# Abstract Syntax, Finnish, and the Languages of the World

Aarne Ranta
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Kielitieteen päivät, Tampere, 2-4 May 2013.

# Overview

1. Language technology: the state of the art
2. Abstract syntax and GF (Grammatical Framework)
3. Case study: clause formation in Finnish
4. Abstract syntax and the diversity of languages

# 1. State of the art in language technology

*Statistical methods have picked a lot of low-hanging fruit but the need of linguistic knowledge is getting recognized again.*

## The 20-year oscillation

20-year periods statistics vs. linguistic

Since early 1990's: statistic

Pendulum swung too far? (Church 2011)

# The success story of statistic NLP

Jelinek, speech recognition ("every time I fire a linguist my recognition rate goes up")

Brown and Jelinek, Statistical Machine Translation

- back to Shannon & Weaver 1948: translation = cryptanalysis
- millions of times more computing power and data
- Google translate

## What favours statistical methods

- Abundant data availability
- Simple morphology (fewer word forms)
- Fixed word order (fewer n-grams)

All this is a perfect match for English!

# A Finnish tradition

Complex language

Multilinguality (society, school teaching)

Emphasis on grammar in education

## Language technology in Finland: linguistics-based

- Lauri Karttunen: unification grammar, Xerox Finite State Tool
- Kimmo Koskenniemi: two-level morphology
- Fred Karlsson: constraint grammar
- Paul Kiparsky: optimality theory

## The statistical credo

Data-driven = objective

No manual work: ”language technology for the lazy”

Coverage rather than precision: ”something is better than nothing”

# Data sparsity: Finnish inflection forms

(Google translate 19 April 2013)

*yö, yön, yötä, yöksi, yönä, yössä, yöstä, yöhön, yöllä, yöltä, yölle, yöttä, yöt, öiden, öitä, öiksi, öinä, öissä, öistä, öihin, öillä, öiltä, öille, öittä, öine, öin*

*night, night, night, night, night, night, night, night, night, night, nights, Yotta, night, nights, nights nights, nights, nies, nights, nights, nights, with, company against loss, nities, öittä, öine, night*

# Data sparsity: long-distance dependencies 1

Agreement disappears

| | |
|---|---|
| *Their children are smart.* | *Heidän lapsensa ovat älykkäitä.* |
| *Their children are too smart.* | *Heidän lapsensa ovat liian fiksu.* |

# Data sparsity: long-distance dependencies 2

Verb compounds are not recognized

| | |
|---|---|
| *Kunta sanoi sopimuksen irti.* | *The municipality terminated the contract.* |
| *Kunta sanoi lopulta sopimuksen irti.* | *The municipality said in the end the contract.* |

## The linguistic credo

”Don’t guess if you know” (Tapanainen and Voutilainen, 1994)

Knowledge acquired in tradition: ”the shoulders of giants”

Try to understand, not just mimick

## Hybrid systems

The emerging wave in NLP

Language is *both* structures *and* distributions

Statistical models on linguistic structures - not just of strings

Attack data sparsity with abstractions

Guess (only) if you don't know

## The borderline

Known, or possible to know

- morphology
- syntax: agreement, word order

Impossible, or too troublesome to know

- all meanings of all words
- the entire context of disambiguation

# Complementary virtues (and vices)

Original English

- *Harris Ravine, executive vice president of customer satisfaction was named executive vice president, finance and administration.*

Grammar-based MT (GF baseline)

- *Harris Ravine, asiakastyydytyksen suorittava pahepresidentti nimitettiin suorittavaksi pahepresidentiksi, finanssiksi ja hallinnoksi.*

Statistical MT (Google)

- *Harris rotko, varatoimitusjohtaja asiakastyytyväisyys nimettiin varatoimitusjohtaja, talous ja hallinto.*

## Putting it together

Grammar from GF, words from Google

- *Harris Ravine, asiakastyytyväisyyden varatoimitusjohtaja nimitettiin varatoimitusjohtajaksi, taloudeksi ja hallinnoksi.*

Choosing a different parse tree

- *Harris Ravine, asiakastyytyväisyyden varatoimitusjohtaja nimitettiin varatoimitusjohtajaksi, talous ja hallinto.*

# 2. GF and abstract syntax

*The mission of GF is to formalize the grammars of the world and make them usable for computer applications.*

