An Abstract Categorial Grammar Approach to the Discourse Modeling

Aleksandre Maskharashvili

25 October 2017
Göteborg, Sweden

Discourse

Discourse

What is 'discourse'

Various kinds of natural language acts

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

Discourse

What is 'discourse'

Various kinds of natural language acts

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

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

π_0 Fred is grumpy,

π_1 because his wife is absent for a week.

π_2 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

π_0 Fred is grumpy,

π_1 because his wife is absent for a week.

π_2 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

π_0 Fred is grumpy,

π_1 because his wife is absent for a week.

π_2 This shows how much he loves her.

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

- 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]$_2$.

- 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
-
-

[Fred is grumpy]$_0$ because [his wife is absent for a week]$_1$. ϵ [This shows how much he loves her]$_2$.

- 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)
-
-

$\underbrace{\text{[Fred is grumpy]}_0 \text{ because [his wife is absent for a week]}_1.}_{\text{DU}}$ ϵ [This shows how much he loves her]$_2$.

- 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
-

$\underbrace{[\text{Fred is grumpy}]_0 \text{ because}^{\textit{Explanation}} [\text{his wife is absent for a week}]_1.}_{\text{DU}}$ $\epsilon^{\textit{Commentary}}$ [This shows how much he loves her]$_2$.

- 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
- Discourse connections create the discourse structure

$\underbrace{\text{[Fred is grumpy]}_0 \text{ because}^{\textit{Explanation}} \text{ [his wife is absent for a week]}_1.}_{\text{DU}}$ $\epsilon^{\textit{Commentary}}$ [This shows how much he loves her]$_2$.

Examples of Discourse Structure

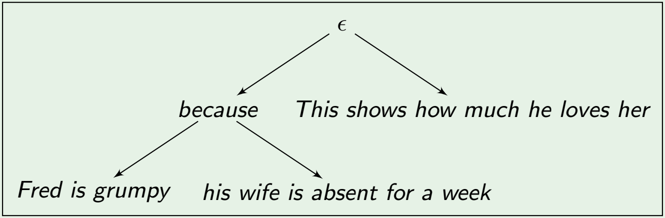
Left-branching trees

(1) [Fred is grumpy]$_0$ because [his wife is absent for a week]$_1$. [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]$_2$.

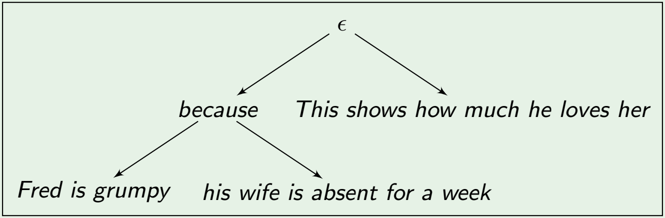


Grammar : trees and substitution

Examples of Discourse Structure

Left-branching trees

(1) [Fred is grumpy]$_0$ because [his wife is absent for a week]$_1$. ϵ [This shows how much he loves her]$_2$.

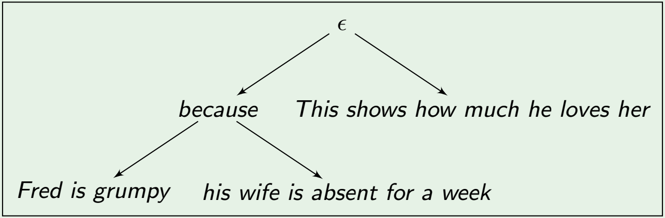


Grammar : trees and substitution

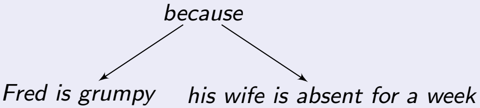
Examples of Discourse Structure

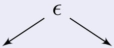
Left-branching trees

(1) **[Fred is grumpy]$_0$ because [his wife is absent for a week]$_1$. ϵ [This shows how much he loves her]$_2$.**



Grammar : trees and substitution

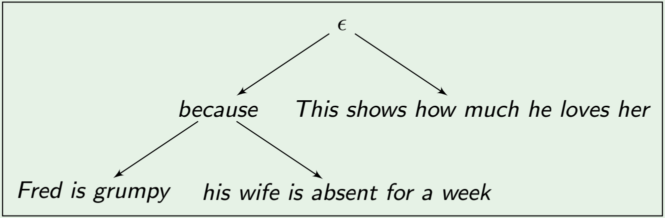




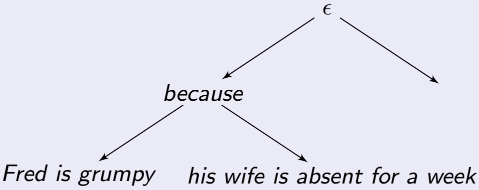
Examples of Discourse Structure

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]$_2$.



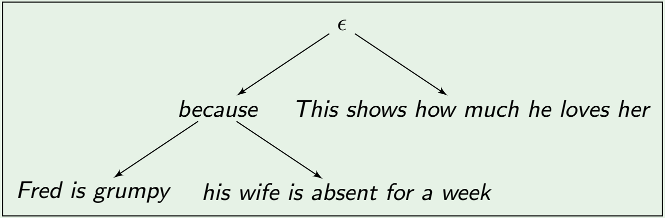
Grammar : trees and substitution



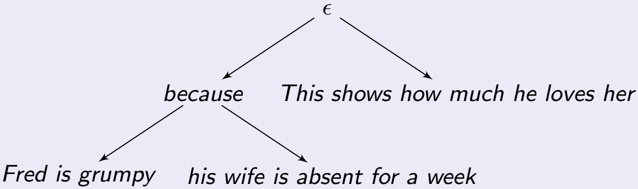
Examples of Discourse Structure

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]$_2$.**



Grammar : trees and substitution



Examples of Discourse Structure

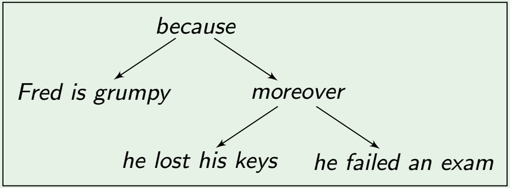
Right-branching trees

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

Examples of Discourse Structure

Right-branching trees

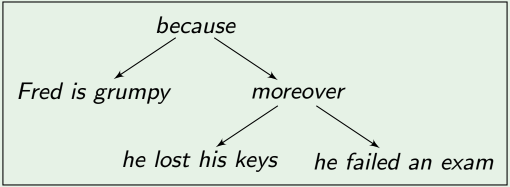
(2) [Fred is grumpy]$_0$ because [he lost his keys]$_1$. Moreover, [he failed an exam]$_2$.



Examples of Discourse Structure

Right-branching trees

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

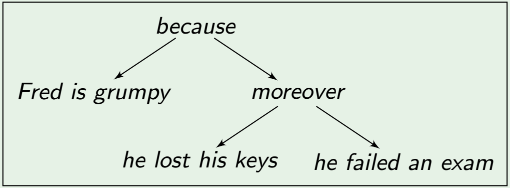


Grammar : trees and substitution

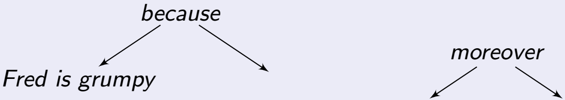
Examples of Discourse Structure

Right-branching trees

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



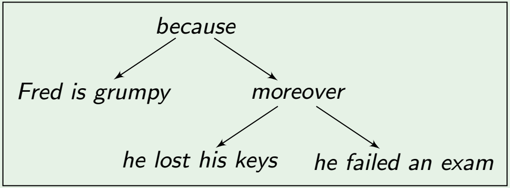
Grammar : trees and substitution



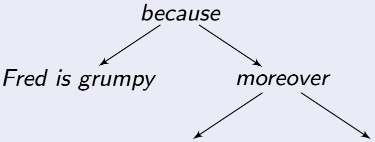
Examples of Discourse Structure

Right-branching trees

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



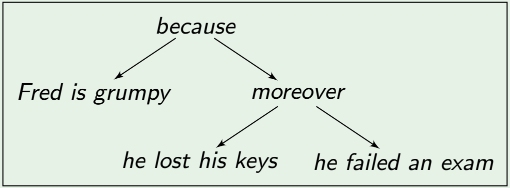
Grammar : trees and substitution



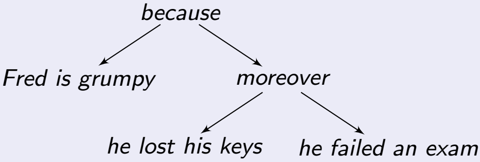
Examples of Discourse Structure

Right-branching trees

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



Grammar : trees and substitution



Examples of Discourse Structure

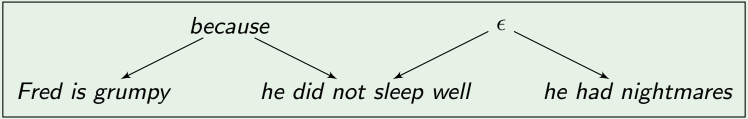
Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]$_0$ because [he did not sleep well]$_1$. [He had nightmares]$_2$.

Examples of Discourse Structure

Directed Acyclic Graphs (DAGs)

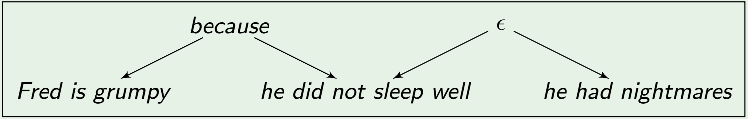
(3) [Fred is grumpy]$_0$ because [he did not sleep well]$_1$. ϵ [He had nightmares]$_2$.



Examples of Discourse Structure

Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]$_0$ because [he did not sleep well]$_1$. ϵ [He had nightmares]$_2$.

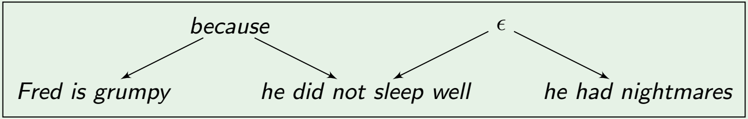


(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.

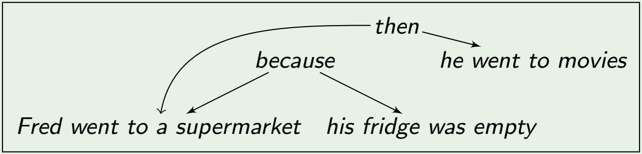
Examples of Discourse Structure

Directed Acyclic Graphs (DAGs)

(3) [Fred is grumpy]$_0$ because [he did not sleep well]$_1$. ϵ [He had nightmares]$_2$.



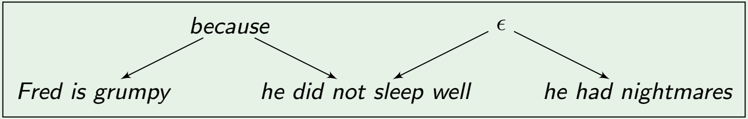
(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.



