An Abstract Categorial Grammar Approach to the Discourse Modeling

Aleksandre Maskharashvili

25 October 2017
Göteborg, Sweden
What is ‘discourse’

Various kinds of natural language acts

- spoken
- written
  - monologues texts
  - dialogs
- multi-modal
- ...
What is ‘discourse’

Various kinds of natural language acts

- spoken
- written
  - monologues texts
  - dialogs
- multi-modal
- ...

Discourse
Discourse

What is ‘discourse’

Various kinds of natural language acts

- spoken
- written
  - monologues texts
  - dialogs
- multi-modal
- ...

Discourse VS Sentence

- Anaphoric expressions
- Interpretations of sentences are not to be considered in isolation
- ...

**Discourse**

### What is ‘discourse’

Various kinds of natural language acts

- spoken
- written
  - monologues texts
  - dialogs
- multi-modal
- ...

### Discourse VS Sentence

- Anaphoric expressions
- Interpretations of sentences are not to be considered in isolation
- ...

### Example

\[ \pi_0 \text{ Fred is grumpy,} \]
\[ \pi_1 \text{ because his wife is absent for a week.} \]
\[ \pi_2 \text{ This shows how much he loves her.} \]
**Discourse**

**What is ‘discourse’**

Various kinds of natural language acts

- spoken
- written
  - monologues texts
  - dialogs
- multi-modal
- ...

**Discourse VS Sentence**

- Anaphoric expressions
- Interpretations of sentences are not to be considered in isolation
- ...

**Example**

\[ \pi_0 \text{ Fred is grumpy,} \]
\[ \pi_1 \text{ because his wife is absent for a week.} \]
\[ \pi_2 \text{ This shows how much he loves her.} \]
**Discourse**

**What is ‘discourse’**

Various kinds of natural language acts

- spoken
- written
  - monologues texts
  - dialogs
- multi-modal
- ...

**Discourse VS Sentence**

- Anaphoric expressions
- Interpretations of sentences are not to be considered in isolation
- ... discourse structure

**Example**

\[ \pi_0 \text{ Fred is grumpy,} \]
\[ \pi_1 \text{ because his wife is absent for a week.} \]
\[ \pi_2 \text{ This shows how much he loves her.} \]
Discourse

Discourse structure

- Rhetorical Structure Theory (RST) (Mann & Thompson, 1987)
- Segmented Discourse Representation Theory (SDRT) (Asher & Lascarides, 2003)
Discourse

Discourse structure

- Rhetorical Structure Theory (RST) (Mann & Thompson, 1987)
- Segmented Discourse Representation Theory (SDRT) (Asher & Lascarides, 2003)

A discourse and its structure

- A discourse consists of utterances - minimal discourse units

[Fred is grumpy]₀  [his wife is absent for a week]₁.  [This shows how much he loves her]₂.
A discourse consists of utterances - minimal discourse units

Utterances are connected to each other

Fred is grumpy because his wife is absent for a week. This shows how much he loves her.
Discourse

Discourse structure

- Rhetorical Structure Theory (RST) (Mann & Thompson, 1987)
- Segmented Discourse Representation Theory (SDRT) (Asher & Lascarides, 2003)

A discourse and its structure

- A discourse consists of utterances - minimal discourse units
- Utterances are connected to each other – this creates larger discourse units (DUs)

\[
Fred \text{ is grumpy}_0 \text{ because } [his \text{ wife is absent for a week}]_1 \in [This \text{ shows how much he loves her}]_2.
\]
Discourse

Discourse structure

- Rhetorical Structure Theory (RST) (Mann&Thompson, 1987)
- Segmented Discourse Representation Theory (SDRT) (Asher&Lascarides, 2003)

A discourse and its structure

- A discourse consists of utterances - minimal discourse units
- Utterances are connected to each other – this creates larger discourse units (DUs)
- A connection between two discourse units is provided by a rhetorical (discourse) relation

[\text{Fred is grumpy}]_0 \text{ because } [\text{his wife is absent for a week}]_1. \text{ Commentary } [\text{This shows how much he loves her}]_2.
A discourse and its structure

- A discourse consists of utterances - minimal discourse units
- Utterances are connected to each other – this creates larger discourse units (DUs)
- A connection between two discourse units is provided by a rhetorical (discourse) relation
- Discourse connections create the discourse structure

[Fred is grumpy]₀ because [his wife is absent for a week]₁. [This shows how much he loves her]₂.
Examples of Discourse Structure
Left-branching trees

(1) \([\text{Fred is grumpy}]_0 \text{ because } [\text{his wife is absent for a week}]_1. \ [\text{This shows how much he loves her}]_2.\)
Examples of Discourse Structure
Left-branching trees

(1) \([\text{Fred is grumpy}_0 \text{ because } \text{[his wife is absent for a week]_1}. \epsilon \text{ [This shows how much he loves her]}_2].\)

Grammar: trees and substitution
Examples of Discourse Structure
Left-branching trees

(1) [Fred is grumpy]₀ because [his wife is absent for a week]₁. ε [This shows how much he loves her]₂.

Grammar: trees and substitution
Examples of Discourse Structure
Left-branching trees

(1) [Fred is grumpy]₀ because [his wife is absent for a week]₁. ε [This shows how much he loves her]₂.
Examples of Discourse Structure
Left-branching trees

(1) [Fred is grumpy]₀ because [his wife is absent for a week]₁. ε [This shows how much he loves her]₂.

Grammar: trees and substitution
Examples of Discourse Structure

Left-branching trees

(1) \([\text{Fred is grumpy}]_0 \) because \([\text{his wife is absent for a week}]_1\). \(\epsilon \) \([\text{This shows how much he loves her}]_2\).

Grammar : trees and substitution
Examples of Discourse Structure
Right-branching trees

(2)  \([\text{Fred is grumpy}]_0 \text{ because } [\text{he lost his keys}]_1. \text{ Moreover, } [\text{he failed an exam}]_2.\)
Examples of Discourse Structure
Right-branching trees

(2) \[[\text{Fred is grumpy}]_0 \text{ because } [\text{he lost his keys}]_1. \text{ Moreover, } [\text{he failed an exam}]_2.\]
Examples of Discourse Structure

Right-branching trees

(2) [Fred is grumpy]₀ because [he lost his keys]₁. Moreover, [he failed an exam]₂.

Grammar: trees and substitution
Examples of Discourse Structure
Right-branching trees

(2) [Fred is grumpy]₀ because [he lost his keys]₁. Moreover, [he failed an exam]₂.
(2) [Fred is grumpy]₀ because [he lost his keys]₁. Moreover, [he failed an exam]₂.
Examples of Discourse Structure
Right-branching trees

(2) \([\text{Fred is grumpy}]_0 \text{ because } [\text{he lost his keys}]_1. \text{ Moreover, } [\text{he failed an exam}]_2.\)