# GF, Grammatical Framework

A programming language for grammars

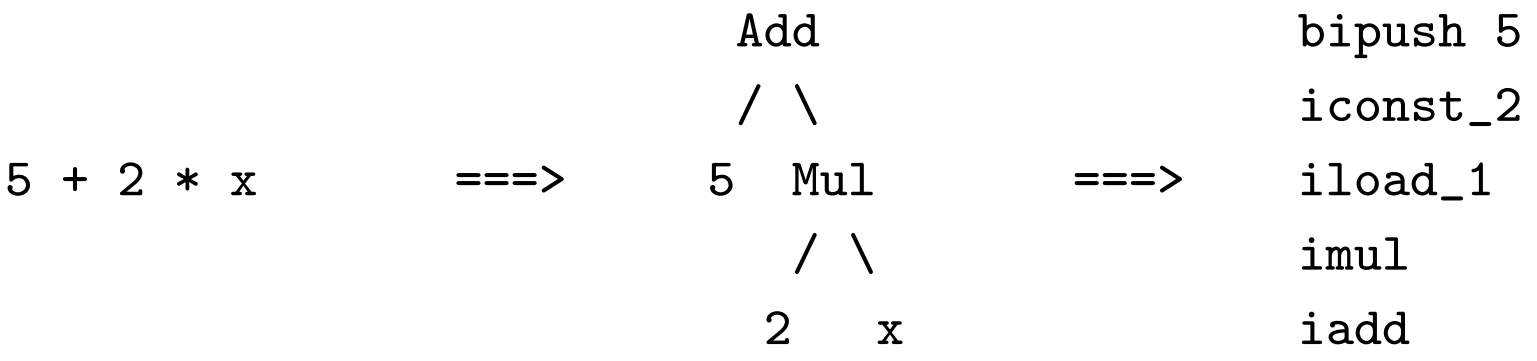
- morphology
- syntax
- semantics

GF grammars are **multilingual**

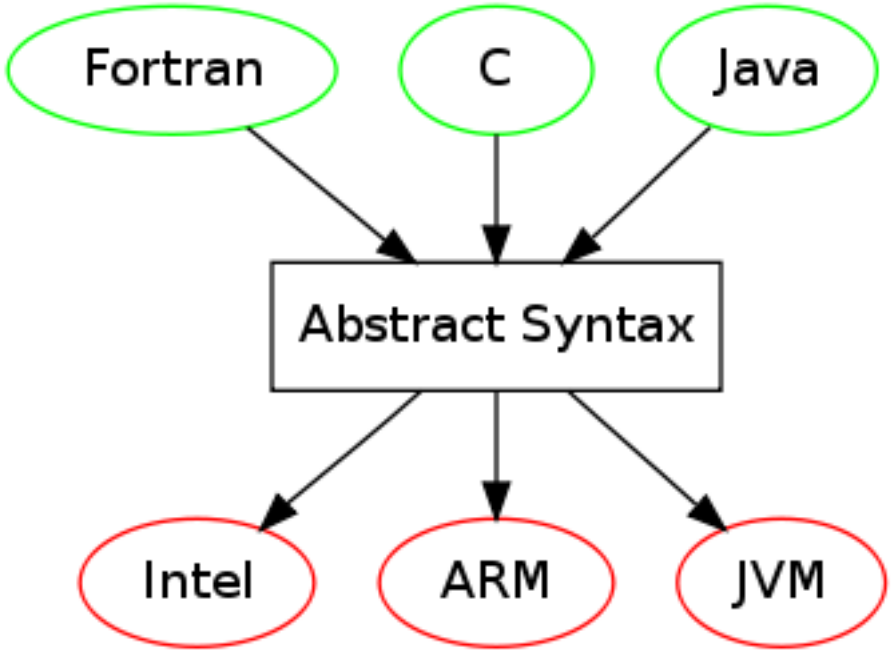
- shared **abstract syntax**
- translation = parsing + linearization