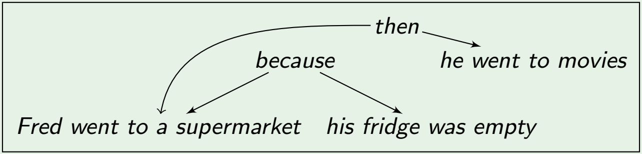
Examples of Discourse Structure

Directed Acyclic Graphs (DAGs)

(3) **[Fred is grumpy]$_0$ because [he did not sleep well]$_1$. ϵ [He had nightmares]$_2$.**



(4) **[Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.**

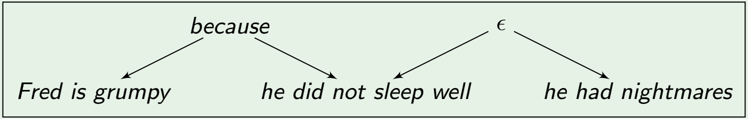


Discourse structure $\neq$ parse tree

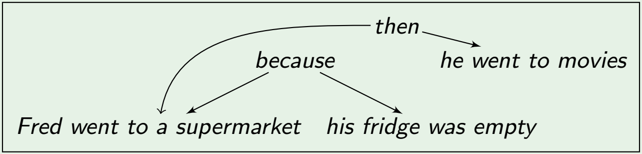
Examples of Discourse Structure

Directed Acyclic Graphs (DAGs)

(3) **[Fred is grumpy]$_0$ because [he did not sleep well]$_1$. ϵ [He had nightmares]$_2$.**



(4) **[Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.**



Discourse structure $\neq$ 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) [Fred went to the supermarket]$_0$. 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 $\underbrace{\textbf{went to the cinema}}_{\textbf{VP}}$]$_1$.

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 $\underbrace{\textbf{went to the cinema}}_{\textbf{VP}}$]$_1$.

- 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 **went to the cinema**]$_1$.

(**went to the cinema** = **VP**)

- 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 $\underbrace{\textbf{went to the cinema}}_{\textbf{VP}}$]$_1$.

- 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

Discourse formalisms and their properties

	Parsing	Generation	Tree	DAG
G-TAG (Danlos, 1998)	✗	✓	✓	✗
D-LTAG (Webber & Joshi, 1998)	✓	✗	✓	✗
D-STAG (Danlos, 2011)	✓	✗	✓	✓

Discourse formalisms and their properties

	Parsing	Generation	Tree	DAG	Clause-medial conn.
G-TAG (Danlos, 1998)	✗	✓	✓	✗	✗
D-LTAG (Webber & Joshi, 1998)	✓	✗	✓	✗	✗
D-STAG (Danlos, 2011)	✓	✗	✓	✓	✗

Discourse formalisms and their properties

	Parsing	Generation	Tree	DAG	Clause-medial conn.
G-TAG (Danlos, 1998)	✗	✓	✓	✗	✗
D-LTAG (Webber & Joshi, 1998)	✓	✗	✓	✗	✗
D-STAG (Danlos, 2011)	✓	✗	✓	✓	✗

Ad hoc encoding of clause-medial connectives

- Prohibits having reversible grammars – both parsing & generation
- Prohibits generalizations

Discourse modeling with ACGs

Goals

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)

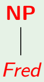
Elementary trees –

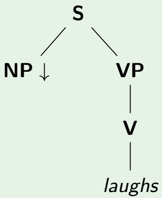
- **Initial trees** : domain of locality

Operations on trees –

Generated structures –

Example





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

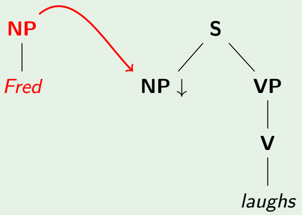
Elementary trees –

- **Initial trees** : domain of locality

Operations on trees – substitution

Generated structures –

Example



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

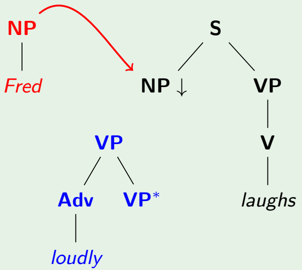
Elementary trees –

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

Operations on trees – substitution

Generated structures –

Example



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

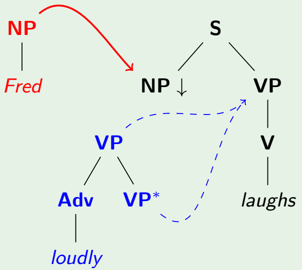
Elementary trees –

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

Operations on trees – substitution and adjunction

Generated structures –

Example



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

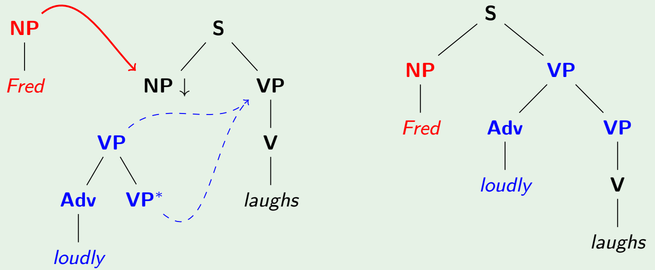
Elementary trees –

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

Operations on trees – substitution and adjunction

Generated structures – derived trees.

Example



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

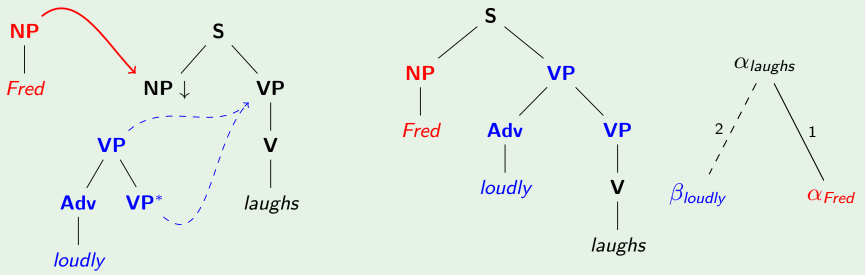
Elementary trees –

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

Operations on trees – substitution and adjunction

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

Example

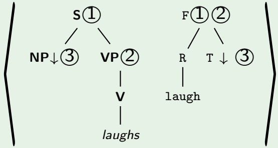


Synchronous TAG (STAG)

(Shieber&Schabes, 1990)

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

Example



(a) α_{laughs}

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_{syn} and t_{sem}
- Parallel operations on corresponding nodes

Example

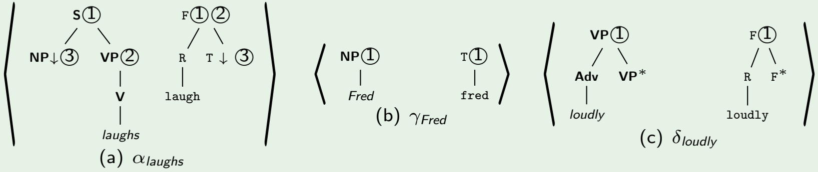
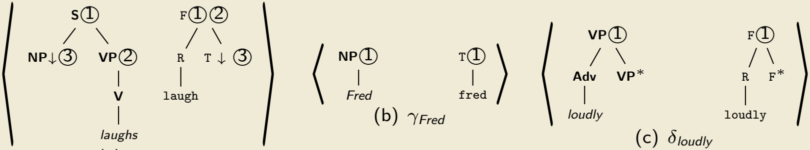


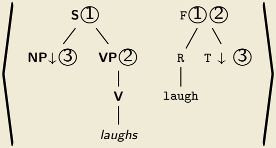
Figure – STAG elementary structures

STAG Example

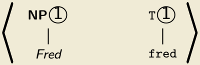


(a) α_{laughs} (b) γ_{Fred} (c) δ_{loudly}

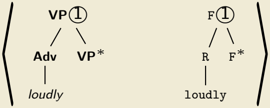
STAG Example



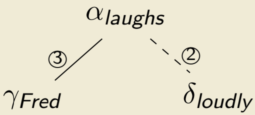
(d) α_{laughs}



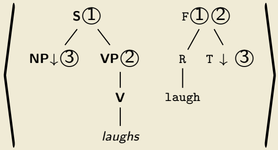
(e) γ_{Fred}



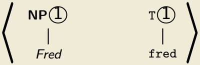
(f) δ_{loudly}



STAG Example

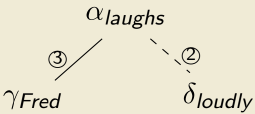


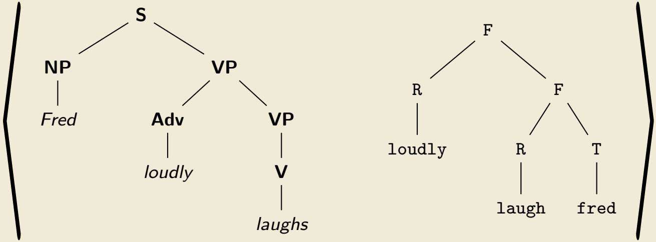
(g) α_{laughs}



(h) γ_{Fred}

(i) δ_{loudly}





Outline

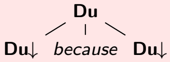
1. TAG
2. **Discourse formalisms**
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG
3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG
4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics

D-LTAG, G-TAG, and D-STAG elementary trees

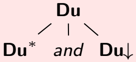
D-LTAG, G-TAG, and D-STAG elementary trees

Discourse-level grammar : D-LTAG

Subordinating conjunctions



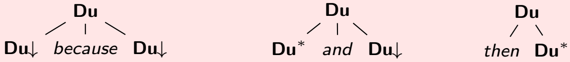
Coordinating conjunctions



D-LTAG, G-TAG, and D-STAG elementary trees

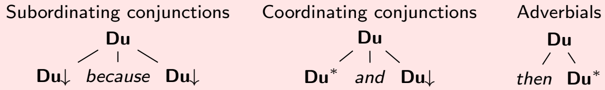
Discourse-level grammar : D-LTAG

Subordinating conjunctions | Coordinating conjunctions | Adverbials

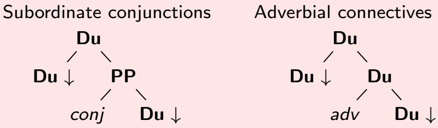


D-LTAG, G-TAG, and D-STAG elementary trees

Discourse-level grammar : D-LTAG

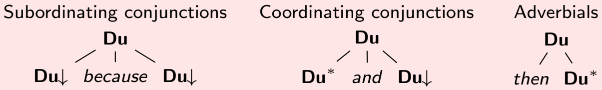


Discourse-level grammar : G-TAG

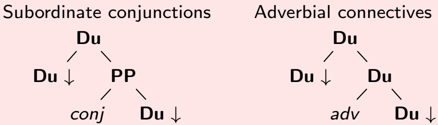


D-LTAG, G-TAG, and D-STAG elementary trees

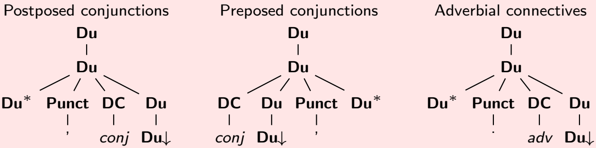
Discourse-level grammar : D-LTAG



Discourse-level grammar : G-TAG



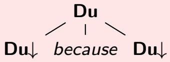
Discourse-level grammar : D-STAG



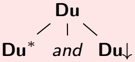
D-LTAG, G-TAG, and D-STAG elementary trees

