Some Observations About Reader Languages

Frank Drewes
Department of Computing Science
Umeå University, Sweden

Joint work with
S. Bensch, H. Jürgensen, B. van der Merwe (1st part) and
S. Bensch, M. Vollenweiler (2nd part)

FASTAR/Espresso Workshop 2012

Today's Menu

Outline

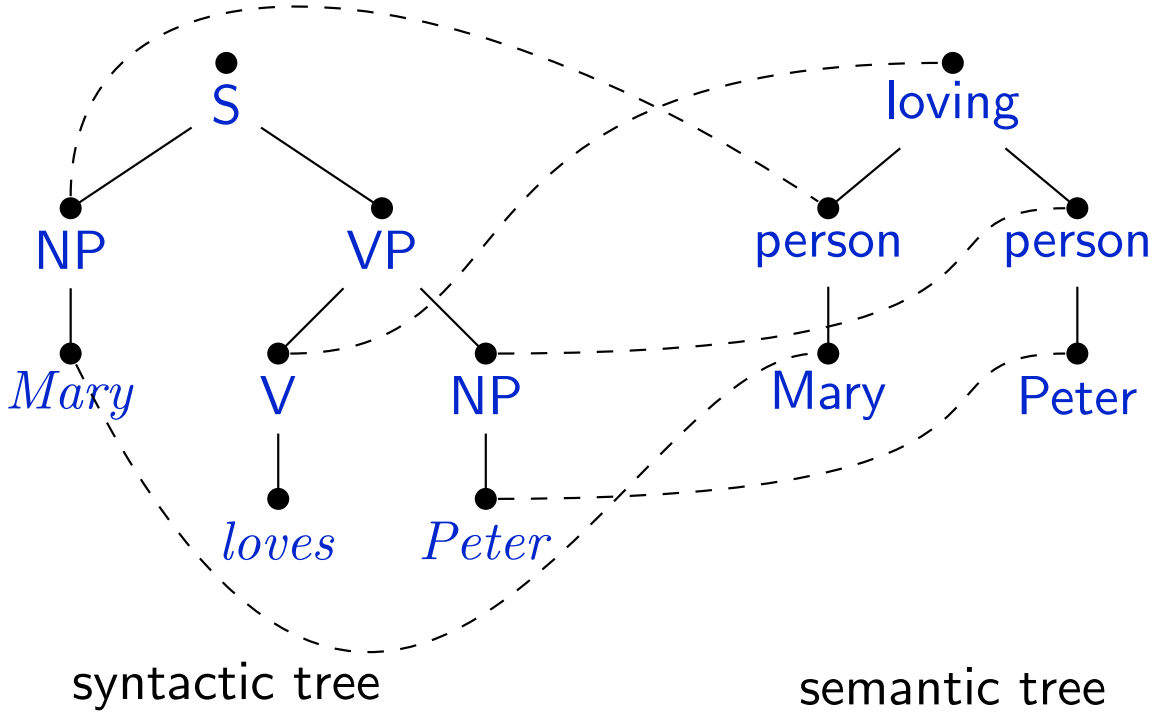
Introduction

Millstream systems are a formal model that tries to capture the interplay between language aspects: morphology, syntax, semantics, pragmatics, ...

- A configuration is a graph representing an analysis of a sentence.
- Readers are graph transformation systems that build a configuration by reading an input string from left to right.
- This is inspired by how we imagine humans to process language.
- One question is which input languages can be accepted by readers?

Outline

A Sample Configuration (Syntax & Semantics)



Components of a Millstream System

The ingredients of a Millstream system:

- k tree languages $L_1, \ldots, L_k$ (representing k language aspects)
- Relation symbols representing different kinds of links.
- Logical formulas restricting the placement of links.

A configuration is any k-tuple of trees $t_1 \in L_1, \ldots, t_k \in L_k$ with additional links, so that the formulas are satisfied.

Outline

Readers Build Configurations Incrementally (1)

- Humans analyze sentences on the fly while hearing them.
- Can we do this algorithmically, building a configuration stepwise as we read a sentence?
- The reader approach:
  - View configurations as graphs.
  - Associate a graph grammar rule with each word.
  - When a word is read, apply the corresponding rule to incorporate the new information into the partial configuration.
  - At the end of the sentence, the configuration should be complete.
- In general, we will have to associate several alternative rules with a word, to cover different roles, ambiguity, etc.