# Abstract syntax in compilers

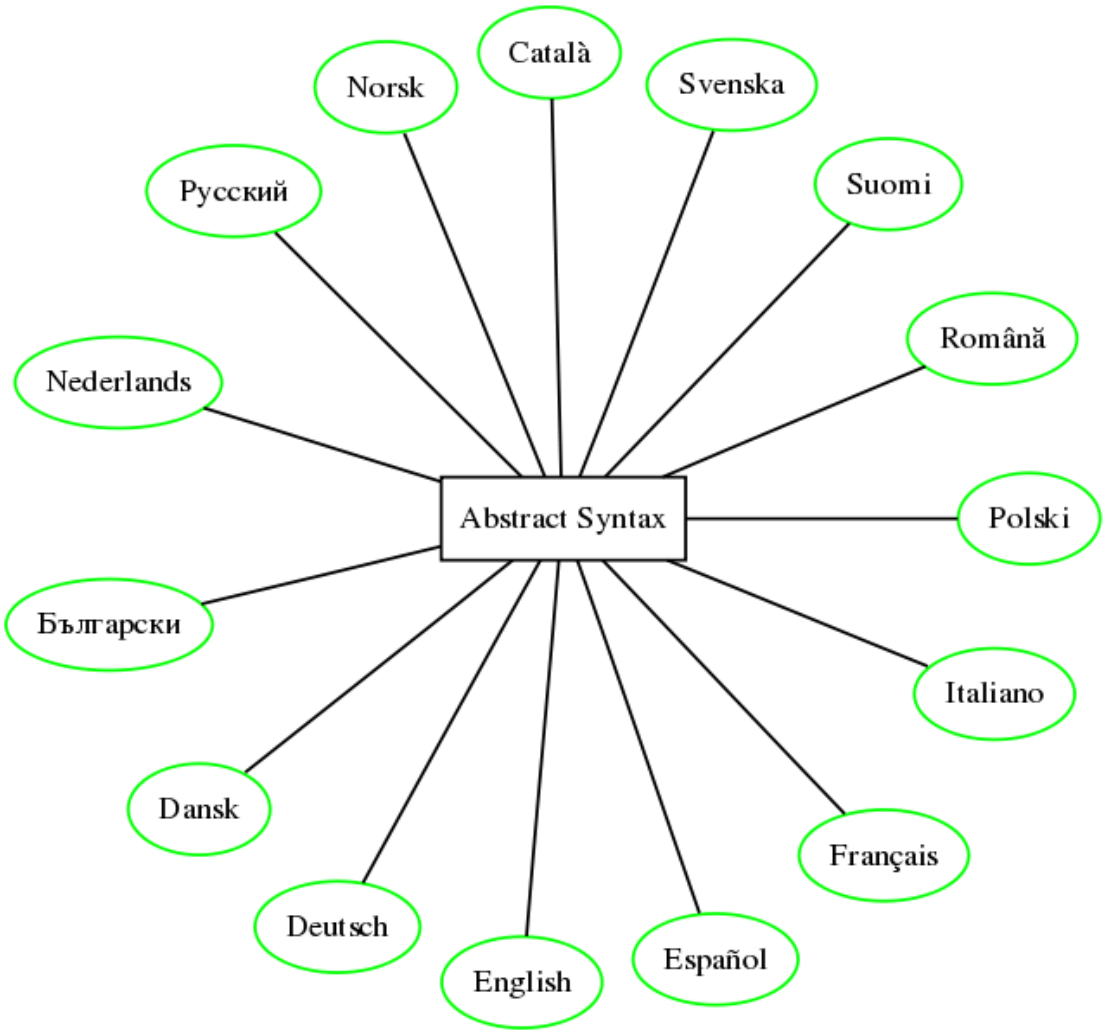
1. Source code (Java)
2. **parsing** to **abstract syntax tree**
3. **linearization** to target code (Java Virtual Machine)



# Multi-source multi-target compilers



# Compiling natural language



# RGL, the GF Resource Grammar Library

Developed since 2001

28 languages

50 authors

Complete morphology, comprehensive syntax

Open source (LGPL/BSD)


Partial coverage
Full coverage

## An introductory demo

http://www.grammaticalframework.org/

GF Cloud -> Minibar -> Grammars: ResourceDemo.pgf

*my new house is small*

# Abstract syntax of the lexicon: word senses

| sense | letter_1 | letter_2 |
|---|---|---|
| English | *letter* | *letter* |
| Finnish | *kirje* | *kirjain* |
| French | *lettre* | *lettre* |
| Swedish | *brev* | *bokstav* |

Similar to **linked wordnets**

# Concrete syntax: add morphology

| sense | letter_1 | letter_2 |
|---|---|---|
| English | *letter, letters* | *letter, letters* |
| Finnish | *kirje, kirjeen, kirjettä,...* | *kirjain, kirjaimen, kirjainta,...* |
| French | *lettre, lettres* FEM | *lettre, letters* FEM |
| Swedish | *brev,brevet,brev,...* NEUTR | *bokstav, bokstaven,...* UTR |