Discourse-level grammar : D-LTAG

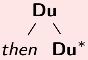
Subordinating conjunctions



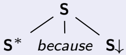
Coordinating conjunctions



Adverbials

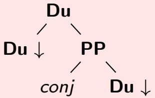


Clause-level grammar

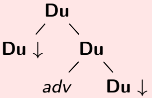


Discourse-level grammar : G-TAG

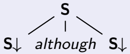
Subordinate conjunctions



Adverbial connectives

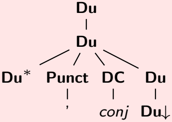


Clause-level grammar

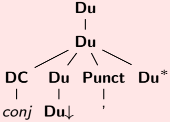


Discourse-level grammar : D-STAG

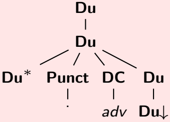
Postposed conjunctions



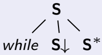
Preposed conjunctions



Adverbial connectives



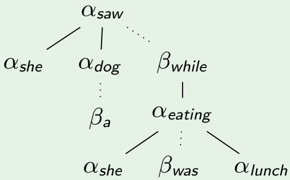
Clause-level grammar



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

While she was eating lunch, she saw a dog

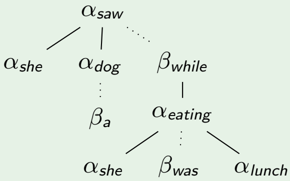
Syntactic analysis



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

While she was eating lunch, she saw a dog

Syntactic analysis

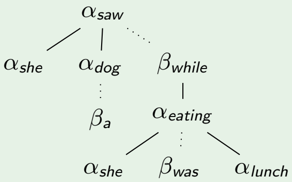


Constructing elementary trees of D-LTAG

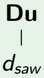
Extraction of trees and construction of an input for a discourse parser

While she was eating lunch, she saw a dog

Syntactic analysis



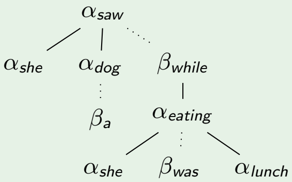
Constructing elementary trees of D-LTAG



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

While she was eating lunch, she saw a dog

Syntactic analysis

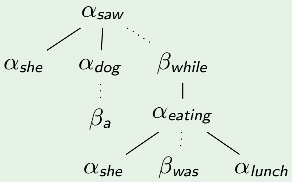


Constructing elementary trees of D-LTAG

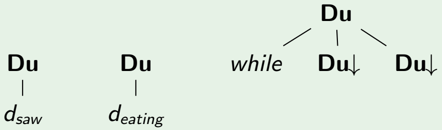
Extraction of trees and construction of an input for a discourse parser

While she was eating lunch, she saw a dog

Syntactic analysis

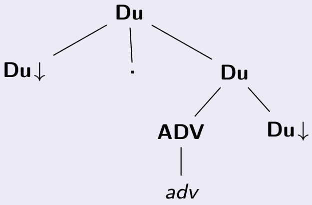


Constructing elementary trees of D-LTAG



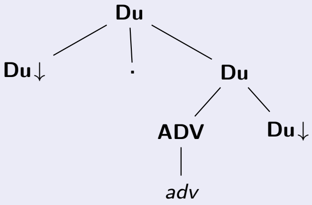
An approach to clause-medial connectives

Elementary tree

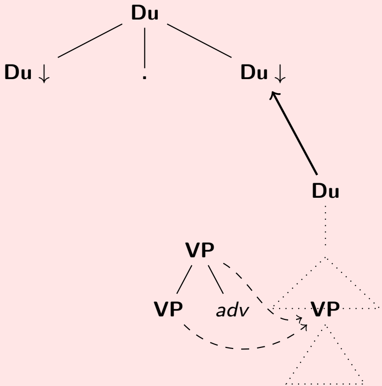


An approach to clause-medial connectives

Elementary tree

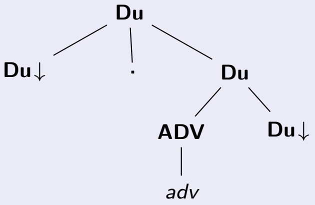


A possible analysis

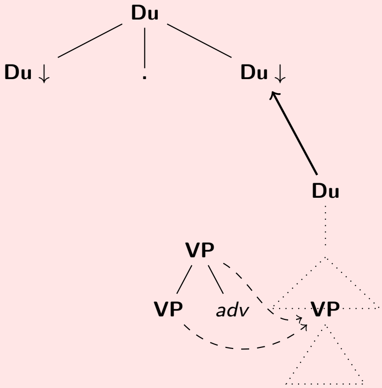


An approach to clause-medial connectives

Elementary tree



A possible analysis



How to encode such a constraint with TAG or sl-MCTAG ?

Discourse Synchronous TAG (D-STAG)

Discourse Synchronous TAG (D-STAG)

Characteristics

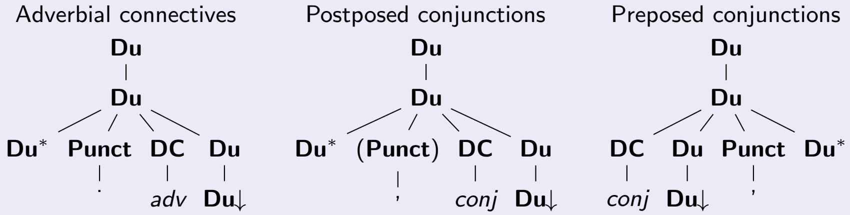
- Based on Synchronous TAG (STAG)
- Generates DAGs as discourse structures

Discourse Synchronous TAG (D-STAG)

Characteristics

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

Elementary trees - Syntax



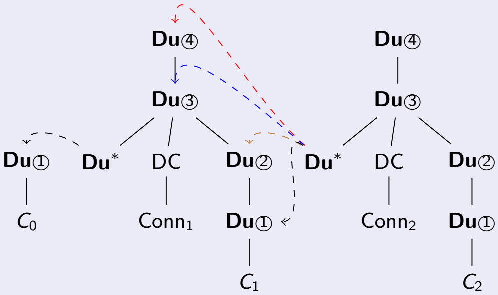
Input (string of clauses and connectives) :

$$C_0 \ \text{Conn}_1 \ C_1 \ \text{Conn}_2 \ C_2$$

D-STAG Discourse Parsing
Ambiguity

Input (string of clauses and connectives) :

$$C_0 \text{ Conn}_1 \; C_1 \text{ Conn}_2 \; C_2$$

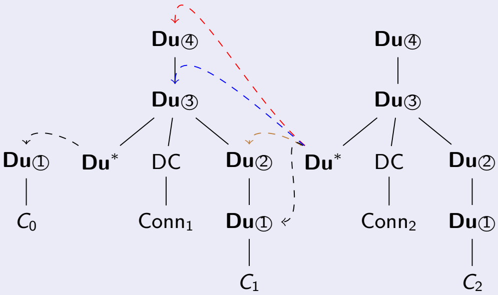


D-STAG Discourse Parsing

Ambiguity

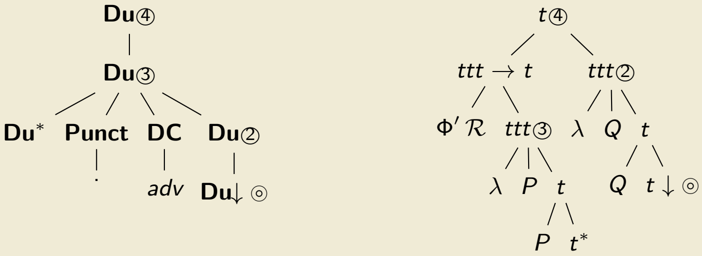
Input (string of clauses and connectives) :

$$C_0 \text{ Conn}_1 \ C_1 \text{ Conn}_2 \ C_2$$



A number of attachment sites with the same yield causes a high ambiguity in parsing. They are needed for obtaining various semantic interpretations.

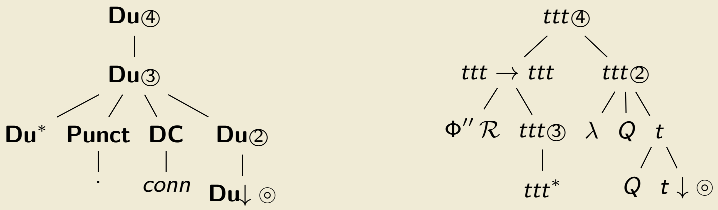
D-STAG semantics



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

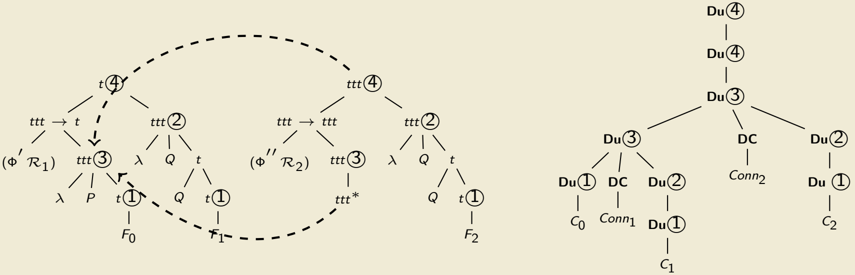
$$(t \rightarrow t) \rightarrow t \triangleq ttt$$

D-STAG semantics

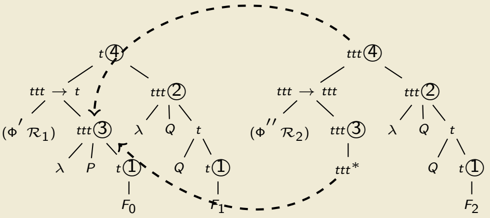


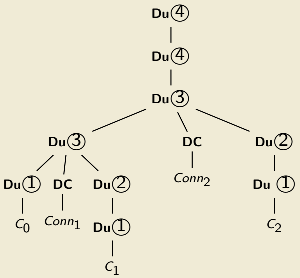
$$\Phi'' \mathcal{R} = \lambda X \; Y \; P.X(\lambda x.Y(\lambda y.\mathcal{R}(x, y) \wedge P(x)))$$

Example : **[Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.**

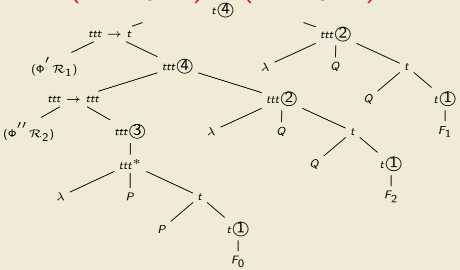


Example : **[Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.**

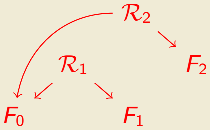


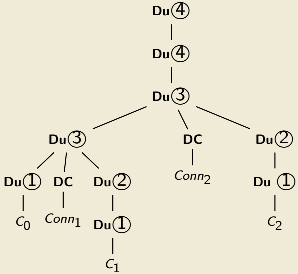


$$(\mathcal{R}_1\, F_0\, F_1) \wedge (\mathcal{R}_2\, F_0\, F_2)$$

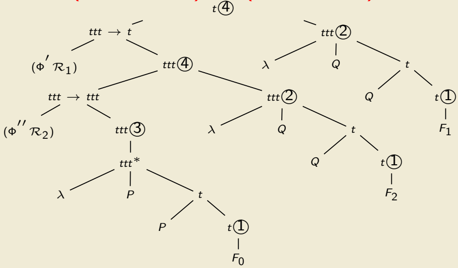


Example : **[Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.**





$$(\mathcal{R}_1\ F_0\ F_1) \wedge (\mathcal{R}_2\ F_0\ F_2)$$



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 $\mathscr{G}$ generates two languages :
  - The abstract language $\mathcal{A}(\mathscr{G})$
  - The object language $\mathcal{O}(\mathscr{G})$

