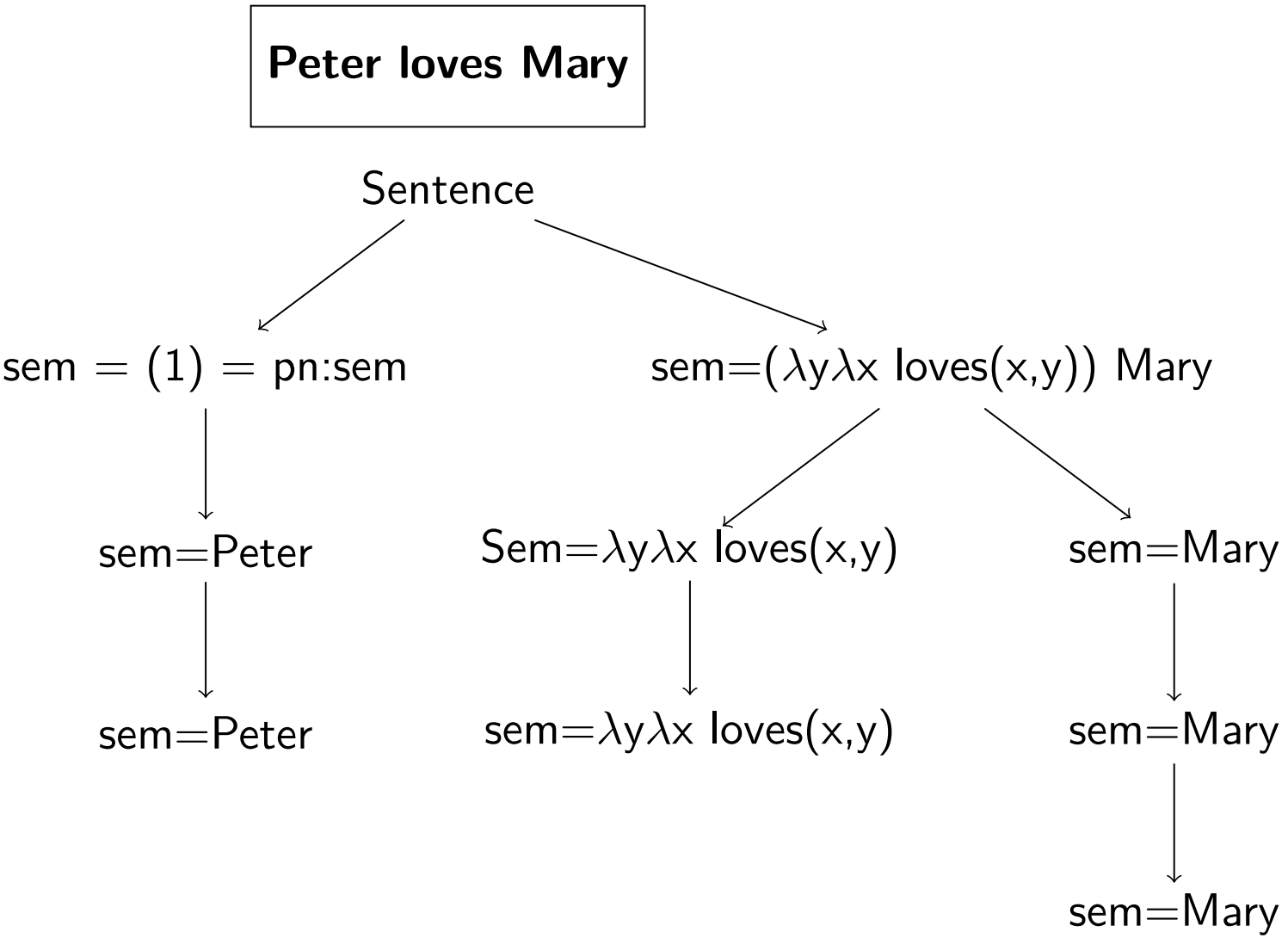
Composition function: example 2



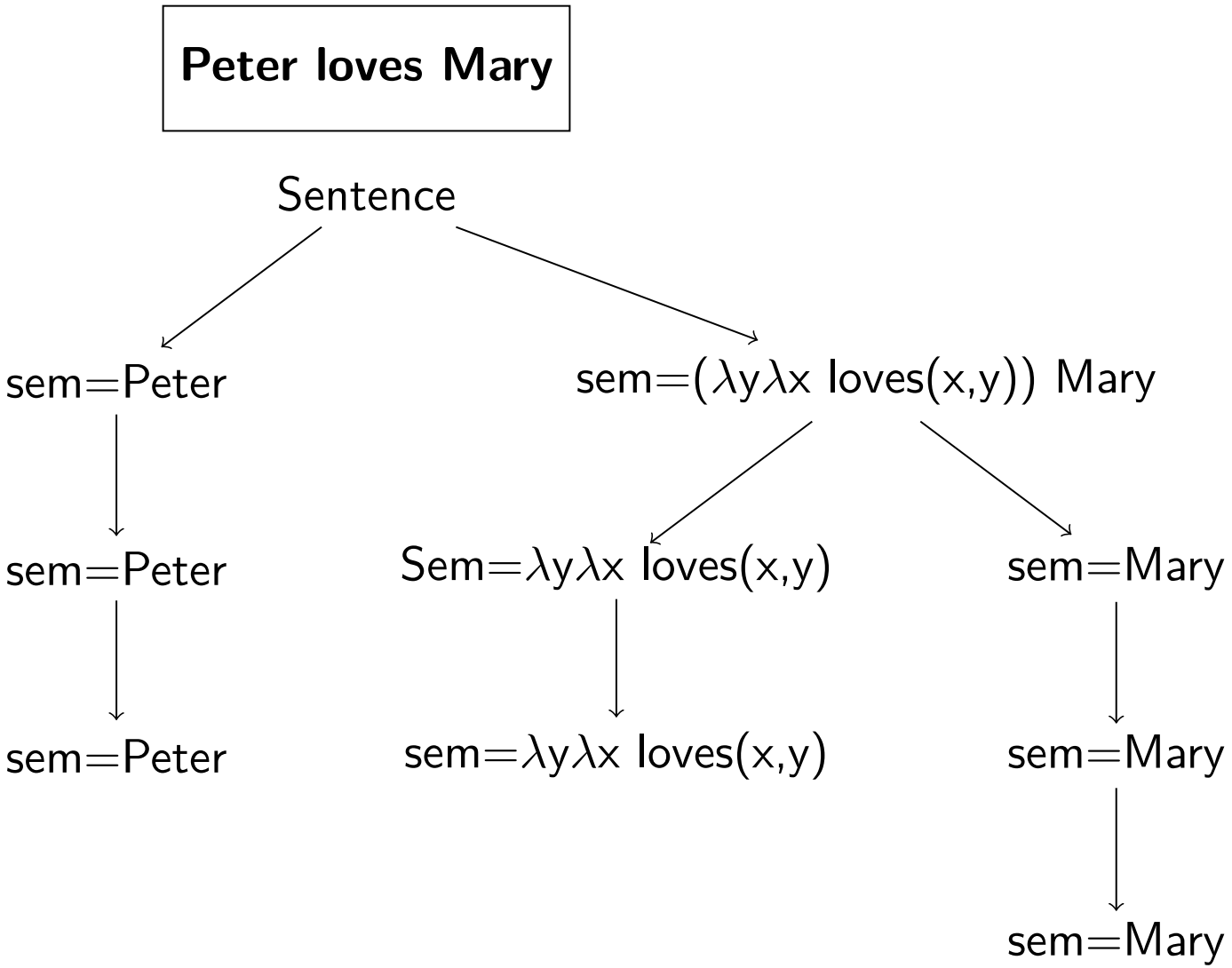
Composition function: example 2



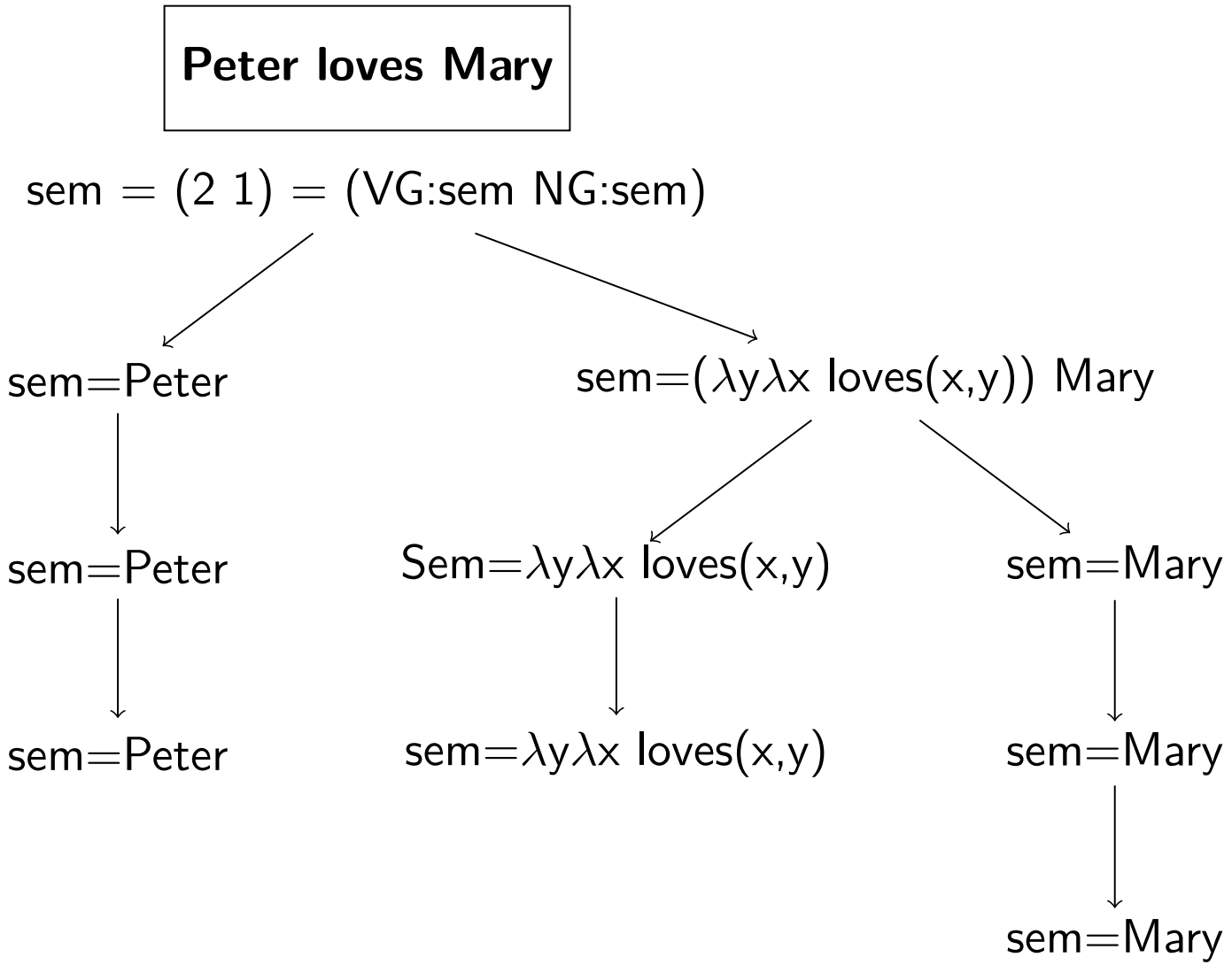
Composition function: example 2