Grammar: trees and substitution

```
because
  Fred is grumpy
  he lost his keys

moreover
  he failed an exam```

Examples of Discourse Structure
Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]₀ because [he did not sleep well]₁. [He had nightmares]₂.
Examples of Discourse Structure
Directed Acyclic Graphs (DAGs)

(3) \([\text{Fred is grumpy}]_0 \text{ because } [\text{he did not sleep well}]_1. \epsilon [\text{He had nightmares}]_2.\)

\[
\begin{array}{c}
\text{Fred is grumpy} \quad \text{because} \quad \text{he did not sleep well} \quad \epsilon \quad \text{he had nightmares}
\end{array}
\]
Examples of Discourse Structure
Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]₀ because [he did not sleep well]₁. ε [He had nightmares]₂.

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
Examples of Discourse Structure
Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]₀ because [he did not sleep well]₁. ∈ [He had nightmares]₂.

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
Examples of Discourse Structure
Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]₀ because [he did not sleep well]₁. ∈ [He had nightmares]₂.

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.

Discourse structure ≠ parse tree
Examples of Discourse Structure
Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]₀ because [he did not sleep well]₁.  ε [He had nightmares]₂.

Discourse structure ≠ parse tree
Problem of the discourse-syntax interface
Clause-medial connectives

Another problem of the discourse-syntax interface
Clause-medial connectives

Another problem of the discourse-syntax interface

Clause-initial connective

(5) \([\text{Fred went to the supermarket}]_0. \text{ Then [he went to the cinema]}_1.\)
Clause-medial connectives
Another problem of the discourse-syntax interface

Clause-initial connective

(5) [Fred went to the supermarket]₀. Then [he went to the cinema]₁.

Clause-medial connective

(6) [Fred went to the supermarket]₀. [He then went to the cinema]₁.
Clause-medial connectives

Another problem of the discourse-syntax interface

Clause-initial connective

(5) [Fred went to the supermarket]₀. Then [he went to the cinema]₁.

Clause-medial connective

(6) [Fred went to the supermarket]₀. [He then [went to the cinema]₁.]

- At the clause-level, a clause-medial connective is a VP modifier.
Clause-medial connectives
Another problem of the discourse-syntax interface

Clause-initial connective

(5) [Fred went to the supermarket]₀. Then [he went to the cinema]₁.

Clause-medial connective

(6) [Fred went to the supermarket]₀. [He then \textit{went to the cinema}]¹.

- At the clause-level, a clause-medial connective is a VP modifier.
- A discourse unit is not only a VP, but the entire clause.
Clause-medial connectives
Another problem of the discourse-syntax interface

Clause-initial connective

(5) [Fred went to the supermarket]₀. Then [he went to the cinema]₁.

Clause-medial connective

(6) [Fred went to the supermarket]₀. [He then \underline{went to the cinema}]₁.

- At the clause-level, a clause-medial connective is a VP modifier.
- A discourse unit is not only a VP, but the entire clause.

Mismatch between the clause-level and discourse-level analyses of clause-medial connectives
Grammatical approach to the discourse-syntax interface

Discourse formalisms
## Discourse formalisms and their properties

<table>
<thead>
<tr>
<th>Discourse formalism</th>
<th>Parsing</th>
<th>Generation</th>
<th>Tree</th>
<th>DAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-TAG</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>(Danlos, 1998)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-LTAG</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>(Webber &amp; Joshi, 1998)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-STAG</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(Danlos, 2011)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Grammatical approach to the discourse-syntax interface

Discourse formalisms

<table>
<thead>
<tr>
<th>Discourse formalisms and their properties</th>
<th>Parsing</th>
<th>Generation</th>
<th>Tree</th>
<th>DAG</th>
<th>Clause-medial conn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-TAG</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(Danlos, 1998)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-LTAG</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(Webber &amp; Joshi, 1998)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-STAG</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>(Danlos, 2011)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ad hoc encoding of clause-medial connectives

Prohibits having reversible grammars – both parsing & generation

Prohibits generalizations
Grammatical approach to the discourse-syntax interface

Discourse formalisms

### Discourse formalisms and their properties

<table>
<thead>
<tr>
<th></th>
<th>Parsing</th>
<th>Generation</th>
<th>Tree</th>
<th>DAG</th>
<th>Clause-medial conn.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>G-TAG</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>(Danlos, 1998)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>D-LTAG</strong></td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>(Webber &amp; Joshi, 1998)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>D-STAG</strong></td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>(Danlos, 2011)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Ad hoc encoding of clause-medial connectives

- Prohibits having reversible grammars – both parsing & generation
- Prohibits generalizations
Discourse modeling with ACGs

Goals

- Overcome the problems related to clause-medial connectives
- Analyze the problems
- Solve them using ACGs
- Study existing discourse formalisms with ACGs + incorporate clause-medial connectives
- Develop tractable encodings
- Discourse parsing
- Discourse generation
Discourse modeling with ACGs

Goals

- Overcome the problems related to clause-medial connectives
  - Analyze the problems
  - Solve them using ACGs
Discourse modeling with ACGs

Goals

- Overcome the problems related to clause-medial connectives
  - Analyze the problems
  - Solve them using ACGs
- Study existing discourse formalisms with ACGs + incorporate clause-medial connectives
Discourse modeling with ACGs

Goals

- Overcome the problems related to clause-medial connectives
  - Analyze the problems
  - Solve them using ACGs
- Study existing discourse formalisms with ACGs + incorporate clause-medial connectives
- Develop tractable encodings
  - Discourse parsing
  - Discourse generation
Plan

1. TAG

2. Discourse formalisms
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG

3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG

4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics
Outline

1. TAG

2. Discourse formalisms
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG

3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG

4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
Operations on trees –
Generated structures –
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
 ▶ Initial trees: domain of locality

Operations on trees –
Generated structures –

Example

```
NP  S  VP
Fred ↓ NP ↓ V
      ↓  
      
laughs
```

NP  S  VP
Fred ↓ NP ↓ V
      ↓  
      
laughs
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
  ► **Initial trees**: domain of locality

Operations on trees – **substitution**

Generated structures –

**Example**

```
NP
   Fred

S
   NP
      ↓
   VP
      ↓
   V
      laughs
```

Example sentence: Fred laughs loudly.
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
- Initial trees: domain of locality
- Auxiliary trees: recursion

Operations on trees – substitution

Generated structures –

Example

```
NP
  ↓
Fred

S
  ↓
NP  VP
  ↓  ↓
NP  VP*
  ↓  ↓
Adv  VP*
    ↓
  loudly

V
  ↓
laughs
```
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
- Initial trees: domain of locality
- **Auxiliary trees**: recursion

Operations on trees – substitution and **adjunction**

Generated structures –

**Example**

```
NP       S
  Fred   NP  VP
       NP  VP
         VP* Adv
            VP
            loud
```

```
  V
  laughs
```
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
- Initial trees: domain of locality
- Auxiliary trees: recursion

Operations on trees – substitution and adjunction

Generated structures – derived trees.

Example

