Syntactic/Semantic Analysis for High-Precision Math Linguistics

CICM 2018

Jan Frederik Schaefer   Michael Kohlhase

Friedrich-Alexander-Universität Erlangen-Nürnberg

August 12, 2018

"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:

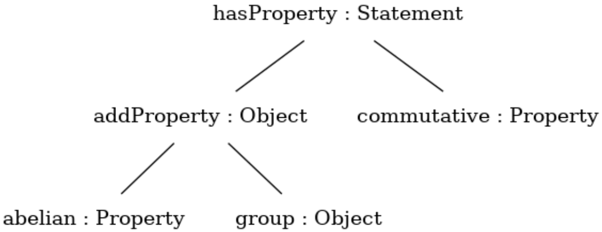
$$\forall n.\mathbf{pos}(n) \wedge \mathbf{int}(n) \Rightarrow (\mathbf{prime}(n) \Leftrightarrow \neg\exists m.\mathbf{int}(m) \wedge \mathbf{divides}(m, n) \wedge \mathbf{less}(1, m) \wedge \mathbf{less}(m, n))$$

GF - Grammatical Framework

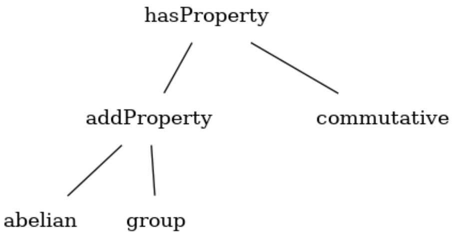
- "A programming language for multilingual grammar applications"
- Natural language as formal language $\Rightarrow$ limited coverage but high precision
- Idea:
  - *Abstract grammar* describes "meaning" we want to express
  - *Concrete grammars* describe how this is expressed in English/German/Logic/...

Abstract Grammar

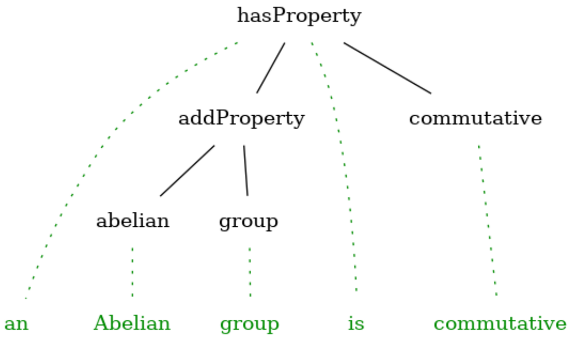
```
abstract Math = {
    cat Object; Property; Statement;
    fun
        group : Object;
        abelian : Property;
        commutative : Property;
        addProperty : Property -> Object -> Object;
        hasProperty : Object -> Property -> Statement;
}
```



Abstract Grammar



Concrete Grammar



```
concrete MathStr of Math = {
  lincat
    Object = Str;
    Property = Str;
    Statement = Str;
  lin
    group = "group";
    abelian = "Abelian";
    commutative = "commutative";
    addProperty prop obj = prop ++ obj;
    hasProperty obj prop = "an" ++ obj ++ "is" ++ prop;
}
```

Problem: "*a*" vs "*an*"

Concrete Grammar - Simple Approach

Problem: "*a*" vs "*an*"

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

Concrete Grammar - Resource Grammar Library

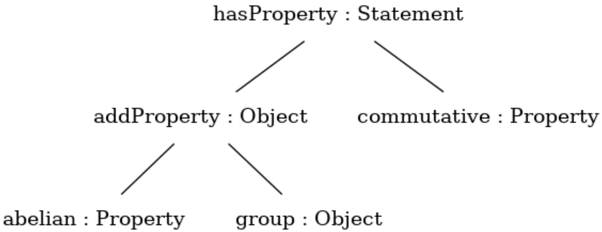
```
concrete MathEng of Math = open SyntaxEng, ParadigmsEng in {
  lincat
    Object = CN;            -- common noun
    Property = AP;          -- adjective phrase
    Statement = S;          -- sentence
  lin
    group = mkCN (mkN "group");
    abelian = mkAP (mkA "Abelian");
    commutative = mkAP (mkA "commutative");
    addProperty prop obj = mkCN prop obj;
    hasProperty obj prop = mkS (mkCl (mkNP aSg_Det obj) prop);
}
```