Readers Build Configurations Incrementally (2)

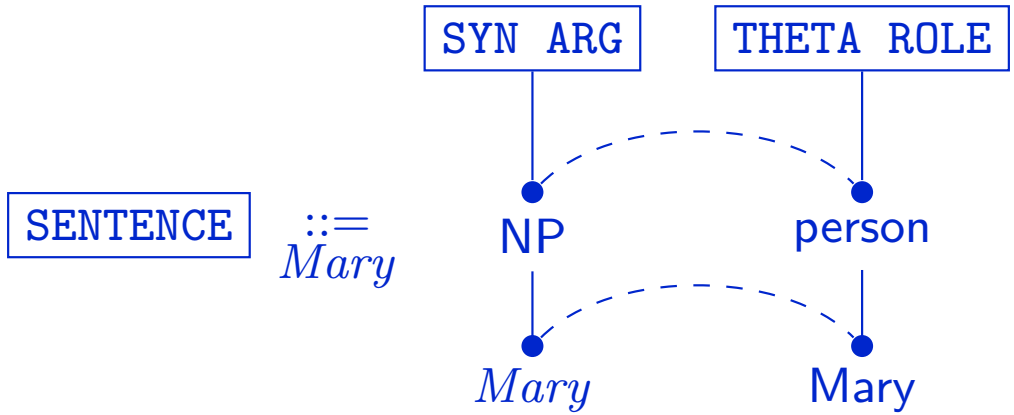
In more detail:

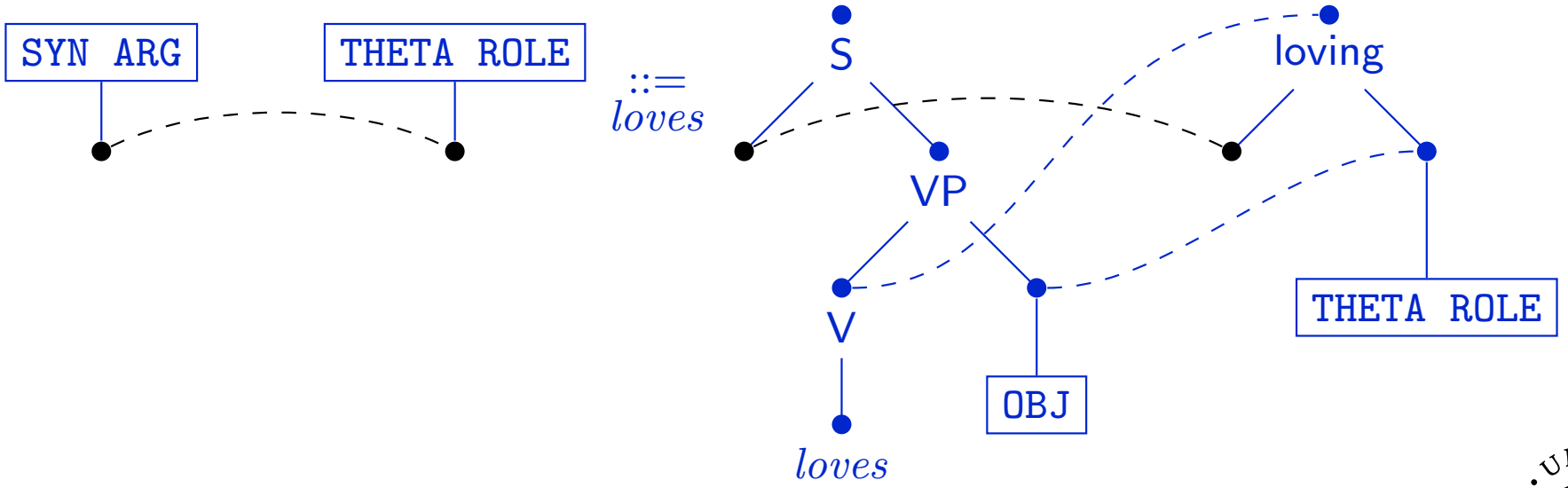
- Each possible input word w is associated with a finite set of graph grammar rules $\Lambda(w)$, its lexicon entries.
- We start from one of finitely many start graphs.
- Reading w means to choose an applicable lexicon entry from $\Lambda(w)$ and apply it to the graph.
- A reading of a sentence $w_1 \cdots w_n$ is a derivation