```
NP
Fred

S
NP ↓ VP

VP
Adv VP*

V laughs

S
NP VP

Adv VP

V laughs
```

Fred loudly laughs

Fred adv loudly V laughs
Tree-Adjoining Grammar (TAG) (Joshi et al., 1975)

Elementary trees –
- Initial trees: domain of locality
- Auxiliary trees: recursion

Operations on trees – substitution and adjunction

Generated structures – derived trees. Their by-products – derivation trees

Example
Synchronous TAG (STAG)
(Shieber & Schabes, 1990)

- Elementary structures - pairs of TAG trees \( \langle t_{\text{syn}}, t_{\text{sem}} \rangle \)
- Correspondence between nodes of \( t_{\text{syn}} \) and \( t_{\text{sem}} \)
- Parallel operations on corresponding nodes

**Example**

- \( \langle S \circled{1} \downarrow 3 \circled{2} \rangle \)
- \( \langle NP \downarrow 3 \circled{3} \rangle \)
- \( \langle VP \circled{2} \rangle \)
- \( \langle V \downarrow \rangle \)
- \( \langle laughs \rangle \)
- \( \langle \alpha_{laughs} \rangle \)

\( (a) \)
Synchronous TAG (STAG)
(Shieber&Schabes, 1990)

- Elementary structures - pairs of TAG trees $\langle t_{syn}, t_{sem} \rangle$
- Correspondence between nodes of $t_{syn}$ and $t_{sem}$
- Parallel operations on corresponding nodes

Example

(a) $\alpha_{\text{laughs}}$

(b) $\gamma_{\text{Fred}}$

(c) $\delta_{\text{loudly}}$

Figure – STAG elementary structures
STAG Example

(a) $\alpha_{\text{laughs}}$

(b) $\gamma_{\text{Fred}}$

(c) $\delta_{\text{loudly}}$
STAG Example

(d) $\alpha_{\text{laughs}}$

(e) $\gamma_{\text{Fred}}$

(f) $\delta_{\text{loudly}}$

$\alpha_{\text{laughs}}$

$\gamma_{\text{Fred}}$

$\delta_{\text{loudly}}$
STAG Example

(g) \( \alpha_{\text{laughs}} \)

(h) \( \gamma_{\text{Fred}} \)

(i) \( \delta_{\text{loudly}} \)

\[ S \rightarrow NP \quad VP \]
\[ NP \rightarrow Fred \]
\[ VP \rightarrow Adv \quad VP^* \]
\[ Adv \rightarrow loudly \]
\[ VP^* \rightarrow F \quad R \quad T \]
\[ F \rightarrow laugh \quad fred \]
\[ Adv \rightarrow loudly \]
\[ VP \rightarrow V \quad laugh \]
\[ NP \rightarrow \text{laughs} \]
\[ VP \rightarrow \text{laughs} \]
Outline

1. TAG

2. Discourse formalisms
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG

3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG

4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics
D-LTAG, G-TAG, and D-STAG elementary trees
D-LTAG, G-TAG, and D-STAG elementary trees

Discourse-level grammar: D-LTAG

Subordinating conjunctions

- \( Du \)
- \( Du \downarrow because \)

Coordinating conjunctions

- \( Du \)
- \( Du^* and Du \downarrow \)

Clause-level grammar

- \( S \)
- \( S \downarrow because \)
- \( S \downarrow although \)
- \( S \downarrow while \)
D-LTAG, G-TAG, and D-STAG elementary trees

Discourse-level grammar: D-LTAG

- Subordinating conjunctions
  - Du
  - Du↓ because Du↓

- Coordinating conjunctions
  - Du* and Du↓

- Adverbials
  - Du
  - / \ then Du*
D-LTAG, G-TAG, and D-STAG elementary trees

**Discourse-level grammar : D-LTAG**

- **Subordinating conjunctions**
  - Du
  - Du ↓ because
  - Du ↓

- **Coordinating conjunctions**
  - Du*
  - and
  - Du ↓

- **Adverbials**
  - Du
  - then
  - Du*

**Discourse-level grammar : G-TAG**

- **Subordinate conjunctions**
  - Du
  - Du ↓ PP
  - conj
  - Du ↓

- **Adverbial connectives**
  - Du
  - adv
  - Du ↓
  - Du ↓
D-LTAG, G-TAG, and D-STAG elementary trees

**Discourse-level grammar : D-LTAG**

Subordinating conjunctions
- **Du**
  - **Du**↓ *because* **Du**↓

Coordinating conjunctions
- **Du**
  - **Du*** and **Du**↓

Adverbials
- **Du**
  - **Du*** then **Du***

**Discourse-level grammar : G-TAG**

Subordinate conjunctions
- **Du**
  - **Du**↓ *PP*
    - PP ↓ *conj* **Du**↓

Adverbial connectives
- **Du**
  - **Du**↓ *adv* **Du**↓

**Discourse-level grammar : D-STAG**

Postposed conjunctions
- **Du**
  - **Du*** Punct DC **Du**↓
    - Punct ↓ *conj* **Du**↓

Preposed conjunctions
- **Du**
  - **Du**↓ *DC* **Du**↓
    - DC ↓ *conj* **Du**↓

Adverbial connectives
- **Du**
  - **Du*** Punct DC **Du**↓
    - Punct ↓ *adv* **Du**↓
D-LTAG, G-TAG, and D-STAG elementary trees

**Discourse-level grammar : D-LTAG**

- **Subordinating conjunctions**
  - Du
  - Du ↓ because Du ↓

- **Coordinating conjunctions**
  - Du
  - Du* and Du ↓

- **Adverbials**
  - Du
  - then Du*

**Clause-level grammar**

- S
  - S* because S ↓

**Discourse-level grammar : G-TAG**

- **Subordinate conjunctions**
  - Du
  - Du ↓ PP conj
  - Du ↓

- **Adverbial connectives**
  - Du
  - Du ↓ adv
  - Du ↓

**Clause-level grammar**

- S
  - S ↓ although S ↓

**Discourse-level grammar : D-STAG**

- **Postposed conjunctions**
  - Du
  - Du
  - Du* Punct DC Du ↓ conj
  - Du ↓

- **Preposed conjunctions**
  - Du
  - DC
  - Du Punct Du* ↓ conj
  - Du ↓

- **Adverbial connectives**
  - Du
  - Du* Punct DC Du ↓
  - Du* Punct DC Du ↓ adv

**Clause-level grammar**

- S
  - while S ↓ S*
While she was eating lunch, she saw a dog

Syntactic analysis

```
  α_saw
  /   \  
α_she  α_dog  β_while
     \   \  
      β_eating
  \     \  
 α_she  β_was  α_lunch
```

Extraction of trees and construction of an input for a discourse parser
Extraction of trees and construction of an input for a discourse parser

While she was eating lunch, she saw a dog

Syntactic analysis

Constructing elementary trees of D-LTAG
While she was eating lunch, she saw a dog

Syntactic analysis

Constructing elementary trees of D-LTAG

Du

\[ d_{\text{saw}} \]
While she was eating lunch, she saw a dog

Syntactic analysis

Constructing elementary trees of D-LTAG
While she was eating lunch, she saw a dog

Syntactic analysis

Constructing elementary trees of D-LTAG
An approach to clause-medial connectives

Elementary tree