Abstract Categorial Grammar (ACG)

(de Groote, 2001)

Main Features

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

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 $\mathscr{G}$ generates two languages :
  - The abstract language $\mathcal{A}(\mathscr{G})$
  - The object language $\mathcal{O}(\mathscr{G})$

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 λ-terms

Abstract Categorial Grammar (ACG)

(de Groote, 2001)

Main Features

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

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 λ-terms :
ACGs can be composed

Abstract Categorial Grammar (ACG)
(de Groote, 2001)

Main Features

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

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 λ-terms : ACGs can be composed

Parsing 2nd order ACGs are reversible (Salvati 2005), (Kanazawa 2007)

From TAG derivation to TAG derived trees

Derivation trees

Their interpretations as derived trees

From TAG derivation to TAG derived trees

Derivation trees

Their interpretations as derived trees

NP
|
Fred

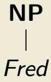
From TAG derivation to TAG derived trees

Derivation trees

C_{Fred} : **NP**

Their interpretations as derived trees

$\mathbf{NP}_1$ *Fred*



From TAG derivation to TAG derived trees

Derivation trees

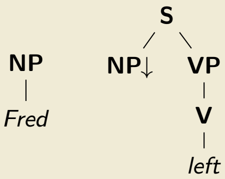
C_{Fred} : $\mathbf{NP}$

C_{left} : $\mathbf{S}_{\mathrm{A}} \multimap \mathbf{VP}_{\mathrm{A}} \multimap \mathbf{NP} \multimap \mathbf{S}$

Their interpretations as derived trees

$\mathbf{NP}_1$ *Fred*

$\lambda S\ A\ np.S(\mathbf{S}_2\ np\ (A\ (\mathbf{VP}_1\ (\mathbf{V}_1\ \mathit{left}))))$



From TAG derivation to TAG derived trees

Derivation trees

$C_{Fred} : \mathbf{NP}$

$C_{left} : \mathbf{S}_{\mathrm{A}} \multimap \mathbf{VP}_{\mathrm{A}} \multimap \mathbf{NP} \multimap \mathbf{S}$

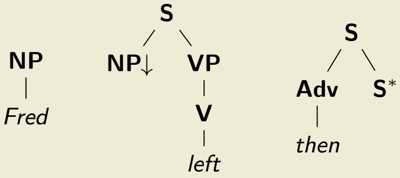
$C^{\mathbf{S}}_{then} : \mathbf{S}_{\mathrm{A}} \multimap \mathbf{S}_{\mathrm{A}}$

Their interpretations as derived trees

$\mathbf{NP}_1\ \mathit{Fred}$

$\lambda S\ A\ np.S(\mathbf{S}_2\ np\ (A\ (\mathbf{VP}_1\ (\mathbf{V}_1\ \mathit{left}))))$

$\lambda A\ x.A\ (\mathbf{S}_2\ (\mathbf{Adv}_1\ \mathit{then})\ x)$



From TAG derivation to TAG derived trees

Derivation trees

C_{Fred} : **NP**

C_{left} : $\mathbf{S}_A \multimap \mathbf{VP}_A \multimap \mathbf{NP} \multimap \mathbf{S}$

$C^{\mathbf{S}}_{then}$: $\mathbf{S}_A \multimap \mathbf{S}_A$

$C^{\mathbf{VP}}_{then}$: $\mathbf{VP}_A \multimap \mathbf{VP}_A$

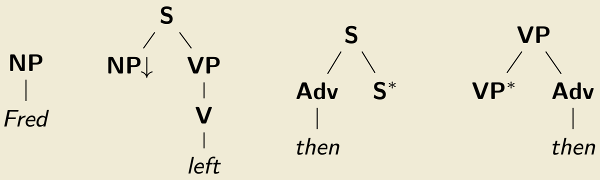
Their interpretations as derived trees

$\mathbf{NP}_1$ *Fred*

$\lambda S\ A\ np.S(\mathbf{S}_2\ np\ (A\ (\mathbf{VP}_1\ (\mathbf{V}_1\ \mathit{left}))))$

$\lambda A\ x.A\ (\mathbf{S}_2\ (\mathbf{Adv}_1\ \mathit{then})\ x)$

$\lambda A\ x.A\ (\mathbf{V}_2\ x\ (\mathbf{Adv}_1\ \mathit{then}))$



From TAG derivation to TAG derived trees

Derivation trees

$C_{\mathit{Fred}} : \mathbf{NP}$

$C_{\mathit{left}} : \mathbf{S}_{\mathrm{A}} \multimap \mathbf{VP}_{\mathrm{A}} \multimap \mathbf{NP} \multimap \mathbf{S}$

$C^{\mathbf{S}}_{\mathit{then}} : \mathbf{S}_{\mathrm{A}} \multimap \mathbf{S}_{\mathrm{A}}$

$C^{\mathbf{VP}}_{\mathit{then}} : \mathbf{VP}_{\mathrm{A}} \multimap \mathbf{VP}_{\mathrm{A}}$

$\mathrm{I}_X : X_A$

Their interpretations as derived trees

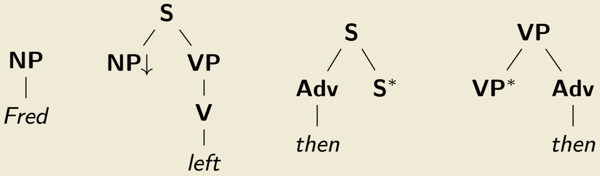
$\mathbf{NP}_1\ \mathit{Fred}$

$\lambda S\ A\ np.S(\mathbf{S}_2\ np\ (A\ (\mathbf{VP}_1\ (\mathbf{V}_1\ \mathit{left}))))$

$\lambda A\ x.A\ (\mathbf{S}_2\ (\mathbf{Adv}_1\ \mathit{then})\ x)$

$\lambda A\ x.A\ (\mathbf{V}_2\ x\ (\mathbf{Adv}_1\ \mathit{then}))$

$\lambda x.x$



From TAG derivation to TAG derived trees

Derivation trees

$C_{Fred} : \mathbf{NP}$

$C_{left} : \mathbf{S}_{\mathrm{A}} \multimap \mathbf{VP}_{\mathrm{A}} \multimap \mathbf{NP} \multimap \mathbf{S}$

$C^{\mathbf{S}}_{then} : \mathbf{S}_{\mathrm{A}} \multimap \mathbf{S}_{\mathrm{A}}$

$C^{\mathbf{VP}}_{then} : \mathbf{VP}_{\mathrm{A}} \multimap \mathbf{VP}_{\mathrm{A}}$

$\mathrm{I}_X : X_A$

Their interpretations as derived trees

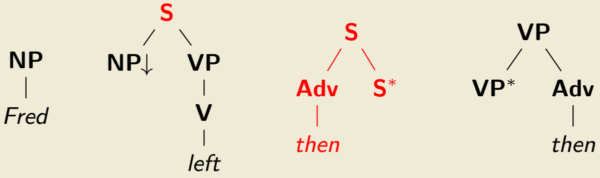
$\mathbf{NP}_1\ Fred$

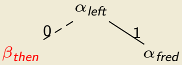
$\lambda S\ A\ np.S(\mathbf{S}_2\ np\ (A\ (\mathbf{VP}_1\ (\mathbf{V}_1\ left))))$

$\lambda A\ x.A\ (\mathbf{S}_2\ (\mathbf{Adv}_1\ then)\ x)$

$\lambda A\ x.A\ (\mathbf{V}_2\ x\ (\mathbf{Adv}_1\ then))$

$\lambda x.x$





$$t_1 = C_{left}\ (C^{\mathbf{S}}_{then}\ \mathrm{I}_{\mathbf{S}})\ \mathrm{I}_{\mathbf{V}}\ C_{Fred}$$

$$\mathscr{G}_{yield} \circ \mathscr{G}_{derived\ trees}(t_1) = then+, +Fred + left$$

From TAG derivation to TAG derived trees

Derivation trees

C_{Fred} : **NP**

C_{left} : $\mathbf{S}_A \multimap \mathbf{VP}_A \multimap \mathbf{NP} \multimap \mathbf{S}$

$C^{\mathbf{S}}_{then}$: $\mathbf{S}_A \multimap \mathbf{S}_A$

$C^{\mathbf{VP}}_{then}$: $\mathbf{VP}_A \multimap \mathbf{VP}_A$

I_X : X_A

Their interpretations as derived trees

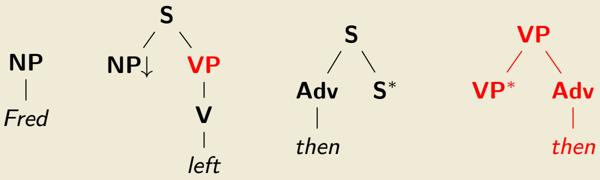
$\mathbf{NP}_1\ Fred$

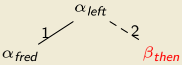
$\lambda S\ A\ np.S(\mathbf{S}_2\ np\ (A\ (\mathbf{VP}_1\ (\mathbf{V}_1\ left))))$

$\lambda A\ x.A\ (\mathbf{S}_2\ (\mathbf{Adv}_1\ then)\ x)$

$\lambda A\ x.A\ (\mathbf{V}_2\ x\ (\mathbf{Adv}_1\ then))$

$\lambda x.x$





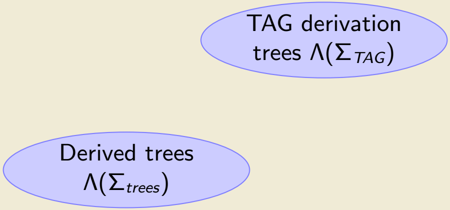
$$t_2 = C_{left}\ \mathsf{I}_{\mathbf{S}}\ (C^{\mathbf{V}}_{then}\ \mathsf{I}_{\mathbf{V}})\ C_{Fred}$$

$$\mathscr{G}_{yield} \circ \mathscr{G}_{derived\ trees}(t_2) = Fred + then + left$$