$$G_0 \Rightarrow_{\Lambda(w_1)} G_1 \Rightarrow_{\Lambda(w_2)} \cdots \Rightarrow_{\Lambda(w_n)} G_n$$

  where G_0 is a start graph and G_n is terminal.
- If there is such a reading, the reader is said to accept $w_1 \cdots w_n$.

Two Examples of Lexicon Entries





Main Result Obtained so Far

The following applies to Millstream systems MS consisting of regular tree grammars and MSO formulas.

Theorem

For nonterminal bounded readers whose generated configurations are of bounded node degree, it is decidable whether the configurations resulting from reading arbitrary sentences are configurations of MS.

Proof idea:

- The lexicon entries of such readers can be turned into (linear) hyperedge-replacement rules (i.e., context-free graph grammar rules).
- Thus, the resulting set of configurations is a hyperedge-replacement graph language.
- For such a graph language, it is decidable whether all graphs it contains satisfy a given MSO formula.

Outline

Reader Languages

Which input sentences (like *Peter likes Mary*) does a reader accept?

Definition (reader language)

The reader language $RL(\mathcal{R})$ of a reader $\mathcal{R}$ consists of all strings accepted by $\mathcal{R}$, i.e., all $w_1 \cdots w_n$ such that

$$G_0 \Rightarrow_{\Lambda(w_1)} G_1 \Rightarrow_{\Lambda(w_2)} \cdots \Rightarrow_{\Lambda(w_n)} G_n$$

for a startgraph G_0 of $\mathcal{R}$ and some terminal graph G_n.

Reader Languages and Szilard Languages

- If we give each rule of a grammar a (unique) name in some alphabet Σ, every derivation is denoted by a string in Σ^*.
- The Szilard language of such a grammar is the set of all strings denoting its terminating derivations.
- Reader languages are similar, where rule names are input words, except that several rules can be associated with the same input word.
- Thus, reader languages are homomorphic images of Szilard languages, using letter-to-letter homomorphisms.
- These languages are also known as label languages.

Reader Languages of Context-Free Readers

Observation: The reader languages of context-free readers are the label languages of context-free grammars.

Consequences (because of known results):

- Reader languages of context-free readers are incomparable with CFL. (Ex.: $a^n b^n$ is not a reader language, but $\{w \mid |w|_a = |w|_b = |w|_c\}$ is.)
- In particular, this class is not closed under intersection with regular languages.
  (Note: General reader languages are closed under intersection with regular languages.)
- $RL(\mathcal{R})$ is regular for linear context-free readers $\mathcal{R}$.

Complexity of Reader Languages of Context-Free Readers

Some easy observations regarding the complexity of reader languages:

- Szilard languages of context-free grammars are in L.
  (Just keep one counter for each nonterminal.)
- Consequently, $RL(\mathcal{R}) \in$ NL for context-free readers.
  (Apply the previous idea while inverting the homomorphism by "guessing".)
- If $\mathcal{R}$ is nonterminal bounded, this uses only constant space
  $\Rightarrow RL(\mathcal{R})$ is regular for nonterminal bounded context-free readers $\mathcal{R}$.

Outline

Most Questions Remain Open (1)

What are the reader languages of non-context-free readers?

- They contain every context-free language – using a reader that turns the input string into a proper parse tree!
- By the earlier remarks, the containment is strict.
- Can we characterize the class of reader languages of general readers?
- . . . or of non-context-free subclasses?
  E.g., what if we only allow for terminal context?

Most Questions Remain Open (2)

Cancellation rules: We may add a set $\Lambda(\epsilon)$ of "ϵ-transitions" with terminal right-hand sides, allowing to cancel optional parts of a sentence.

- Does this make readers more powerful?
- If so, does it also enlarge the set of reader languages?
- What if the reader is context-free? (For label languages of context-free grammars, ϵ-labels do not increase power.)

Thank you for listening!