Linked wordnet + morphological lexica

## Expressing this in GF

Abstract syntax: **categories**, **functions**

```
cat N
fun letter_1 : N
fun letter_2 : N
```

Concrete syntax: **linearization types**, **linearization rules**

# Linearization

```
lincat N = {
  s : Number => Str ;    -- table from number to string
  g : Gender             -- inherent gender
  }

lin letter_1 = {
  s = table {
    Sg => "lettre" ;     -- singular form
    Pl => "lettres"      -- plural form
    } ;
  g = Fem                -- the gender is feminine
  }
```

# Linearizing nouns in Finnish

Complex parameter types: 2 * 12 + 2 = 26 noun forms

```
param Case =
    Nom   | Gen   | Part  | Ess   | Transl
  | Iness | Elat  | Illat
  | Adess | Ablat | Allat | Abess

param NForm =
   NCase Number Case | NComit | NInstr
```

```
lincat N = {s : NForm => Str}

lin letter_1 = {s =
  table {
    NCase Sg Nom  => "kirje" ;
    NCase Sg Gen  => "kirjeen" ;
    NCase Sg Part => "kirjettä" ;
    ...
    NInstr        => "kirjein"
    }
 }
```

## Paradigms

Functions from strings to inflection tables

```
kotus1   : Str -> N   -- like "rako"
kotus33  : Str -> N   -- like "avain"
kotus48  : Str -> N   -- like "tarve"
```

Lexicon, more compactly

```
letter_1 = kotus48 "kirje"
letter_2 = kotus33 "kirjain"
```

kotus = paradigm from the word list of *Kotimaisten Kielten Tutkimuskeskus*

# Smart paradigms

Inferring the table from its base form

```
mkN sana = case sana of {
  _  + "o"  => kotus1 sana ;
  _  + "e"  => kotus48 sana ;
  _  + "in" => kotus33 sana ;
  ...
  }
```

In the lexicon, one form is usually enough

```
lin letter_1 = mkN "kirje"
lin letter_2 = mkN "kirjain"
```

Sometimes another form is needed

```
lin teddy = mkN "nalle" "nallen"

lin row   = mkN "rivi" "rivejä"
lin stone = mkN "kivi" "kiviä"
```

87% of nouns and 96% of verbs need only one form.

## Multilingual grammar

One abstract syntax

- common categories
- common functions

Many concrete syntaxes

- different linearization types (number of forms, inherent features)
- different linearization rules

## From words to syntax

Adjectival modification: *arrogant* + *letter* = *arrogant letter*

Abstract syntax: a function that takes two arguments

```
fun Mod : A -> N -> N
```

Concrete syntax: table of complex phrases

English:

```
lin Mod adj noun = {s = table {<n,c> => adj.s ++ noun.s ! n ! c}}
```

Thus we have the table

| **form** | **Sg** | **Pl** |
|---|---|---|
| Nom | *arrogant letter* | *arrogant letters* |
| Gen | *arrogant letter's* | *arrogant letters'* |

| **form** | **Sg** | **Pl** |
|---|---|---|
| Nom | *röyhkeä kirje* | *röyhkeät kirjeet* |
| Gen | *röyhkeän kirjeen* | *röyhkeiden kirjeiden* |
| Part | *röyhkeää kirjettä* | *röyhkeitä kirjeitä* |
| Ess | *röyhkeänä kirjeenä* | *röyhkeinä kirjeinä* |
| Transl | *röyhkeäksi kirjeeksi* | *röyhkeiksi kirjeiksi* |
| Iness | *röyhkeässä kirjeessä* | *röyhkeissä kirjeissä* |
| Elat | *röyhkeästä kirjeestä* | *röyhkeistä kirjeistä* |
| Illat | *röyhkeään kirjeeseen* | *röyhkeihin kirjeisiin* |
| Adess | *röyhkeällä kirjeellä* | *röyhkeillä kirjeillä* |
| Ablat | *röyhkeältä kirjeeltä* | *röyhkeiltä kirjeiltä* |
| Allat | *röyhkeälle kirjeelle* | *röyhkeille kirjeille* |
| Abess | *röyhkettä kirjeettä* | *röyhkeittä kirjeittä* |
| Comit | *röyhkeine* | *kirjeine* |
| Instr | *röyhkein* | *kirjein* |

# Context-free expansion

English: 4 rules

```
N_Sg_Nom ::= A N_Sg_Nom
N_Sg_Gen ::= A N_Sg_Gen
N_Pl_Nom ::= A N_Pl_Nom
N_Pl_Nom ::= A N_Pl_Gen
```

French: 4 rules

```
N_Masc_Sg ::= A_Masc_Sg N_Masc_Sg
```

...