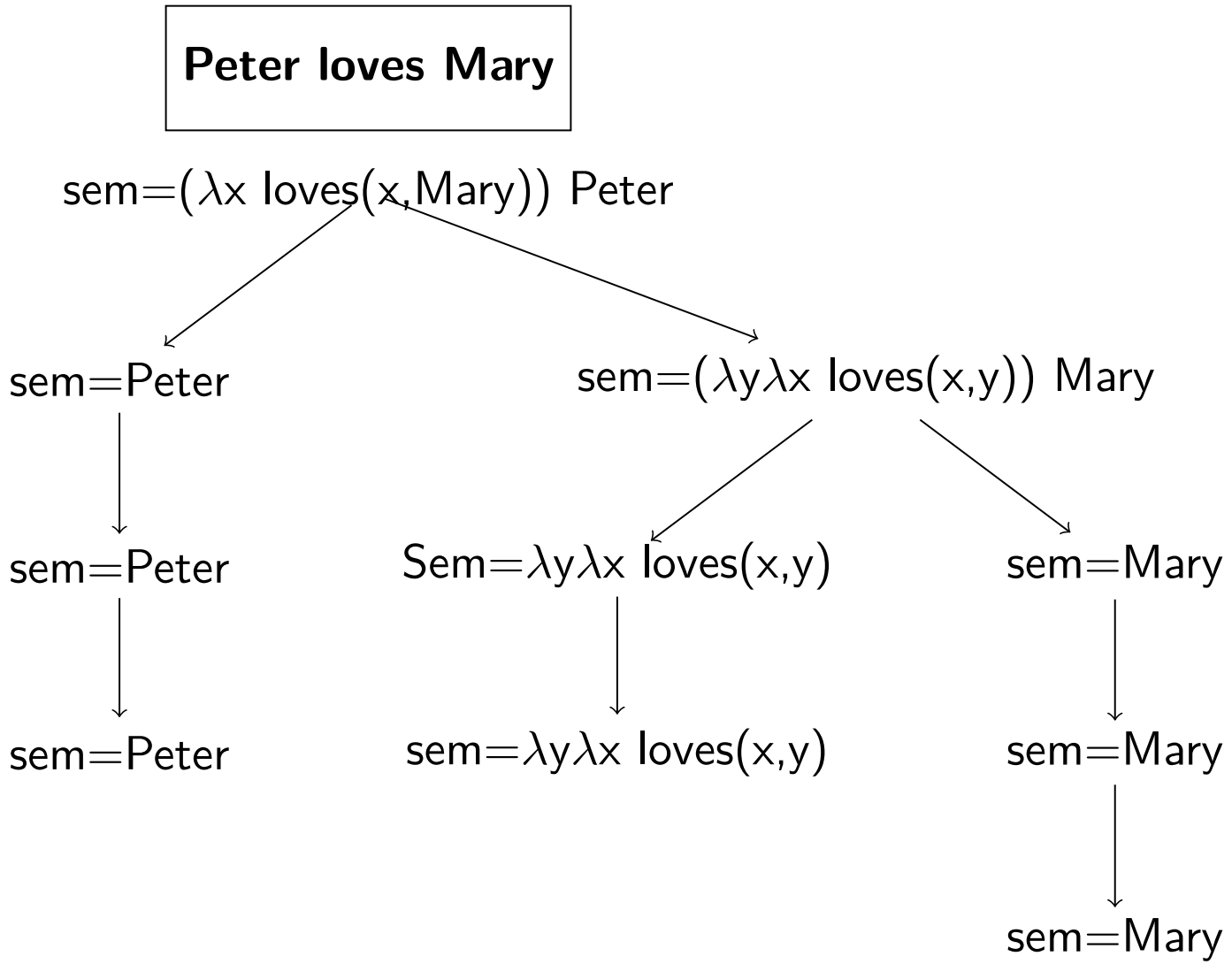
Composition function: example 2



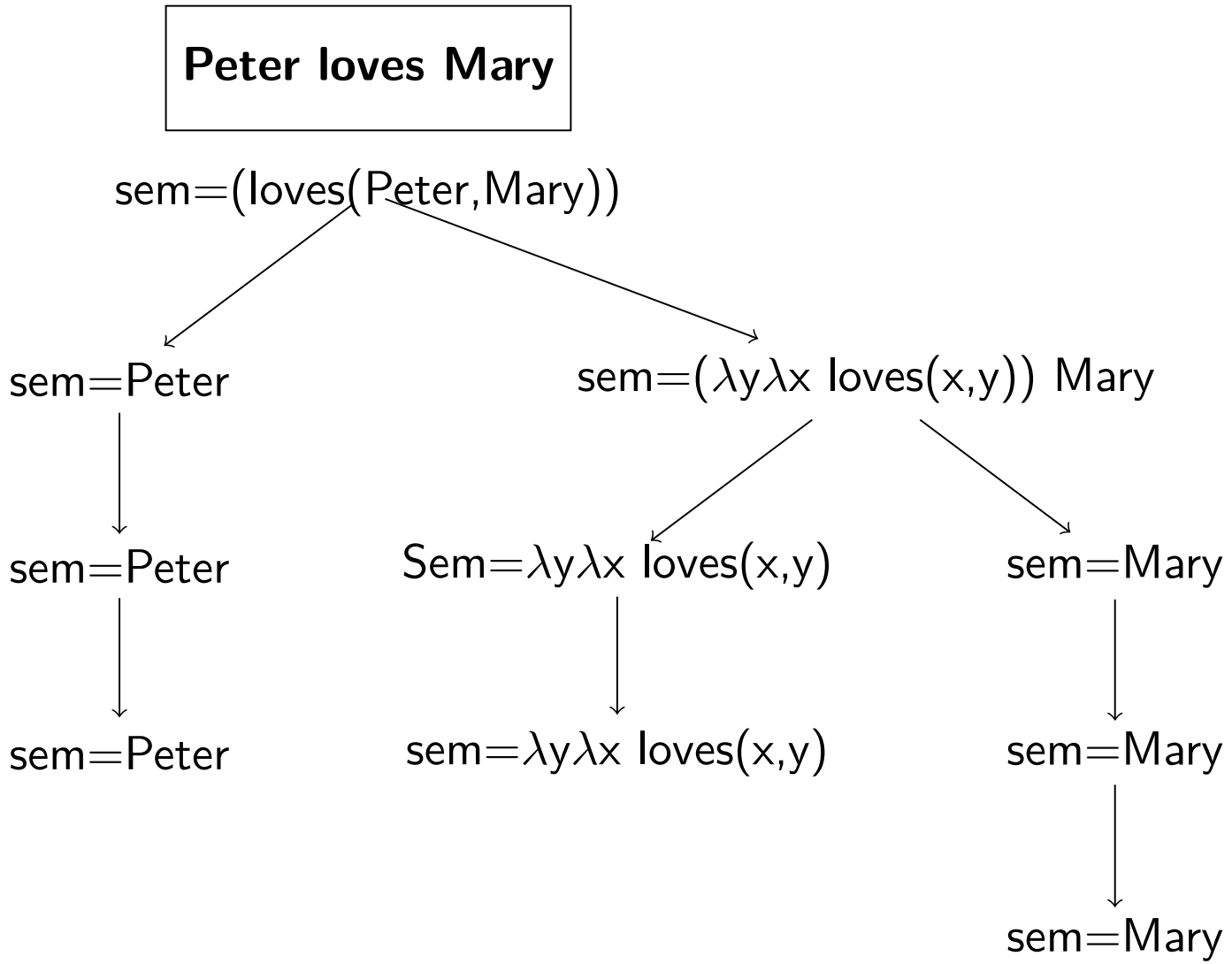
Composition function: example 2



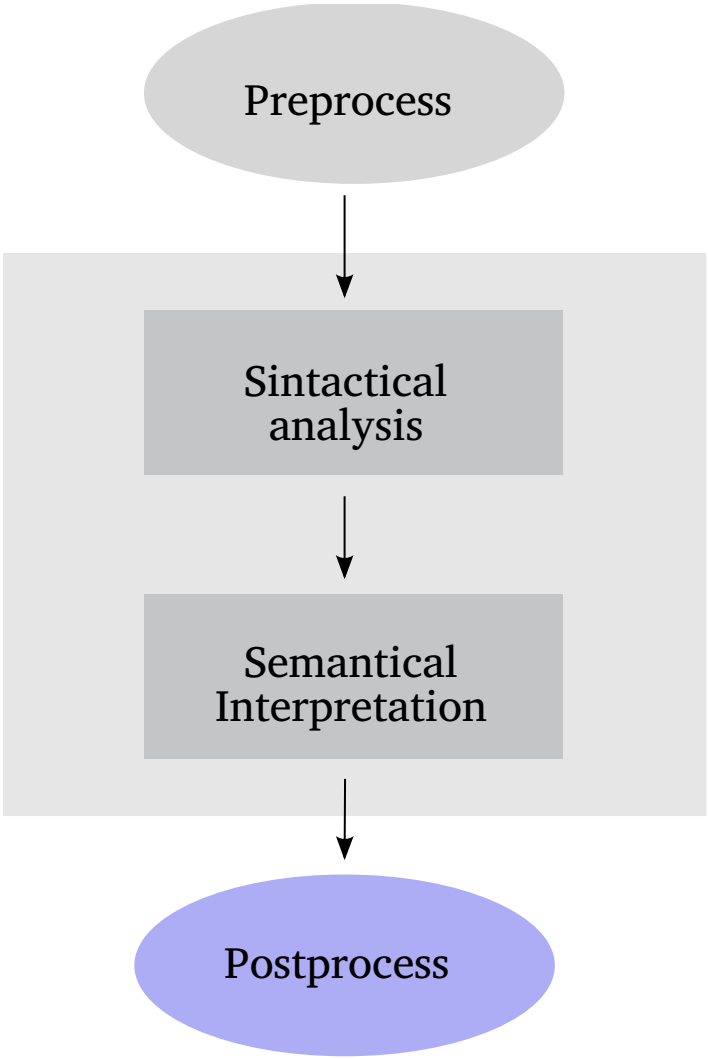
Composition function: example 2



Composition function: example 2



Natural Language Processing



What is the capital of France?

```
(sentence
 (interrogative_sentence
   (interrogative_pronoun (what))
   (verb (is)
     (sj (the capital of France)))
   )
 )
)
```

```
question(X), capital(X,france)
```

Postprocess

- Reference resolution
- Semantical-pragmatic analysis
- Illocutive analysis: Intention recognizing

NLP now (1990-)

- **Empirical** NLP
- Based on textual **corpora**
- **Statistical** processing vs. **Linguistic** processing
- **Combination** of linguistics and statistical models
- Application of **Machine Learning** methods to model language

Classical applications

- Based on **text**
  - Automatic translation
  - Extraction of information from text
- Based on **dialogs**
  - Natural language interfaces
  - Data base queries
  - Telephonic information services
  - Intelligent tutor systems

