Extending a Surface Realizer to Generate Coherent Discourse

Eva Banik

Department of Computing
The Open University

June 5th, 2009

Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

1. Referential Coherence

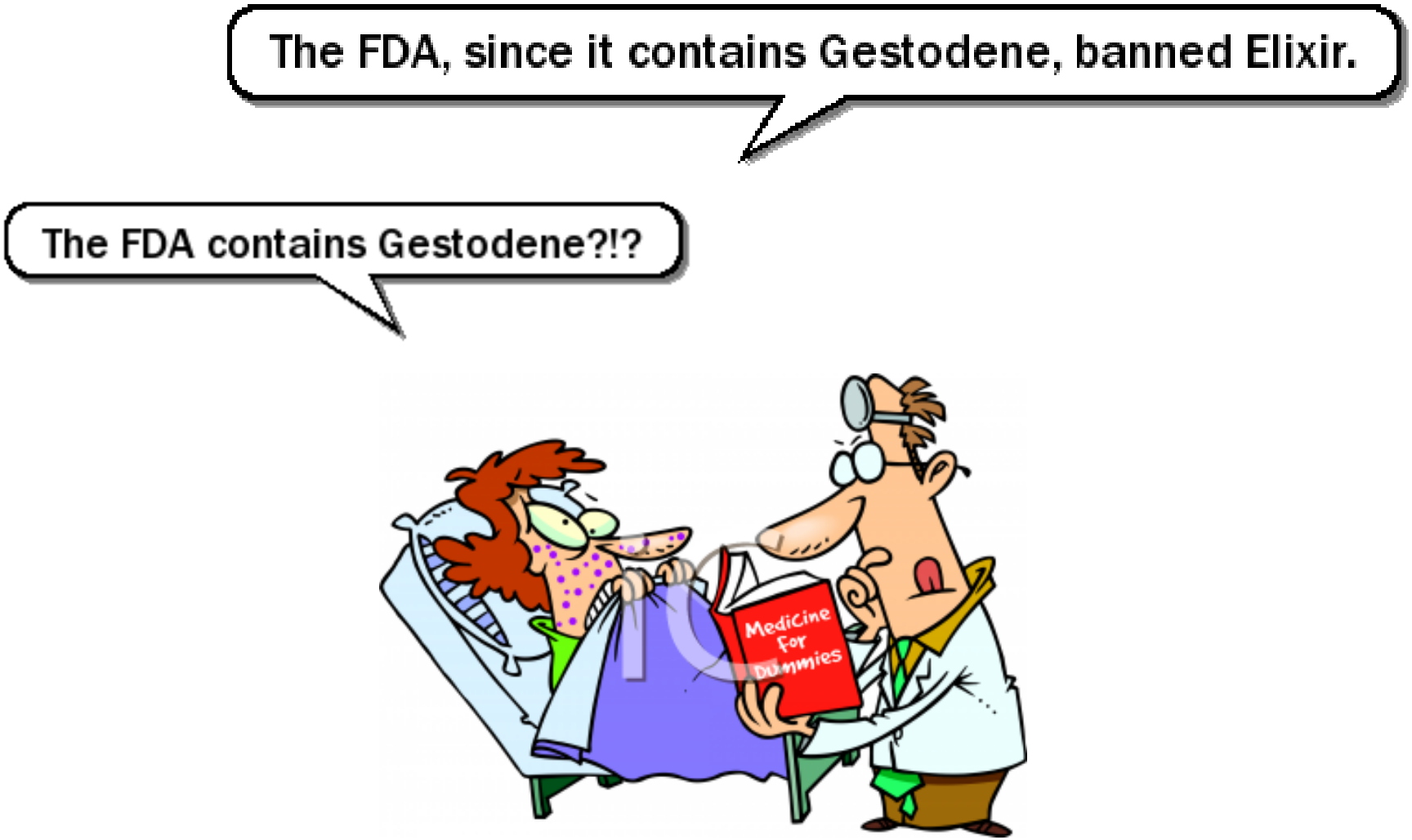
1. Referential Coherence

2. Correct Emphasis

3. Pronominalization

3. Pronominalization

3. Pronominalization



3. Pronominalization

Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

NLG system architectures

- What to say? — Content Determination, Content Selection, Strategic Planner
- How to say it? — Tactical component

How to say it?

Content Planning

- Discourse Planning

Microplanning

- Sentence Aggregation
- Lexicalization
- Referring Expression Generation

Realization

- Linguistic Realization
- Morphology
- Formatting

- The difficulty: modelling constraints across grammatical levels

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- Add extra modules to the pipeline

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- Add extra modules to the pipeline
  - Shaw(2002)
    RefExp generation + Aggregation + RefExp generation (again!)

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- Add extra modules to the pipeline
  - Shaw(2002)
    RefExp generation + Aggregation + RefExp generation (again!)
  - Robin(1994)
    Revision + Revision Correction

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- Add extra modules to the pipeline
  - Shaw(2002)
    RefExp generation + Aggregation + RefExp generation (again!)
  - Robin(1994)
    Revision + Revision Correction
- Ranking constraints based on centering theory
  - ICONOCLAST : Kibble and Power(2004)

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- the research challenges in NLG become system engineering tasks:

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- the research challenges in NLG become system engineering tasks:
  - what modules should a system have?

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- the research challenges in NLG become system engineering tasks:
  - what modules should a system have?
  - how should these modules be ordered?

- The difficulty: modelling constraints across grammatical levels

Previous Approaches

- the research challenges in NLG become system engineering tasks:
  - what modules should a system have?
  - how should these modules be ordered?
  - what interaction should be allowed between modules?

In this talk:

Microplanning

- Sentence Aggregation
- Lexicalization
- Referring Expression Generation

Realization

- Linguistic Realization
- Morphology
- Formatting

A few grammar engineering tricks...

can make it possible to generate coherent discourse

with a surface realizer...

efficiently!

A few grammar engineering tricks...

can make it possible to generate coherent discourse

with a surface realizer...

efficiently!

Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

GenI — Kow (2007)

- uses a Feature-Based Lexicalized Tree Adjoining Grammar
- input: Flat Semantics
- output: all possible realizations of the input semantics

The Generation Algorithm

- Tree Selection
- Tree Assembly
  1. substitution
  2. adjunction
- Linearization

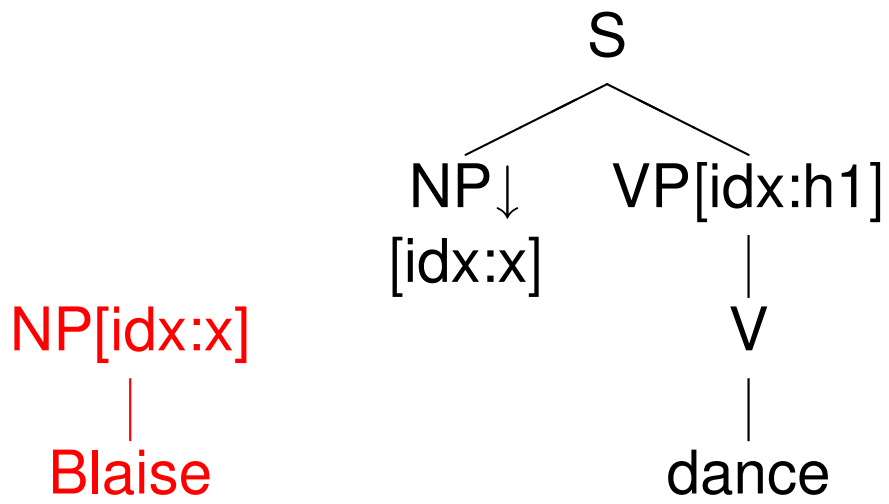
Input:

```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

Tree Selection:

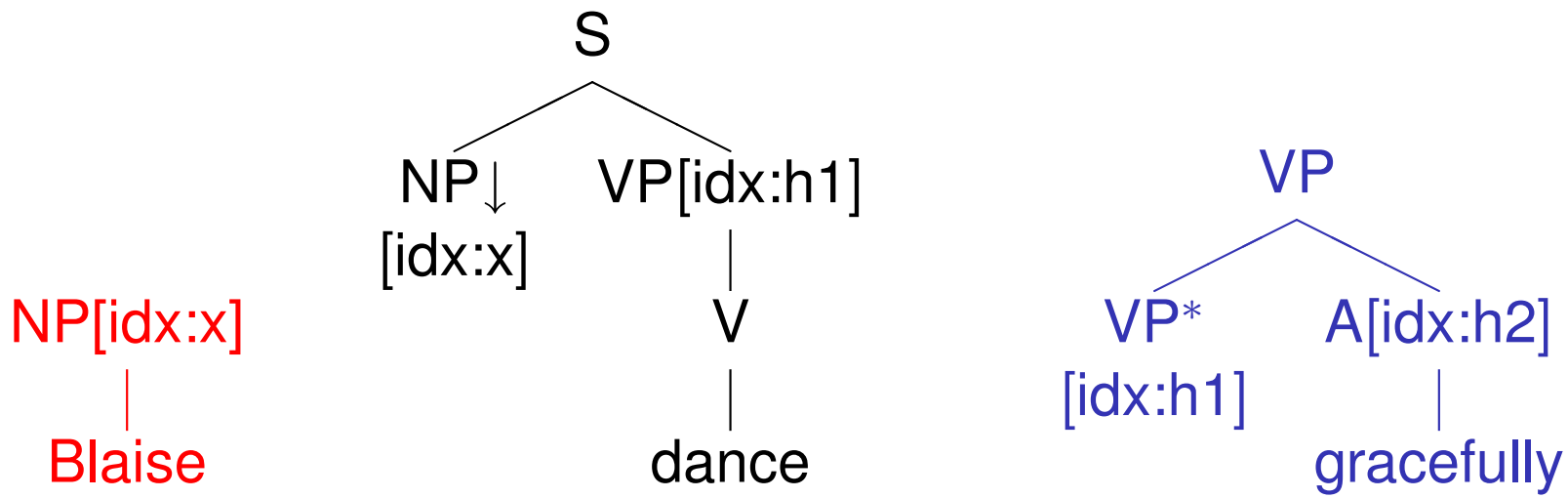

```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

Tree Selection:



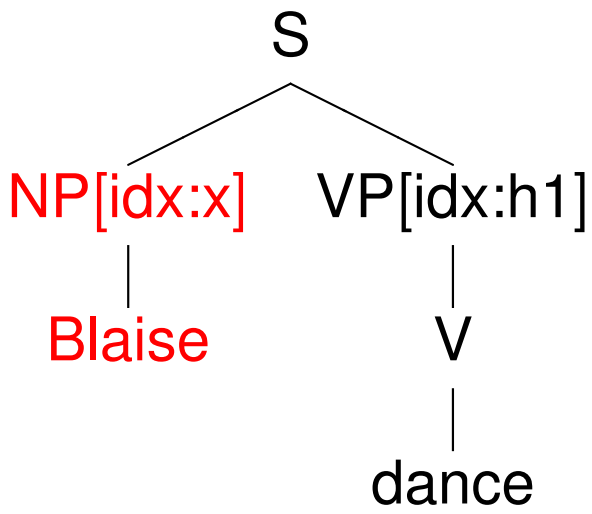

```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

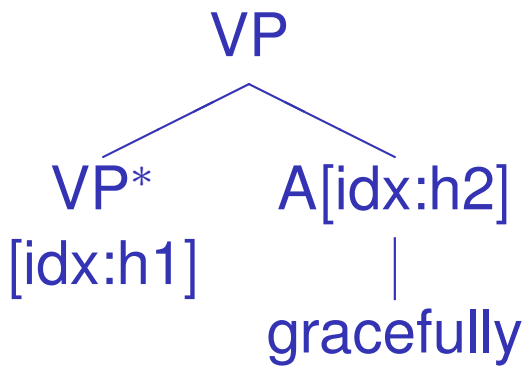
Tree Selection:




```
h0:blaise(x)      h1:dance(x)      h2:gracefully(h1)
```

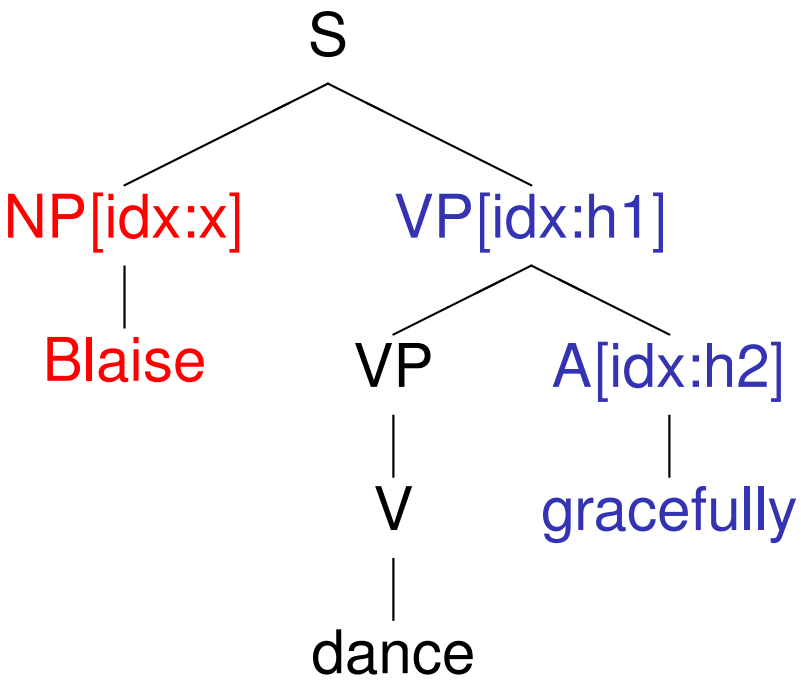
Tree Assembly Phase 1: Substitution





`h0:blaise(x)` `h1:dance(x)` `h2:gracefully(h1)`

Tree Assembly Phase 2: Adjunction




```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

Linearization + Morphology:

Blaise dances gracefully

`h0:blaise(x)` `h1:dance(x)` `h2:gracefully(h1)`

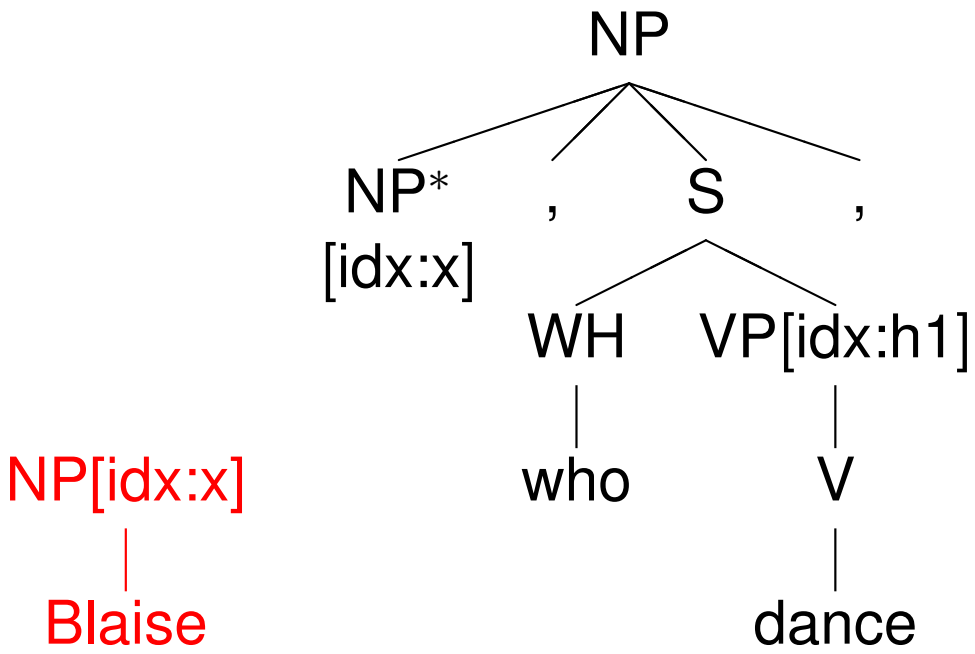
Input:

```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

Tree Selection:

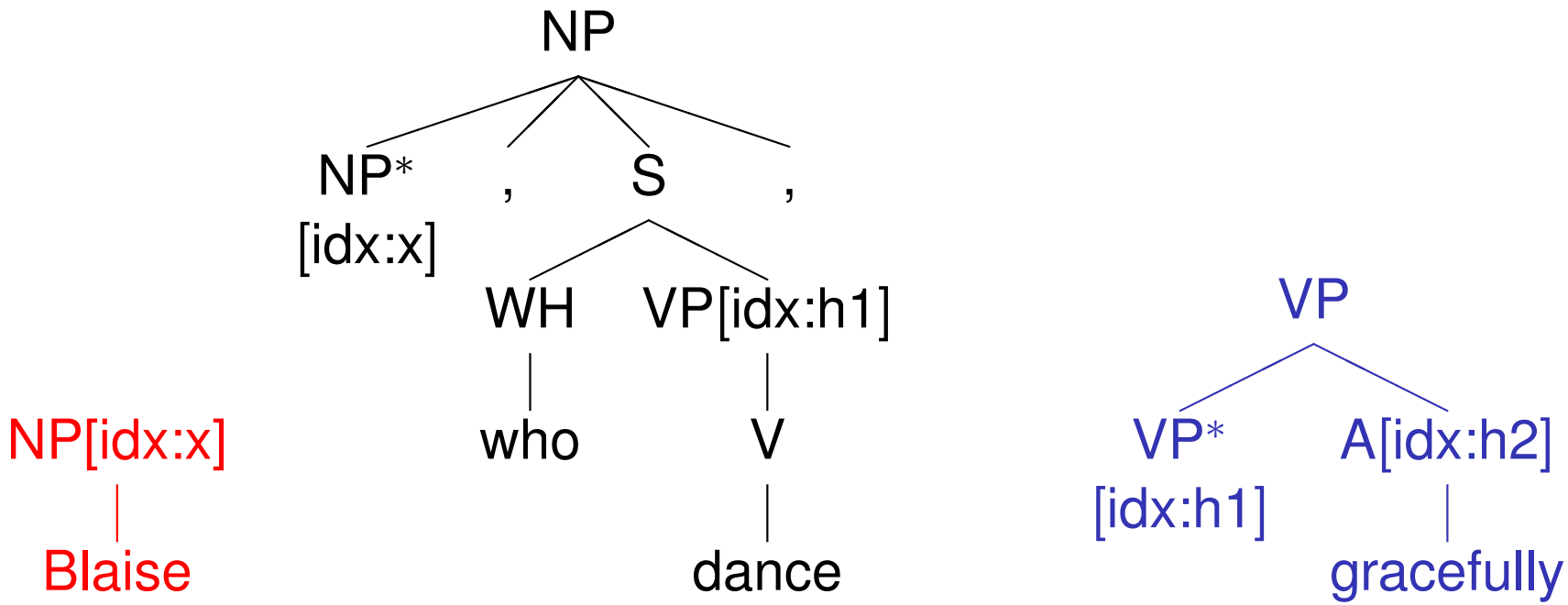