Finnish: 26 rules

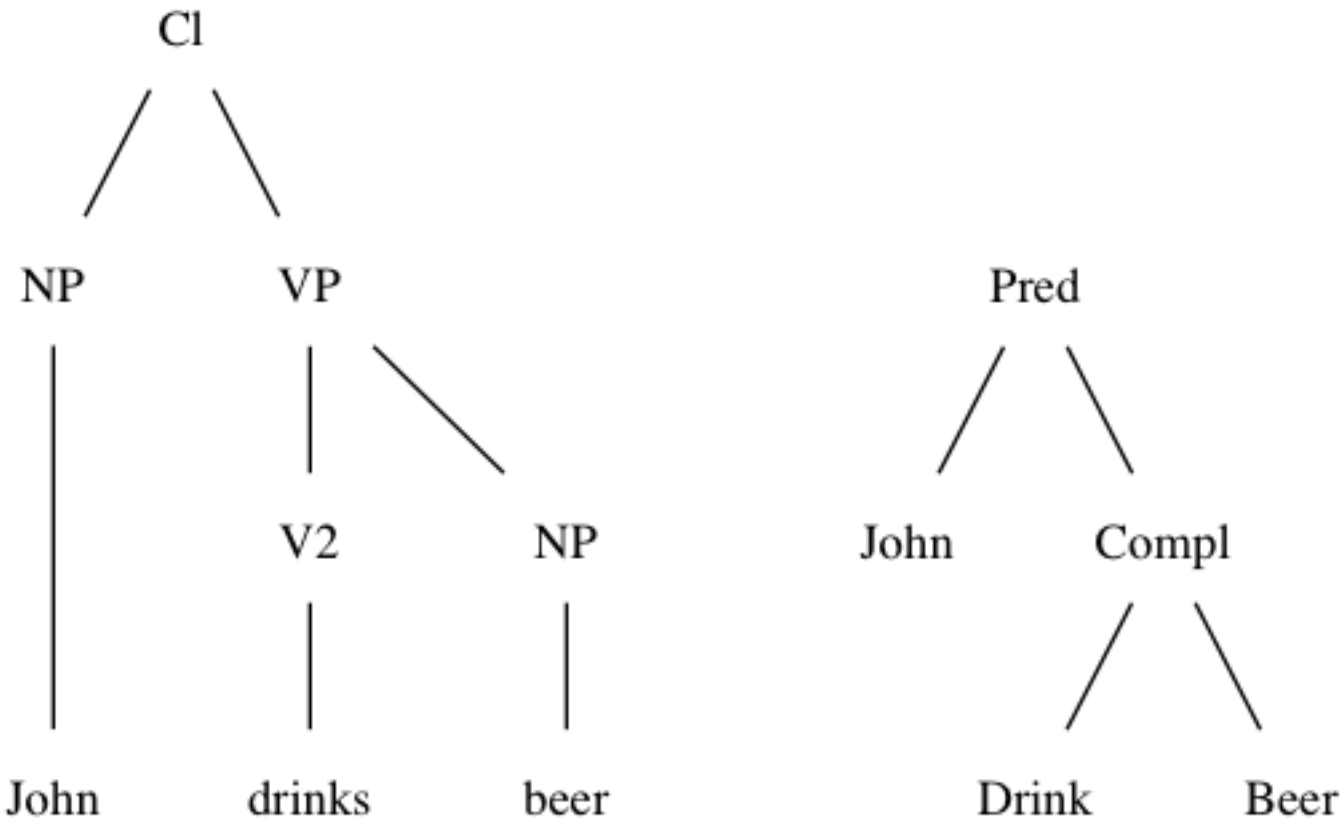
# 3. Case study: subject-verb-object clause formation

*A Finnish clause can appear in thousands of forms.*

# Clause formation: abstract syntax

```
cat Cl  -- clause          e.g. John drinks beer
cat NP  -- noun phrase     e.g. John
cat VP  -- verb phrase     e.g. drink beer
cat V2  -- 2-place verb    e.g. drink

fun Pred  : NP -> VP -> Cl  -- John (drinks beer)
fun Compl : V2 -> NP -> VP  -- drinks beer
```


