CS395-T
Topics in Natural Language Processing

LECTURE 5

Sanda Harabagiu
Department of Computer Sciences
University of Texas at Austin
Tu, Th 11AM-12:20 PM
sanda@cs.utexas.edu
http://www.cs.utexas.edu/users/sanda/cs395T.html
http://www.engr.smu.edu/~sanda/cs395T.html

1

Parsing Viewed through a Constraint-based Formalism

- Representation mode: feature structures
- Operation: unification

Integration into a grammatical formalism.

2

Feature Structures

- Simple way to encode properties
  - Sets of feature-value pairs
- What are the features ?
  - Atomic symbols
  - Representation: attribute-value matrix AVM

$$\begin{bmatrix} \text{Feature}_1 & \text{Value}_1 \\ \vdots & \\ \text{Feature}_n & \text{Value}_n \end{bmatrix}$$

3

Concrete Example

- Representing the 3sg NP category

$$\begin{bmatrix} \text{CAT} & \text{NP} \\ \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix}$$

- Representing the 3pl NP category

$$\begin{bmatrix} \text{CAT} & \text{NP} \\ \text{NUMBER} & \text{PL} \\ \text{PERSON} & 3 \end{bmatrix}$$

4

Feature-Values

- Features are not limited to atomic symbols as their values

$$\begin{bmatrix} \text{CAT} & \text{NP} \\ \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix} \end{bmatrix}$$

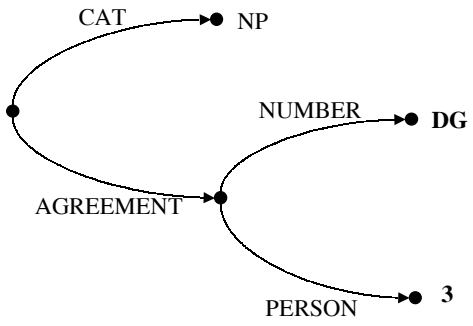
another feature becomes a value !

5

Feature PAth

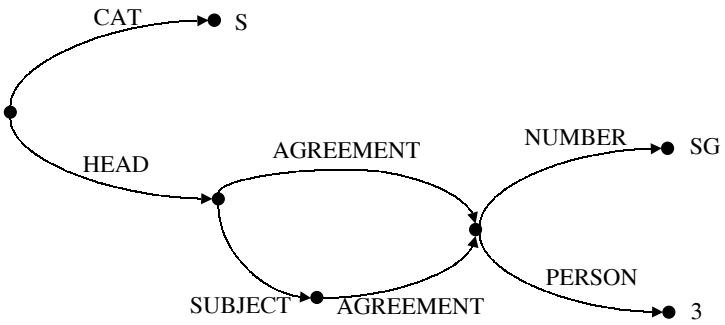
- A list of features through a feature structure leading to a particular value

  alternative graphical illustration of feature structures



6

Re-entrancy



A feature structure that shares **values**.

7

The Matrix Version

$$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{HEAD} & \begin{bmatrix} \text{AGREEMENT} & \boxed{1} & \begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix} \\ \text{SUBJECT} & & \begin{bmatrix} \text{AGREEMENT} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Why use them ?
They allow us to express linguistic generalization in surprisingly compact and elegant ways

8

Unification of Feature Structures

- Two principal operations can be performed on feature structures
  a) Merge information content
  b) Reject the merger of incompatible structures

  One computational technique: unification

Example:

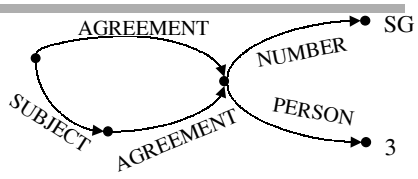
[NUMBER SG] ∪ [NUMBER SG] = [NUMBER SG]

$$\begin{cases} \text{[NUMBER SG]} \cup \text{[NUMBER PL]} \textit{ Fails !} \\ \text{[NUMBER SG]} \cup \text{[NUMBER []]} = \text{[NUMBER SG]} \\ \text{[NUMBER SG]} \cup \text{[PERSON 3]} = \begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix} \end{cases}$$