```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

Tree Selection:



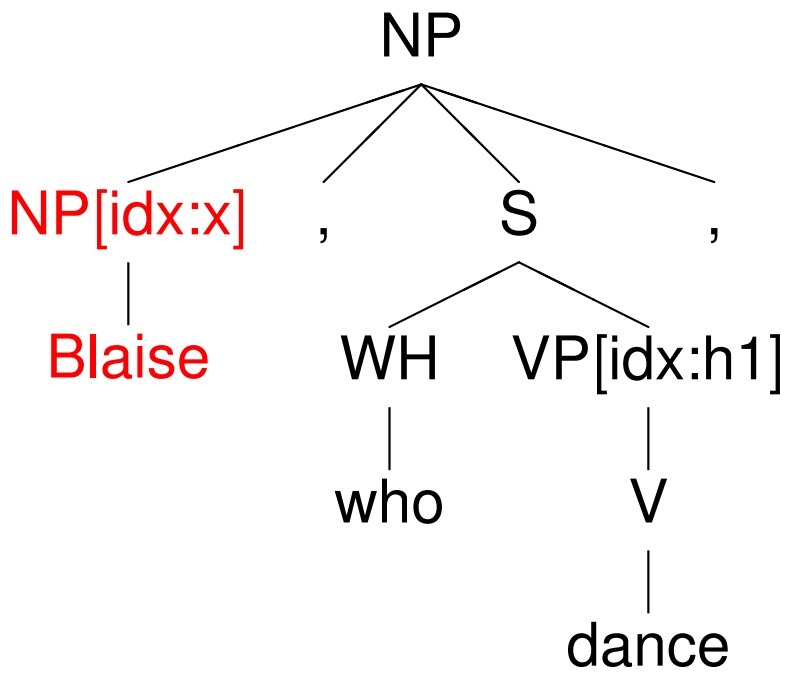

```
h0:blaise(x)        h1:dance(x)        h2:gracefully(h1)
```

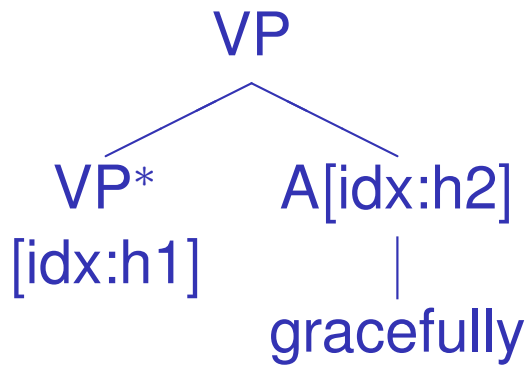
Tree Selection:



`h0:blaise(x)` `h1:dance(x)` `h2:gracefully(h1)`

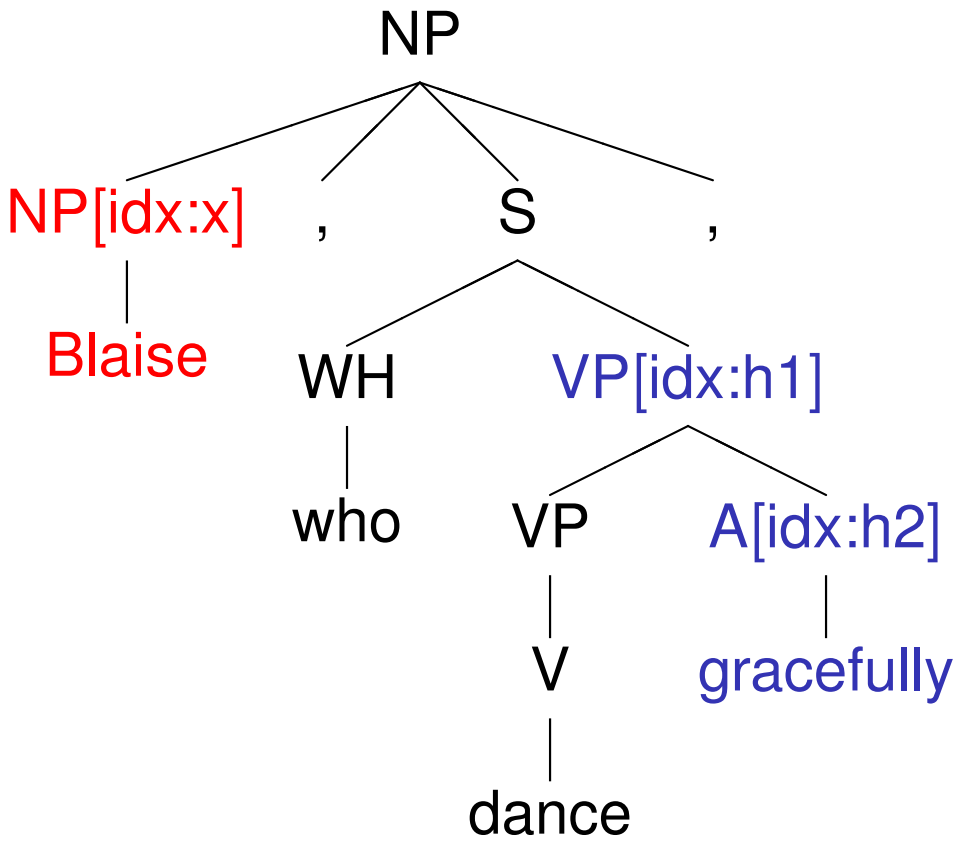
Tree Assembly Phase 2: Adjunction






```
h0:blaise(x)        h1:dance(x)        h2:gracefully(h1)
```

Tree Assembly Phase 2: Adjunction




```
h0:blaise(x)    h1:dance(x)    h2:gracefully(h1)
```

Linearization + Morphology:

Blaise, who dances gracefully,

`h0:blaise(x)` `h1:dance(x)` `h2:gracefully(h1)`

The Generation Algorithm

- Tree Selection
- Tree Assembly
  1. substitution
  2. adjunction
- Linearization

The Generation Algorithm

- Tree Selection
- Tree Assembly ⇐ Very expensive!
  1. substitution
  2. adjunction
- Linearization

The Generation Algorithm

- Tree Selection + Polarity Filtering
- Tree Assembly
  1. substitution
  2. adjunction
- Linearization

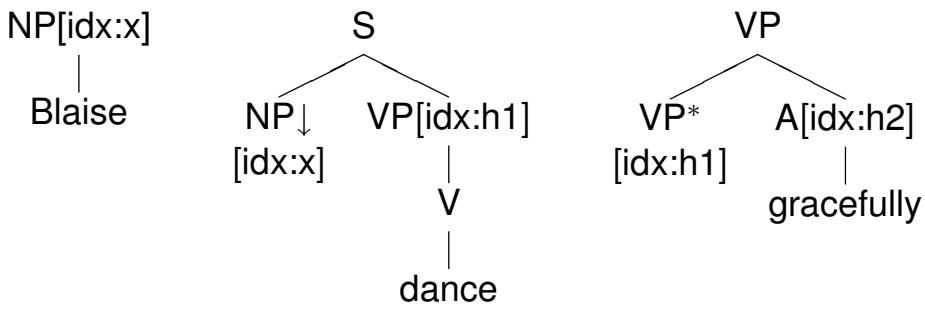
Polarity filtering

Tree set 1.

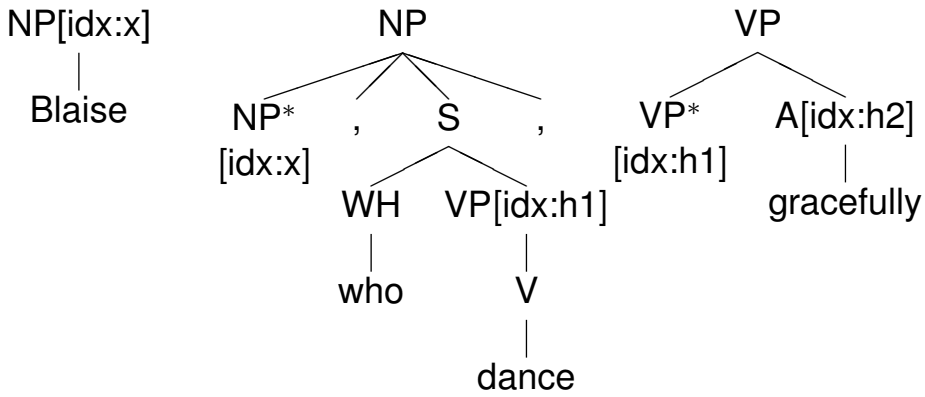
Tree set 2.

Polarity filtering

Tree set 1.



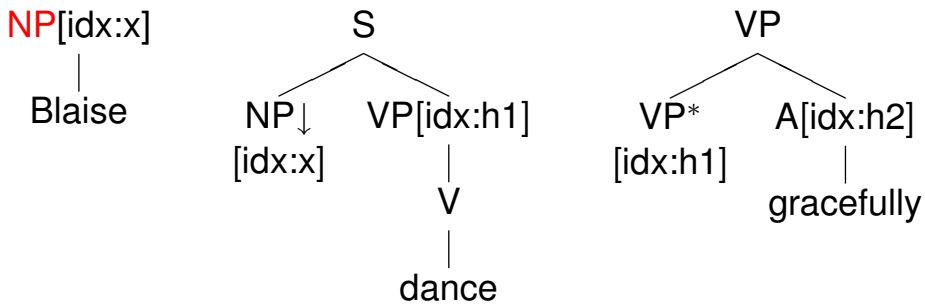
Tree set 2.



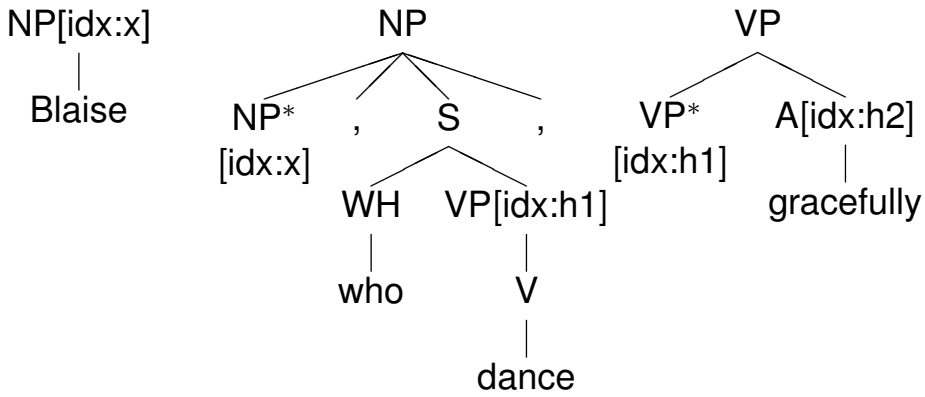
Polarity filtering

+NP

Tree set 1.

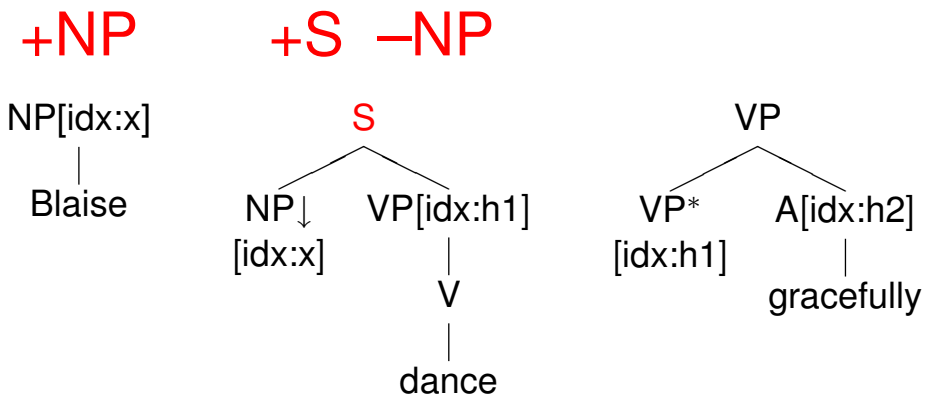


Tree set 2.

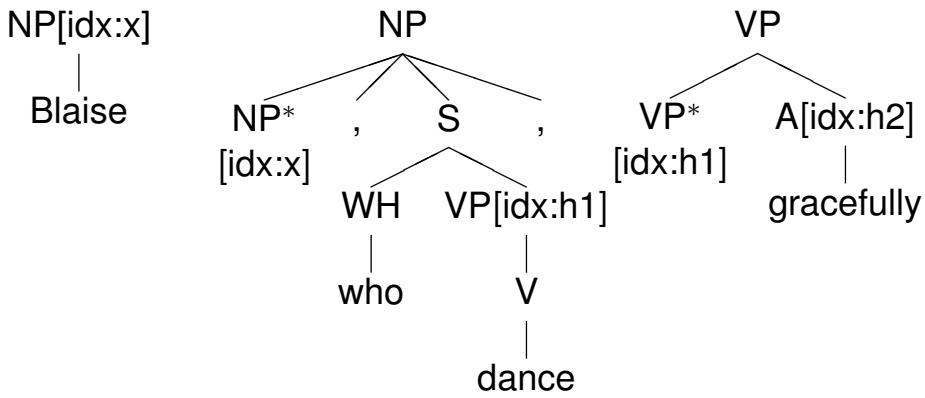


Polarity filtering

Tree set 1.

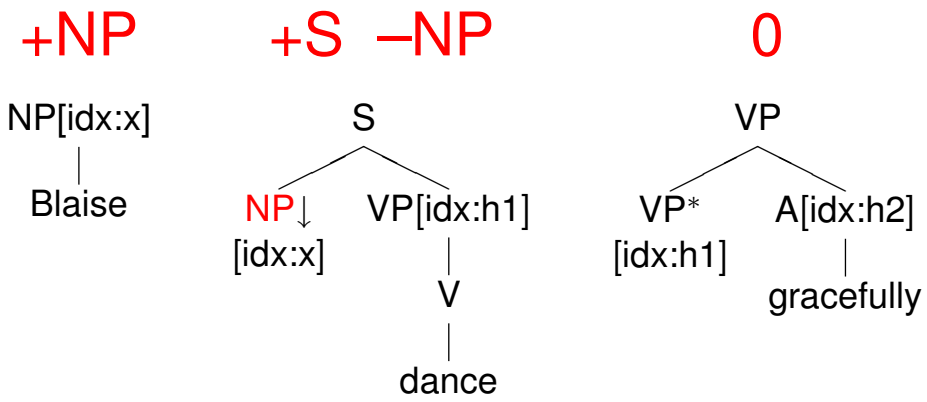


Tree set 2.

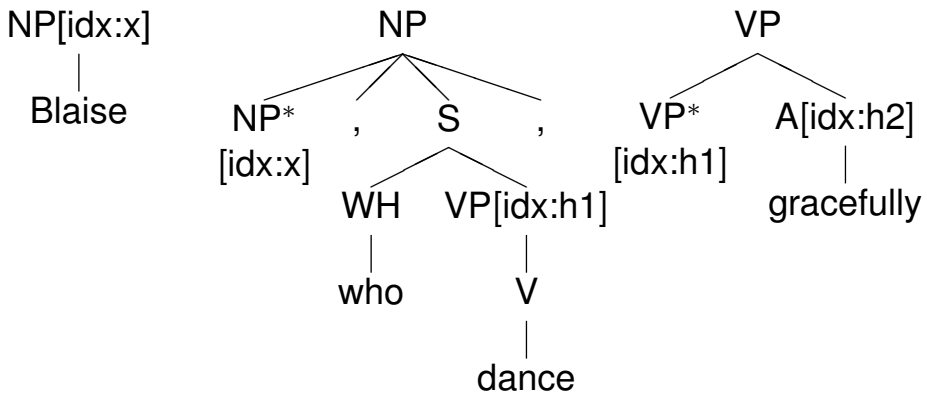


Polarity filtering

Tree set 1.



