Syntactic/Semantic Analysis for High-Precision Math Linguistics

CICM 2018

Jan Frederik Schaefer Michael Kohlhase

Friedrich-Alexander-Universität Erlangen-Nürnberg

August 13, 2018

Example

"A positive integer n is called prime, iff there is no integer $1 < m < n$ *such that* $m|n$"

Translation to (from) German:

"Eine positive ganze Zahl n ist prim genau dann, wenn es keine ganze Zahl $1 < m < n$ *gibt, sodass* $m|n$"

Formalization:

$\forall n.\mathbf{pos}(n) \wedge \mathbf{int}(n) \Rightarrow (\mathbf{prime}(n) \Leftrightarrow$
$\neg\exists m.\mathbf{int}(m) \wedge \mathbf{divides}(m, n) \wedge \mathbf{less}(1, m) \wedge \mathbf{less}(m, n))$

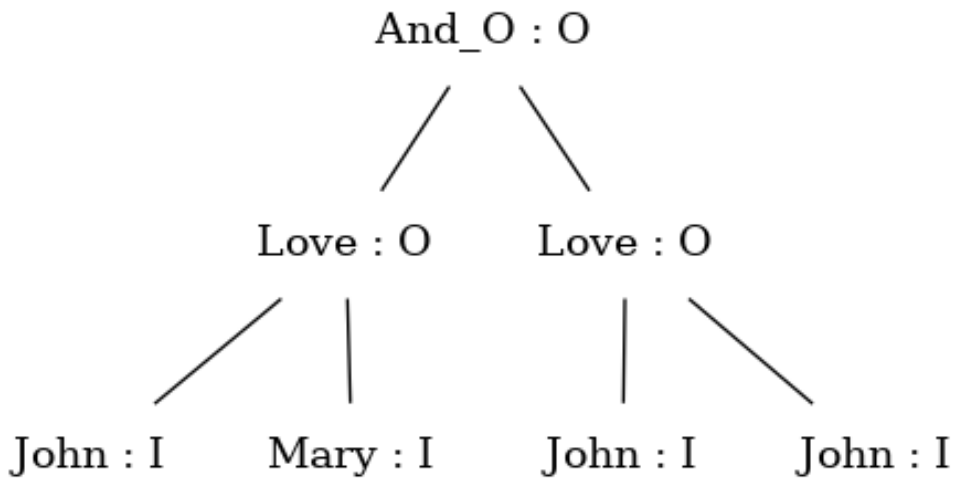
GF - Grammatical Framework

- "A programming language for multilingual grammar applications"
- Natural language as formal language $\Rightarrow$ limited coverage but high precision
- Idea:
  - *Abstract grammar* describes "meaning" we want to express
  - *Concrete grammars* describe how this is expressed in English/German/Logic/...

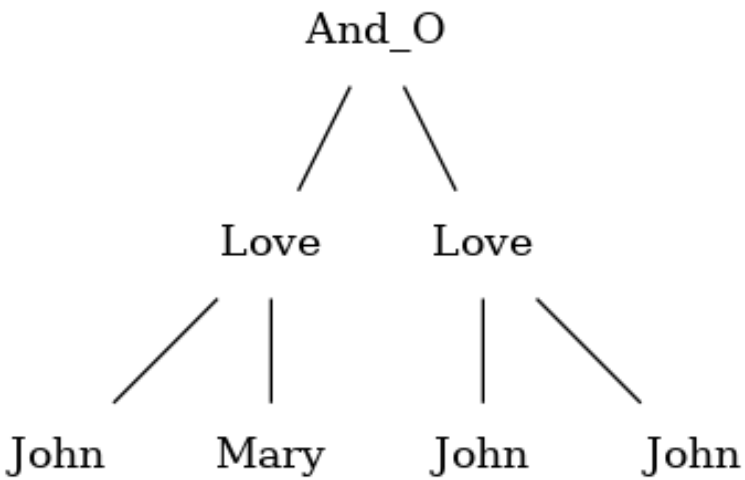
Abstract Grammar

```
abstract Gossip = {
    cat
        O;    -- propositions
        I;    -- individuals
    fun
        John, Mary : I;
        Love : I -> I -> O;
        And_O : O -> O -> O;
        And_I : I -> I -> I;
}
```