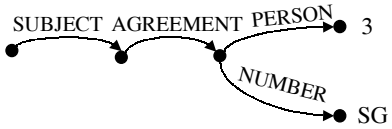
9

More Complicated Unifications

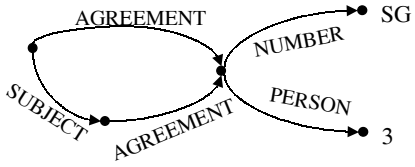
$$\begin{bmatrix} \text{AGREEMENT } \boxed{1} \begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix} \\ \text{SUBJECT} \quad \begin{bmatrix} \text{AGREEMENT } \boxed{1} \end{bmatrix} \end{bmatrix}$$



$$\cup \begin{bmatrix} \text{SUBJECT} \quad \begin{bmatrix} \text{AGREEMENT} \begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$



$$= \begin{bmatrix} \text{AGREEMENT } \boxed{1} \begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix} \\ \text{SUBJECT} \quad \begin{bmatrix} \text{AGREEMENT } \boxed{1} \end{bmatrix} \end{bmatrix}$$



10

Features That Share

$$\begin{bmatrix} \text{AGREEMENT} & [\text{NUMBER SG}] \\ \text{SUBJECT} & [\text{AGREEMENT} \quad [\text{NUMBER SG}]] \end{bmatrix}$$

$$\cup \begin{bmatrix} \text{SUBJECT} & [\text{AGREEMENT} \quad [\text{PERSON 3}, \text{NUMBER SG}]] \end{bmatrix}$$

$$= \begin{bmatrix} \text{AGREEMENT} & [\text{NUMBER SG}] \\ \text{SUBJECT} & [\text{AGREEMENT} \quad [\text{NUMBER SG}, \text{PERSON 3}]] \end{bmatrix}$$

11

A failing Example

$$\begin{bmatrix} \text{AGREEMENT} \ \boxed{1} & [\text{NUMBER SG}, \text{PERSON 3}] \\ \text{SUBJECT} & [\text{AGREEMENT} \ \boxed{1}] \end{bmatrix}$$

$$\cup \begin{bmatrix} \text{AGREEMENT} & [\text{NUMBER SG}, \text{PERSON 3}] \\ \text{SUBJECT} & [\text{AGREEMENT} \quad [\text{NUMBER PL}, \text{PERSON 3}]] \end{bmatrix}$$

Fails !

12

Feature Subsumption

- Represented by $\subseteq$
- Formal definition:

$$F \subseteq G \Leftrightarrow$$

  - For every feature x in F, $F(x) \subseteq G(x)$
  - For all paths p and q in F s.t.

$$F(p)=F(q) \Rightarrow G(p) = G(q)$$

13

Examples

(A) $\begin{bmatrix} \text{NUMBER} & \text{SG} \end{bmatrix}$ (B) $\begin{bmatrix} \text{PERSON} & 3 \end{bmatrix}$

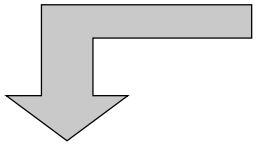
(C) $\begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix}$ (D) $\begin{bmatrix} \text{CAT} & \text{VP} \\ \text{AGREEMENT} & 1 \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & 1 \end{bmatrix} \end{bmatrix}$

(E) $\begin{bmatrix} \text{CAT} & \text{VP} \\ \text{AGREEMENT} & 1 \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

(F) $\begin{bmatrix} \text{CAT} & \text{VP} \\ \text{AGREEMENT} & 1 \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & 1\begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

14

examples



(A) $\begin{bmatrix} \text{NUMBER} & \text{SG} \end{bmatrix}$ (B) $\begin{bmatrix} \text{PERSON} & 3 \end{bmatrix}$

(C) $\begin{bmatrix} \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix}$ (D) $\begin{bmatrix} \text{CAT} & \text{VP} \\ \text{AGREEMENT} & 1 \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & 1 \end{bmatrix} \end{bmatrix}$

(E) $\begin{bmatrix} \text{CAT} & \text{VP} \\ \text{AGREEMENT} & 1 \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

(F) $\begin{bmatrix} \text{CAT} & \text{VP} \\ \text{AGREEMENT} & 1 \\ \text{SUBJECT} & \begin{bmatrix} \text{AGREEMENT} & 1 \begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

- $A \subseteq C$
- $B \subseteq C$
- $D \subseteq E$
- $D \subseteq F$

15

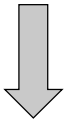
Feature structures in the grammar

- Used to express syntactic constraints that would be difficult by using context-free grammars alone.
- **Question:** How can feature structures and unification be integrated in the specification of a grammar?
  - By augmenting the rules of a regular grammar with attachments that specify feature structures for the constituents of the rules.
  - Is this all? No – use also appropriate unification operations.