Cl
NP
VP
V2
NP
John
drinks
beer
Pred
John
Compl
Drink
Beer

# Clause formation: concrete syntax

A clause can vary in tense, polarity, and sentence type

```
lincat Cl = {s : Tense => Pol => SType => Str}
```

| **form** | **Decl** | **Quest** |
|---|---|---|
| Pres Simul Pos | *John drinks beer* | *does John drink beer* |
| Pres Simul Neg | *John doesn't drink beer* | *doesn't John drink beer* |
| Cond Anter | *John would have drunk beer* | *would John have drunk bee* |

etc, (2 * 4) * 2 * 2 = 32 forms

# Tense and mood

```
param TMood =
   IndPres     -- juon     "I drink"
 | IndImpf     -- join     "I drank"
 | Condit      -- joisin   "I would drink"
 | Potent      -- juonen   "I may drink"

param Anteriority =
   Simul       -- juon/join/joisin/juonen
 | Anter       -- olen/olin/olisin/lienen juonut
```

# From noun to noun phrase

NP case system: morphological noun cases + Accusative

```
param NPForm = NPCase Case | NPAcc
```

Mapping the accusative to noun cases depends on polarity, number, and "personality"

| **features** | **noun accusative** | **example** | **English** |
|---|---|---|---|
| neg | partitive | *en osta autoa* | I don't buy a car |
| pos, pl | nominative | *ostan autot* | I buy the cars |
| pos, sg, personal | genitive | *ostan auton* | I buy a car |
| pos, nonpersonal | nominative | *osta auto* | buy a car |

# Verb valency

Subject case:

| | | |
|---|---|---|
| *minä nukun* | "I sleep" | nominative |
| *minulla on auto* | "I have a car" | adessive |
| *minun täytyy nukkua* | "I must sleep" | genitive |

Object case + pre/postposition:

| | | |
|---|---|---|
| *minä näen sinut* | "I see you" | accusative |
| *minä rakastan sinua* | "I love you" | partitive |
| *minä katson sinun perääsi* | "I look after you" | genitive + *perään* |

# Constituent order

Constituent order is not **free**:

- not all orders are possible
- changing order typically changes meaning

The order is **variable**, to express contrast, topicalization, etc.

| | |
|---|---|
| *hän osti auton* | "she bought a car" |
| *auton hän osti* | "it was a car she bought" |
| *auton osti hän* | "it was she who bought the car" |
| *osti+ko hän auton* | "did she buy a car" |

# Topological structure

GF models variable order by a **record**, instead of a fixed string

```
lincat Cl = {
  s : Tense => Pol => {
    subj : Str ;
    verb : Str ;
    obj  : Str ;
    }
  }
```

# Top-level sentences

The fields are combined to a string when a sentence is formed:

| | |
|---|---|
| SVO (unmarked) | `subj verb obj` |
| OSV (fronted object) | `obj subj verb` |
| OVS | `obj verb subj` |
| VSO question (unmarked) | `verb+k0 subj obj` |

(`k0` is the question particle)

# A part of a clause inflection table

| | |
|---|---|
| Pres Pos SVO | *Jussi juo olutta* |
| Past Pos SVO | *Jussi joi olutta* |
| Cond Anter Neg VSO-Quest | *eikö Jussi olisi juonut olutta* |
| Pot Pos SVO-Quest | *Jussiko olutta juonee* |

(2*4 tenses) * (2 polarities) * (2 * 3 orders) * (2 sentence types) = 192 forms...

...multiplied by dozens with adverbial fields and discourse particles

# 4. Abstract Syntax and the Diversity of Languages

*The observed complexity of languages can be compressed by a factor of thousands.*

## How complex is a grammar?

Superficial measure: lines of code and development time for RGL

| language | syntax | morphology | total | months | remarks |
|---|---|---|---|---|---|
| Chinese | 1000 | 100 | 1100 | 1 | not quite finished |
| English | 1000 | 800 | 1800 | 6 | the first RGL |
| Finnish | 1500 | 1500 | 3000 | 6 | one of the first |
| French | 1800 | 1800 | 3600 | 6 | shared Romance |
| Swedish | 1600 | 700 | 2300 | 4 | shared Scandinavian |

The effort for new languages is typically 2 to 4 kLoC, 3 to 6 pm.