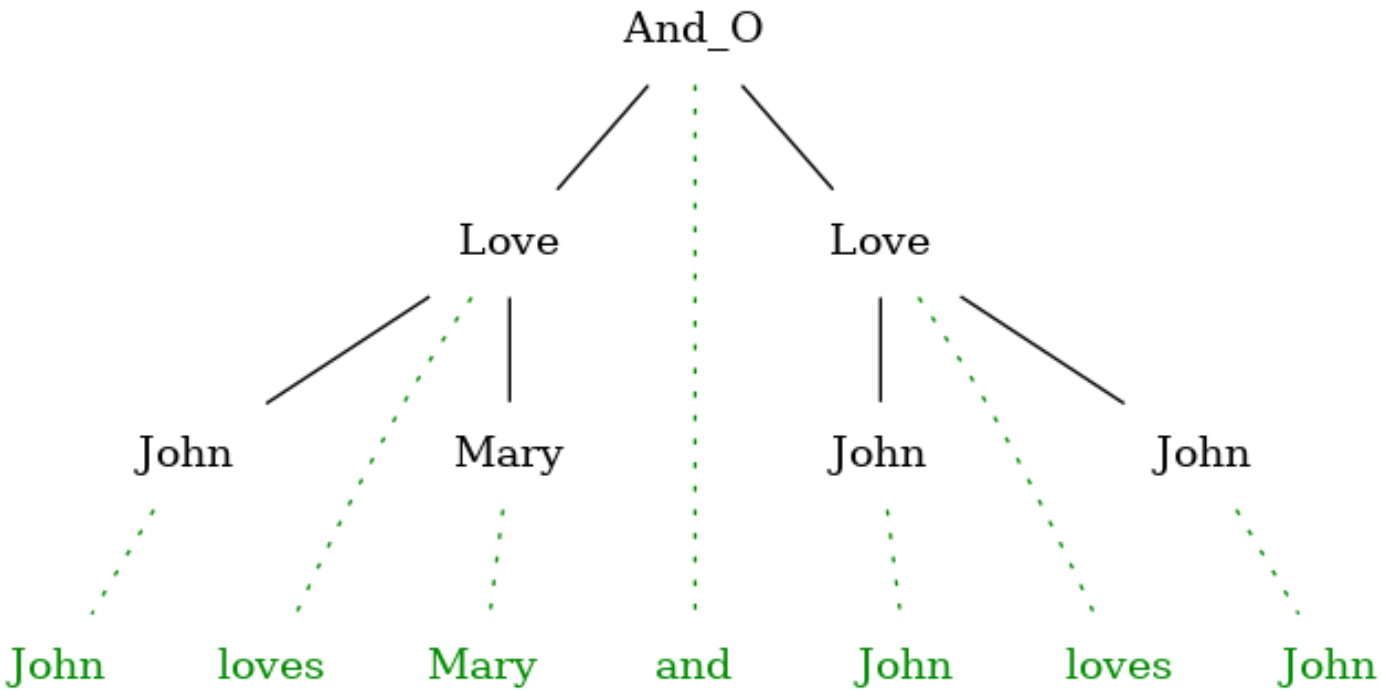
Example: And_O (Loves John Mary) (Loves John John)



Abstract Grammar



Concrete Grammar



Concrete Grammar - Simple Approach

```
concrete GossipStr of Gossip = {
    lincat
        I = Str;
        O = Str;
    lin
        John = "John";
        Mary = "Mary";
        Love a b = a ++ "loves" ++ b;
        And_O a b = a ++ "and" ++ b;
        And_I a b = a ++ "and" ++ b;
}
```