Tree set 2.

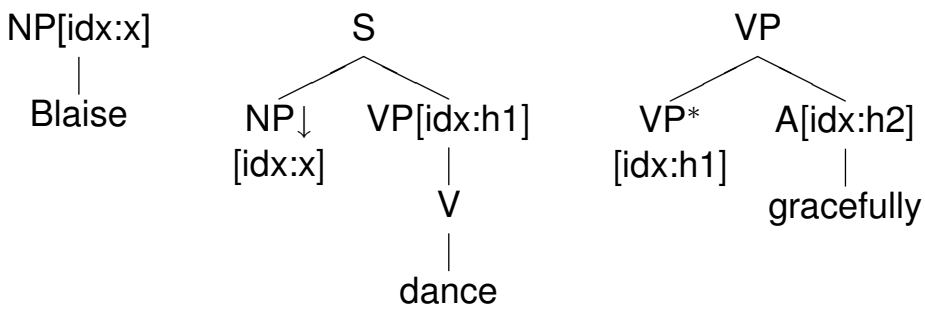


Polarity filtering

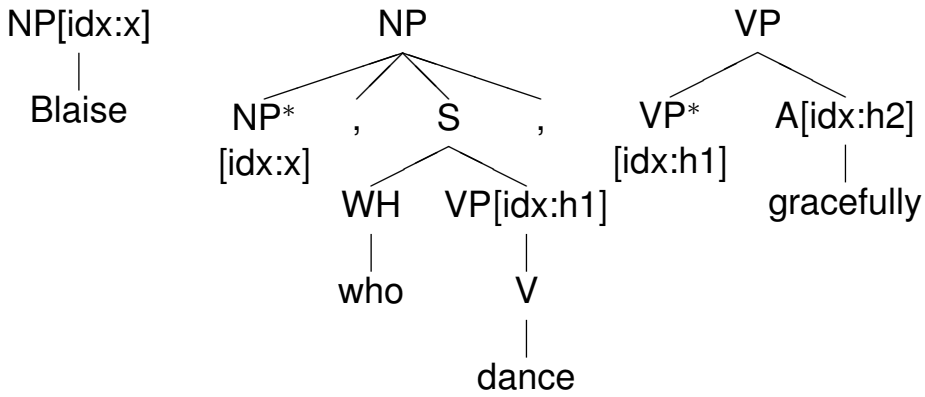
Sum: +S

+NP +S –NP 0

Tree set 1.



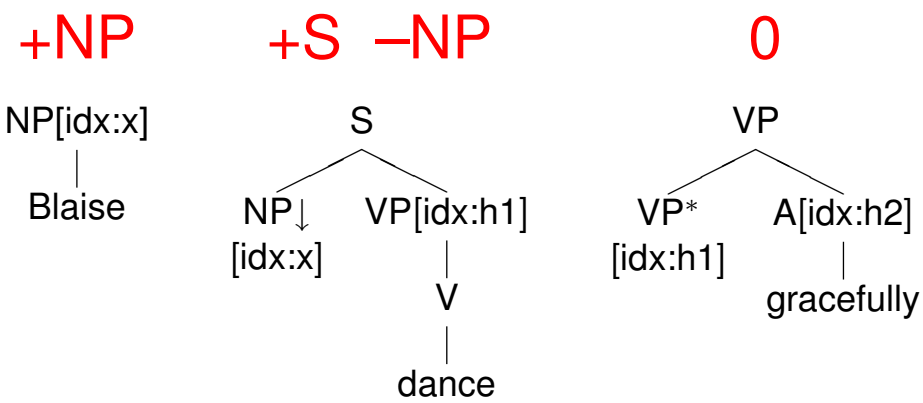
Tree set 2.



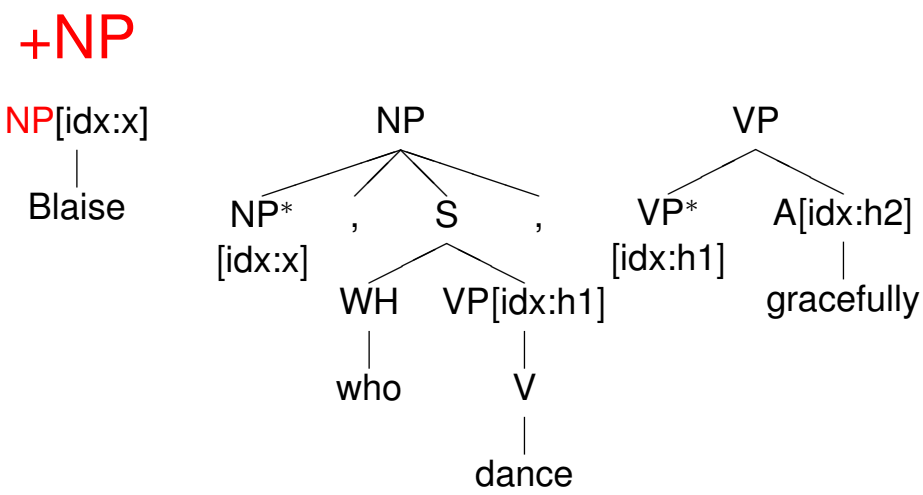
Polarity filtering

Sum: +S

Tree set 1.



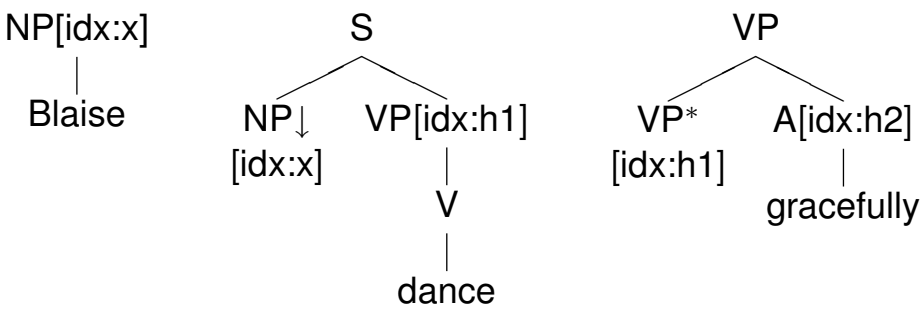
Tree set 2.



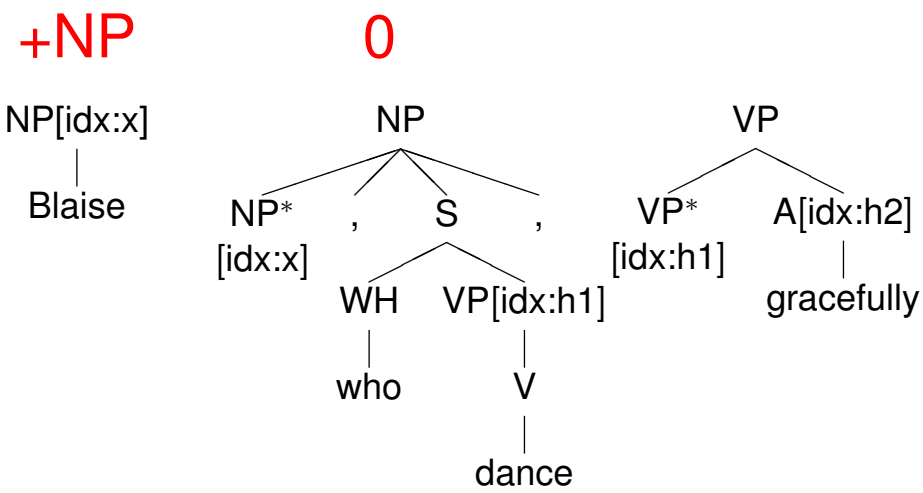
Polarity filtering

Sum: +S

Tree set 1.



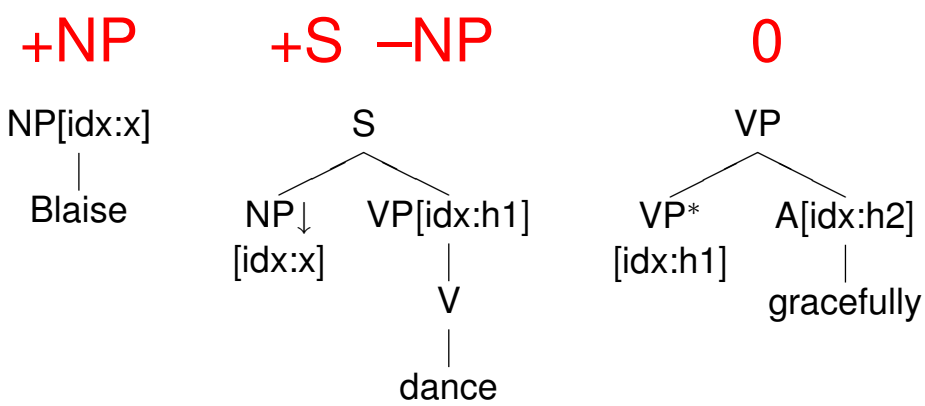
Tree set 2.



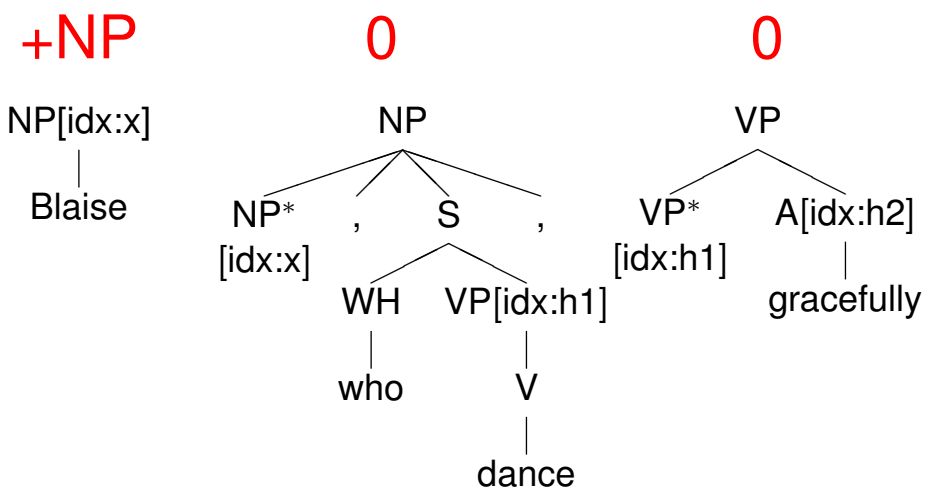
Polarity filtering

Sum: +S

Tree set 1.



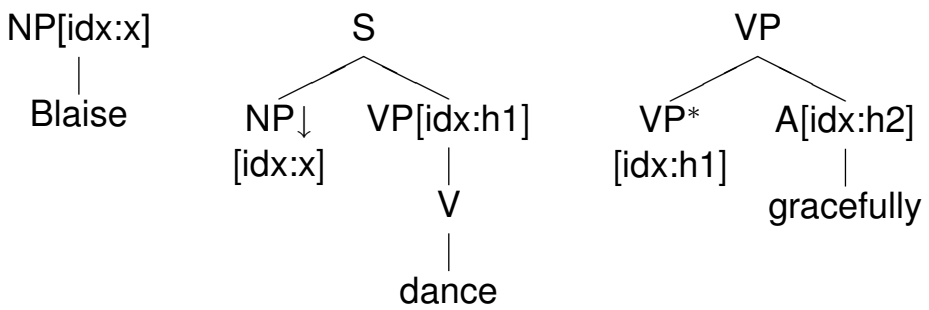
Tree set 2.



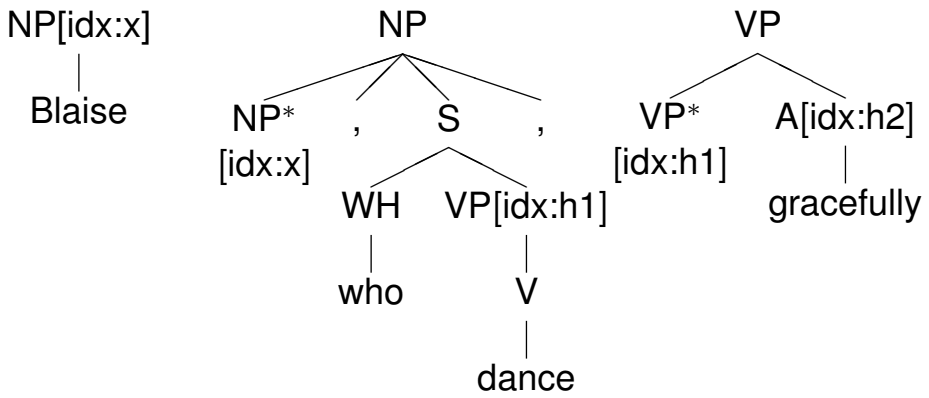
Polarity filtering

Sum:

Tree set 1. +NP +S –NP 0 +S



Tree set 2. +NP 0 0 +NP



Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

A few grammar engineering tricks...

can make it possible to generate coherent discourse

with a surface realizer...

efficiently!

A few grammar engineering tricks...

1. extend the *domain of locality* of elementary trees
2. annotate nodes with discourse entities
3. separate discourse entities from predicates

can make it possible to generate coherent discourse

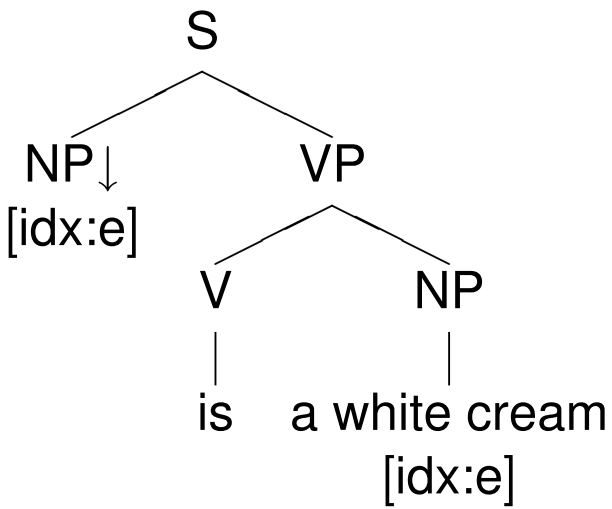
with a surface realizer...

efficiently!

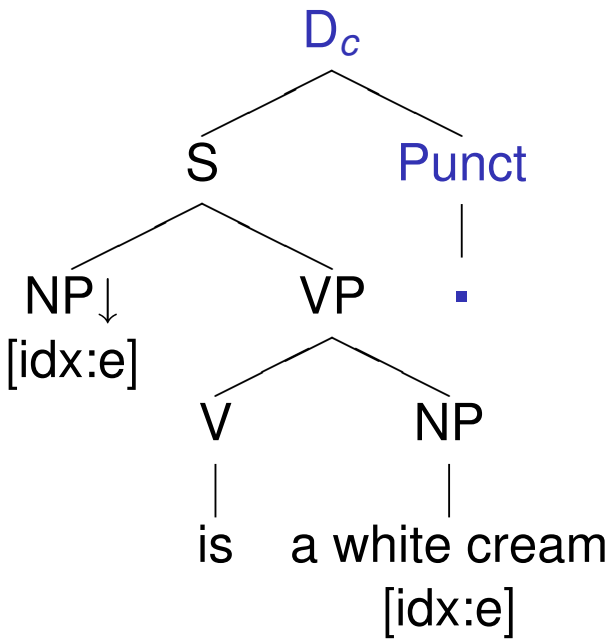
Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

1. Extend the *domain of locality* of elementary trees to the discourse level

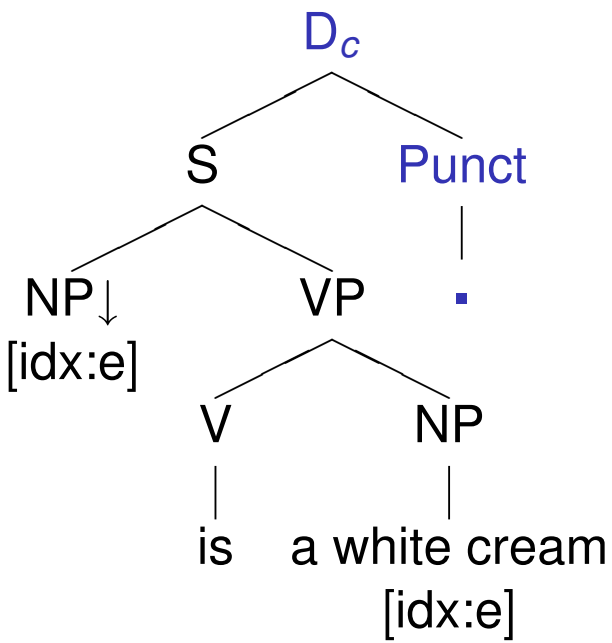


1. Extend the *domain of locality* of elementary trees to the discourse level



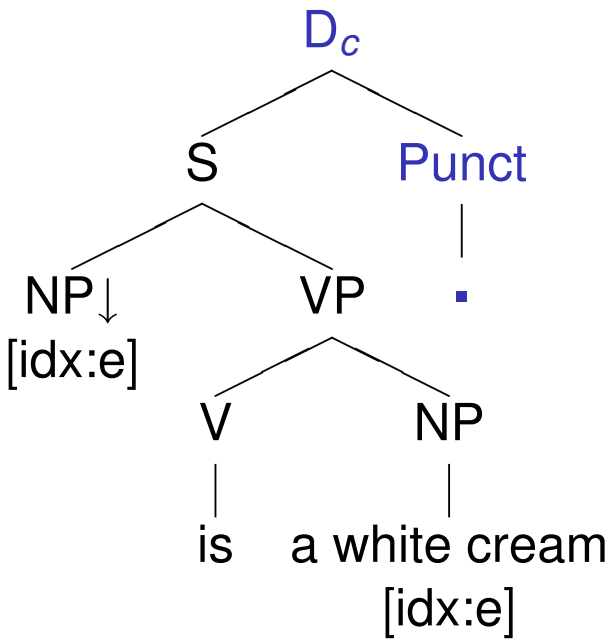
1. Extend the *domain of locality* of elementary trees to the discourse level