Applications based on document collections

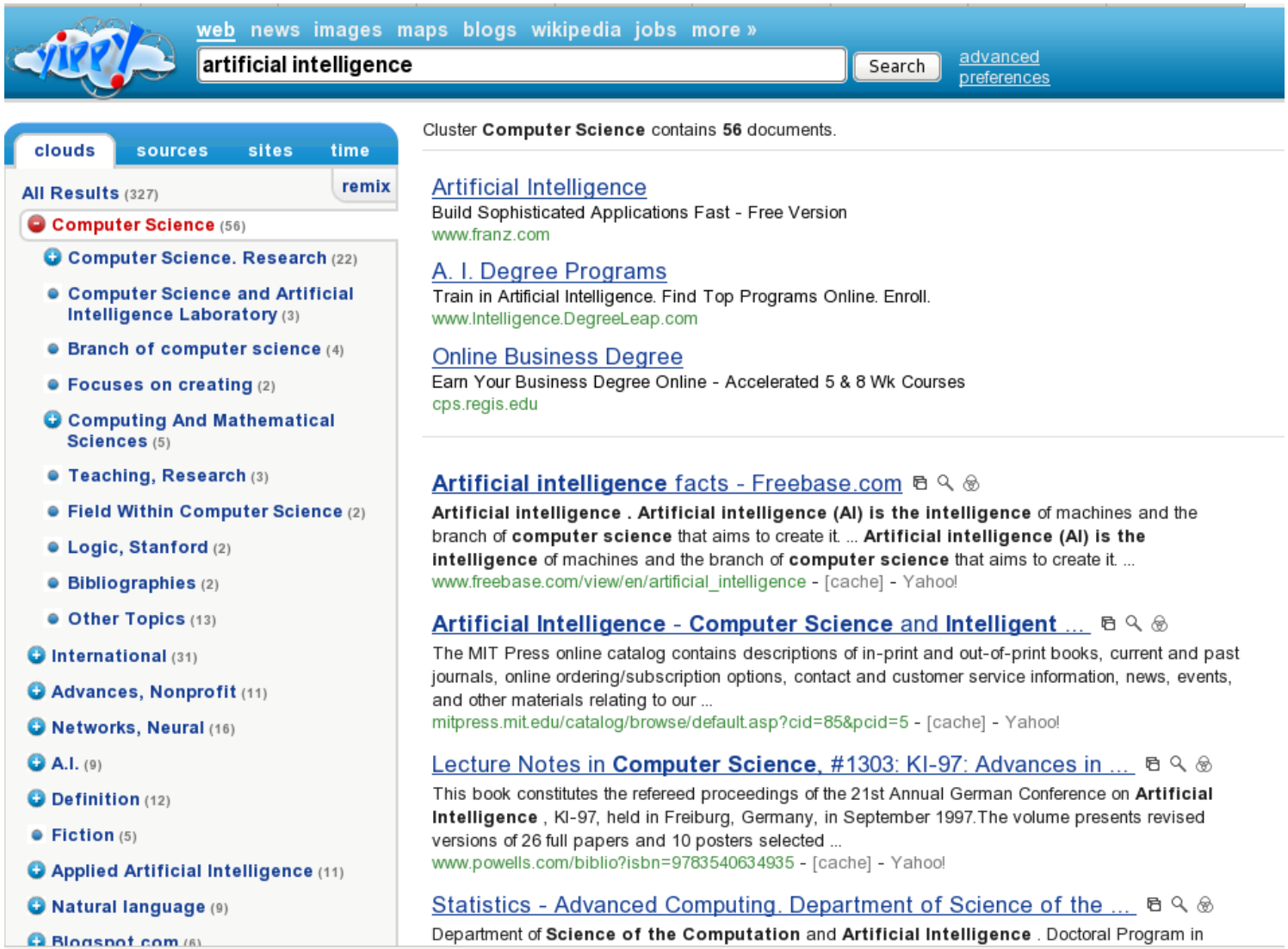
- Sources
  - Document corpora
  - Internet
- Information retrieval
  - The linguistic component improves precision
  - Language independent access and retrieval
- Document categorization
  - Document routing: Press agencies, e-mail classification, etc.
  - Document filtering: Commercial e-mail, netnews, etc.
  - Automatic (Re)organization of document collections: Textual databases, Internet, etc.
  - Topic Detection and Tracking

Applications based on document collections

- Extraction of information from the Web
  - Integration of information
  - Information search
  - Clustering and detection of relations inter/intra document
  - Routing and classification of documents
  - Document filtering
  - Corpora compilation
- Automatic summarization
- Question Answering