Problem: "*John and Mary loves John*"

Concrete Grammar - Simple Approach

Problem: Ending of verb depends on subject

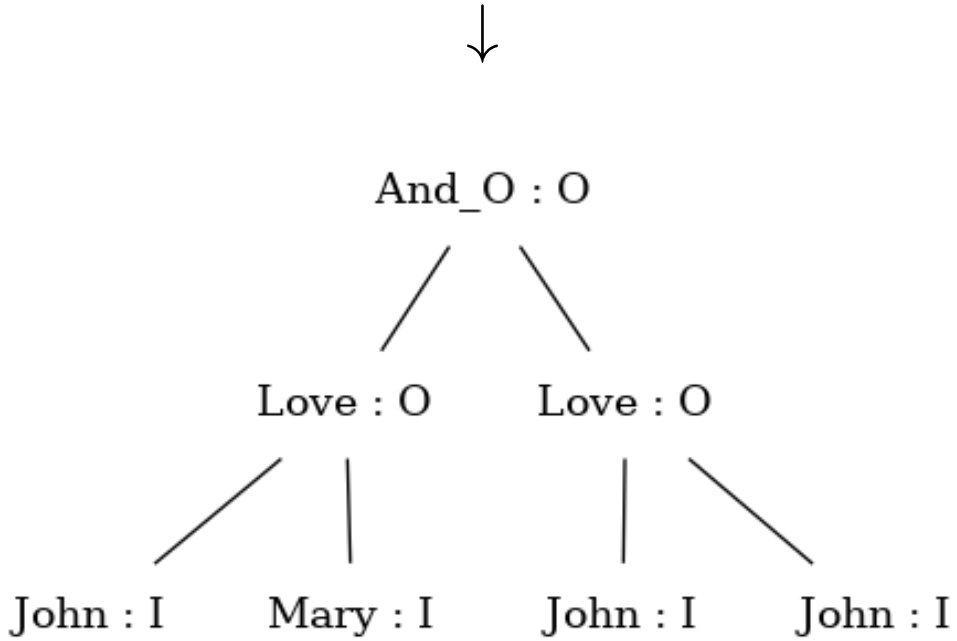
- Idea: Use record types
- This is a common problem $\rightsquigarrow$ GF's *resource grammar library*

Concrete Grammar - Resource Grammar Library

```
concrete GossipEng of Gossip = open SyntaxEng,ParadigmsEng in {
    lincat
        I = NP;   -- noun phrase
        O = S;    -- sentence
    lin
        John = mkNP (mkPN "John");
        Mary = mkNP (mkPN "Mary");
        Love a b = mkS (mkCl a (mkV2 "love") b);
        And_O a b = mkS and_Conj a b;
        And_I a b = mkNP and_Conj a b;
}
```

Concrete Grammar - Translation

"*Johann liebt Maria und Johann liebt Johann*"