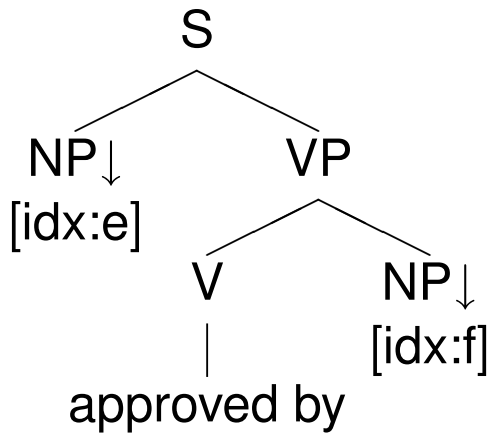
Discourse Initial



1. Extend the *domain of locality* of elementary trees to the discourse level

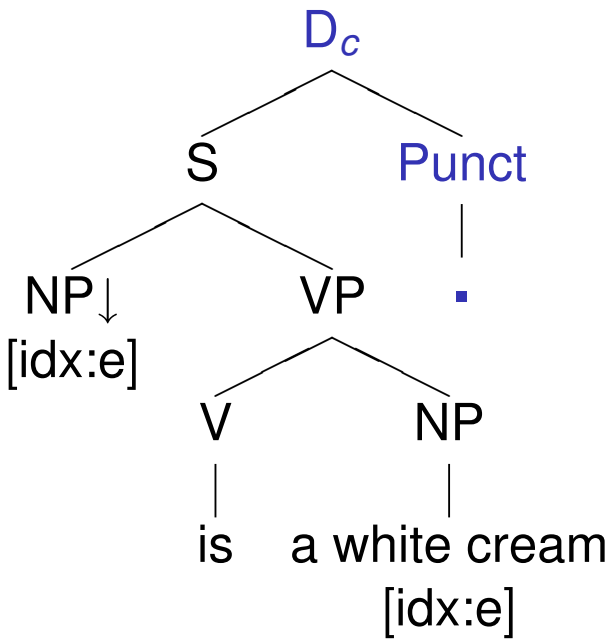
Discourse Initial

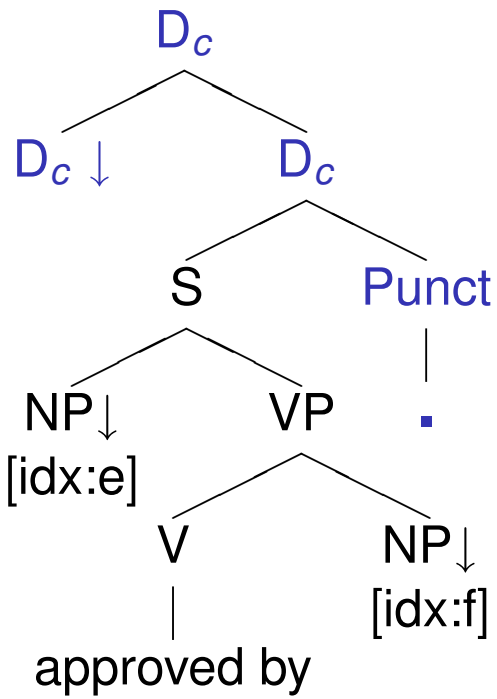




1. Extend the *domain of locality* of elementary trees to the discourse level

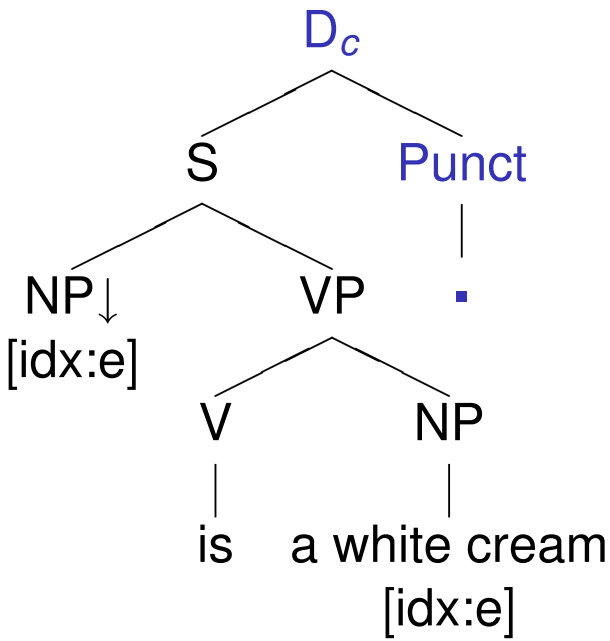
Discourse Initial



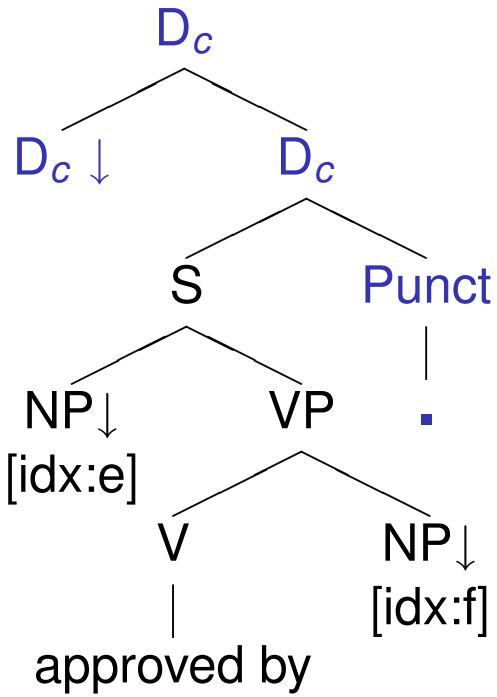


1. Extend the *domain of locality* of elementary trees to the discourse level

Discourse Initial



Discourse Continuing



Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment

U1: Elixir$_e$ is a white cream.
U2: It$_e$ is approved by the FDA$_f$.

Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment

U1: Elixir$_e$ is a white cream. [e]
U2: It$_e$ is approved by the FDA$_f$. [e,f]

Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment
- Each utterance has a set of *forward looking* centers and exactly one *backward looking* center...

U1: Elixir$_e$ is a white cream.
U2: It$_e$ is approved by the FDA$_f$.

Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment
- Each utterance has a set of *forward looking* centers and exactly one *backward looking* center...

U1: Elixir$_e$ is a white cream. `Cf = [e]`
U2: It$_e$ is approved by the FDA$_f$. `Cf = [e,f]`

Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment
- Each utterance has a set of *forward looking* centers and exactly one *backward looking* center...
- except for the discourse-initial utterance, which has no backward looking center.

U1: Elixir$_e$ is a white cream. `Cf = [e]`
U2: It$_e$ is approved by the FDA$_f$. `Cb = [e] Cf = [e,f]`

Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment
- Each utterance has a set of *forward looking* centers and exactly one *backward looking* center...
- except for the discourse-initial utterance, which has no backward looking center.
- The backward looking center of Un+1 *connects* with one of the forward-looking centers of Un

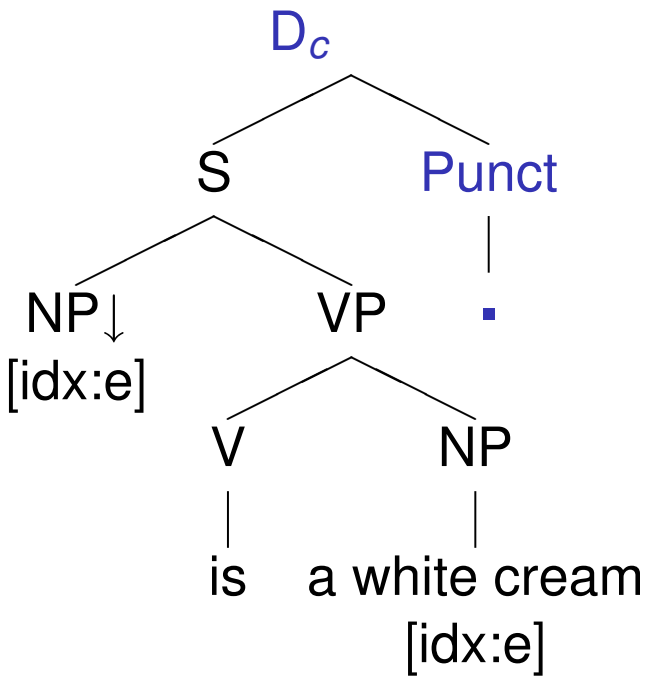
U1: Elixir$_e$ is a white cream. `Cf = [e]`
U2: It$_e$ is approved by the FDA$_f$. `Cb = [e]Cf = [e,f]`

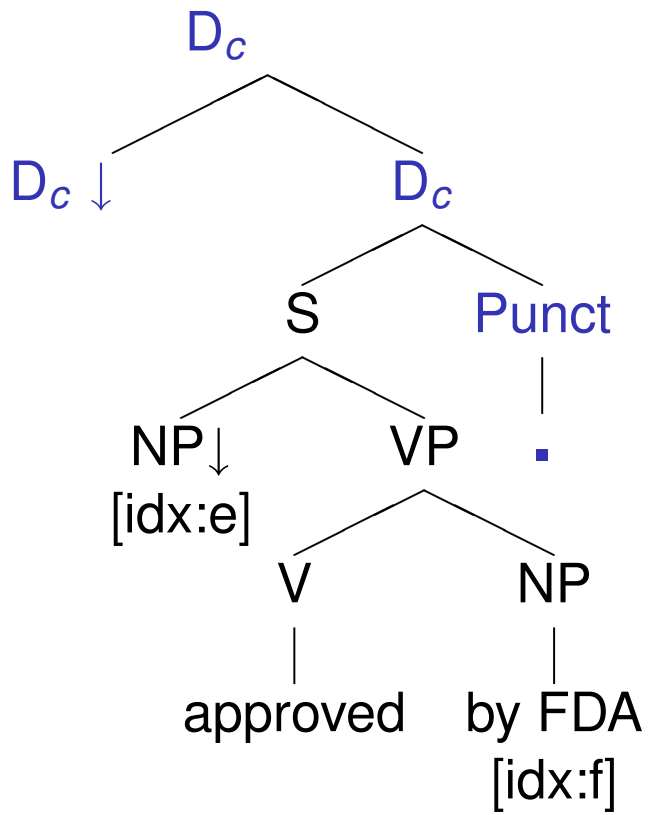
Ideas from Centering Theory (Grosz et al., 1995)

- The *centers* of an utterance Un are those entities that link Un to other utterances in the discourse segment
- Each utterance has a set of *forward looking* centers and exactly one *backward looking* center...
- except for the discourse-initial utterance, which has no backward looking center.
- The backward looking center of Un+1 *connects* with one of the forward-looking centers of Un

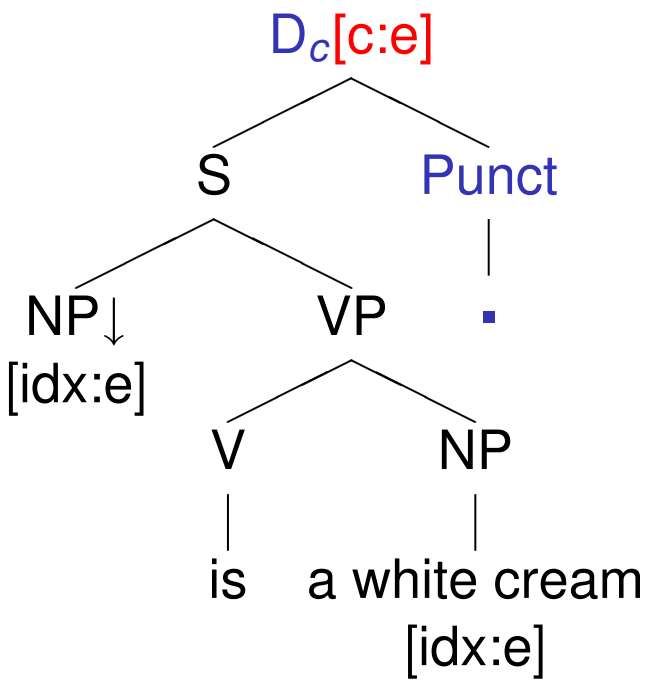
U1: Elixir$_e$ is a white cream. Cf = [e]
U2: It$_e$ is approved by the FDA$_f$. Cb = [e] Cf = [e,f]

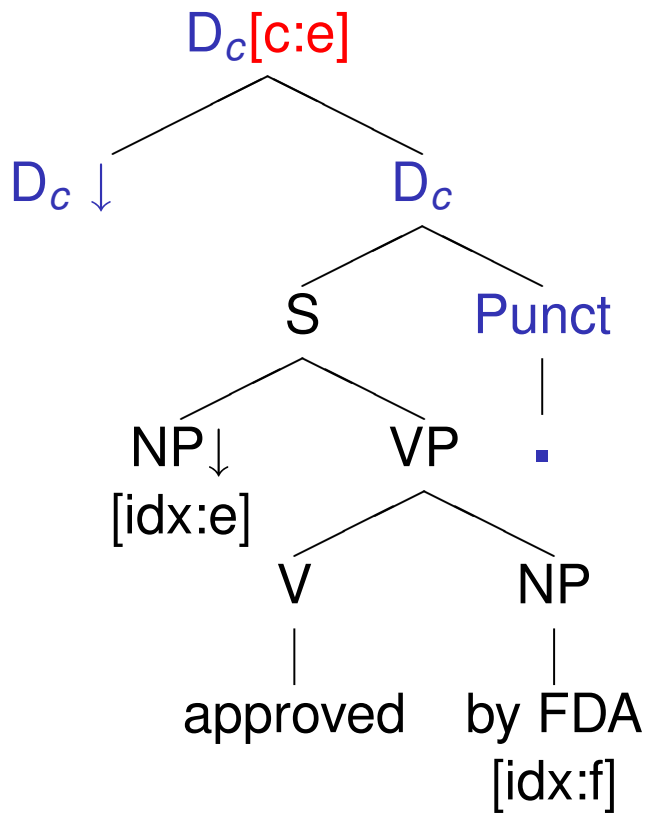
2. Annotate discourse-level root nodes and substitution nodes with discourse entities



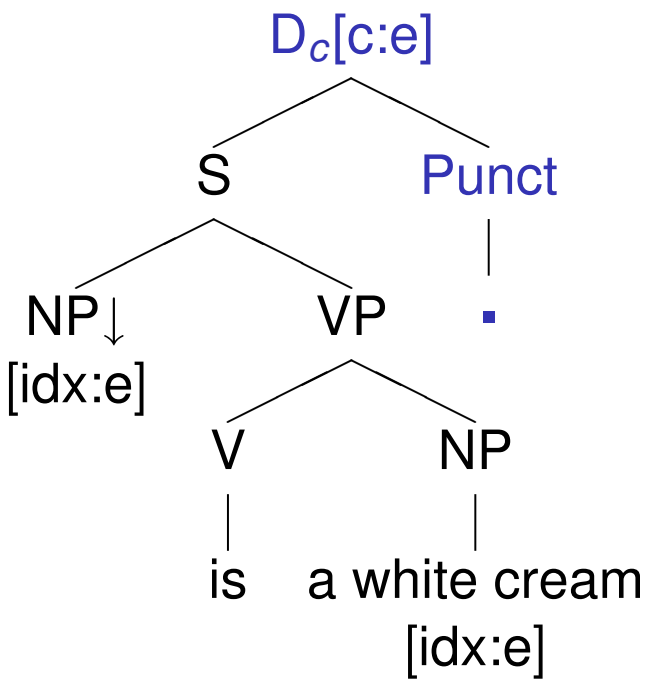


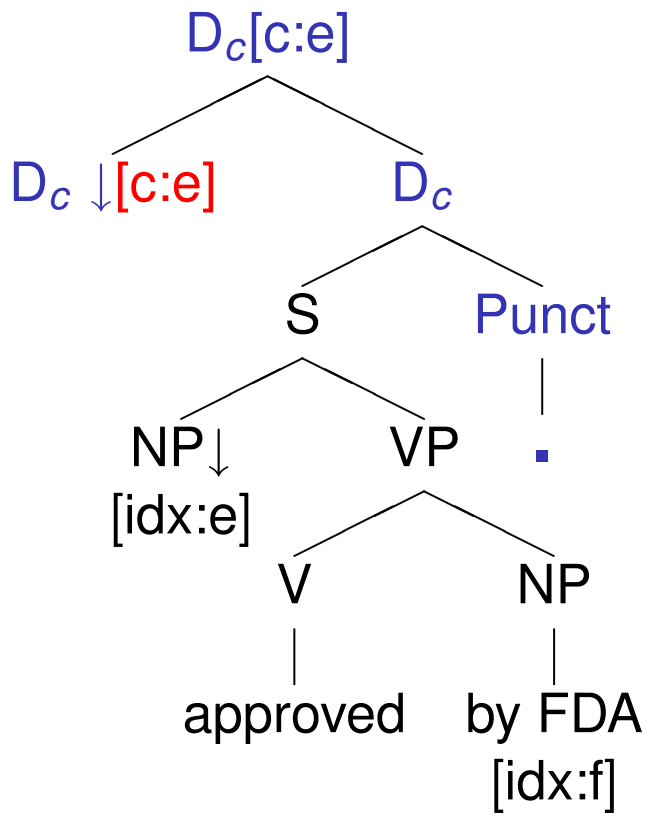
2. Annotate discourse-level root nodes and substitution nodes with discourse entities