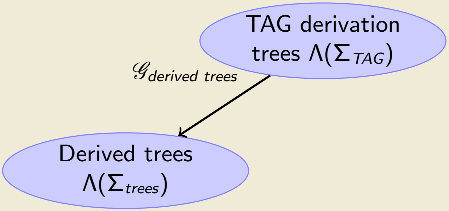
TAG as ACGs

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

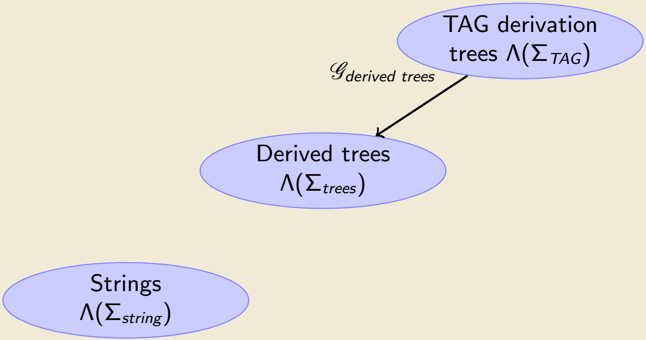
TAG as ACGs


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

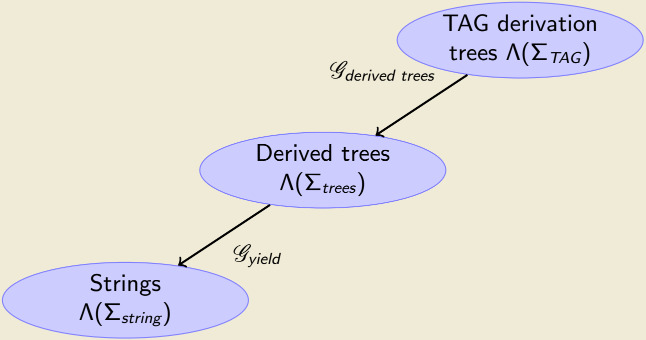
TAG as ACGs


TAG derivation trees $\Lambda(\Sigma_{TAG})$
$\mathscr{G}_{derived\ trees}$
Derived trees $\Lambda(\Sigma_{trees})$

TAG as ACGs

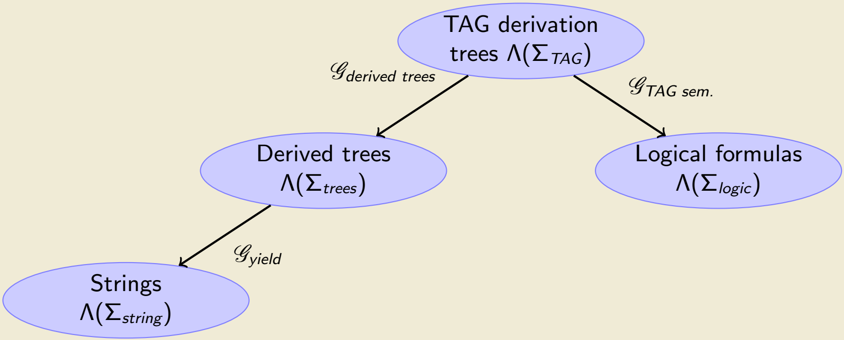

TAG derivation trees $\Lambda(\Sigma_{TAG})$
$\mathscr{G}_{derived\ trees}$
Derived trees $\Lambda(\Sigma_{trees})$
Strings $\Lambda(\Sigma_{string})$

TAG as ACGs



TAG derivation trees $\Lambda(\Sigma_{TAG})$
$\mathscr{G}_{derived\ trees}$
Derived trees $\Lambda(\Sigma_{trees})$
$\mathscr{G}_{yield}$
Strings $\Lambda(\Sigma_{string})$

TAG as ACGs + Montague semantics (Pogodalla, 2004)



Vocabulary Σ_{logic} for Montague semantics

`fred`	$: e$	`loudly`	$: t \multimap t$
`laugh`	$: e \multimap t$	`grumpy`	$: e \multimap t$

Vocabulary Σ_{logic} for Montague semantics

fred	$: e$	loudly	$: t \multimap t$
laugh	$: e \multimap t$	grumpy	$: e \multimap t$
$\wedge$	$: t \multimap t \multimap t$	$\vee$	$: t \multimap t \multimap t$
$\Rightarrow$	$: t \multimap t \multimap t$	$\neg$	$: t \multimap t$
$\exists$	$: (e \to t) \multimap t$	$\forall$	$: (e \to t) \multimap t$

TAG as ACG

Interpretation of TAG derivation trees into Montague semantics

C_{fred} : **NP** $\lambda^{o} P. P\, \texttt{fred}$

TAG as ACG

Interpretation of TAG derivation trees into Montague semantics

$C_{\textsf{fred}} : \mathbf{NP}$ $\qquad \lambda^{o} P. P\, \texttt{fred}$

$C_{\textsf{laugh}} : \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{NP} \multimap \mathbf{S}$ $\qquad \lambda^{o} s_a\, v_a\, subje.\, s_a\, (subje\, (v_a\, (\lambda^{o} x.\, ((\texttt{lough}\, x)))))$

TAG as ACG

Interpretation of TAG derivation trees into Montague semantics

$C_{\text{fred}} : \mathbf{NP}$	$\lambda^{o} P.\, P\, \texttt{fred}$
$C_{\text{laugh}} : \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{NP} \multimap \mathbf{S}$	$\lambda^{o} s_a\, v_a\, subje.\, s_a\,(subje\,(v_a\,(\lambda^{o} x.\,((\texttt{lough}\, x)))))$
$C_{\text{loudly}} : \mathbf{VP_A} \multimap \mathbf{VP_A}$	$\lambda^{o} vp_a\, r.\, vp_a\,(\lambda^{o} x.\,\texttt{loudly}\,(rx))$
$I_{\mathbf{X_A}}$	$\lambda x.x$

TAG as ACG

Interpretation of TAG derivation trees into Montague semantics

$C_{\text{fred}} : \mathbf{NP}$	$\lambda^{\circ} P.\, P\, \texttt{fred}$
$C_{\text{laugh}} : \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{NP} \multimap \mathbf{S}$	$\lambda^{\circ} s_a\, v_a\, subje.\, s_a\, (subje\, (v_a\, (\lambda^{\circ} x.\, ((\texttt{lough}\, x)))))$
$C_{\text{loudly}} : \mathbf{VP_A} \multimap \mathbf{VP_A}$	$\lambda^{\circ} vp_a\, r.\, vp_a\, (\lambda^{\circ} x.\, \texttt{loudly}\, (rx))$
$\mathrm{I}_{\mathbf{X_A}}$	$\lambda x.x$

Example

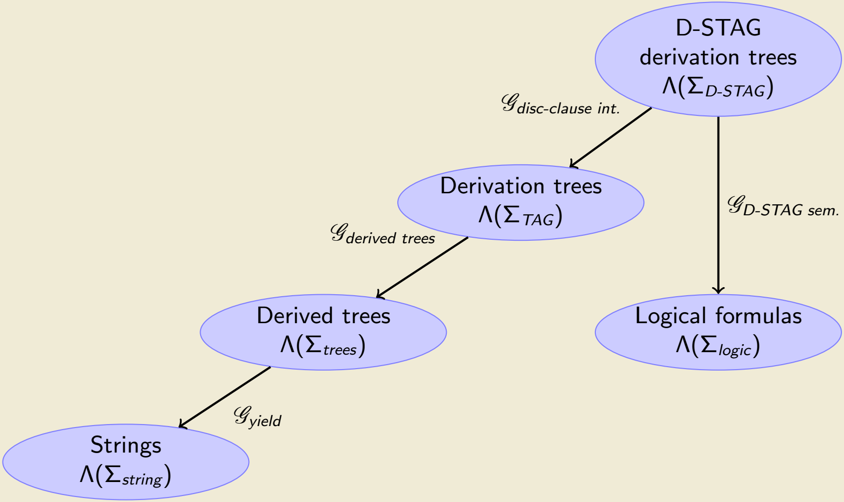
Fred loudly laughs.

$$\mathscr{L}^{\text{sem}}_{\text{TAG}}(C_{\text{laugh}}\ \mathrm{I}_{\mathbf{S_A}}\ (C_{\text{loudly}}\ \mathrm{I}_{\mathbf{VP_A}})\ C_{\text{Fred}}) \twoheadrightarrow_{\beta} \texttt{loudly}\ (\texttt{laugh Fred})\ :\ t$$

Outline

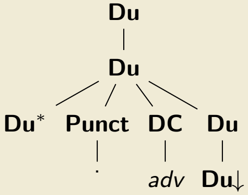
1. TAG
2. Discourse formalisms
   - Properties of grammars of D-LTAG, G-TAG, D-STAG
   - Discourse parsing
   - Problem of clause-medial connectives & a possible analysis
   - D-STAG
3. ACG
   - Definition & basic properties
   - TAG as ACG
   - TAG with semantics as ACG
4. D-STAG as ACG
   - D-STAG as ACGs + clause-medial connectives
   - D-STAG as ACGs with labeled semantics

D-STAG as ACGs



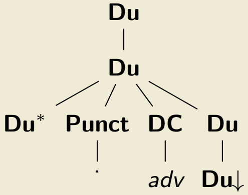
Discourse grammar with ACGs

Adverbial Connectives :



Discourse grammar with ACGs

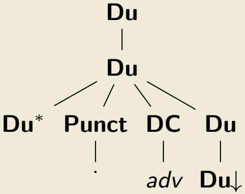
Adverbial Connectives :



d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

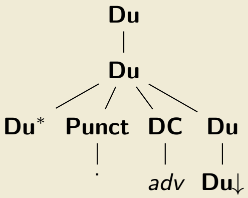
Adverbial Connectives :



$$d_{then}: \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$$

Discourse grammar with ACGs

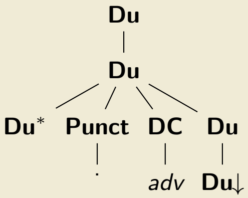
Adverbial Connectives :

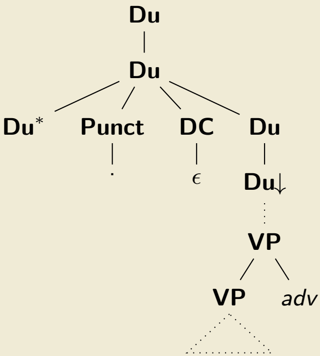


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

Adverbial Connectives :

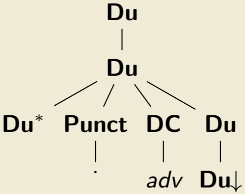


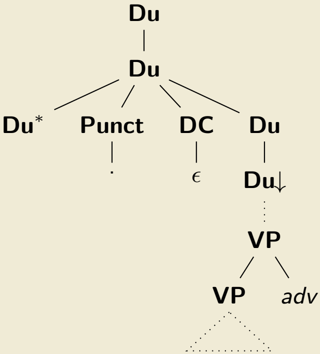


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

Adverbial Connectives :



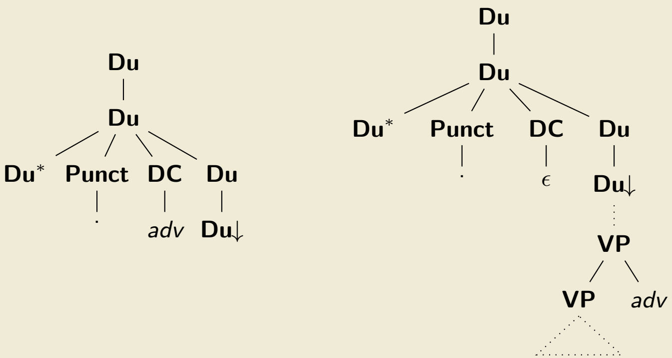


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

Adverbial Connectives :