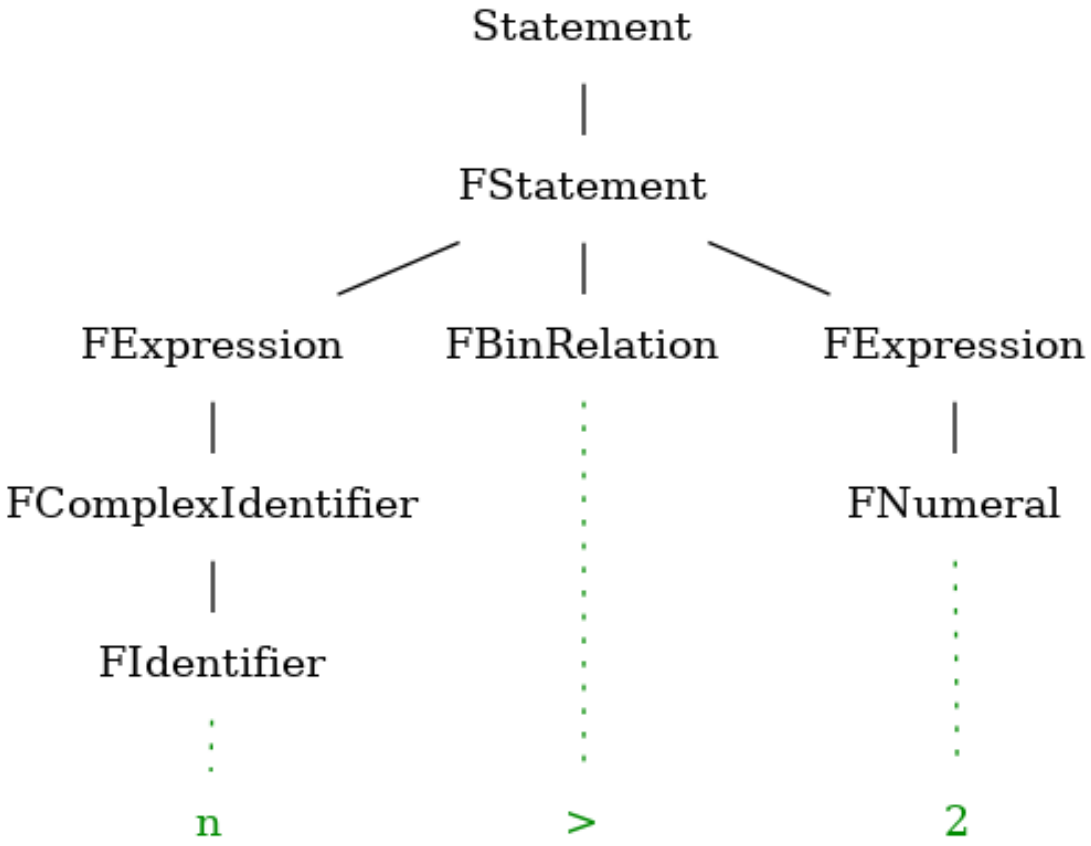
"*John loves Mary and John loves John*"

Using GF for Mathematics - Challenges

- Parsing formulae
- Different grammatical roles of formulae in a sentence
  - "*if* $n > 1$"
  - "*if* $n + k$ *is even*"
- Other idiosyncracies in mathematical language not covered by the resource grammar library, like
  - "*let n be a...*"
  - "*an integer is called prime iff...*"
- Finding the right abstract grammar (syntactic vs semantic)

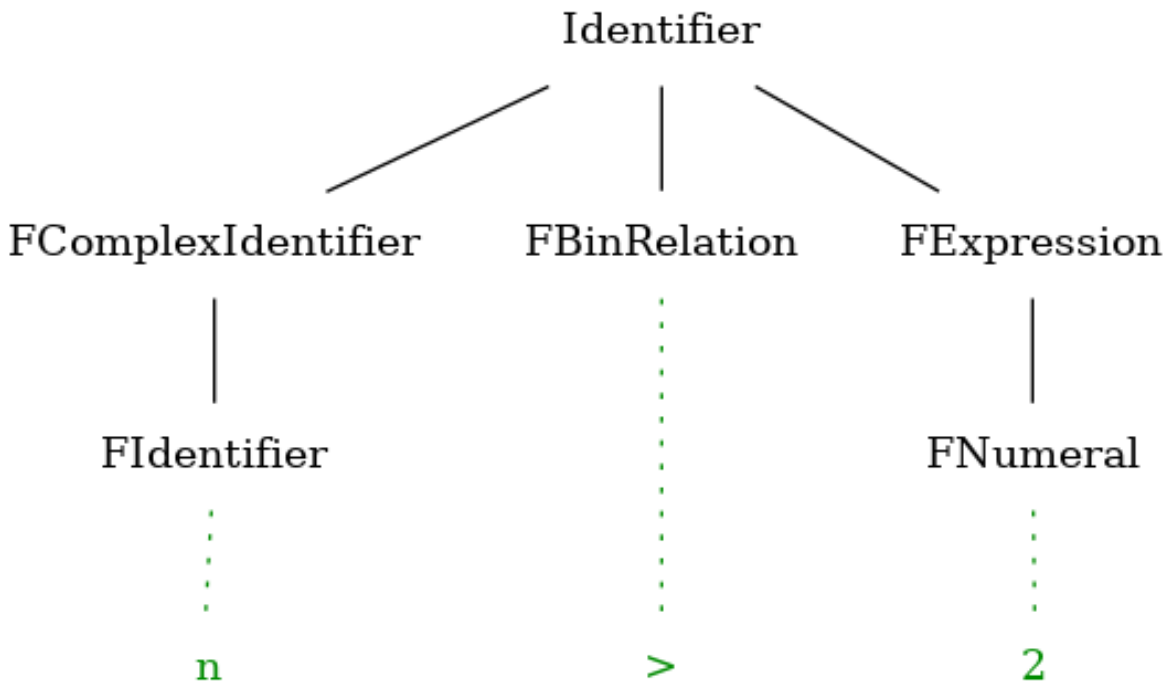
Using GF for Mathematics - Formula as Statement