Concrete Grammar - Resource Grammar Library

```
concrete MathFre of Math = open SyntaxFre, ParadigmsFre in {
  lincat
    Object = CN;           -- common noun
    Property = AP;         -- adjective phrase
    Statement = S;         -- sentence
  lin
    group = mkCN (mkN "groupe" masculine);
    abelian = mkAP (mkA "abélien");
    commutative = mkAP (mkA "commutatif");
    addProperty prop obj = mkCN prop obj;
    hasProperty obj prop = mkS (mkCl (mkNP aSg_Det obj) prop);
}
```

"*un groupe abélien est commutatif*"

$\downarrow$



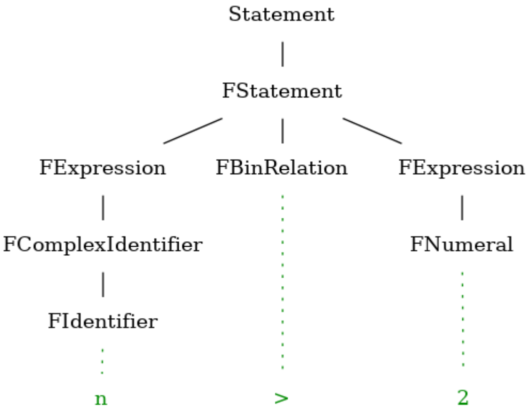
$\downarrow$

"*an Abelian group is commutative*"

Using GF for Mathematics - Challenges

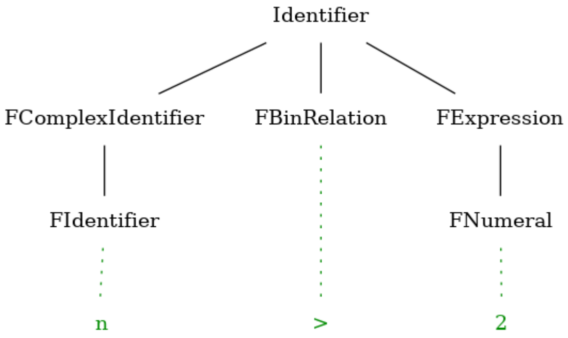
- Parsing formulae
- Different grammatical roles of formulae in a sentence
  - "*if* $n > 1$"
  - "*if* $n + k$ *is even*"
- Other idiosyncracies in mathematical language not covered by the resource grammar library, like
  - "*let n be a...*"
  - "*an integer is called prime iff...*"
- Finding the right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

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



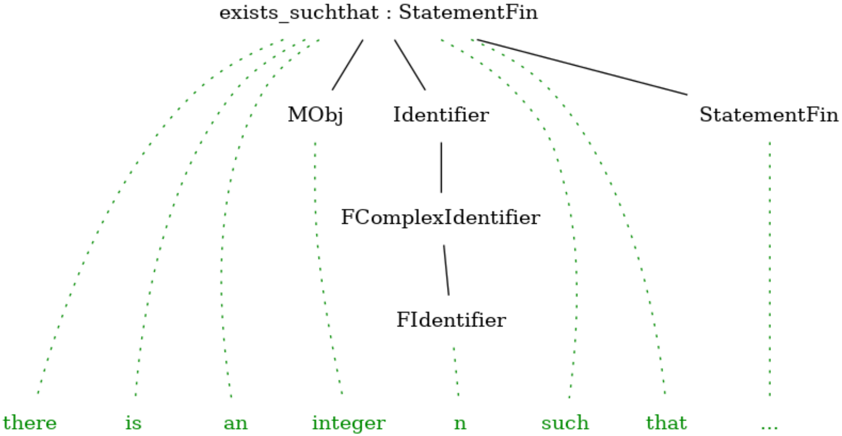
Using GF for Mathematics - Formula as Identifier

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



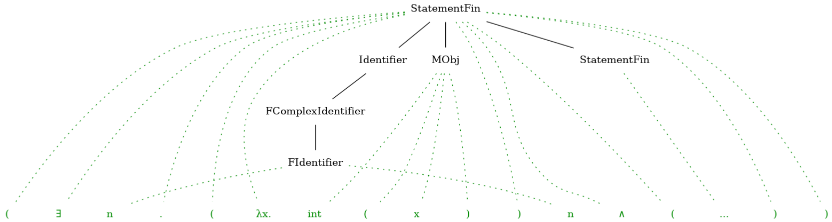
Using GF for Mathematics - Using Identifier in Statement

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



Using GF for Mathematics - Using Identifier in Statement

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



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

$$\downarrow_\beta$$

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

"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