## How complex are the grammar rules

A more exact measure: size of source code syntax tree

| language | morphology | syntax | morphology # | syntax # |
|---|---|---|---|---|
| Chinese | I | II | 316 | 666 |
| English | III | IIIII | 1029 | 1468 |
| Finnish | IIIIIIIIIIIIIIIIIIII | IIIIIIIII | 7191 | 3499 |
| French | IIIIIIIII | IIIIIIII | 2999 | 2720 |
| Swedish | IIIIII | IIIII | 2115 | 1404 |

I = 300 nodes. Morphology = mkN+mkV. Syntax = Pred + Compl

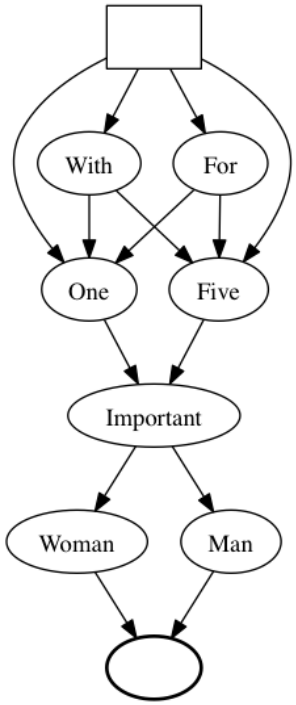
# Context-free expansion

| language | CF rules | CF/GF |
|---|---|---|
| Chinese | 219 | 5 |
| English | 8319 | 189 |
| Finnish | 887297 | 20166 |
| French | 631477 | 14352 |
| Swedish | 3351 | 76 |

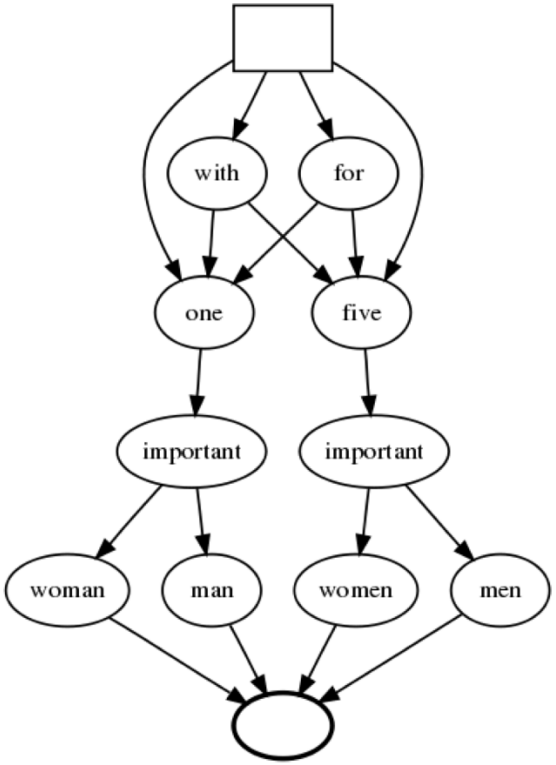
Measured with Miniresource, the core of RGL, 44 GF rules.