"*we know that* $n > 2$"



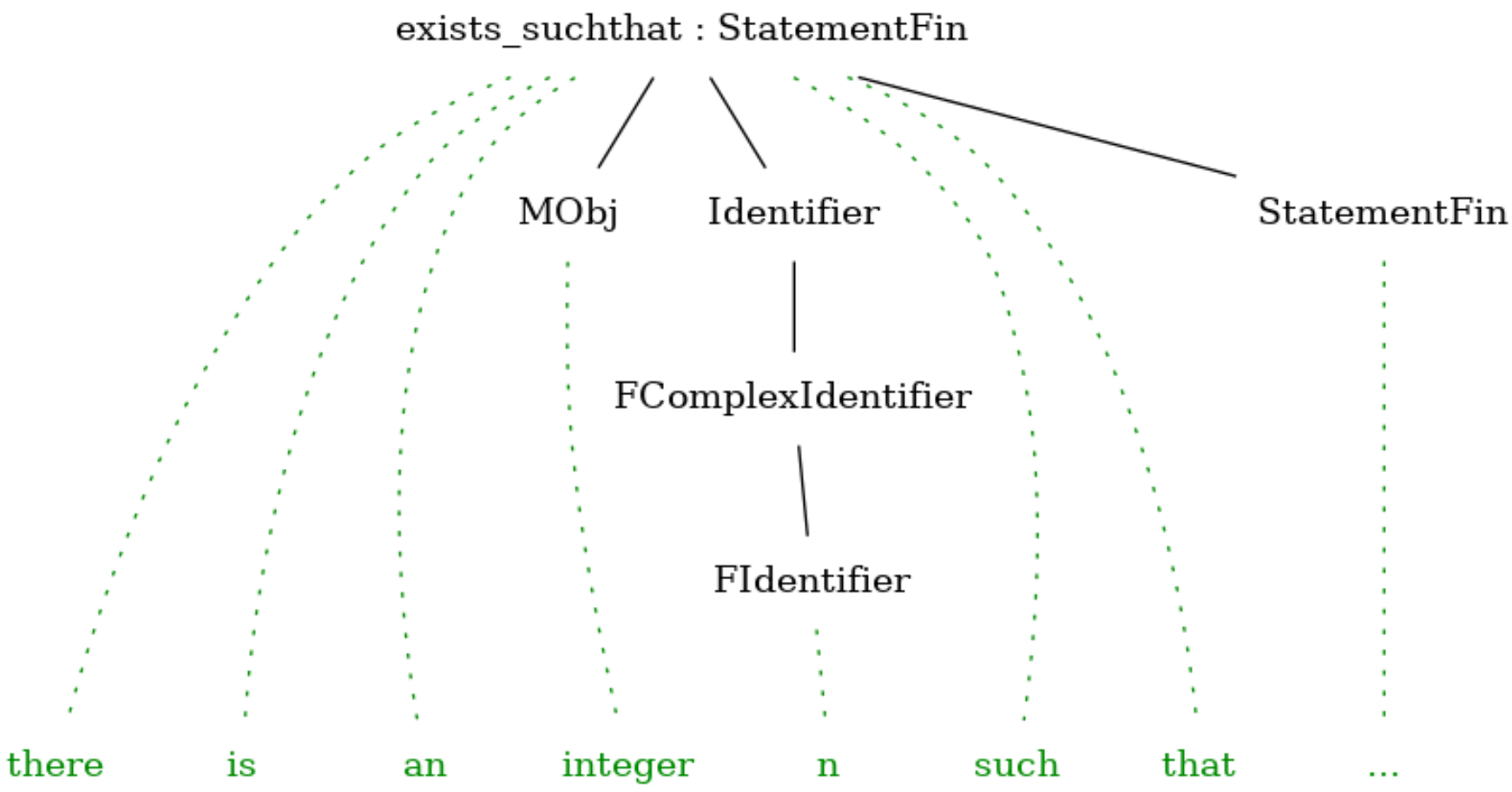
Using GF for Mathematics - Formula as Identifier