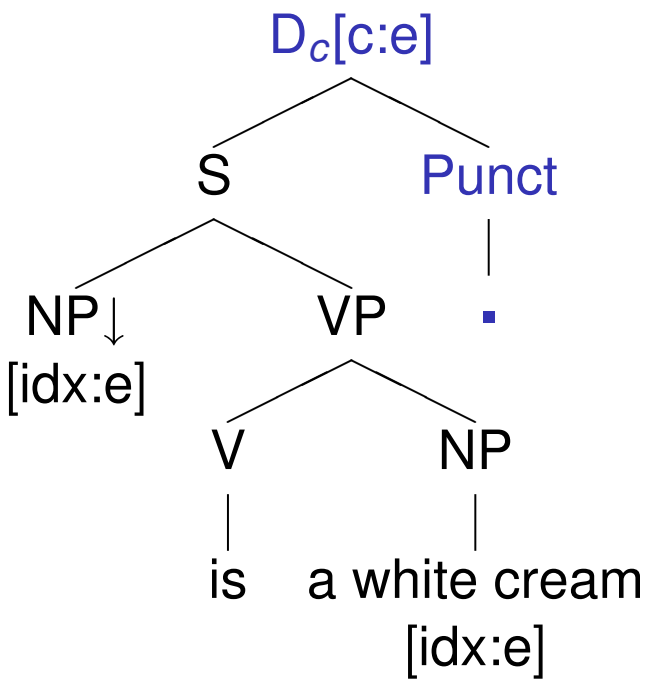


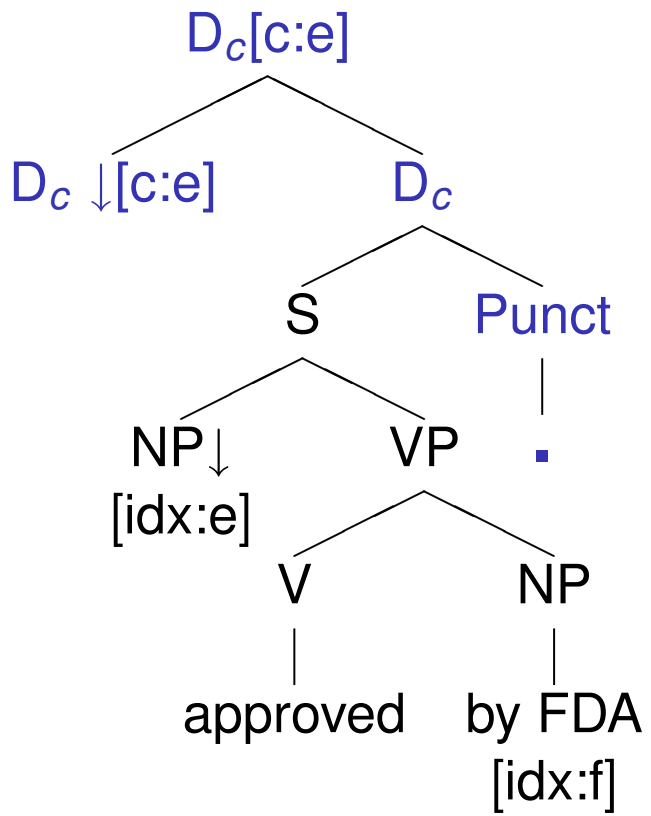
2. Annotate discourse-level root nodes and substitution nodes with discourse entities





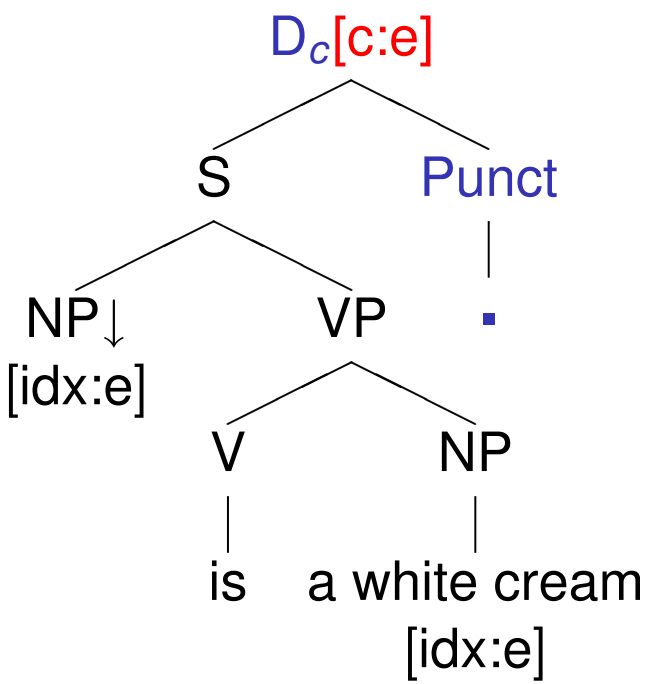
2. Annotate discourse-level root nodes and substitution nodes with discourse entities = Discourse Polarities

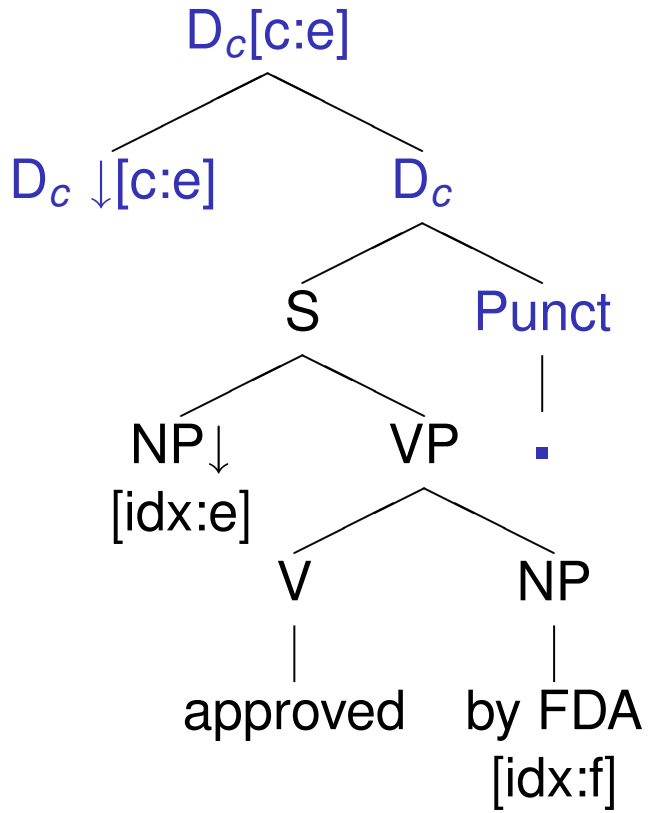




2. Annotate discourse-level root nodes and substitution nodes with discourse entities = Discourse Polarities

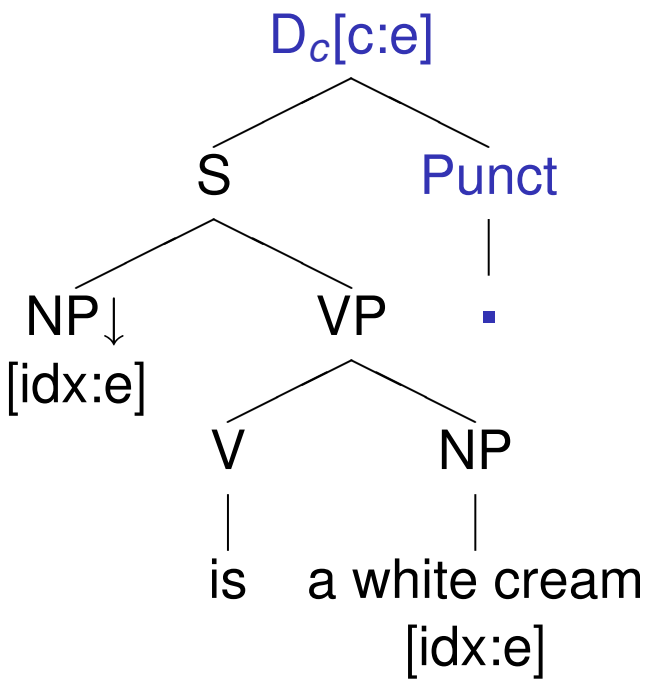
+e



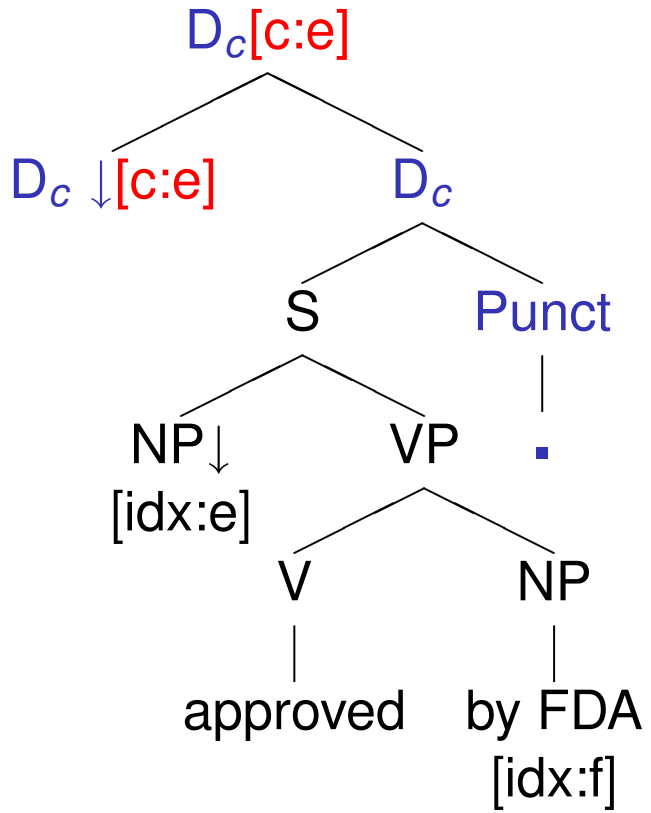


2. Annotate discourse-level root nodes and substitution nodes with discourse entities = Discourse Polarities

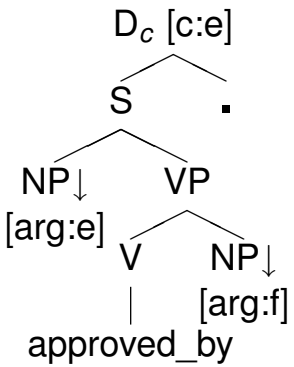
+e

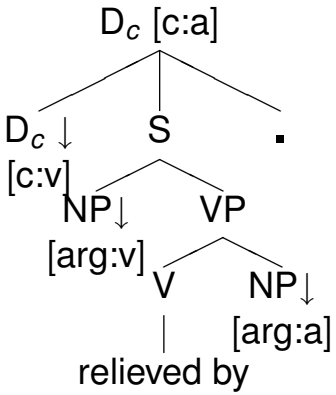


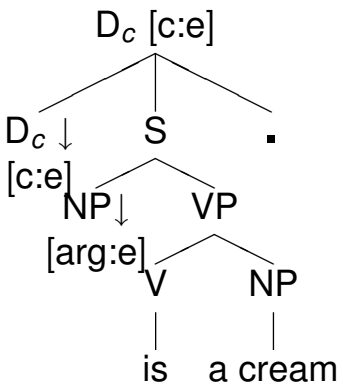
+e -e

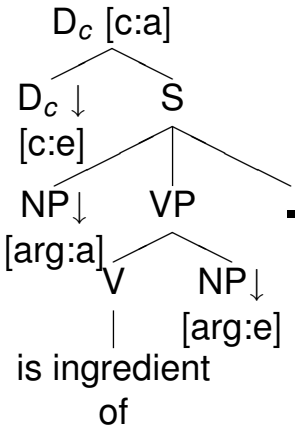


```
h3:approve(f,e)  h6:relieve(a,v)   h0:cream(e)     h4:contain(e,a)
```



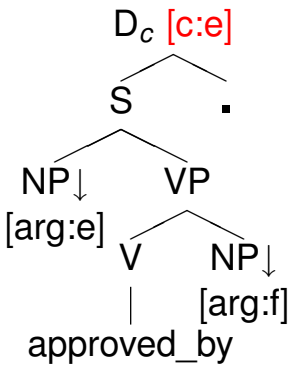


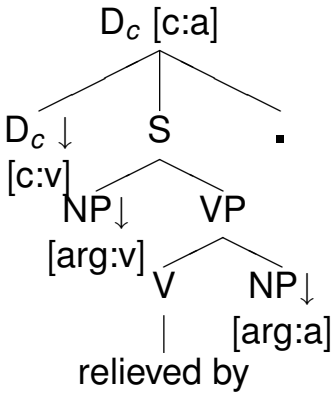


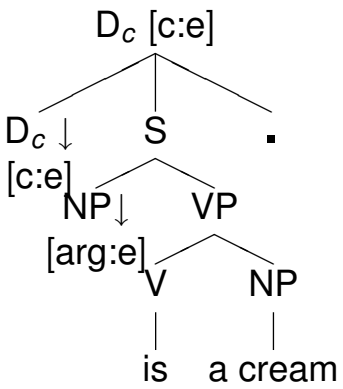


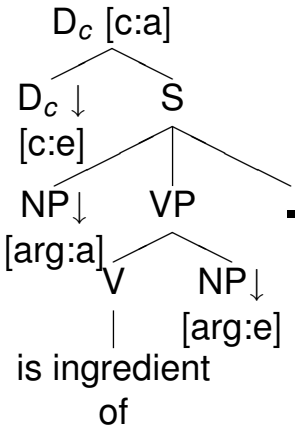

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+e



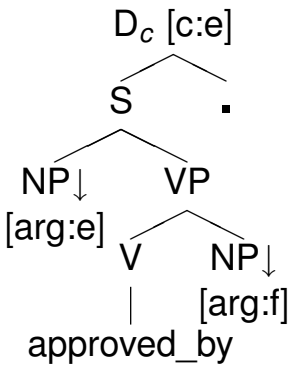




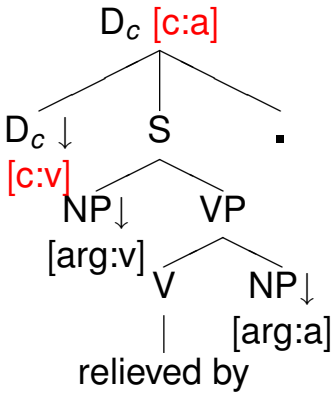


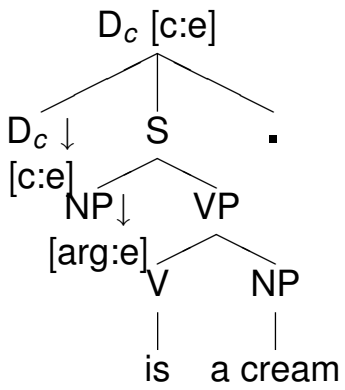

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

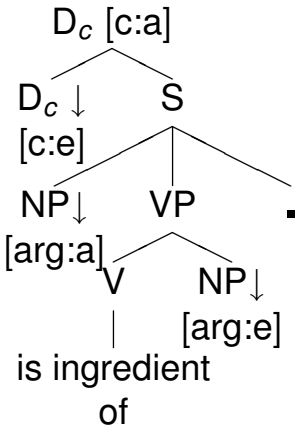
+e



+a −v

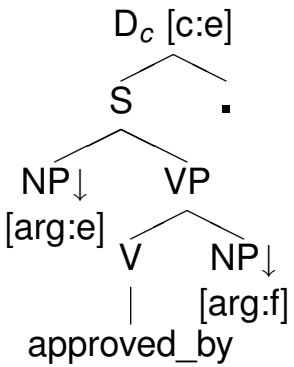




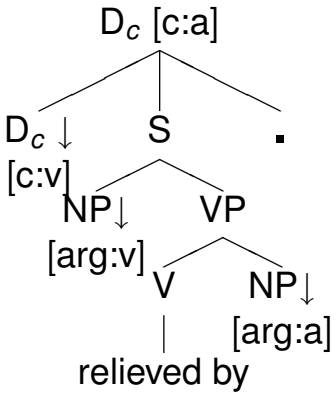



```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

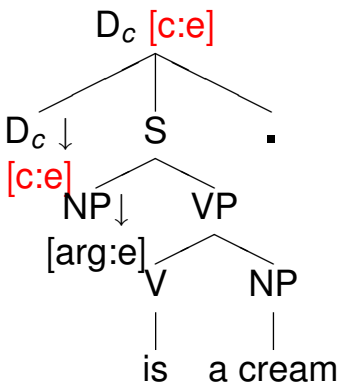
+e

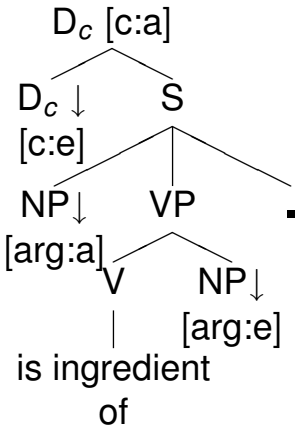


+a −v



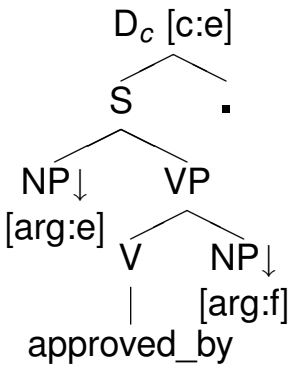
+e −e



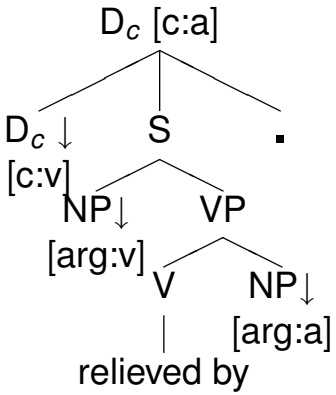