16

Example

- S → NP VP
- Only if the number of the NP is equal to the number of the VP



- S → NP VP
- <NP NUMBER> = <VP NUMBER>

17

Head features

- Observation:
  - a general phenomenon of constraint-based grammars
  - → the features of most grammatical categories are copied from one of the children to the parent.

  What child? The head of the phrase.
- **Examples**:
  - VP → Verb NP
    - <VP AGREEMENT> = <Verb AGREEMENT>
  - NP → Det Nominal
    - <Det AGREEMENT> = <Nominal AGREEMENT>
    - <NP AGREEMENT> = <Nominal AGREEMENT>
  - Nominal → Noun
    - <Nominal AGREEMENT> = <Noun AGREEMENT>

18

Rewrite the rules with a HEAD feature

- VP → Verb NP
  - <VP HEAD> = <Verb HEAD >
- NP → Det Nominal
  - <NP HEAD > = <Nominal HEAD >
  - <Det HEAD AGREEMENT> = <Nominal HEAD AGREEMENT>
- Nominal → Noun
  - <Nominal HEAD > = <Noun HEAD >

19

Subcategorization

- Recap: verbs can be picky about the patterns of arguments they will allow themselves to appear with.
- SUBCAT feature
  - Verb → serves
    - <Verb HEAD AGREEMENT NUMBER> = SG
    - <Verb HEAD SUBCAT> = TRANS
  - VP → Verb
    - <VP HEAD> = <Verb HEAD >
    - <VP HEAD SUBCAT> = INTRANS
  - VP→ Verb NP
    - <VP HEAD> = <Verb HEAD >
    - <VP HEAD SUBCAT> = TRANS
  - VP→ Verb NP NP
    - <VP HEAD> = <Verb HEAD >
    - <VP HEAD SUBCAT> = DITRANS

20

Subcategorization frames

Noun Phrase Types		
There	nonreferential there	**There** *is still much to learn*
It	nonreferential it	**It** *was evident that my ideas*
NP	noun phrase	*As he was relating* **his story**
Preposition Phrase Types		
PP	preposition phrase	*couch their message* **in terms**
Pping	gerundive PP	*censured him* **for not having intervened**
PPpart	particle	*turn it* **off**
Verb Phrase Types		
VPbrst	bare stem VP	*she could* **discuss it**
VPto	to-marked infin. VP	*Why do you want* **to know***?*
VPwh	wh-VP	**it is worth considering how to write**
VPing	gerundive VP	*I would consider* **using it**

21

Subcategorization frames

Complement Clause Types		
Finite Clause		
Sfin	finite clause	*maintain* **that the situation was unsatisfactory**
Swh	wh-clause	*it tells us* **where we are**
Sif	whether/if clause	*ask* **whether Aristophanes is depicting a**
Nonfinite clause		
Sing	gerundive clause	*see* **some attention being given**
Sto	to-marked clause	*know* **themselves to be relatively unhealthy**
Sforto	for-to clause	*She was waiting* **for him to make some reply**
Sbrst	bare stem clause	*commanded* **that his sermons be published**
Other Types		
AjP	adjective phrase	*thought it* **possible**
Quo	quotes	*asked* **"What was it like?"**

22

Example: subcategorization patterns for the verb ask

Subcat	Example
Quo	asked [$_{Quo}$"What was it like?"]
NP	asking [$_{NP}$ a question]
Swh	asked [$_{Swh}$ what trades you're interested in]
Sto	ask [$_{Sto}$ him to tell you]
PP	that means asking [$_{PP}$ at home]
Vto	asked [$_{Vto}$ to see a girl called Evelyn]
NP Sif	asked [$_{NP}$ him] [$_{Sif}$ whether he could make]
NP NP	asked [$_{NP}$ myself] [$_{NP}$ a question]
NP Swh	asked [$_{NP}$ him] [$_{Swh}$ why he took time off]

23

Long-Distance Dependencies

- **Examples**:
  - What cities does Continental serve?
  - What flights do you have from Boston to Baltimore?
  - What time does that flight leave Atlanta?