```
Du
  ↓
Du
  ↓
ADV
  ↓
adv
```

A possible analysis

How to encode such a constraint with TAG or sl-MCTAG?
An approach to clause-medial connectives

Elementary tree

A possible analysis

How to encode such a constraint with TAG or sl-MCTAG?
An approach to clause-medial connectives

How to encode such a constraint with TAG or sl-MCTAG?
Discourse Synchronous TAG (D-STAG)
Discourse Synchronous TAG (D-STAG)

Characteristics

- Based on Synchronous TAG (STAG)
- Generates DAGs as discourse structures
Discourse Synchronous TAG (D-STAG)

Characteristics
- Based on Synchronous TAG (STAG)
- Generates DAGs as discourse structures

Elementary trees - Syntax

Adverbial connectives

\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]

Postposed conjunctions

\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]

Preposed conjunctions

\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
\[ \text{Du} \]
Input (string of clauses and connectives):

\[ C_0 \text{ Conn}_1 \ C_1 \text{ Conn}_2 \ C_2 \]
A number of attachment sites with the same yield causes a high ambiguity in parsing. They are needed for obtaining various semantic interpretations.
A number of attachment sites with the same yield causes a high ambiguity in parsing. They are needed for obtaining various semantic interpretations.
D-STAG semantics

\[ \Phi' \mathcal{R} = \lambda X. Y. X(\lambda x. Y(\lambda y. \mathcal{R}(x, y))) \]

\[ (t \rightarrow t) \rightarrow t \triangleq ttt \]
D-STAG semantics

\[ \Phi'' R = \lambda X \ Y \ \mathcal{P}.X(\lambda x. Y(\lambda y. \mathcal{R}(x, y) \land P(x))) \]
Example: [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
Example: [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
Example: [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
Outline

1. TAG

2. Discourse formalisms
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG

3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG

4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics
Abstract Categorial Grammar (ACG)
(de Groote, 2001)

Main Features

- ACGs are a grammatical framework
Abstract Categorial Grammar (ACG)  
(de Groote, 2001)

Main Features

- ACGs are a grammatical framework
- An ACG \( G \) generates two languages:
  - The abstract language \( A(G) \)
  - The object language \( O(G) \)
Abstract Categorial Grammar (ACG)
(de Groote, 2001)

**Main Features**

- ACGs are a **grammatical framework**
- An ACG $\mathcal{G}$ generates **two languages**:
  - The **abstract language** $\mathcal{A}(\mathcal{G})$
  - The **object language** $\mathcal{O}(\mathcal{G})$

**Abstract language**: Admissible structures (parse structures, derivations)

**Object language**: An interpretation of the abstract language
Abstract Categorial Grammar (ACG)
(de Groote, 2001)

Main Features

- ACGs are a grammatical framework
- An ACG $\mathcal{G}$ generates two languages:
  - The abstract language $A(\mathcal{G})$
  - The object language $O(\mathcal{G})$

Abstract language: Admissible structures (parse structures, derivations)
Object language: An interpretation of the abstract language

Basic properties

Modularity: Both languages are of the same nature – sets of linear $\lambda$-terms
Abstract Categorial Grammar (ACG)
(de Groote, 2001)

<table>
<thead>
<tr>
<th>Main Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ACGs are a <strong>grammatical framework</strong></td>
</tr>
<tr>
<td>• An ACG ( \mathcal{G} ) generates <strong>two languages</strong> :</td>
</tr>
<tr>
<td>▶ The <strong>abstract language</strong> ( \mathcal{A}(\mathcal{G}) )</td>
</tr>
<tr>
<td>▶ The <strong>object language</strong> ( \mathcal{O}(\mathcal{G}) )</td>
</tr>
</tbody>
</table>

**Abstract language**: Admissible structures (parse structures, derivations)

**Object language**: An **interpretation** of the abstract language

<table>
<thead>
<tr>
<th>Basic properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modularity</strong></td>
</tr>
<tr>
<td>Both languages are of the same nature – sets of linear ( \lambda )-terms :</td>
</tr>
<tr>
<td>ACGs can be composed</td>
</tr>
</tbody>
</table>
Abstract Categorial Grammar (ACG)
(de Groote, 2001)

Main Features

- ACGs are a grammatical framework
- An ACG $\mathcal{G}$ generates two languages:
  - The abstract language $\mathcal{A}(\mathcal{G})$
  - The object language $\mathcal{O}(\mathcal{G})$

Abstract language: Admissible structures (parse structures, derivations)
Object language: An interpretation of the abstract language

Basic properties

Modularity: Both languages are of the same nature – sets of linear $\lambda$-terms:
ACGs can be composed

Parsing: 2nd order ACGs are reversible (Salvati 2005), (Kanazawa 2007)
From TAG derivation to TAG derived trees

<table>
<thead>
<tr>
<th>Derivation trees</th>
<th>Their interpretations as derived trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
<tr>
<td>S</td>
<td>A</td>
</tr>
<tr>
<td>λ</td>
<td>A</td>
</tr>
<tr>
<td>S</td>
<td>Adv</td>
</tr>
<tr>
<td>λ</td>
<td>x</td>
</tr>
<tr>
<td>S</td>
<td>VP</td>
</tr>
<tr>
<td>λ</td>
<td>x</td>
</tr>
</tbody>
</table>
From TAG derivation to TAG derived trees

Derivation trees  Their interpretations as derived trees

NP
  |  Fred
From TAG derivation to TAG derived trees

Derivation trees

\[ C_{Fred} : \text{NP} \]

Their interpretations as derived trees

\[ \text{NP}_1 Fred \]

\[ \text{NP} \]

\[ Fred \]
From TAG derivation to TAG derived trees

Derivation trees

- $C_{Fred} : NP$
- $C_{left} : S_A \rightarrow VP_A \rightarrow NP \rightarrow S$

Their interpretations as derived trees

- $NP_1 Fred$
- $\lambda S A np.S(S_2 np (A (VP_1 (V_1 left))))$
From TAG derivation to TAG derived trees

Derivation trees

\[ C_{Fred} : NP \]
\[ C_{left} : S_A \rightarrow VP_A \rightarrow NP \rightarrow S \]
\[ C_{then} : S_A \rightarrow S_A \]

Their interpretations as derived trees

\[ NP_1 Fred \]
\[ \lambda S \ A \ np. S(S_2 \ np \ (A (VP_1 (V_1 left)))) \]
\[ \lambda A \ x. A (S_2 (Adv_1 then) x) \]
From TAG derivation to TAG derived trees

Derivation trees

- $C_{Fred} : \text{NP}$
- $C_{left} : S_A \rightarrow VP_A \rightarrow \text{NP} \rightarrow S$
- $C_{then}^S : S_A \rightarrow S_A$
- $C_{then}^{VP} : VP_A \rightarrow VP_A$

Their interpretations as derived trees

- $\text{NP}_1 Fred$
- $\lambda S\ A\ np.\ S(S_2\ np\ (A\ (VP_1\ (V_1\ left))))$
- $\lambda A\ x.\ A\ (S_2\ (Adv_1\ then)\ x)$
- $\lambda A\ x.\ A\ (V_2\ x\ (Adv_1\ then))$
From TAG derivation to TAG derived trees

Derivation trees