```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

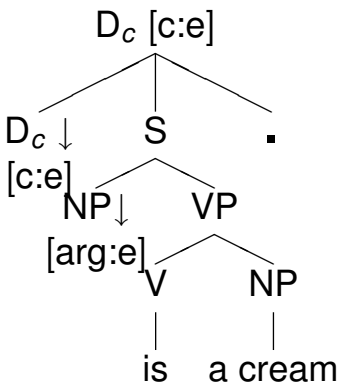
+e



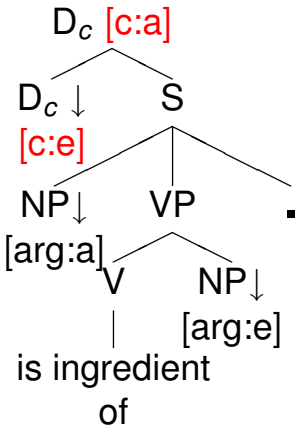
+a −v



+e −e

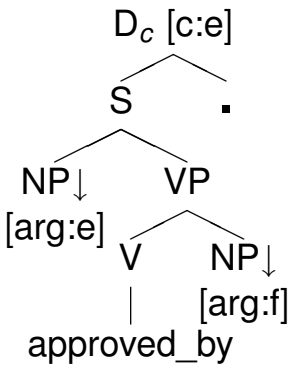


+a −e

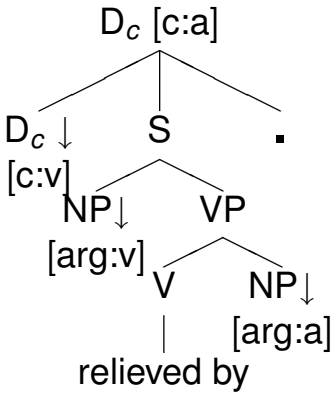



```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

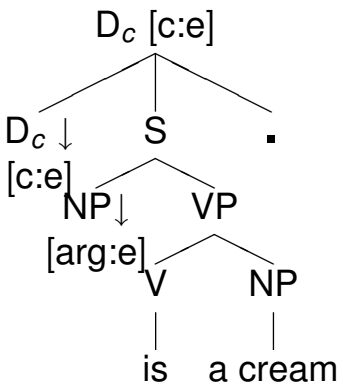
+e



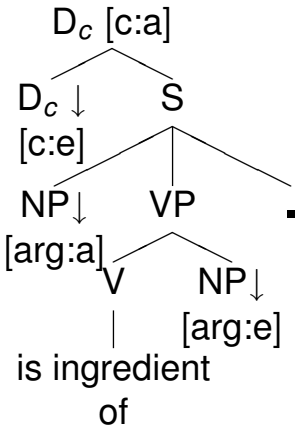
+a −v



+e −e



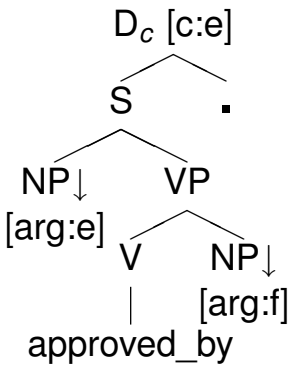
+a −e



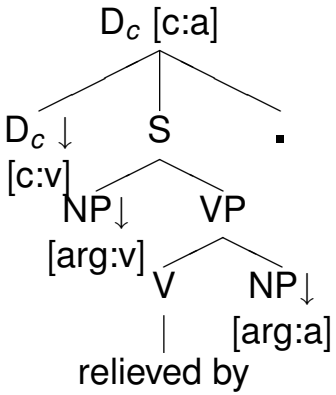

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+2a -v

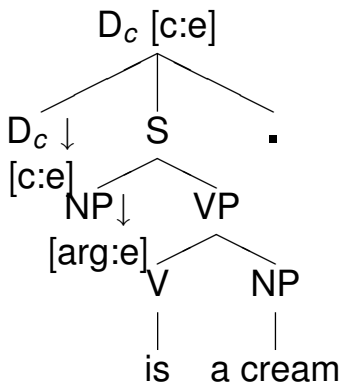
+e



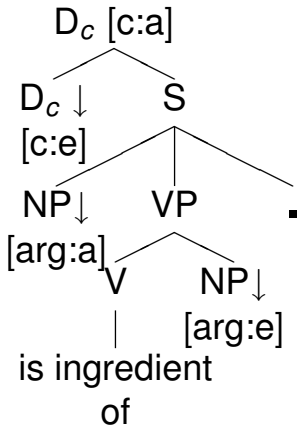
+a −v



+e −e



+a −e

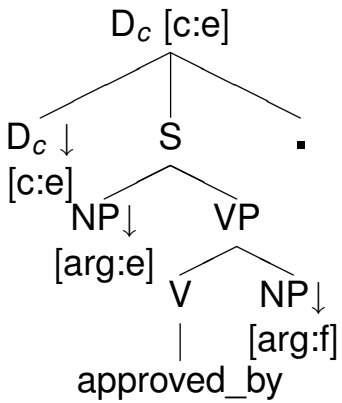


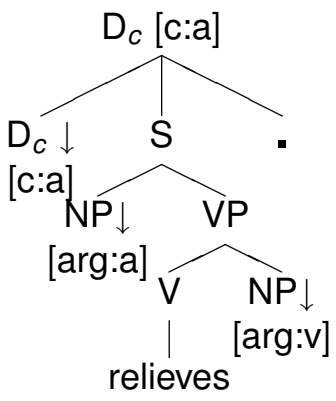

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

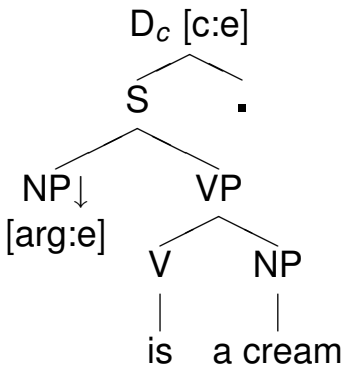
+2a -v

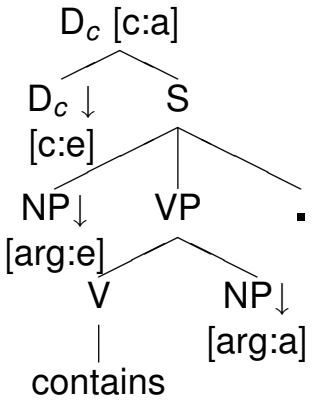
Elixir is approved by the FDA. Viral skin disorders are relieved by Aliprosan. Elixir is a white cream. Aliprosan is an ingredient of Elixir.

```
h3:approve(f,e)  h6:relieve(a,v)   h0:cream(e)     h4:contain(e,a)
```



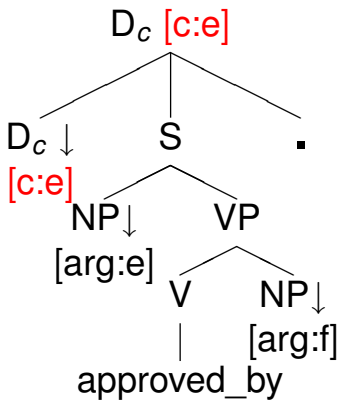


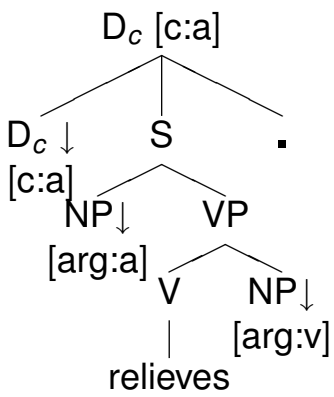


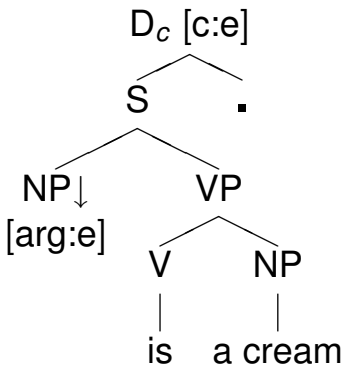


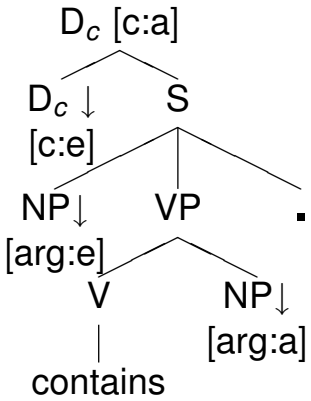

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+e −e



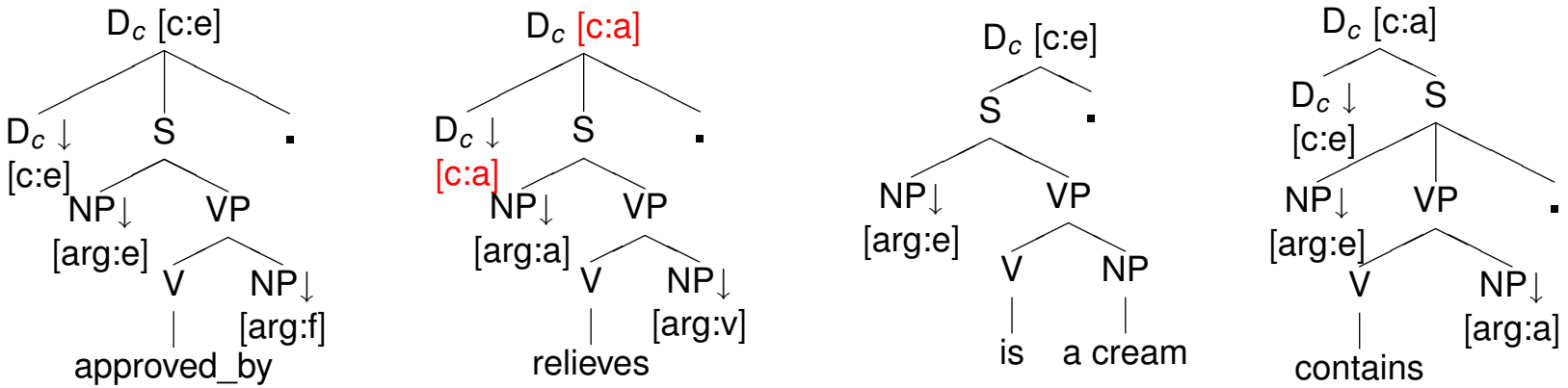






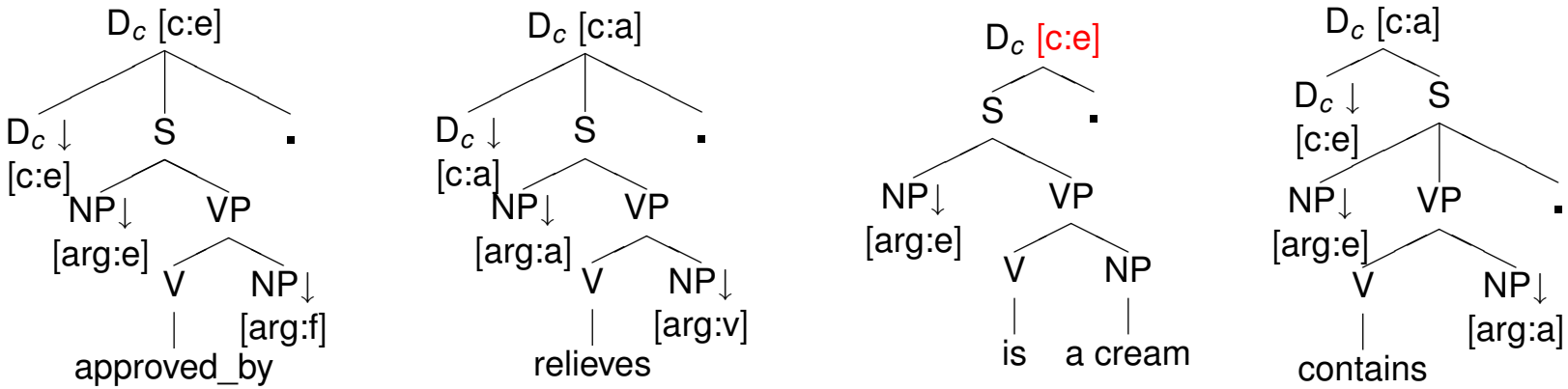

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+e −e +a −a



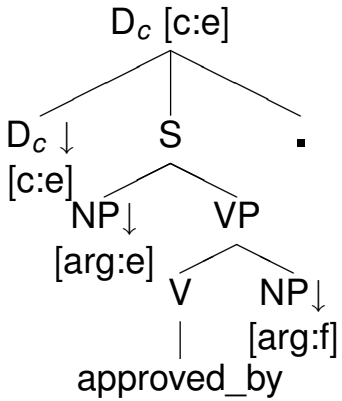

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+e −e +a −a +e

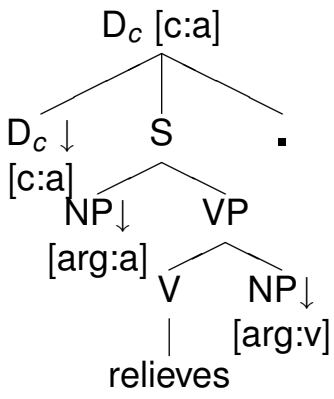



```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

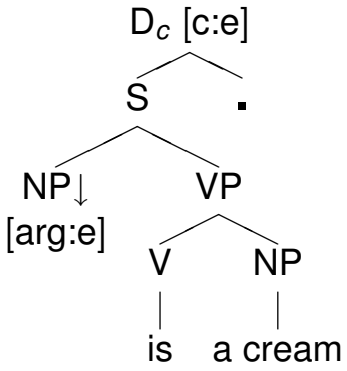
+e −e



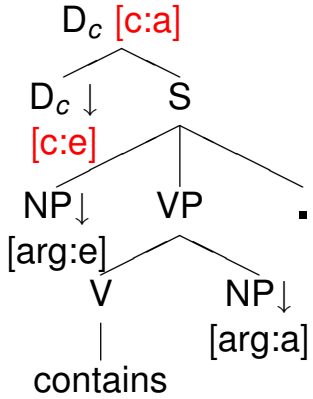
+a −a



+e

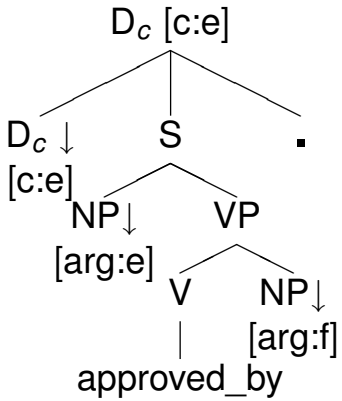


+a −e

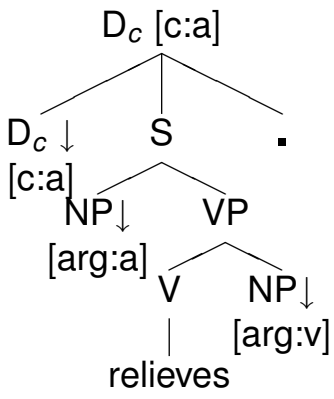



```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

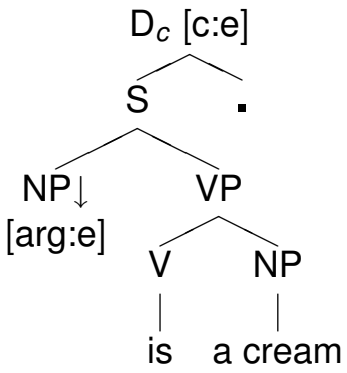
+e −e



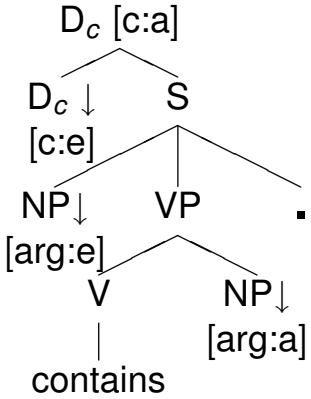
+a −a



+e



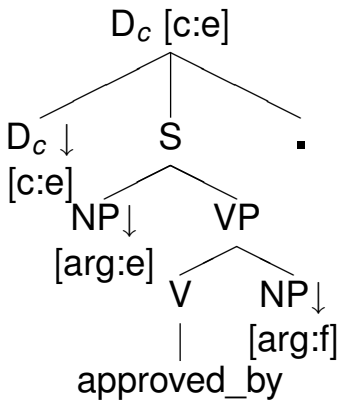
+a −e



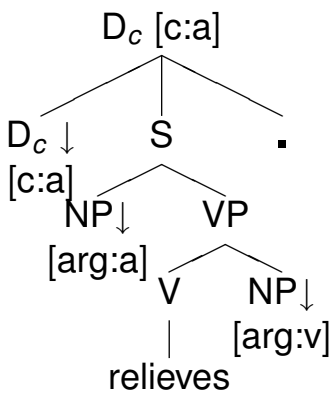