24

Implementing Unification

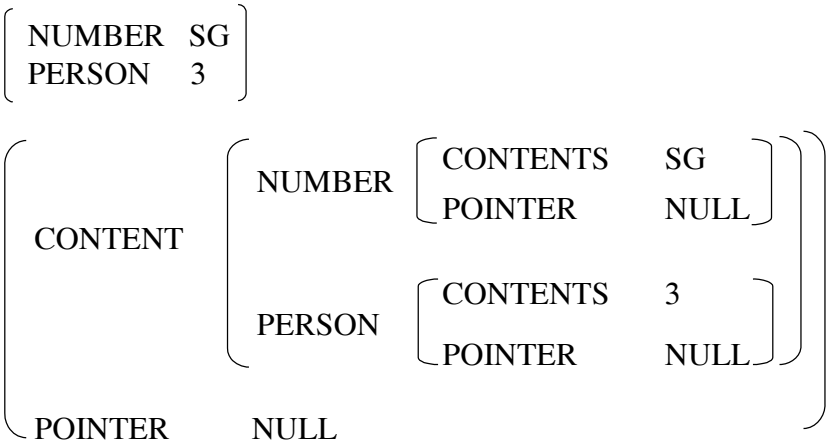
- Features can be represented as directed acyclic graphs (DAGs)
  - features = directed edges
  - values = atoms | DAGs

⇒The unification is a graph matching algorithm.

- Unification Data Structures
  - extended DAG – content field
    - pointer field