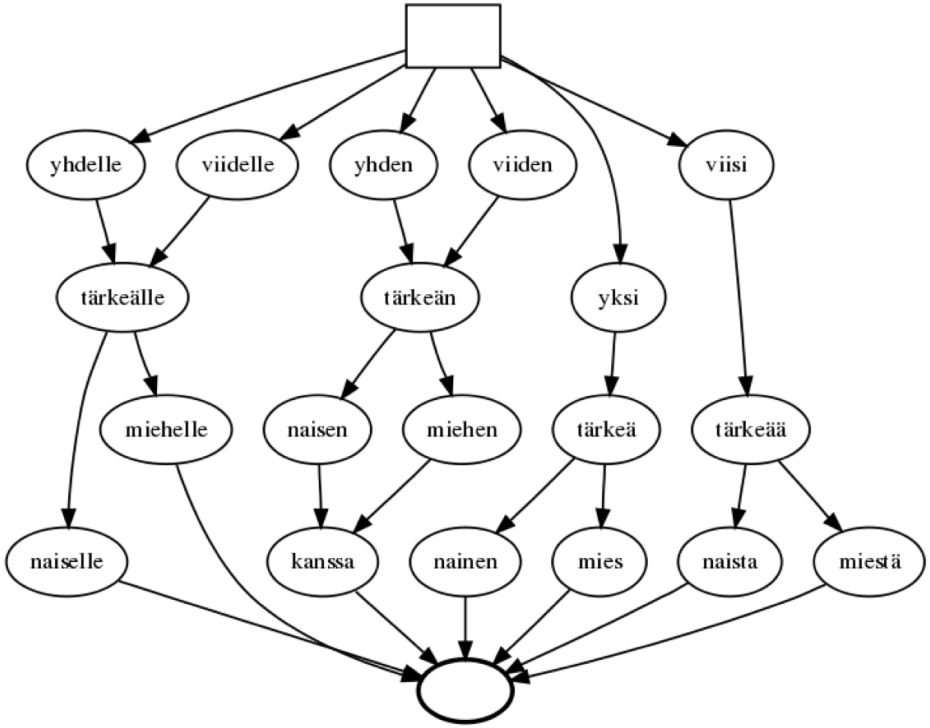
## Another visualization

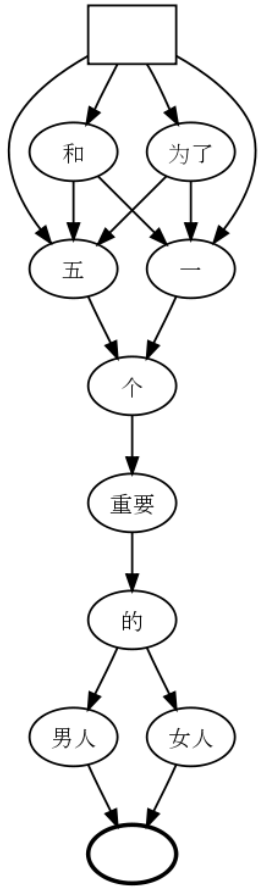
(-|*for*|*with*) (*one*|*five*) *important* (*man*|*woman*)



# Concrete syntax variation



## Using grammar information in statistical models

Factored models

- instead of *öittä*, consider *yö*+N+Pl+Abess

Tree-based models

- instead of *Jussi juo olutta*, consider `Pred Jussi (Compl Juoda Olut)`

Porting treebank statistics

- abstract treebanks can be shared

# Ongoing work in GF

**More languages**: Amharic, Arabic, Estonian, Hebrew, Latin, Nahuatl, Swahili,...

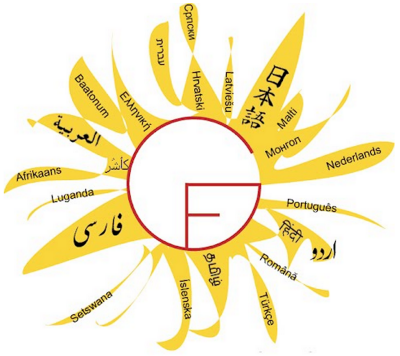
**Large-scale lexical resources**: in particular, associated with linked wordnets

**Large-scale hybrid translation systems**: in particular, English to Finnish, Hindi,...

**Typological studies based on the RGL**

# Contributions welcome

**3rd GF Summer School 2013**
***Scaling up Grammatical Resources***




**Frauenchiemsee island, Bavaria**
**18th–30th August, 2013**

http://school.grammaticalframework.org/2013/

CSLI Studies in
Computational Linguistics

**GRAMMATICAL FRAMEWORK** is a programming language designed for writing grammars, which has the capability of addressing several languages in parallel. This thorough introduction demonstrates how to write grammars in Grammatical Framework and use them in applications such as tourist phrasebooks, spoken dialogue systems, and natural language interfaces. The examples and exercises presented here address several languages, and the readers are shown how to look at their own languages from the computational perspective.

Since the book requires no previous knowledge of linguistics, it can be an effective and useful resource for computer scientists and programmers, while introducing linguists to a novel approach to multilingual grammars inspired by the theory of programming languages.

**Aarne Ranta** is professor of computer science at the University of Gothenburg, Sweden. He is the acting coordinator of the European Union research project MOLTO (Multilingual On-Line Translation), which develops techniques for high-quality translation among fifteen languages.

CSLI
PUBLICATIONS
Center for the Study of
Language and Information
Stanford, California

ISBN-13 978-1-57586-626-0
ISBN-10 1-57586-626-9

Aarne Ranta

**Grammatical Framework**
**Programming with Multilingual Grammars**

CSLI

CSLI Studies in
Computational Linguistics

# Grammatical Framework

## Programming with Multilingual Grammars

Aarne Ranta

# Conclusion

A lot is known about grammar: inflection, agreement, constructions

This knowledge should be formalized, implemented, and shared

Abstract syntax is both practically useful and theoretically interesting