```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+a

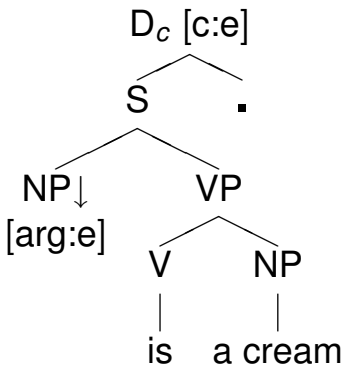
+e −e



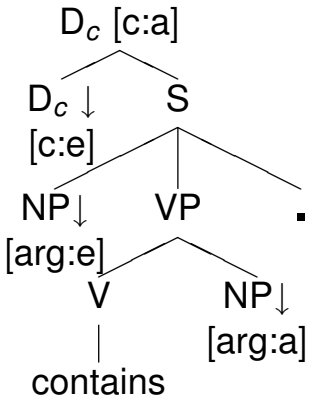
+a −a



+e



+a −e




```
h3:approve(f,e)  h6:relieve(a,v)  h0:cream(e)  h4:contain(e,a)
```

+a

Elixir is a white cream. Elixir is approved by the FDA. Elixir contains Aliprosan. Aliprosan relieves viral skin disorders.

Some Statistics

- 1536 possible ways to combine the trees, of these the grammar syntactically allows 192 solutions
- If we annotate trees with discourse entities we get 16 solutions
- All of the 16 solutions are *referentially* coherent
- Of the 1536 possible tree sets, 1320 were discarded by polarity filtering (i.e. the realizer attempted to combine only 216 tree sets)

Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

Pronominalization

- a sub-task of referring expression generation
- usually the job of a referring expression module, which decides whether a reference to a discourse entity should be
  - a name
  - a pronoun
  - a description

Pronominalization

- a sub-task of referring expression generation
- usually the job of a referring expression module, which decides whether a reference to a discourse entity should be
  - a name
  - a pronoun
  - a description

However, some pronouns are determined by syntax

A *referring expression* is any noun phrase, or surrogate for a noun phrase, if its communicative purpose is to identify an object to the hearer.

Contrast:

The drug is illegal.

Elixir is *an illegal drug*.

A *referring expression* is any noun phrase, or surrogate for a noun phrase, if its communicative purpose is to identify an object to the hearer.

Contrast:

The drug is illegal.

Elixir is *an illegal drug*.

Argument NPs

A *referring expression* is any noun phrase, or surrogate for a noun phrase, if its communicative purpose is to identify an object to the hearer.

Contrast:

The drug is illegal.

Elixir is *an illegal drug*.

Argument NPs

Predicative NPs

TEXT = predicative skeleton + referring expressions

RX(e) is a white cream. RX(e) is approved by RX(f) . RX(e) contains RX(a) which relieves RX(v) .

TEXT = predicative skeleton + referring expressions

Elixir is a white cream. RX(e) is approved by RX(f) . RX(e) contains RX(a) which relieves RX(v) .

TEXT = predicative skeleton + referring expressions

Elixir is a white cream. It is approved by the FDA . RX(e) contains RX(a) which relieves RX(v) .

TEXT = predicative skeleton + referring expressions

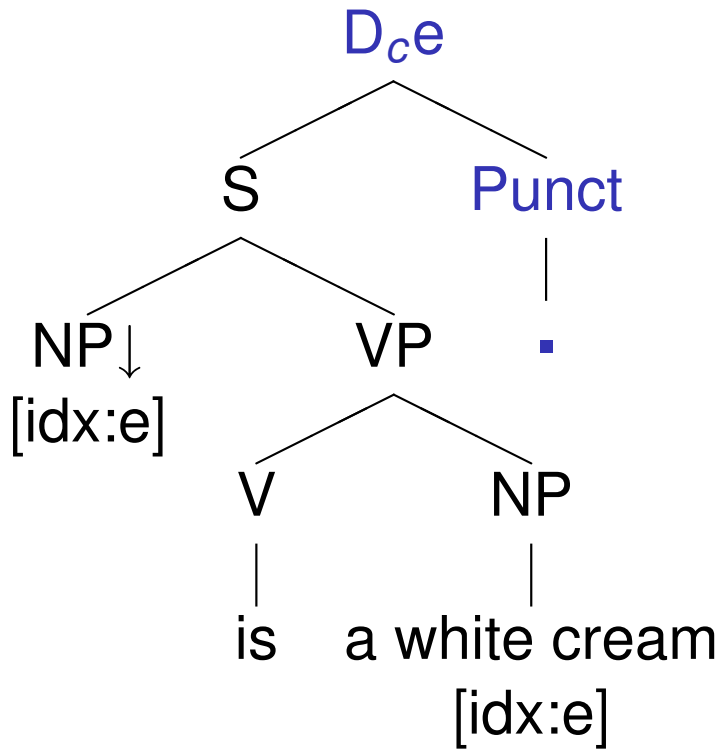
Elixir is a white cream. It is approved by the FDA . It contains Aliprosan which relieves viral skin disorders .

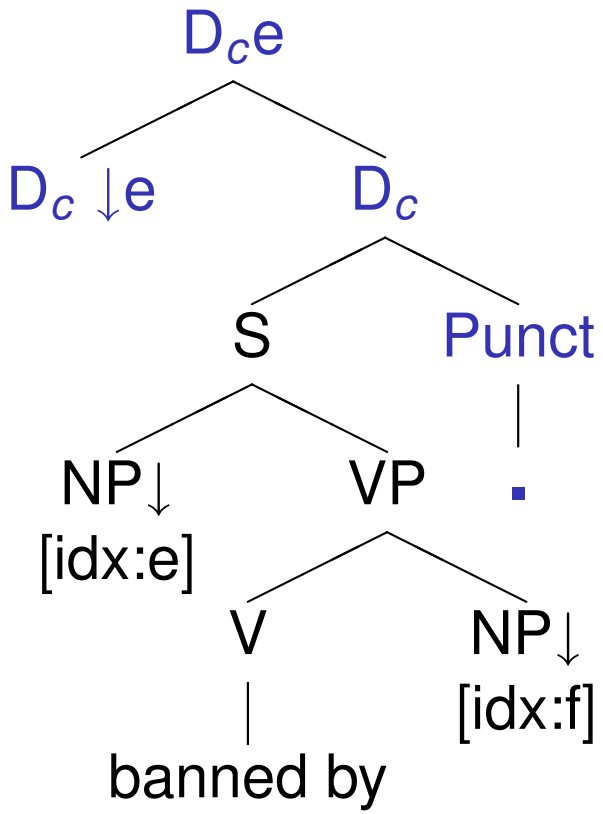
TEXT = predicative skeleton + referring expressions

Elixir is a white cream. It is approved by the FDA . It contains Aliprosan which relieves viral skin disorders .

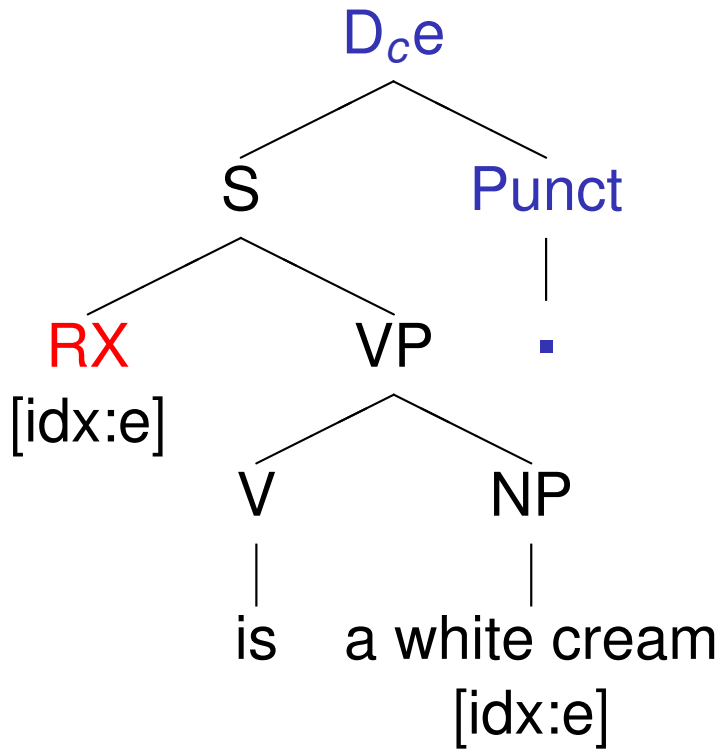
- Surface realizer determines the predicative skeleton
- It is not the surface realizer's job to produce referring expressions

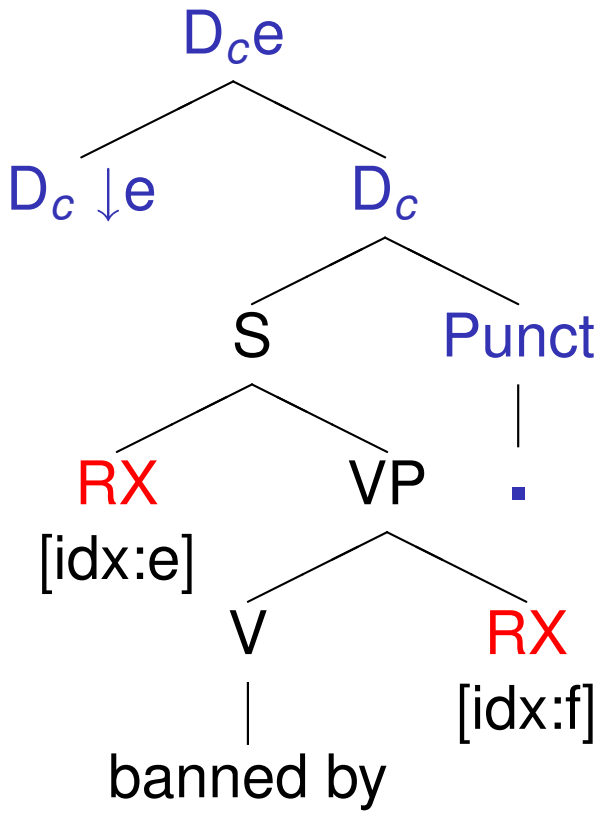
3 Separating Out Referring Expressions





3 Separating Out Referring Expressions





Syntactic constraints on pronominalization

- Pronoun not allowed:

Elixir, an illegal drug, is banned by the FDA .

Syntactic constraints on pronominalization

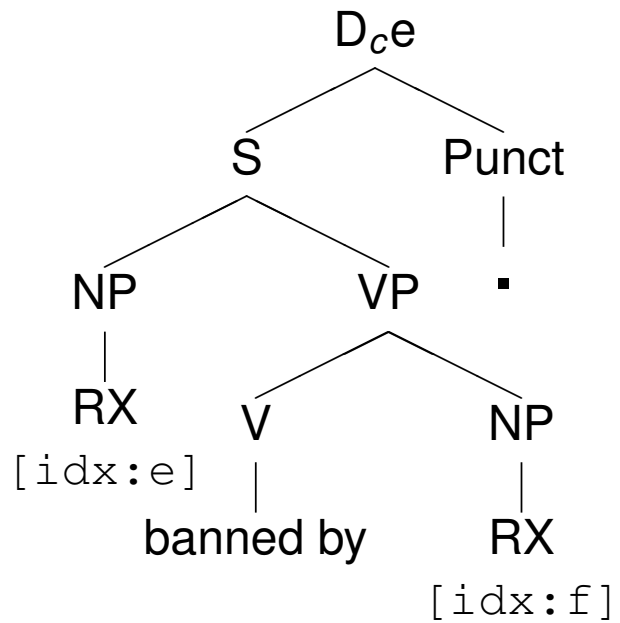
- Pronoun not allowed:

* It, an illegal drug, is banned by the FDA .

Syntactic constraints on pronominalization

- Pronoun not allowed:

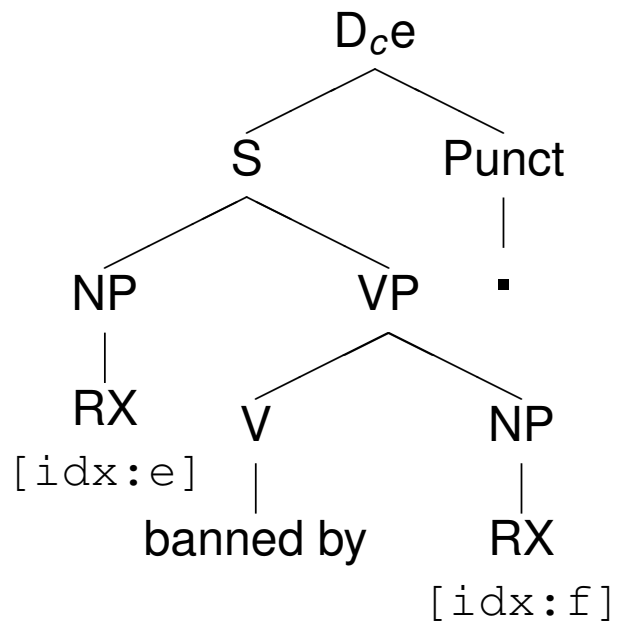
RX(e)	is banned by RX(f)	.

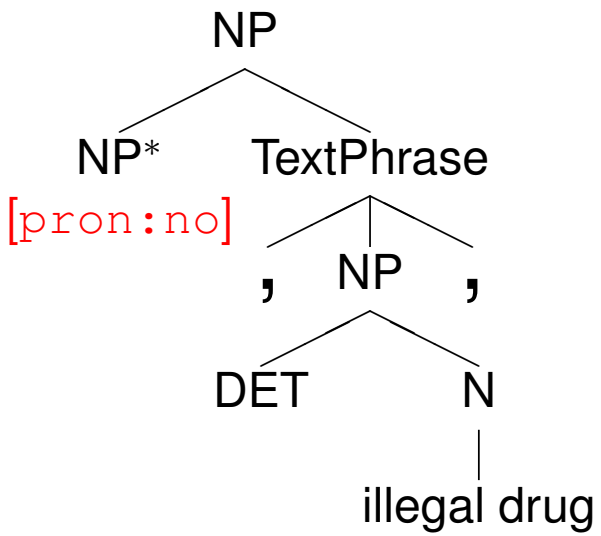


Syntactic constraints on pronominalization

- Pronoun not allowed:

RX(e)	is banned by RX(f) .