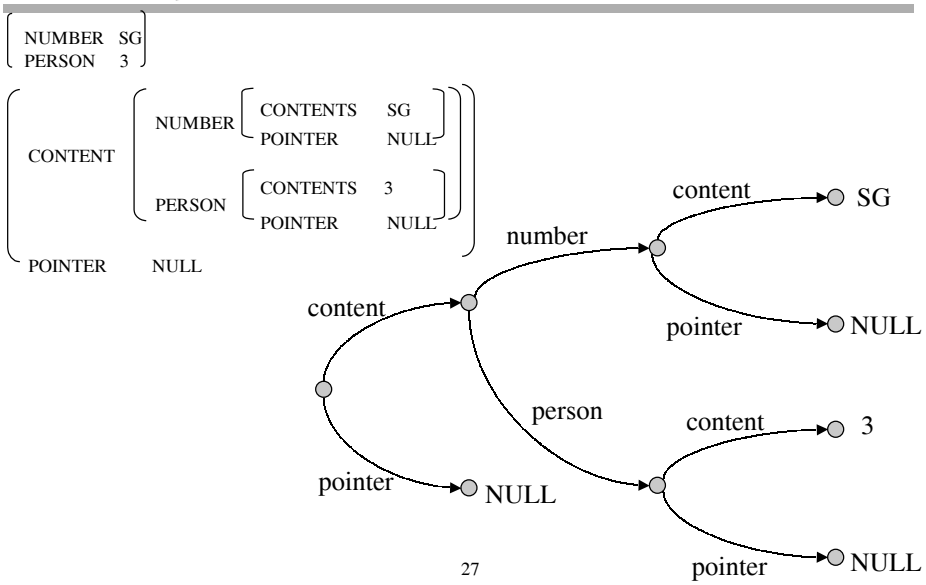
25

Example



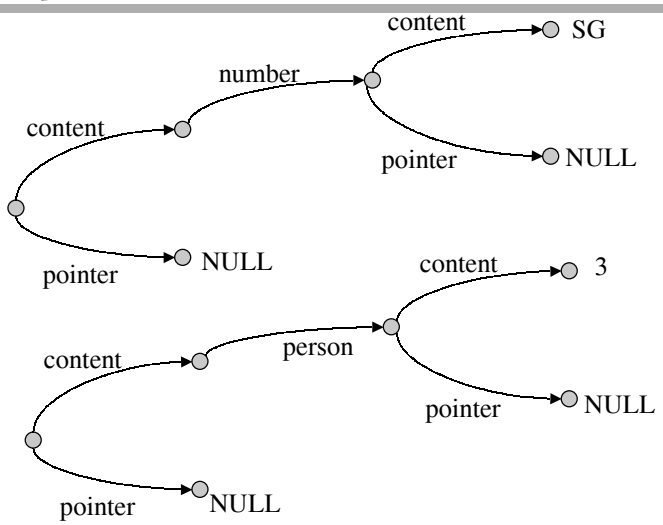
26

The graph



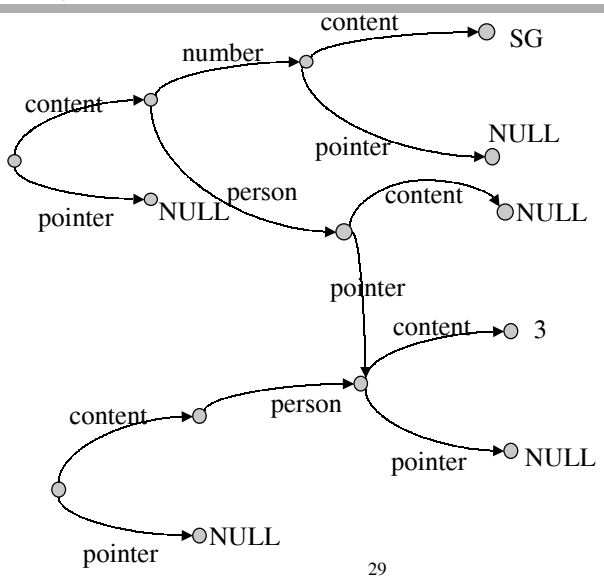
27

Example



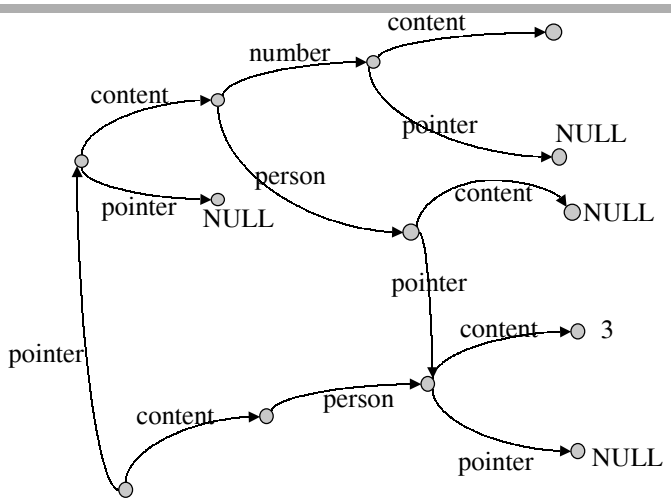
28

Using pointers



29

The result



30

Dereferencing

- If the POINTER field of an extracted feature structure is not NULL, the real content of that structure is found by following the pointer from the POINTER field

  → also called Real Contents

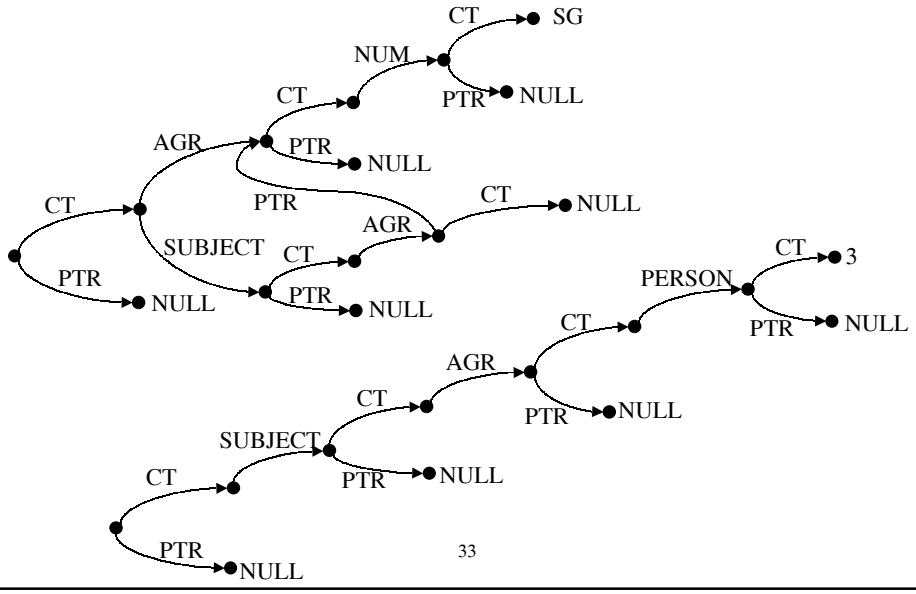
31

The Unification algorithm

```
function UNIFY(f1,f2) returns fstructure or failure
 f1-real ← Real contents of f1
 f2-real ← Real contents of f2
 if f1-real is null then
    f1.pointer ← f2
    return f2
 else if f2-real is null then
        f2.pointer ← f1
        return f1
      else if f1-real and f2-real are identical then
             f1.pointer ← f2
             return f2
           else if both f1-real and f2-real are complex feature structures then
                  f2.pointer ← f1
                  for each feature in f2-real do
                      other-feature ← Find or create a feature corresponding to feature f1-real
                      if UNIFY(feature.value, other-feature.value) returns failure then
                         return failure
                  return f1
                else return failure
```