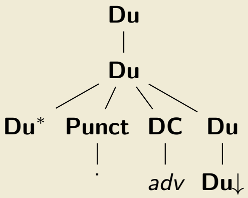


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

Adverbial Connectives :



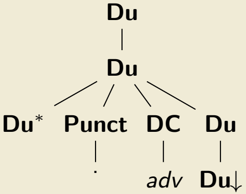
d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

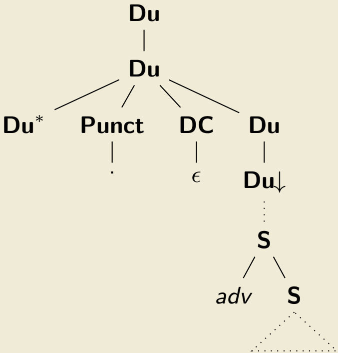
$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

$d^{\mathbf{S}}_{then}$

Discourse grammar with ACGs

Adverbial Connectives :

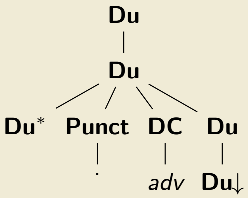


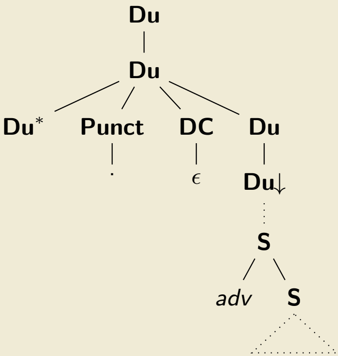


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$
$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$
$d^{\mathbf{S}}_{then}$

Discourse grammar with ACGs

Adverbial Connectives :

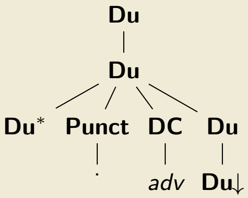


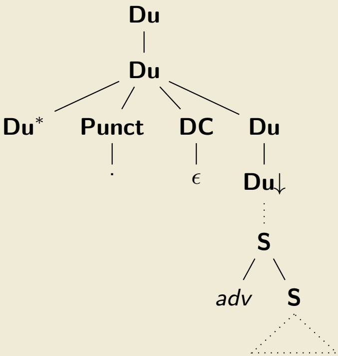


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$
$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$
$d^{\mathbf{S}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{S_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

Adverbial Connectives :

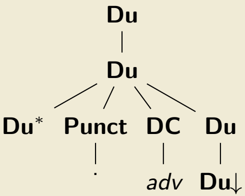


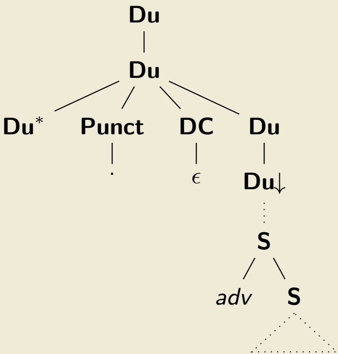


d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$
$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$
$d^{\mathbf{S}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{S_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

Discourse grammar with ACGs

Adverbial Connectives :





d_{then}: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A}$

$d^{\mathbf{V}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{VP_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

$d^{\mathbf{S}}_{then}$: $\mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap (\mathbf{S_A} \multimap \mathbf{Du}) \multimap \mathbf{Du_A}$

3rd order ACG – no polynomial parsing property !

D-STAG as ACGs

2nd order ACGs

From D-STAG derivation trees to TAG derivation trees

$\mathbf{Du} := \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{S}$

D-STAG as ACGs

2nd order ACGs

From D-STAG derivation trees to TAG derivation trees

$\mathbf{Du} := \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{S}$
$(\mathbf{DC} \triangleq \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A})$

From D-STAG derivation trees to TAG derivation trees

$\mathbf{Du} := \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{S}$

$(\mathbf{DC} \triangleq \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A})$

$d^{\mathbf{S}}_{then} : \mathbf{DC} \quad := \lambda^{o} d_1\ d_2\ d_3\ d_u.\ \mathrm{cons}\ d_1\ d_2\ d_3\ (d_u\ C^{\mathbf{S}}_{then}\ \mathrm{I}_{\mathbf{VP}})$

From D-STAG derivation trees to TAG derivation trees

$\mathbf{Du} := \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{S}$
$(\mathbf{DC} \triangleq \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A})$

$d^{\mathbf{S}}_{then} : \mathbf{DC} \quad := \lambda^{o} d_1\ d_2\ d_3\ d_u.\ \mathrm{cons}\ d_1\ d_2\ d_3\ (d_u\ C^{\mathbf{S}}_{then}\ \mathrm{I}_{\mathbf{VP}})$

$d^{\mathbf{V}}_{then} : \mathbf{DC} \quad := \lambda^{o} d_1\ d_2\ d_3\ d_u.\ \mathrm{cons}\ d_1\ d_2\ d_3\ (d_u\ \mathrm{I}_{\mathbf{S}}\ C^{\mathbf{VP}}_{then})$

From D-STAG derivation trees to TAG derivation trees

$\mathbf{Du} := \mathbf{S_A} \multimap \mathbf{VP_A} \multimap \mathbf{S}$

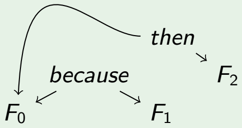
$(\mathbf{DC} \triangleq \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du_A} \multimap \mathbf{Du} \multimap \mathbf{Du_A})$

$d^{\mathbf{S}}_{then} : \mathbf{DC} \quad := \lambda^{\circ} d_1\ d_2\ d_3\ d_u.\ \mathrm{cons}\ d_1\ d_2\ d_3\ (d_u\ C^{\mathbf{S}}_{then}\ \mathrm{I}_{\mathbf{VP}})$

$d^{\mathbf{V}}_{then} : \mathbf{DC} \quad := \lambda^{\circ} d_1\ d_2\ d_3\ d_u.\ \mathrm{cons}\ d_1\ d_2\ d_3\ (d_u\ \mathrm{I}_{\mathbf{S}}\ C^{\mathbf{VP}}_{then})$

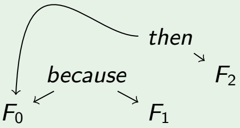
D-STAG as ACGs – semantic interpretations

(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]$_2$.

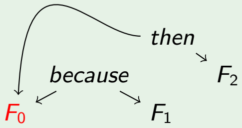


D-STAG as ACGs

$$
\begin{array}{l}
(\texttt{EXPLANATION} \\
\quad (\exists! x.\ (\texttt{supermarket}\ x)\ \wedge\ (\texttt{go-to fred}\ x)) \\
\quad (\exists! x.\ (\texttt{fridge}\ x)\ \wedge\ (\texttt{empty}\ x))) \\
\wedge \\
(\texttt{NARRATION} \\
\quad (\exists! x.(\texttt{supermarket}\ x)\ \wedge\ (\texttt{go-to fred}\ x)) \\
\quad (\exists! x.\ (\texttt{movies}\ x)\ \wedge\ (\texttt{go-to fred}\ x)))
\end{array}
$$

D-STAG as ACGs – semantic interpretations

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to movies]$_2$.

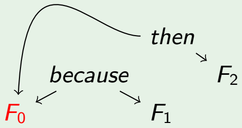


D-STAG as ACGs

$$
\begin{array}{l}
(\texttt{EXPLANATION} \\
\quad (\exists! x.\ (\texttt{supermarket}\ x) \ \wedge\ (\texttt{go-to fred}\ x)) \\
\quad (\exists! x.\ (\texttt{fridge}\ x) \ \wedge\ (\texttt{empty}\ x))) \\
\wedge \\
(\texttt{NARRATION} \\
\quad (\exists! x.(\texttt{supermarket}\ x) \ \wedge\ (\texttt{go-to fred}\ x)) \\
\quad (\exists! x.\ (\texttt{movies}\ x) \ \wedge\ (\texttt{go-to fred}\ x)))
\end{array}
$$

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 ?

$$
\begin{array}{l}
(\texttt{EXPLANATION} \\
\quad (\exists ! x.\ (\texttt{supermarket}\ x)\ \wedge\ (\texttt{go-to fred}\ x)) \\
\quad (\exists ! x.\ (\texttt{fridge}\ x)\ \wedge\ (\texttt{empty}\ x))) \\
\wedge \\
(\texttt{NARRATION} \\
\quad (\exists ! x.(\texttt{supermarket}\ x)\ \wedge\ (\texttt{go-to fred}\ x)) \\
\quad (\exists ! x.\ (\texttt{movies}\ x)\ \wedge\ (\texttt{go-to fred}\ x)))
\end{array}
$$

Labeled semantic interpretations

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

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

Labeled semantic interpretations

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

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

Predicates have one additional argument of type ℓ 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 ℓ for labels

Argument of discourse relations are of type ℓ (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 ℓ for labels

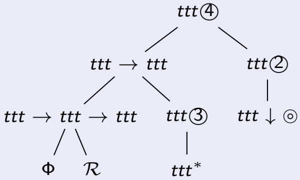
Argument of discourse relations are of type ℓ (labels)

$\mathtt{fred}, \mathtt{he} : e$
$\mathtt{sleep}, \mathtt{bad\text{-}mood}, \mathtt{exam} : e \to \ell \to t$
$\mathtt{love}, \mathtt{miss}, \mathtt{fail} : e \to e \to \ell \to t$
$\forall, \exists, \exists! : (e \to t) \to t$
$\dots$

$\mathtt{EXPLANATION} : \ell \to \ell \to \ell \to t$
$\mathtt{CONTINUATION} : \ell \to \ell \to \ell \to t$
$\mathtt{NARRATION} : \ell \to \ell \to \ell \to t$
$\exists_l : (\ell \to t) \to t$
$\dots$

Labeled semantic trees

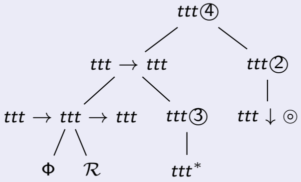
Unlabeled $(t \to t) \to t \triangleq ttt$



$$\Phi'' \mathcal{R} = \lambda X.\, \lambda Y.\, \lambda P.\, X\,(\lambda x.\,(Y\,(\lambda y.\,(\mathcal{R}\,x\,y) \wedge P(x))))$$

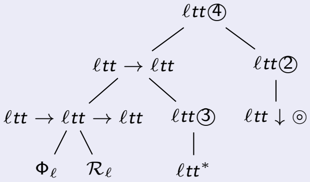
Labeled semantic trees

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



$$\Phi^{''}\mathcal{R} = \lambda X.\lambda Y.\lambda P.X\,(\lambda x.(Y\,(\lambda y.(\mathcal{R}\,x\,y) \wedge P(x))))$$

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



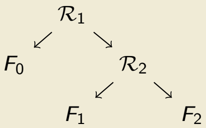
$$\Phi_l^{''}\mathcal{R}_\ell = \; X\;Y\;P.\,\exists_l\;l.\,X(\lambda\,x.\;Y(\lambda\,y.\,(P\,x)\;\wedge\;(\mathcal{R}_\ell\;x\;y\;l)))$$

Example

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

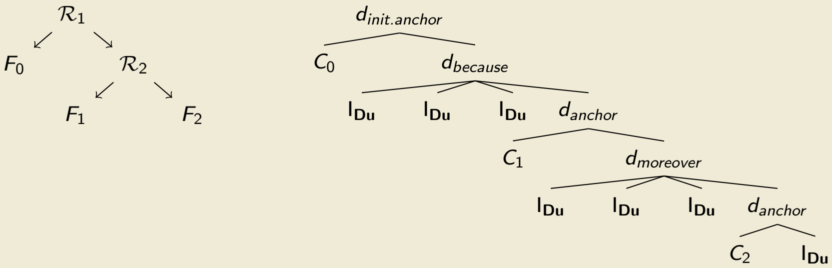
Example

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



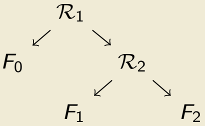
Example

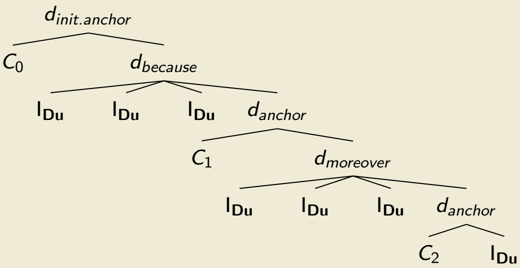
(2) [Fred is grumpy]$_0$ because [he lost his keys]$_1$. Moreover, [he failed an exam]$_2$.