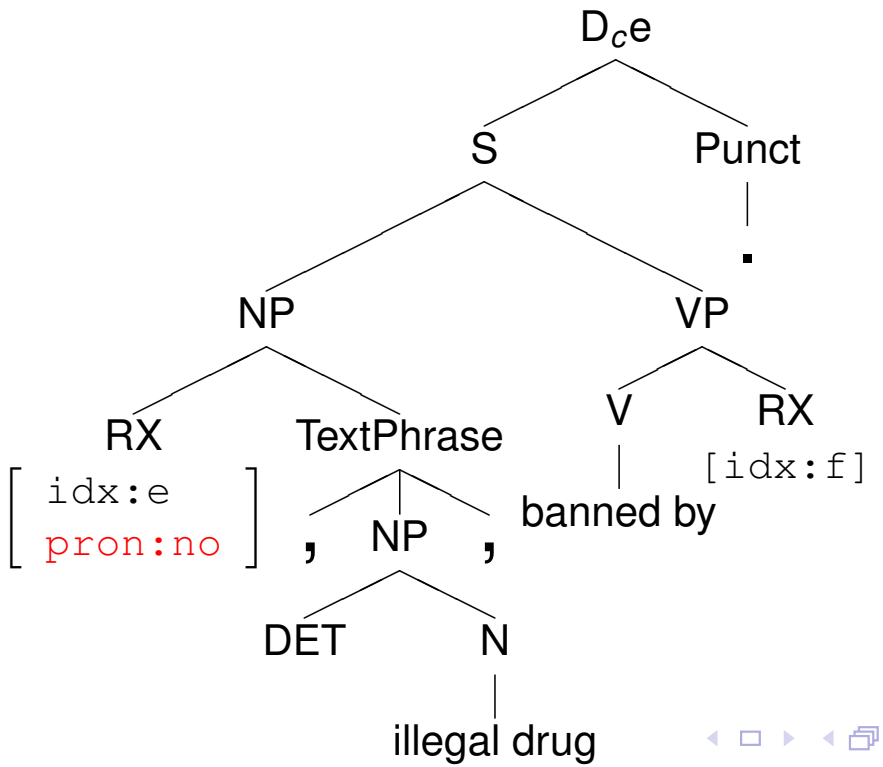




Syntactic constraints on pronominalization

- Pronoun not allowed:

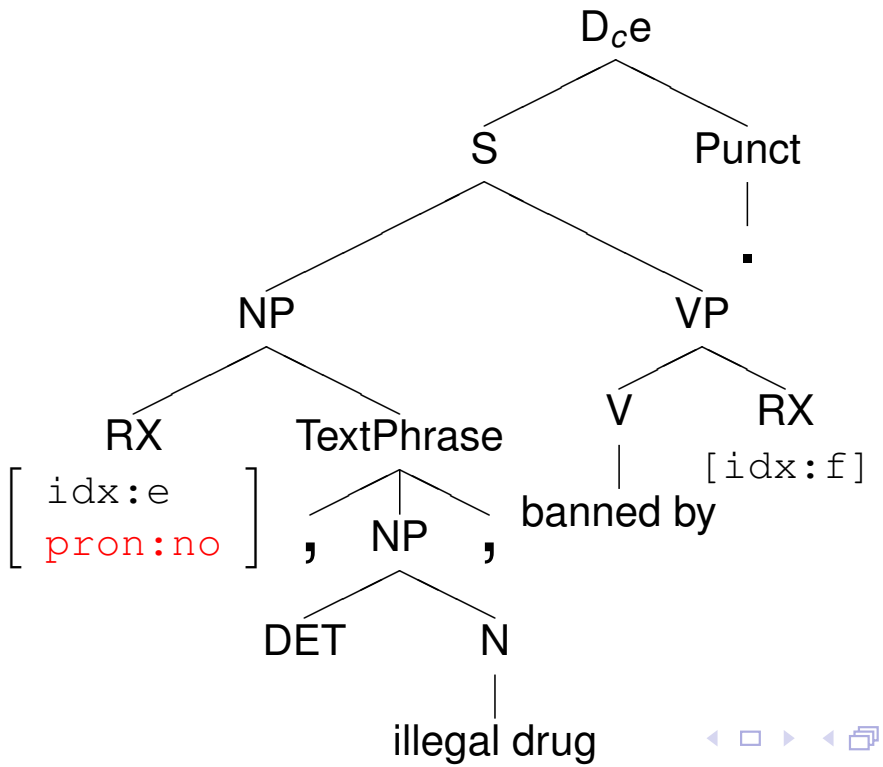
RX(e)[pron:no], an illegal drug, is banned by RX(f) .



Syntactic constraints on pronominalization

- Pronoun not allowed:

Elixir, an illegal drug, is banned by the FDA .



Syntactic constraints on pronominalization

- Pronoun *obligatory*:

Elixir, since	it	contains Gestodene, is banned by the FDA.

Syntactic constraints on pronominalization

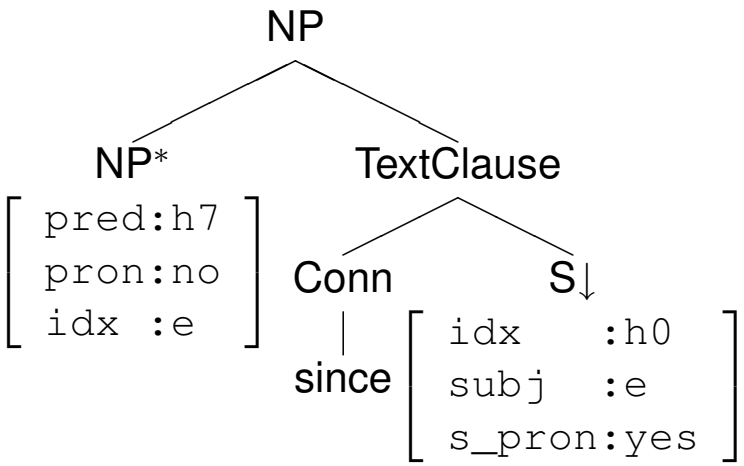
- Pronoun *obligatory*:

* Elixir, since Elixir contains Gestodene, is banned by the FDA.

Syntactic constraints on pronominalization

- Pronoun *obligatory*:

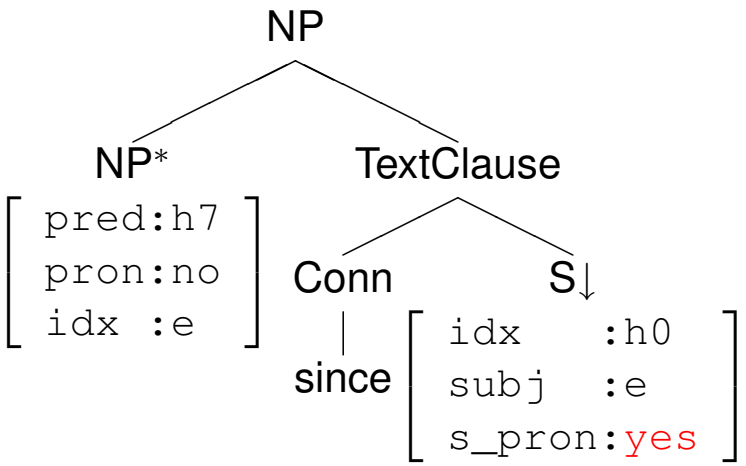
Elixir, since	RX(e)	contains Gestodene, is banned by the FDA.



Syntactic constraints on pronominalization

- Pronoun *obligatory*:

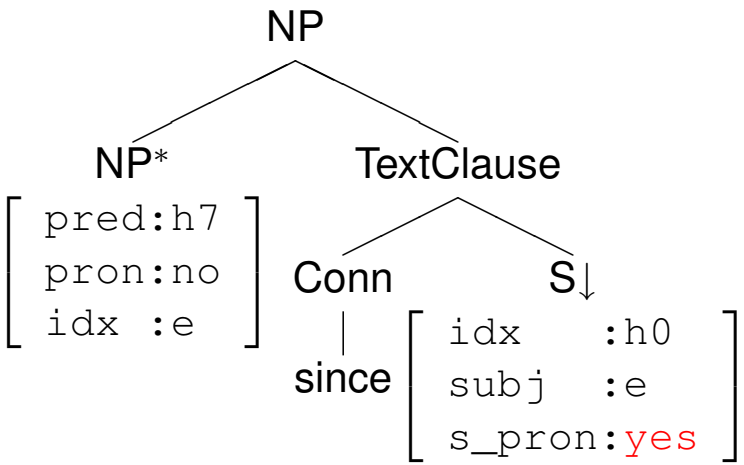
Elixir, since	RX(e)	contains Gestodene, is banned by the FDA.

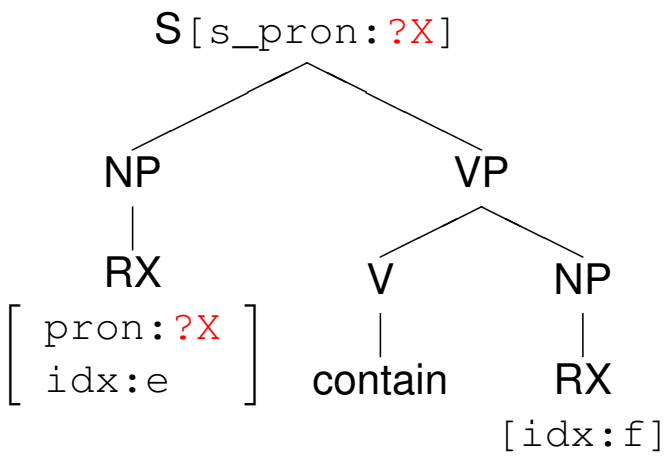


Syntactic constraints on pronominalization

- Pronoun *obligatory*:

Elixir, since	RX(e)	contains Gestodene, is banned by the FDA.

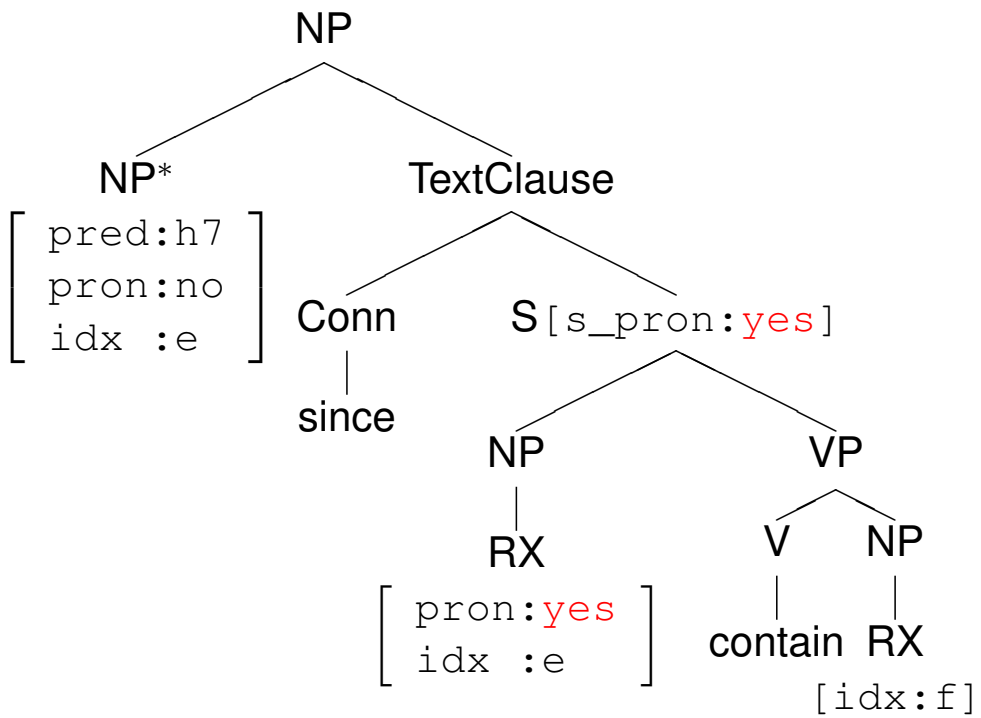




Syntactic constraints on pronominalization

- Pronoun *obligatory*:

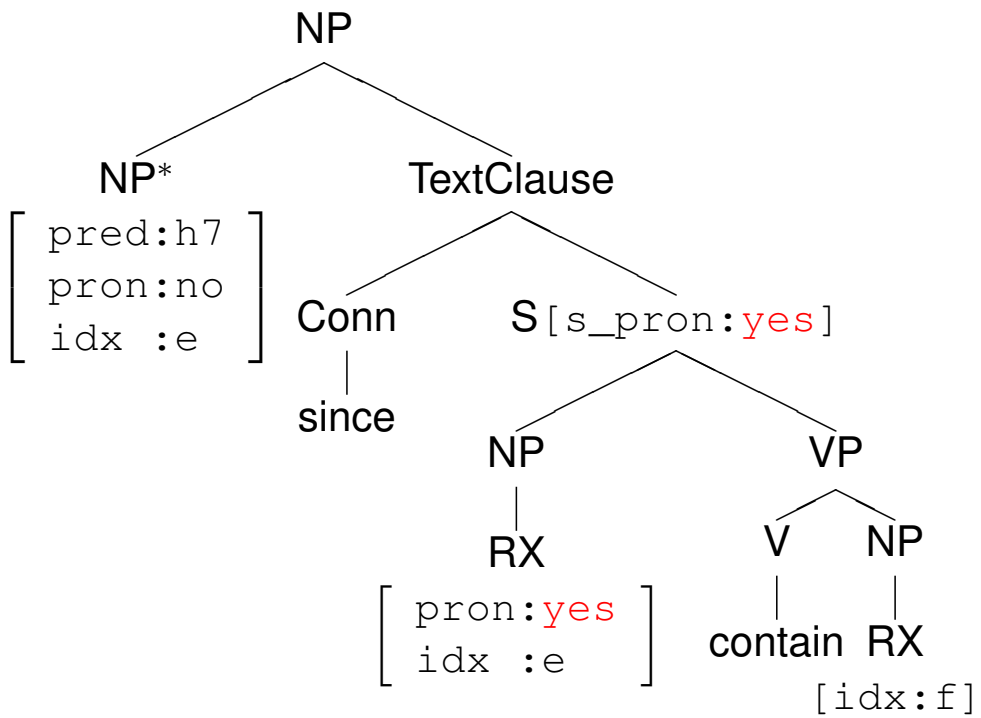
Elixir, since RX(e)[pron:yes] contains Gestodene, is banned by the FDA.



Syntactic constraints on pronominalization

- Pronoun *obligatory*:

Elixir, since	it	contains Gestodene, is banned by the FDA.



Outline

1. Setting the Scene
   - What Makes Discourse Coherent?
   - Generating Coherent Discourse
   - GenI and TAG
2. An Integrated Syntax-Discourse Grammar
   - Modeling Referential Coherence
   - Pronominalization
   - Emphasis

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+a +e +f or **+v**

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+a +e +f or **+v**

---

94 solutions

- Viral skin disorders are relieved by Aliprosan, which is an ingredient of Elixir (a white cream, which the FDA approves).
- Elixir (a white cream) contains Aliprosan, which relieves viral skin disorders. It is approved by the FDA.
- Aliprosan, which relieves viral skin disorders, is an ingredient of Elixir. Elixir is a white cream, which the FDA approves.
- The FDA approves Elixir. Elixir is a white cream. It contains Aliprosan, which relieves viral skin disorders.

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+a

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+a

---

30 solutions

- Aliprosan, which is an ingredient of Elixir (a white cream, which the FDA approves) relieves viral skin disorders.
- Elixir is a white cream, which the FDA approves. It contains Aliprosan, which relieves viral skin disorders.
- Elixir (a white cream, which the FDA approves) contains Aliprosan, which relieves viral skin disorders.
- The FDA approves Elixir. Elixir is a white cream. It contains Aliprosan, which relieves viral skin disorders.

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+a
firstMention:e

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+a
firstMention:e

---

21 solutions

- Elixir (a white cream) is approved by the FDA. It contains Aliprosan. Aliprosan relieves viral skin disorders.
- Elixir is a white cream. It is approved by the FDA. It contains Aliprosan, which relieves viral skin disorders.
- Elixir (a white cream, which the FDA approves) contains Aliprosan, which relieves viral skin disorders.

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)[Important]
h6:relieve(a v)
```

+a
firstMention:e

```
h0:white_cream(e)[notImportant]
h3:approve(f e)[notImportant]
h4:contain(e a)[Important]
h6:relieve(a v)[notImportant]
```

+a
firstMention:e

```
h0:white_cream(e)[notImportant]
h3:approve(f e)[notImportant]
h4:contain(e a)[Important]
h6:relieve(a v)[notImportant]
```

+a
firstMention:e

---

1 solution

- Elixir (a white cream, which the FDA approves) contains Aliprosan, which relieves viral skin disorders.

```
h0:white_cream(e)
h3:approve(f e)[Important]
h4:contain(e a)[Important]
h6:relieve(a v)
```

+a
firstMention:e

```
h0:white_cream(e)[notImportant]
h3:approve(f e)[Important]
h4:contain(e a)[Important]
h6:relieve(a v)[noImportant]
```

+a
firstMention:e

```
h0:white_cream(e)[notImportant]
h3:approve(f e)[Important]
h4:contain(e a)[Important]
h6:relieve(a v)[noImportant]
```

+a
firstMention:e

---

1 solution

- Elixir (a white cream) is approved by the FDA. It contains Aliprosan, which relieves viral skin disorders.

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+V
firstMention:v

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+V
firstMention:v

---

2 solutions

- Viral skin disorders are relieved by Aliprosan, which Elixir (a white cream, which the FDA approves) contains.
- Viral skin disorders are relieved by Aliprosan, which is an ingredient of Elixir (a white cream, which the FDA approves).

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+f
firstMention:f

```
h0:white_cream(e)
h3:approve(f e)
h4:contain(e a)
h6:relieve(a v)
```

+f
firstMention:f

---

1 solutions

- The FDA approves Elixir (a white cream, which contains Aliprosan, which relieves viral skin disorders).

Summary

1. Integrated syntax-discourse grammar to generate many paraphrases
   - discourse-level domain of locality
   - discourse-entities on discourse-level nodes
   - no NP substitution nodes
2. Tree properties can be used to select the solution that
   - fits the discourse context
   - has the intended emphasis
3. Discourse-level features can be used to remove tree sets that would lead to referrentially incoherent solutions

Summary

1. Integrated syntax-discourse grammar to generate many paraphrases
   - discourse-level domain of locality
   - discourse-entities on discourse-level nodes
   - no NP substitution nodes
2. Tree properties can be used to select the solution that
   - fits the discourse context
   - has the intended emphasis
3. Discourse-level features can be used to remove tree sets that would lead to referrentially incoherent solutions