[C_{Fred} : NP]
[C_{left} : S_A \rightarrow VP_A \rightarrow NP \rightarrow S]
[C_{then}^S : S_A \rightarrow S_A]
[C_{then}^{VP} : VP_A \rightarrow VP_A]
[I_X : X_A]

Their interpretations as derived trees

NP_1 Fred
\lambda S \ A \ np. S(S_2 \ np \ (A (VP_1 (V_1 left))))
\lambda A \ x. A (S_2 (Adv_1 then) x)
\lambda A \ x. A (V_2 x (Adv_1 then))
\lambda x. x
From TAG derivation to TAG derived trees

Derivation trees
\[ C_{Fred} : \text{NP} \]
\[ C_{left} : S_A \rightarrow VP_A \rightarrow \text{NP} \rightarrow S \]
\[ C_{then} : S_A \rightarrow S_A \]
\[ C_{VP} : VP_A \rightarrow VP_A \]
\[ I_X : X_A \]

Their interpretations as derived trees
\[ \text{NP}_1 \text{ Fred} \]
\[ \lambda S \ A \ np. S(S_2 \ np \ (A (VP_1 (V_1 \ left)))) \]
\[ \lambda A \ x.A \ (S_2 \ (Adv_1 \ then) \ x) \]
\[ \lambda A \ x.A \ (V_2 \ x \ (Adv_1 \ then)) \]
\[ \lambda x.x \]

\[ t_1 = C_{left} (C^S_{then} I_s) I_V C_{Fred} \]

\[ \mathcal{G}_{yield} \circ \mathcal{G}_{derived \ trees}(t_1) = \text{then}+, +Fred + \text{left} \]
From TAG derivation to TAG derived trees

Derivation trees

- $C_{Fred} : NP$
- $C_{left} : S_A \rightarrow VP_A \rightarrow NP \rightarrow S$
- $C_{then} : S_A \rightarrow S_A$
- $C_{VP} : VP_A \rightarrow VP_A$
- $I_X : X_A$

Their interpretations as derived trees

- $NP_1 Fred$
- $\lambda S A np. S(S_2 np (A (VP_1 (V_1 left))))$
- $\lambda A x. A (S_2 (Adv_1 then) x)$
- $\lambda A x. A (V_2 x (Adv_1 then))$
- $\lambda x. x$

$t_2 = C_{left} I_S (C_{then} I_V) C_{Fred}$

$\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees}(t_2) = Fred + then + left$
TAG as ACGs

TAG derivation trees $\Lambda(\Sigma_{TAG})$
TAG as ACGs

TAG derivation trees $\Lambda(\Sigma_{TAG})$

Derived trees $\Lambda(\Sigma_{trees})$
TAG as ACGs

$G_{\text{derived trees}} \Lambda(\Sigma_{TAG})$

$\Lambda(\Sigma_{trees})$

$\Lambda(\Sigma_{TAG})$
TAG as ACGs

\[ \Lambda(\Sigma_{\text{trees}}) \rightarrow \mathcal{G}_{\text{derived trees}} \rightarrow \Lambda(\Sigma_{\text{TAG}}) \]

\[ \Lambda(\Sigma_{\text{string}}) \rightarrow \text{Derived trees} \rightarrow \Lambda(\Sigma_{\text{TAG}}) \]
TAG as ACGs

\[ \Lambda(\Sigma_{\text{trees}}) \quad \Lambda(\Sigma_{\text{TAG}}) \]

\[ \mathcal{G}_{\text{derived trees}} \]

\[ \mathcal{G}_{\text{yield}} \]
TAG as ACGs + Montague semantics (Pogodalla, 2004)

- TAG derivation trees $\Lambda(\Sigma_{TAG})$
- Derived trees $\Lambda(\Sigma_{trees})$
- Logical formulas $\Lambda(\Sigma_{logic})$
- Strings $\Lambda(\Sigma_{string})$

$\mathcal{G}_{\text{derived trees}}$ to $\mathcal{G}_{\text{TAG sem.}}$

$\mathcal{G}_{\text{yield}}$ from $\mathcal{G}_{\text{derived trees}}$ to $\mathcal{G}_{\text{yield}}$
Vocabulary $\Sigma_{\text{logic}}$ for Montague semantics

<table>
<thead>
<tr>
<th>term</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>fred</td>
<td>$e$</td>
</tr>
<tr>
<td>laugh</td>
<td>$e \to t$</td>
</tr>
<tr>
<td>loudly</td>
<td>$t \to t$</td>
</tr>
<tr>
<td>grumpy</td>
<td>$e \to t$</td>
</tr>
</tbody>
</table>
Vocabulary $\Sigma_{\text{logic}}$ for Montague semantics

<table>
<thead>
<tr>
<th>Word</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>fred</td>
<td>$e$</td>
<td></td>
</tr>
<tr>
<td>laugh</td>
<td>$e \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>$&amp;$</td>
<td>$t \rightarrow t \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>$\Rightarrow$</td>
<td>$t \rightarrow t \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>$\exists$</td>
<td>$(e \rightarrow t) \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>loudly</td>
<td>$t \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>grumpy</td>
<td>$e \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>$\lor$</td>
<td>$t \rightarrow t \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>$\neg$</td>
<td>$t \rightarrow t$</td>
<td></td>
</tr>
<tr>
<td>$\forall$</td>
<td>$(e \rightarrow t) \rightarrow t$</td>
<td></td>
</tr>
</tbody>
</table>
TAG as ACG
Interpretation of TAG derivation trees into Montague semantics

\[ C_{\text{fred}} : \text{NP} \quad \chi^o P. P_{\text{fred}} \]
TAG as ACG
Interpretation of TAG derivation trees into Montague semantics

\[ C_{\text{fred}} : \text{NP} \quad \chi^o P. P_{\text{fred}} \]

\[ C_{\text{laugh}} : S_a \rightarrow \text{VP}_a \rightarrow \text{NP} \rightarrow S \quad \chi^o s_a v_a \text{ subje. } s_a (\text{subje } (v_a (\chi^o x. ((\text{lough } x)))))) \]
TAG as ACG
Interpretation of TAG derivation trees into Montague semantics

$C_{\text{fred}} : \text{NP}$
$\lambda^o P. P \text{ fred}$

$C_{\text{laugh}} : \text{S} \rightarrow \text{VP} \rightarrow \text{NP} \rightarrow \text{S}$
$\lambda^o s_a v_a \text{ subje. } s_a (\text{ subje (} v_a (\lambda^o x. ((\text{ lough } x))) ) ) )$

$C_{\text{loudly}} : \text{VP} \rightarrow \text{VP}$
$\lambda^o v p_a r. v p_a (\lambda^o x. \text{ loudly (} r x ) )$

$I_{\chi_A}$
$\lambda x. x$
TAG as ACG
Interpretation of TAG derivation trees into Montague semantics

\[ C_{\text{fred}} : \text{NP} \quad \quad \quad \lambda^o P. P \text{ fred} \]
\[ C_{\text{laugh}} : \text{S}_A \rightarrow \text{VP}_A \rightarrow \text{NP} \rightarrow \text{S} \quad \quad \lambda^o s_a v_a \text{ subje. } s_a (\text{subje} (v_a (\lambda^o x. ((\text{lough } x))))) \]
\[ C_{\text{loudly}} : \text{VP}_A \rightarrow \text{VP}_A \quad \quad \lambda^o v p_a r. \text{ v } p_a (\lambda^o x. \text{ loudly } (r x)) \]
\[ I_{X_A} \quad \quad \quad \lambda x. x \]

Example
Fred loudly laughs.

\[ L^{\text{sem}}_{\text{TAG}} (C_{\text{laugh}} I_{S_A} (C_{\text{loudly}} I_{\text{VP}_A}) C_{\text{Fred}}) \rightarrow_\beta \text{ loudly (laugh Fred)} : t \]
Outline

1. TAG

2. Discourse formalisms
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG

3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG

4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics
D-STAG as ACGs

- **Strings** $\Lambda(\Sigma_{\text{string}})$
- Derived trees $\Lambda(\Sigma_{\text{trees}})$
- Derivation trees $\Lambda(\Sigma_{\text{TAG}})$
- D-STAG derivation trees $\Lambda(\Sigma_{\text{D-STAG}})$
- Logical formulas $\Lambda(\Sigma_{\text{logic}})$

Diagrams:
- $\mathcal{G}_{\text{yield}}$ from **Strings** to Derived trees
- $\mathcal{G}_{\text{derived trees}}$ from Derived trees to Derivation trees
- $\mathcal{G}_{\text{disc-clause int.}}$ from Derivation trees to D-STAG derivation trees
- $\mathcal{G}_{\text{D-STAG sem.}}$ from D-STAG derivation trees to Logical formulas
Discourse grammar with ACGs

Adverbial Connectives:

```
Du
  |   DC  |  Du
  |   |   |   |
Du  |   |   |   |
  |   |   |   |
Du*  Punct  adv  Du
```

3rd order ACG – no polynomial parsing property!
Discourse grammar with ACGs

Adverbial Connectives:

\[
d_{\text{then}}: D_u A \rightarrow D_u A \rightarrow D_u A \rightarrow D_u \rightarrow D_u A
\]
Discourse grammar with ACGs

Adverbial Connectives:

\[ d_{\text{then}}: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du} \rightarrow \text{Du}_A \]
Discourse grammar with ACGs

Adverbial Connectives:

\[ d_{then} : D_u A \rightarrow D_u A \rightarrow D_u A \rightarrow D_u \rightarrow D_u A \]
Discourse grammar with ACGs

Adverbial Connectives:

\[ d_{\text{then}}: Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A \]
Discourse grammar with ACGs

Adverbial Connectives:

\[
\begin{array}{c}
\text{Du} \\
\text{Du} \\
\text{Du*} \rightarrow \text{Punct} \rightarrow \text{DC} \rightarrow \text{Du} \\
\text{Du} \rightarrow \text{Du} \rightarrow \text{Du} \rightarrow \text{VP} \rightarrow \text{VP} \rightarrow \text{adv} \rightarrow \text{Du}
\end{array}
\]

\[
d_{\text{then}}: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du} \rightarrow \text{Du}_A
\]

\[
d_{\text{then}}^\text{v}: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow (\text{VP}_A \rightarrow \text{Du}) \rightarrow \text{Du}_A
\]
Discourse grammar with ACGs

Adverbial Connectives:

\[ d_{\text{then}}: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du} \rightarrow \text{Du}_A \]

\[ d'_{\text{then}}: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow (\text{VP}_A \rightarrow \text{Du}) \rightarrow \text{Du}_A \]
Discourse grammar with ACGs

Adverbial Connectives:

```
<table>
<thead>
<tr>
<th></th>
<th>Du</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Du*</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Punct</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td></td>
</tr>
<tr>
<td>Du</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>adv</td>
<td>DU</td>
</tr>
</tbody>
</table>
```

$d_{then}$: $Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A$

$d_{then}^V$: $Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow (VP_A \rightarrow Du) \rightarrow Du_A$

$d_{then}^S$
Discourse grammar with ACGs

Adverbial Connectives:

\[ d_{then}: Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A \]
\[ d^v_{then}: Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow (VP_A \rightarrow Du) \rightarrow Du_A \]
\[ ds_{then} \]
Discourse grammar with ACGs

Adverbial Connectives:

\[
\begin{align*}
\text{D}_1 & \rightarrow \text{D}_1 \\
\text{D}_1^* & \rightarrow \text{Punct} \\
\text{DC} & \rightarrow \text{adv} \\
\text{S} & \rightarrow \text{adv} \\
\end{align*}
\]

*d_\text{then}*: \text{D}_{\text{A}} \rightarrow \text{D}_{\text{A}} \rightarrow \text{D}_{\text{A}} \rightarrow \text{D}_1 \rightarrow \text{D}_{\text{A}}

*d_\text{then}^\text{V}*: \text{D}_{\text{A}} \rightarrow \text{D}_{\text{A}} \rightarrow \text{D}_{\text{A}} \rightarrow (\text{VP}_\text{A} \rightarrow \text{D}_1) \rightarrow \text{D}_{\text{A}}

*d_\text{then}^\text{S}*: \text{D}_{\text{A}} \rightarrow \text{D}_{\text{A}} \rightarrow \text{D}_{\text{A}} \rightarrow (\text{S}_\text{A} \rightarrow \text{D}_1) \rightarrow \text{D}_{\text{A}}
Discourse grammar with ACGs

Adverbial Connectives:

\[
\begin{aligned}
&\text{Du} \\
&\text{Du} \\
&\text{Du}^* \quad \text{Punct} \quad \text{DC} \quad \text{Du} \\
&\text{Du}^* \\
&\text{Punct} \\
&\text{DC} \\
&\text{Du} \\
&\text{Du} \\
&\text{S} \\
&\text{adv} \\
&\text{S}
\end{aligned}
\]

\[
\begin{aligned}
d_{\text{then}} &: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du} \rightarrow \text{Du}_A \\
d_{\text{then}}^V &: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow (\text{VP}_A \rightarrow \text{Du}) \rightarrow \text{Du}_A \\
d_{\text{then}}^S &: \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow (\text{S}_A \rightarrow \text{Du}) \rightarrow \text{Du}_A
\end{aligned}
\]
Discourse grammar with ACGs

Adverbial Connectives:

\[
\begin{align*}
\text{Du}^* & \quad \text{Punct} & \quad \text{DC} & \quad \text{Du} \\
\text{Du} & \quad . & \quad \text{adv} & \quad \text{Du}
\end{align*}
\]

3rd order ACG – no polynomial parsing property!
D-STAG as ACGs

2nd order ACGs
D-STAG as ACGs

2nd order ACGs

From D-STAG derivation trees to TAG derivation trees

\[ D_u \equiv S_A \rightarrow VP_A \rightarrow S \]
D-STAG as ACGs

2nd order ACGs

From D-STAG derivation trees to TAG derivation trees

\[ D_u := S_A \circ VP_A \circ S \]

\[(DC \triangleq D_{u_A} \circ D_{u_A} \circ D_{u_A} \circ D_{u} \circ D_{u_A}) \]
D-STAG as ACGs

2nd order ACGs

From D-STAG derivation trees to TAG derivation trees

\[
\text{Du} := S_A \rightarrow VP_A \rightarrow S
\]

\[
(DC \triangleq Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A)
\]

\[
d^S_{then}: DC := \lambda d_1 \ d_2 \ d_3 \ d_u. \ cons \ d_1 \ d_2 \ d_3 \ (d_u \ C^S_{then} \ l_{VP})
\]
**D-STAG as ACGs**

2nd order ACGs

---

**From D-STAG derivation trees to TAG derivation trees**

\[ \text{Du} := S_A \rightarrow VP_A \rightarrow S \]

\[(\text{DC} \triangleq \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du}_A \rightarrow \text{Du} \rightarrow \text{Du}_A)\]

\[d^S_{\text{then}}: \text{DC} := \lambda^o d_1 d_2 d_3 d_u. \text{cons d}_1 d_2 d_3 (d_u C^S_{\text{then}} l_{VP})\]

\[d^V_{\text{then}}: \text{DC} := \lambda^o d_1 d_2 d_3 d_u. \text{cons d}_1 d_2 d_3 (d_u l_S C^{\text{VP}}_{\text{then}})\]
D-STAG as ACGs

2nd order ACGs

From D-STAG derivation trees to TAG derivation trees

\[ Du := S_A \rightarrow VP_A \rightarrow S \]

\[ (DC \triangleq Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A) \]

\[ d^S_{then}: DC \quad := \lambda d_1 \ d_2 \ d_3 \ d_u. \ cons \ d_1 \ d_2 \ d_3 \ (d_u \ C^S_{then} \ l_{VP}) \]

\[ d^V_{then}: DC \quad := \lambda d_1 \ d_2 \ d_3 \ d_u. \ cons \ d_1 \ d_2 \ d_3 \ (d_u \ l_{S} \ C^{VP}_{then}) \]
(4)  [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
D-STAG as ACGs – semantic interpretations

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.
D-STAG as ACGs – semantic interpretations

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.

\[ (\exists! x. \text{(supermarket} x) \land (\text{go-to fred} x)) \land \]
\[ (\exists! x. \text{(fridge} x) \land (\text{empty} x))) \land \]
\[ (\exists! x. (\text{movies} x) \land (\text{go-to fred} x))) \]

D-STAG as ACGs

Then

because

\[ F_0 \]
\[ F_1 \]
\[ F_2 \]
D-STAG as ACGs – semantic interpretations

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to movies]₂.

D-STAG as ACGs – How to express that the two are the same?

(EXPLANATION)

(∃!x. (supermarket x) ∧ (go-to fred x))
(∃!x. (fridge x) ∧ (empty x)))
∧

(NARRATION)

(∃!x.(supermarket x) ∧ (go-to fred x))
(∃!x. (movies x) ∧ (go-to fred x)))
Labeled semantic interpretations

$\Sigma^\ell_{\text{logic}}$ - a signature for labeled semantics

Atomic types: \{e, t, $\ell$\}
Labeled semantic interpretations

$\Sigma_{\text{logic}}^{\ell}$ - a signature for labeled semantics

Atomic types: $\{e, t, \ell\}$

Predicates have one additional argument of type $\ell$ for labels
Labeled semantic interpretations

$\Sigma^\ell_{\text{logic}}$ - a signature for labeled semantics

Atomic types: $\{e, \ t, \ \ell\}$

Predicates have one additional argument of type $\ell$ for labels

Argument of discourse relations are of type $\ell$ (labels)
Labeled semantic interpretations

$\Sigma^\ell_{\text{logic}}$ - a signature for labeled semantics

Atomic types: $\{e, t, \ell\}$

Predicates have one additional argument of type $\ell$ for labels

Argument of discourse relations are of type $\ell$ (labels)

fred, he : e

EXPLANATION : $\ell \rightarrow \ell \rightarrow \ell \rightarrow t$

sleep, bad-mood, exam : $e \rightarrow \ell \rightarrow t$

CONTINUATION : $\ell \rightarrow \ell \rightarrow \ell \rightarrow t$

love, miss, fail : $e \rightarrow e \rightarrow \ell \rightarrow t$

NARRATION : $\ell \rightarrow \ell \rightarrow \ell \rightarrow t$

$\forall, \exists, \exists! : (e \rightarrow t) \rightarrow t$

$\exists_l : (\ell \rightarrow t) \rightarrow t$

...
Labeled semantic trees

Unlabeled \((t \rightarrow t) \rightarrow t \triangleright ttt\)

\[
\begin{align*}
\Phi'' \mathcal{R} &= \lambda X. \lambda Y. \lambda P. X (\lambda x. (Y (\lambda y. (\mathcal{R} x y) \land P(x)))))
\end{align*}
\]
Labeled semantic trees

Unlabeled \((t \to t) \to t \triangleq ttt\)

\[
\begin{align*}
\Phi'' \mathcal{R} &= \lambda X. \lambda Y. \lambda P. X (\lambda x. (Y (\lambda y. (\mathcal{R} x y) \land P(x))))
\end{align*}
\]

Labeled \((\ell \to t) \to t \triangleq \ell tt\)

\[
\begin{align*}
\Phi' \mathcal{R}_\ell &= X Y P. \exists I. X(\lambda x. Y(\lambda y. (P x) \land (\mathcal{R}_\ell x y I)))
\end{align*}
\]
Example

(2) [Fred is grumpy]_0 because [he lost his keys]_1. Moreover, [he failed an exam]_2.
Example

(2) [Fred is grumpy]_0 because [he lost his keys]_1. Moreover, [he failed an exam]_2.
Example

(2)  [Fred is grumpy]_0 because [he lost his keys]_1. Moreover, [he failed an exam]_2.
Example

(2) [Fred is grumpy]₀ because [he lost his keys]₁. Moreover, [he failed an exam]₂.

\[
d_1 = d_{init.\text{anchor}} \ C_0 \ (d_{because} \ I_Du \ I_Du \ I_Du \ (d_{anchor} \ C_1 \ (d_{moreover} \ I_Du \ I_Du \ I_Du \ (d_{anchor} \ C_2 \ I_Du))))
\]
Example

(2) [Fred is grumpy]₀ because [he lost his keys]₁. Moreover, [he failed an exam]₂.

\[
d_1 = d_{\text{init.anchor}} C_0 (d_{\text{because}} l_{\text{Du}} l_{\text{Du}} l_{\text{Du}} (d_{\text{anchor}} C_1 (d_{\text{moreover}} l_{\text{Du}} l_{\text{Du}} l_{\text{Du}} l_{\text{Du}} (d_{\text{anchor}} C_2 l_{\text{Du}}))))
\]

\[
= \exists \ell l_0 l_{R_1} l_0: \text{grumpy}(fred) \land (\exists \ell l_1 l_{R_2} (\exists ! x. l_1: \text{keys}(x) \land l_1: \text{lose}(fred, x)) \land
(\exists \ell l_2 (\exists ! x. l_2: \text{exam}(x) \land l_2: \text{fail}(fred, x) \land (l_{R_2}: \phi_{\text{Cont.}}(l_1, l_2) \land l_{R_1}: \phi_{\text{Expl.}}(l_0, l_{R_2})))))