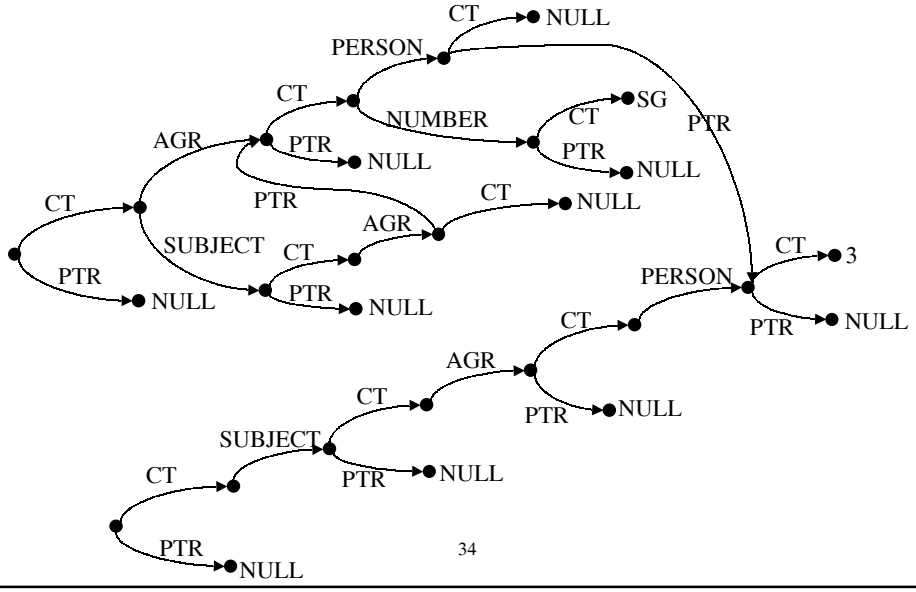
32

Walk Through an Example



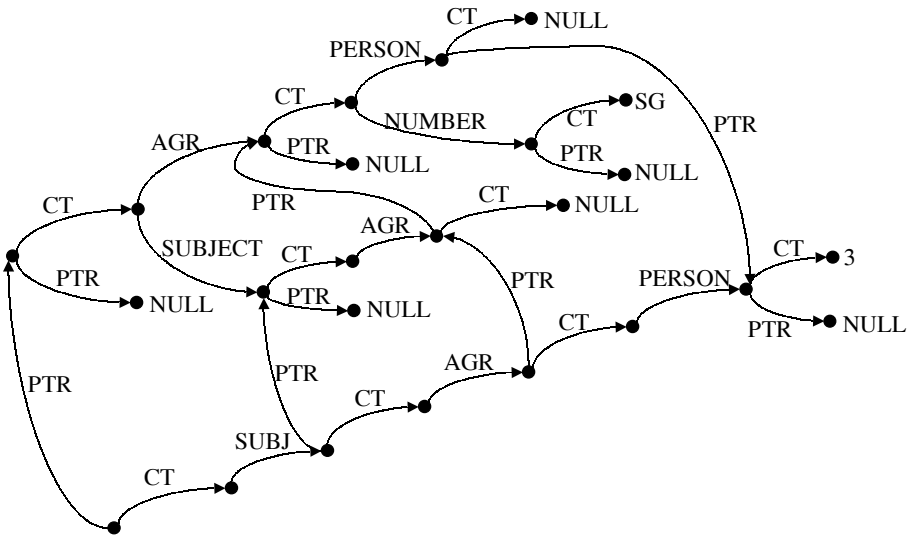
33

Walk Through an Example



34

Walk Through an Example



35

Parsing with Unification Constraints

1) Integrating unification into an Earley Parser.

   One of the topics of the Current Issues Discussion.

   1. Presenter
   2. Argues

   ⇒ Volunteers ?

   Textbook pages 427 – 437

36