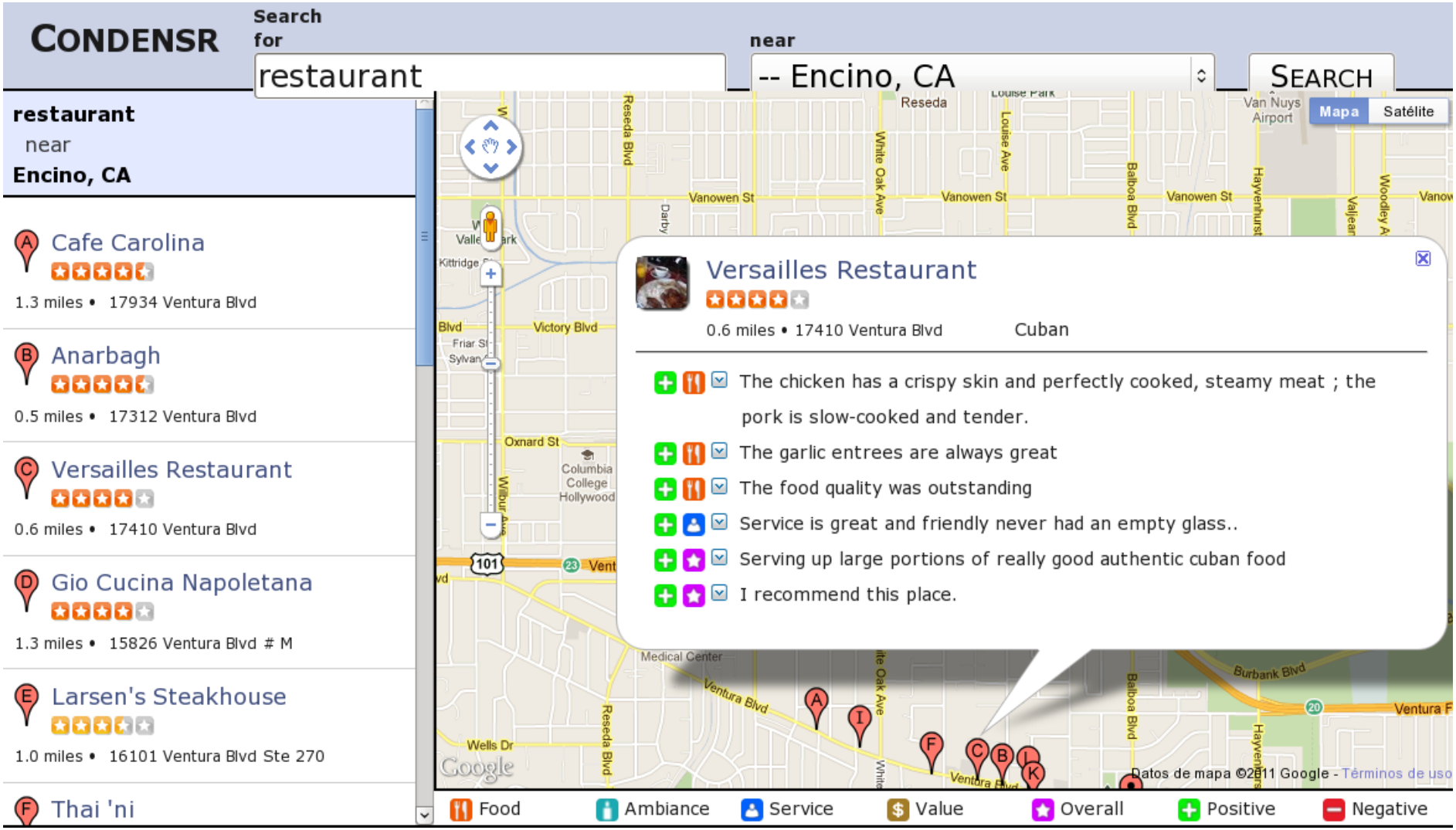
Extraction of information from the Web

yippy.com



Automatic summarization

condensr.com



Question Answering

Question Answering

- Wolfram Alpha
- Answers.com
- Start
- Qualim
- TextMap