"*let* $n > 2$ *be an integer*"



Using GF for Mathematics - Using Identifier in Statement

"*there is an integer n such that . . .*"



Using GF for Mathematics - Using Identifier in Statement

"there is an integer n such that . . ."

exists_suchthat : PosNegPol -> MObj -> Identifier -> StatementFin -> StatementFin;

$(\exists n.(\lambda x.\mathbf{int}(x))n \wedge (\ldots))$

$\downarrow_\beta$

$\exists n.\mathbf{int}(n) \wedge \ldots$

Using GF for Mathematics - Using Identifier in Statement

"there isn't a positive integer $n > 2$ such that ..."

exists_suchthat : PosNegPol -> MObj -> Identifier -> StatementFin -> StatementFin;

$$(\neg\exists n.(\lambda x.\mathbf{pos}(x) \wedge \mathbf{int}(x))n \wedge \mathbf{greater}(n, 2) \wedge (\ldots))$$

$$\downarrow_\beta$$

$$\neg\exists n.\mathbf{pos}(n) \wedge \mathbf{int}(n) \wedge \mathbf{greater}(n, 2), \wedge \ldots$$

Using GF for Mathematics - Example

"*A positive integer n is called prime, iff there is no integer* $1 < m < n$ *such that* $m|n$"

Translation to (from) German:

"*Eine positive ganze Zahl n ist prim genau dann, wenn es keine ganze Zahl* $1 < m < n$ *gibt, sodass* $m|n$"

Formalization:

$$(\forall n.((\lambda x.\mathbf{pos}(x) \wedge \mathbf{int}(x))n) \Rightarrow ((\lambda x.\mathbf{prime}(x))n \Leftrightarrow$$
$$(\neg\exists m.(\lambda x.\mathbf{int}(x))m \wedge \mathbf{less}(1, m) \wedge \mathbf{less}(m, n) \wedge (\mathbf{divides}(m, n)))))$$

$$\downarrow_\beta$$

$$\forall n.\mathbf{pos}(n) \wedge \mathbf{int}(n) \Rightarrow (\mathbf{prime}(n) \Leftrightarrow$$
$$\neg\exists m.\mathbf{int}(m) \wedge \mathbf{divides}(m, n) \wedge \mathbf{less}(1, m) \wedge \mathbf{less}(m, n))$$

Next Steps

- Extend grammars for larger coverage
- Extend lexica for larger coverage
- Switch to DRT