Example

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

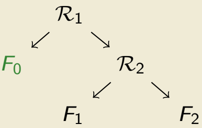


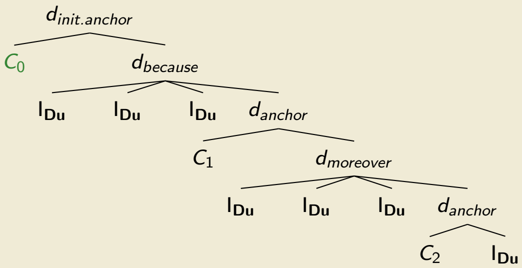


$$d_1 = d_{init.anchor}\ C_0\ (d_{because}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_1\ (d_{moreover}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_2 \mathbf{I_{Du}}))))$$

Example

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





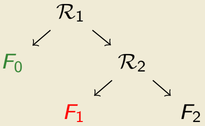
$$d_1 = d_{init.anchor}\ C_0\ (d_{because}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_1\ (d_{moreover}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_2 \mathbf{I_{Du}}))))$$

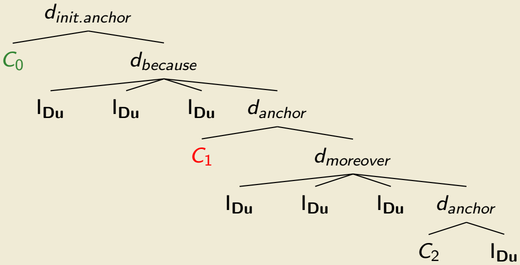
$$:= \exists_\ell l_0 l_{\mathcal{R}_1}.l_0{:}\,\texttt{grumpy}(\texttt{fred}) \wedge (\exists_\ell l_1 l_{\mathcal{R}_2}.(\exists! x. l_1{:}\,\texttt{keys}(x) \wedge l_1{:}\,\texttt{lose}(\texttt{fred}, x)) \wedge$$

$$(\exists_\ell l_2.(\exists! x. l_2{:}\,\texttt{exam}(x) \wedge l_2{:}\,\texttt{fail}(\texttt{fred}, x) \wedge (l_{\mathcal{R}_2}{:}\,\phi_{\text{Cont.}}(l_1, l_2) \wedge l_{\mathcal{R}_1}{:}\,\phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2}))))))$$

Example

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





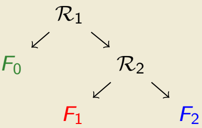
$$d_1 = d_{init.anchor}\ C_0\ (d_{because}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_1\ (d_{moreover}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_2 \mathbf{I_{Du}}))))$$

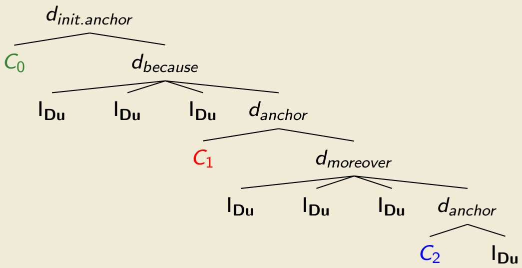
$$:= \exists_\ell l_0 l_{\mathcal{R}_1}.l_0{:}\,\texttt{grumpy}(\texttt{fred}) \wedge (\exists_\ell l_1 l_{\mathcal{R}_2}.(\exists! x.l_1{:}\,\texttt{keys}(x) \wedge l_1{:}\,\texttt{lose}(\texttt{fred}, x)) \wedge$$

$$(\exists_\ell l_2.(\exists! x.l_2{:}\,\texttt{exam}(x) \wedge l_2{:}\,\texttt{fail}(\texttt{fred}, x) \wedge (l_{\mathcal{R}_2}{:}\,\phi_{\text{Cont.}}(l_1, l_2) \wedge l_{\mathcal{R}_1}{:}\,\phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2}))))))$$

Example

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





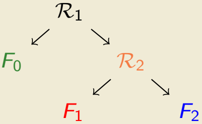
$$d_1 = d_{init.anchor}\ C_0\ (d_{because}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_1\ (d_{moreover}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_2 \mathbf{I_{Du}}))))$$

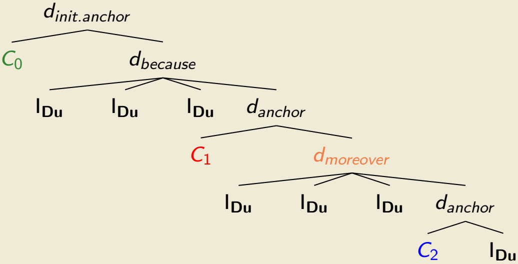
$$:= \exists_\ell l_0 l_{\mathcal{R}_1}.l_0{:}\,\texttt{grumpy}(\texttt{fred}) \wedge (\exists_\ell l_1 l_{\mathcal{R}_2}.(\exists! x.l_1{:}\,\texttt{keys}(x) \wedge l_1{:}\,\texttt{lose}(\texttt{fred}, x)) \wedge$$

$$(\exists_\ell l_2.(\exists! x.l_2{:}\,\texttt{exam}(x) \wedge l_2{:}\,\texttt{fail}(\texttt{fred}, x) \wedge (l_{\mathcal{R}_2}{:}\,\phi_{\text{Cont.}}(l_1, l_2) \wedge l_{\mathcal{R}_1}{:}\,\phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2}))))))$$

Example

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



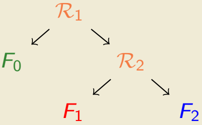


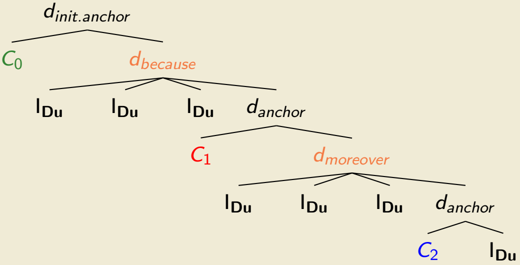
$$d_1 = d_{init.anchor}\ C_0\ (d_{because}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ (d_{anchor}\ C_1\ (d_{moreover}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ (d_{anchor}\ C_2 \mathrm{I}_{\mathbf{Du}}))))$$

$$:= \exists_\ell\, l_0\, l_{\mathcal{R}_1}.\, l_0\!: \mathtt{grumpy}(\mathtt{fred}) \wedge (\exists_\ell\, l_1\, l_{\mathcal{R}_2}.(\exists! x.\, l_1\!: \mathtt{keys}(x) \wedge l_1\!: \mathtt{lose}(\mathtt{fred}, x)) \wedge$$
$$(\exists_\ell\, l_2.(\exists! x.\, l_2\!: \mathtt{exam}(x) \wedge l_2\!: \mathtt{fail}(\mathtt{fred}, x) \wedge (l_{\mathcal{R}_2}\!: \phi_{\text{Cont.}}(l_1, l_2) \wedge l_{\mathcal{R}_1}\!: \phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2}))))))$$

Example

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





$$d_1 = d_{init.anchor}\ C_0\ (d_{because}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ (d_{anchor}\ C_1\ (d_{moreover}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ \mathrm{I}_{\mathbf{Du}}\ (d_{anchor}\ C_2 \mathrm{I}_{\mathbf{Du}}))))$$

$$:= \exists_\ell\, l_0\, l_{\mathcal{R}_1}.l_0{:}\,\mathtt{grumpy}(\mathtt{fred}) \wedge (\exists_\ell\, l_1\, l_{\mathcal{R}_2}.(\exists! x.l_1{:}\,\mathtt{keys}(x) \wedge l_1{:}\,\mathtt{lose}(\mathtt{fred}, x)) \wedge$$

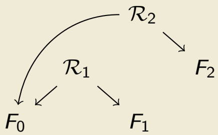
$$(\exists_\ell\, l_2.(\exists! x.l_2{:}\,\mathtt{exam}(x) \wedge l_2{:}\,\mathtt{fail}(\mathtt{fred}, x) \wedge (l_{\mathcal{R}_2}{:}\,\phi_{\text{Cont.}}(l_1, l_2) \wedge l_{\mathcal{R}_1}{:}\,\phi_{\text{Expl.}}(l_0, l_{\mathcal{R}_2}))))))$$

Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.

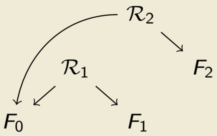
Example

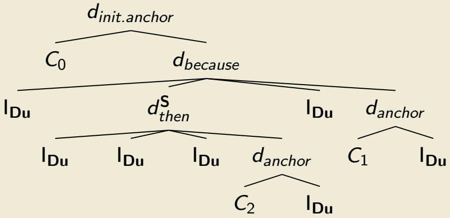
(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.



Example

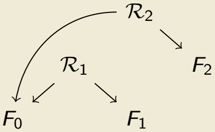
(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.

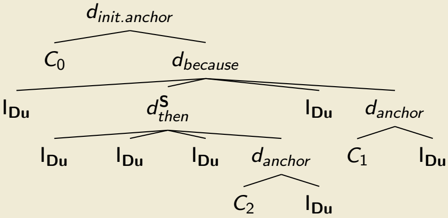




Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.