\]
Example

(2) \([\text{Fred is grumpy}]_0 \) because \([\text{he lost his keys}]_1 \). Moreover, \([\text{he failed an exam}]_2 \).

\[
d_1 = d_{\text{init.anchor}} \ C_0 \ (d_{\text{because}} \ l_{\text{Du}} \ l_{\text{Du}} \ l_{\text{Du}} \ (d_{\text{anchor}} \ C_1 \ (d_{\text{moreover}} \ l_{\text{Du}} \ l_{\text{Du}} \ l_{\text{Du}} \ (d_{\text{anchor}} \ C_2 \ l_{\text{Du}}))))
\]
\[
:= \exists \ell \ l_0 \ l_{\mathcal{R}_1}. \ l_0: \text{grumpy}(\text{fred}) \land (\exists \ell \ l_1 \ l_{\mathcal{R}_2}. (\exists! x. \ l_1: \text{keys}(x) \land l_1: \text{lose}(\text{fred}, x)) \land \\
(\exists \ell \ l_2. (\exists! x. \ l_2: \text{exam}(x) \land l_2: \text{fail}(\text{fred}, x) \land (l_{\mathcal{R}_2}: \phi_{\text{Cont.}}(\ell_1, \ell_2) \land l_{\mathcal{R}_1}: \phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2})))))
\]
Example

(2)  [Fred is grumpy]_0 because [he lost his keys]_1. Moreover, [he failed an exam]_2.

\[d_1 = d_{\text{init. anchor}} \ C_0 \ (d_{\text{because}} \ l_{ Du} \ l_{ Du} \ l_{ Du} \ (d_{\text{anchor}} \ C_1 \ (d_{\text{moreover}} \ l_{ Du} \ l_{ Du} \ l_{ Du} \ (d_{\text{anchor}} \ C_2 \ l_{ Du}))))\]

\[:= \exists \ell l_0 l_{\mathcal{R}_1}. l_0: \text{grumpy}(\text{fred}) \land (\exists \ell l_1 l_{\mathcal{R}_2}. (\exists! x. l_1: \text{keys}(x) \land l_1: \text{lose}(\text{fred}, x)) \land (\exists \ell l_2. (\exists! x. l_2: \text{exam}(x) \land l_2: \text{fail}(\text{fred}, x) \land (l_{\mathcal{R}_2}: \phi_{\text{Cont.}}(l_1, l_2) \land l_{\mathcal{R}_1}: \phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2}))))\]
Example

(2) \([\text{Fred is grumpy}]_0 \text{ because } [\text{he lost his keys}]_1\). Moreover, \([\text{he failed an exam}]_2\).

\[
d_1 = d_{\text{init.anchor}} C_0 (d_{\text{because}} l_{\text{Du}} l_{\text{Du}} l_{\text{Du}} (d_{\text{anchor}} C_1 (d_{\text{moreover}} l_{\text{Du}} l_{\text{Du}} l_{\text{Du}} (d_{\text{anchor}} C_2 l_{\text{Du}}))))
\]
\[
:= \exists \ell_0 l_{R_1}. \ell_0 : \text{grumpy}(\text{fred}) \land (\exists \ell_1 l_{R_2}. (\exists ! x. l_1 : \text{keys}(x) \land l_1 : \text{lose}(\text{fred}, x)) \land
(\exists \ell_2. (\exists ! x. l_2 : \text{exam}(x) \land l_2 : \text{fail}(\text{fred}, x) \land (l_{R_2} : \phi_{\text{Cont.}} (l_1, l_2) \land l_{R_1} : \phi_{\text{Expl.}} (l_0, l_{R_2})))))
\]
Example

(2) [Fred is grumpy]₀ because [he lost his keys]₁. Moreover, [he failed an exam]₂.

\[ d₁ = d_{\text{init.anchor}} \ C₀ \ (d_{\text{because}} \ l_{\text{Du}} \ l_{\text{Du}} \ l_{\text{Du}} \ (d_{\text{anchor}} \ C₁ \ (d_{\text{moreover}} \ l_{\text{Du}} \ l_{\text{Du}} \ l_{\text{Du}} \ (d_{\text{anchor}} \ C₂ \ l_{\text{Du}})))) \]

\[ := \exists l₀ l₁ R₁ l₀: \text{grumpy}(fred) \land (\exists l₁ l₂ R₂ (\exists ! x. l₁: \text{keys}(x) \land l₁: \text{lose}(fred, x)) \land (\exists l₂ (\exists ! x. l₂: \text{exam}(x) \land l₂: \text{fail}(fred, x) \land (l₂ C₀: \phi_{\text{Cont.}}(l₁, l₂) \land l₁ R₁: \phi_{\text{Expl.}}(l₀, l₂)))))) \]
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.

\[
d_3 = d_{init.\, anchor} \, C_0 \, (d_{because} \, l_{Du} \, (d_S \, l_{Du} \, l_{Du} \, (d_{anchor} \, C_2 \, l_{Du}) \, l_{Du} \, (d_{anchor} \, C_1 \, l_{Du})))
\]
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.

\[
d_3 = d_{\text{init.anchor}} \ C_0 \ (d_{\text{because}} \ l_{Du} \ (d_{\text{then}} \ l_{Du} \ l_{Du} \ (d_{\text{anchor}} \ C_2 \ l_{Du})) \ l_{Du} \ (d_{\text{anchor}} \ C_1 \ l_{Du}))
\]
\[
:= \exists \ell \ l_0 \ l_{R_1} \ l_{R_2}. \exists! x. \ l_0: \text{supermarket}(x) \land l_0: \text{go to}(\text{fred}, x) \land (\exists \ell \ l_2. (\exists! x. l_2: \text{movies}(x) \land l_2: \text{go to}(\text{fred}, x)) \land ((\exists \ell \ l_1. (\exists! x. l_1: \text{fridge}(x) \land l_1: \text{empty}(x)) \land (l_{R_1}: \phi_{\text{Expl.}}(l_0, l_1) \land \top)) \land l_{R_2}: \phi_{\text{Nar.}}(l_0, l_2)))
\]
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.

\[d_3 = d_{init.\, anchor} \ C_0 \ (d_{because} \ l_{Du} \ (d_{then} \ l_{Du} \ l_{Du} \ l_{Du} \ (d_{anchor} \ C_2 \ l_{Du})) \ l_{Du} \ (d_{anchor} \ C_1 \ l_{Du}))\]
\[:= \exists \ell \ l_0 \ l_{\mathcal{R}_1} \ l_{\mathcal{R}_2}. \exists ! x. \ l_0:\text{supermarket}(x) \land l_0:\text{go\_to}(fred, x) \land (\exists \ell \ l_2. (\exists ! x. l_2:\text{movies}(x) \land l_2:\text{go\_to}(fred, x)) \land ((\exists \ell \ l_1. (\exists ! x. l_1:\text{fridge}(x) \land l_1:\text{empty}(x)) \land (l_{\mathcal{R}_1}: \phi_{\text{Expl.}}(l_0, l_1) \land \top)) \land l_{\mathcal{R}_2}: \phi_{\text{Nar.}}(l_0, l_2)))\]
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.

\[
d₃ = d_{\text{init. anchor}} \ C₀ \ (d_{\text{because}} \ I₃ \ (d_{\text{then}} \ I₃ \ I₄ \ (d_{\text{anchor}} \ I₄ \ I₄))) \ I₄ \ (d_{\text{anchor}} \ I₃ \ I₄))
\]

\[
:= \exists \ell_0 \ I₃ \ (d_{\text{then}} \ I₃ \ I₄ \ ((\exists \ell_1. (\exists !x. \ \ell_1: \text{supermarket}(x) \land \ell_1: \text{go to}(\text{fred}, x) \land (\exists \ell_2. (\exists !x. \ \ell_2: \text{movies}(x) \land \ell_2: \text{go to}(\text{fred}, x)) \land ((\exists \ell_1. (\exists !x. \ \ell_1: \text{fridge}(x) \land \ell_1: \text{empty}(x)) \land (d_{\text{anchor}} \ C₂ \ I₄))) \land I₄ \ (d_{\text{anchor}} \ C₁ \ I₄)))
\]
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.

\[
d_3 = d_{\text{init.anchor}} C_0 (d_{\text{because}} l_{\text{Du}} (d_{\text{then}} l_{\text{Du}} l_{\text{Du}} l_{\text{Du}} (d_{\text{anchor}} C_2 l_{\text{Du}})) l_{\text{Du}} (d_{\text{anchor}} C_1 l_{\text{Du}})) \]
\[
:= \exists \ell l_0 l_{\mathcal{R}_1} l_{\mathcal{R}_2}. \exists! x. l_0: \text{supermarket}(x) \land l_0: \text{go.to}(\text{fred}, x) \land (\exists \ell l_2. (\exists! x. l_2: \text{movies}(x) \land l_2: \text{go.to}(\text{fred}, x)) \land ((\exists \ell l_1. (\exists! x. l_1: \text{fridge}(x) \land l_1: \text{empty}(x))) \land (l_{\mathcal{R}_1}: \phi_{\text{Expl.}}(l_0, l_1) \land \top)) \land l_{\mathcal{R}_2}: \phi_{\text{Nar.}}(l_0, l_2)))
\]
Example

(4) [Fred went to a supermarket]₀ because [his fridge was empty]₁. Then, [he went to the cinema]₂.

\[ d_3 = d_{\text{init.Anchor}} \ C_0 \ (d_{\text{because}} \ l_{\text{Du}} \ (d_{\text{then}} \ l_{\text{Du}} \ l_{\text{Du}} \ l_{\text{Du}} \ (d_{\text{anchor}} \ C_2 \ l_{\text{Du}})) \ l_{\text{Du}} \ (d_{\text{anchor}} \ C_1 \ l_{\text{Du}})) \]
\[ := \exists \ell l_0 \ l_{\mathcal{R}_1} \ l_{\mathcal{R}_2} . \exists x. \ l_0: \text{supermarket}(x) \land l_0: \text{go_to}(fred, x) \land (\exists \ell l_2. (\exists x. l_2: \text{movies}(x) \land l_2: \text{go_to}(fred, x)) \land ((\exists \ell l_1. (\exists x. l_1: \text{fridge}(x) \land l_1: \text{empty}(x)) \land (l_{\mathcal{R}_1}: \phi_{\text{Expl.}}(l_0, l_1) \land \top)) \land l_{\mathcal{R}_2}: \phi_{\text{Nar.}}(l_0, l_2))) \]
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
- 2nd order ACGs
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
- 2nd order ACGs – polynomial parsing and generation

Future work

- Encode D-LTAG as ACGs
- Referring expression generation
- Discourse connectives and attitude verbs
- Disambiguation in parsing
## Conclusion

### Summary of the current results
- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
- 2nd order ACGs – polynomial parsing and generation

### Future work
- Encode D-LTAG as ACGs
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
- 2nd order ACGs – polynomial parsing and generation

Future work

- Encode D-LTAG as ACGs
- Referring expression generation
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
- 2nd order ACGs – polynomial parsing and generation

Future work

- Encode D-LTAG as ACGs
- Referring expression generation
- Discourse connectives and attitude verbs
Conclusion

Summary of the current results

- Purely grammatical approach to clause-medial connectives
- Uniform encoding of clause-initial and clause-medial connectives
- D-STAG as ACGs + labeled interpretations
- A single grammar incorporating clause-level and discourse-level grammars
- 2nd order ACGs – polynomial parsing and generation

Future work

- Encode D-LTAG as ACGs
- Referring expression generation
- Discourse connectives and attitude verbs
- Disambiguation in parsing
Thank you!