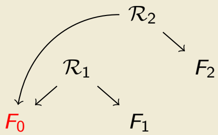


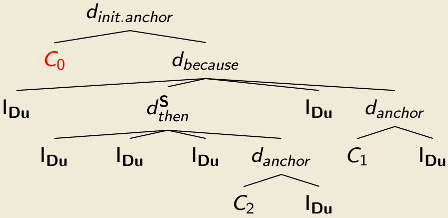


$$d_3 = d_{init.anchor}\ C_0\ (d_{because}\ \mathsf{I}_{\mathbf{Du}}\ (d^{\mathbf{S}}_{then}\ \mathsf{I}_{\mathbf{Du}}\ \mathsf{I}_{\mathbf{Du}}\ \mathsf{I}_{\mathbf{Du}}\ (d_{anchor}\ C_2\ \mathsf{I}_{\mathbf{Du}}))\ \mathsf{I}_{\mathbf{Du}}\ (d_{anchor}\ C_1\ \mathsf{I}_{\mathbf{Du}}))$$

Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.



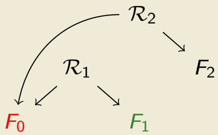


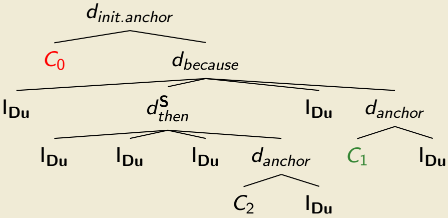
$$d_3 = d_{init.anchor}\ C_0\ (d_{because}\ \mathsf{I_{Du}}\ (d^{\mathsf{S}}_{then}\ \mathsf{I_{Du}}\ \mathsf{I_{Du}}\ \mathsf{I_{Du}}\ (d_{anchor}\ C_2\ \mathsf{I_{Du}}))\ \mathsf{I_{Du}}\ (d_{anchor}\ C_1\ \mathsf{I_{Du}}))$$

$$:= \exists_\ell l_0\ l_{\mathcal{R}_1}\ l_{\mathcal{R}_2}.\exists! x.\ l_0\colon \mathtt{supermarket}(x) \wedge l_0\colon \mathtt{go\_to}(\mathtt{fred}, x) \wedge (\exists_\ell l_2.(\exists! x. l_2\colon \mathtt{movies}(x) \wedge l_2\colon \mathtt{go\_to}(\mathtt{fred}, x)) \wedge ((\exists_\ell l_1.(\exists! x. l_1\colon \mathtt{fridge}(x) \wedge l_1\colon \mathtt{empty}(x)) \wedge (l_{\mathcal{R}_1}\colon \phi_{\mathrm{Expl.}}(l_0, l_1) \wedge \top)) \wedge l_{\mathcal{R}_2}\colon \phi_{\mathrm{Nar.}}(l_0, l_2)))$$

Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.



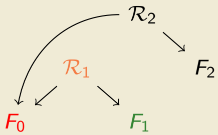


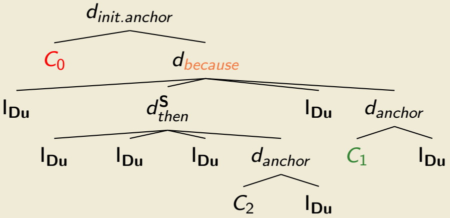
$$d_3 = d_{init.anchor}\ C_0\ (d_{because}\ \mathbf{I_{Du}}\ (d^{\mathbf{S}}_{then}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ \mathbf{I_{Du}}\ (d_{anchor}\ C_2\ \mathbf{I_{Du}}))\ \mathbf{I_{Du}}\ (d_{anchor}\ C_1\ \mathbf{I_{Du}}))$$

$$:= \exists_\ell l_0\ l_{\mathcal{R}_1}\ l_{\mathcal{R}_2}.\exists! x.\ l_0 : \texttt{supermarket}(x) \land l_0 : \texttt{go\_to}(\texttt{fred}, x) \land (\exists_\ell l_2.(\exists! x. l_2 : \texttt{movies}(x) \land l_2 : \texttt{go\_to}(\texttt{fred}, x)) \land ((\exists_\ell l_1.(\exists! x. l_1 : \texttt{fridge}(x) \land l_1 : \texttt{empty}(x)) \land (l_{\mathcal{R}_1} : \phi_{\text{Expl.}}(l_0, l_1) \land \top)) \land l_{\mathcal{R}_2} : \phi_{\text{Nar.}}(l_0, l_2)))$$

Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.



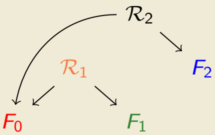


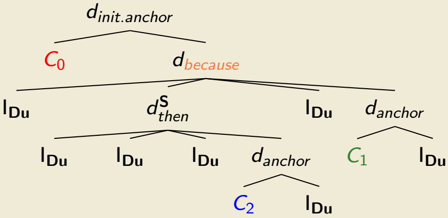
$$d_3 = d_{init.anchor}\ C_0\ (d_{because}\ \mathsf{I_{Du}}\ (d^{\mathsf{S}}_{then}\ \mathsf{I_{Du}}\ \mathsf{I_{Du}}\ \mathsf{I_{Du}}\ (d_{anchor}\ C_2\ \mathsf{I_{Du}}))\ \mathsf{I_{Du}}\ (d_{anchor}\ C_1\ \mathsf{I_{Du}}))$$

$$:= \exists_\ell l_0\ l_{\mathcal{R}_1}\ l_{\mathcal{R}_2}.\exists! x.\ l_0\colon \mathtt{supermarket}(x) \wedge l_0\colon \mathtt{go\_to}(\mathtt{fred}, x) \wedge (\exists_\ell l_2.(\exists! x. l_2\colon \mathtt{movies}(x) \wedge l_2\colon \mathtt{go\_to}(\mathtt{fred}, x)) \wedge ((\exists_\ell l_1.(\exists! x. l_1\colon \mathtt{fridge}(x) \wedge l_1\colon \mathtt{empty}(x)) \wedge (l_{\mathcal{R}_1}\colon \phi_{\mathsf{Expl.}}(l_0, l_1) \wedge \top)) \wedge l_{\mathcal{R}_2}\colon \phi_{\mathsf{Nar.}}(l_0, l_2)))$$

Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.



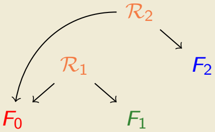


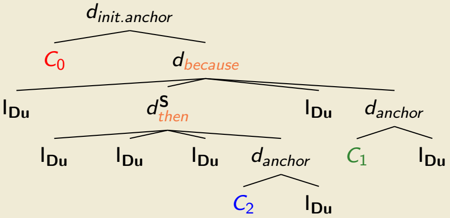
$$d_3 = d_{init.anchor}\ C_0\ (d_{because}\ \mathsf{I}_{\mathbf{Du}}\ (d^{\mathbf{S}}_{then}\ \mathsf{I}_{\mathbf{Du}}\ \mathsf{I}_{\mathbf{Du}}\ \mathsf{I}_{\mathbf{Du}}\ (d_{anchor}\ C_2\ \mathsf{I}_{\mathbf{Du}}))\ \mathsf{I}_{\mathbf{Du}}\ (d_{anchor}\ C_1\ \mathsf{I}_{\mathbf{Du}}))$$

$$:= \exists_\ell l_0\ l_{\mathcal{R}_1}\ l_{\mathcal{R}_2}.\exists! x.\ l_0\colon \texttt{supermarket}(x) \wedge l_0\colon \texttt{go\_to}(\texttt{fred}, x) \wedge (\exists_\ell l_2.(\exists! x. l_2\colon \texttt{movies}(x) \wedge l_2\colon \texttt{go\_to}(\texttt{fred}, x)) \wedge ((\exists_\ell l_1.(\exists! x. l_1\colon \texttt{fridge}(x) \wedge l_1\colon \texttt{empty}(x)) \wedge (l_{\mathcal{R}_1}\colon \phi_{\text{Expl.}}(l_0, l_1) \wedge \top)) \wedge l_{\mathcal{R}_2}\colon \phi_{\text{Nar.}}(l_0, l_2)))$$

Example

(4) [Fred went to a supermarket]$_0$ because [his fridge was empty]$_1$. Then, [he went to the cinema]$_2$.





$$d_3 = d_{init.anchor}\ C_0\ (d_{because}\ \mathsf{I_{Du}}\ (d^{\mathsf{S}}_{then}\ \mathsf{I_{Du}}\ \mathsf{I_{Du}}\ \mathsf{I_{Du}}\ (d_{anchor}\ C_2\ \mathsf{I_{Du}}))\ \mathsf{I_{Du}}\ (d_{anchor}\ C_1\ \mathsf{I_{Du}}))$$

$$:= \exists_\ell l_0\ l_{\mathcal{R}_1}\ l_{\mathcal{R}_2}.\exists! x.\ l_0\colon \mathtt{supermarket}(x) \wedge l_0\colon \mathtt{go\_to}(\mathtt{fred}, x) \wedge (\exists_\ell l_2.(\exists! x. l_2\colon \mathtt{movies}(x) \wedge l_2\colon \mathtt{go\_to}(\mathtt{fred}, x)) \wedge ((\exists_\ell l_1.(\exists! x. l_1\colon \mathtt{fridge}(x) \wedge l_1\colon \mathtt{empty}(x)) \wedge (l_{\mathcal{R}_1}\colon \phi_{\mathrm{Expl.}}(l_0, l_1) \wedge \top)) \wedge l_{\mathcal{R}_2}\colon \phi_{\mathrm{Nar.}}(l_0, l_2)))$$

Conclusion

Summary of the current results

- Purely grammatical approach to 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

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

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 !

References I

Asher, Nicholas et Alex Lascarides (2003). *Logics of Conversation*. Cambridge University Press.

Danlos, Laurence (2011). “D-STAG : a Formalism for Discourse Analysis based on SDRT and using Synchronous TAG”. In : *14th conference on Formal Grammar - FG 2009*. Sous la dir. de Philippe de Groote, Markus Egg et Laura Kallmeyer. T. 5591. LNCS/LNAI. Springer, p. 64–84. doi : 10.1007/978-3-642-20169-1_5. url : http://webloria.loria.fr/~degroote/FG09/Danlos.pdf.

De Groote, Philippe (2001). “Towards Abstract Categorial Grammars”. In : *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, p. 148–155. acl : P01-1033. url : http://aclweb.org/anthology/P/P01/P01-1033.

Joshi, Aravind K., Leon S. Levy et Masako Takahashi (1975). “Tree Adjunct Grammars”. In : *Journal of Computer and System Sciences* 10.1, p. 136–163. doi : 10.1016/S0022-0000(75)80019-5.

Mann, William C. et Sandra A. Thompson (1987). *Rhetorical Structure Theory : A Theory of Text Organization*. Rapp. tech. ISI/RS-87-190. Information Sciences Institute.

Pogodalla, Sylvain (2009). “Advances in Abstract Categorial Grammars : Language Theory and Linguistic Modeling. ESSLLI 2009 Lecture Notes, Part II”. url : http://hal.inria.fr/hal-00749297.

References II

Shieber, Stuart M. et Yves Schabes (1990). "Synchronous Tree-adjoining Grammars". In : *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*. COLING '90. Helsinki, Finland : Association for Computational Linguistics, p. 253–258. isbn : 952-90-2028-7. doi : 10.3115/991146.991191.

Webber, Bonnie et al. (1999). "Discourse Relations : A Structural and Presuppositional Account Using Lexicalised TAG". In : *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. College Park, Maryland, USA : Association for Computational Linguistics, p. 41–48. url : http://www.aclweb.org/anthology/P99-1006.