Language Technologies Institute

Carnegie Mellon University

Advanced NLP

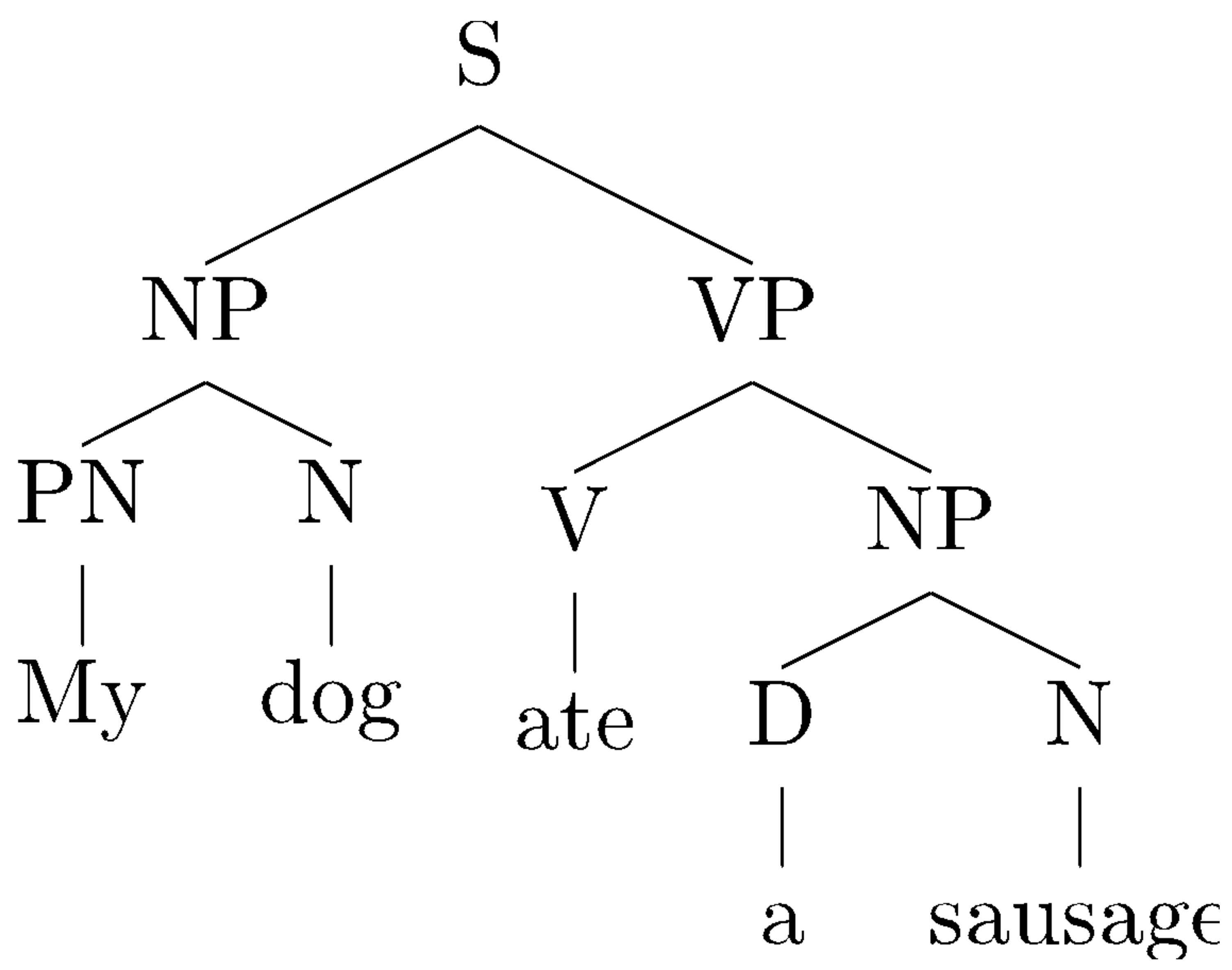
11-711 · October 2023

Syntax and parsing 2; Semantics 1

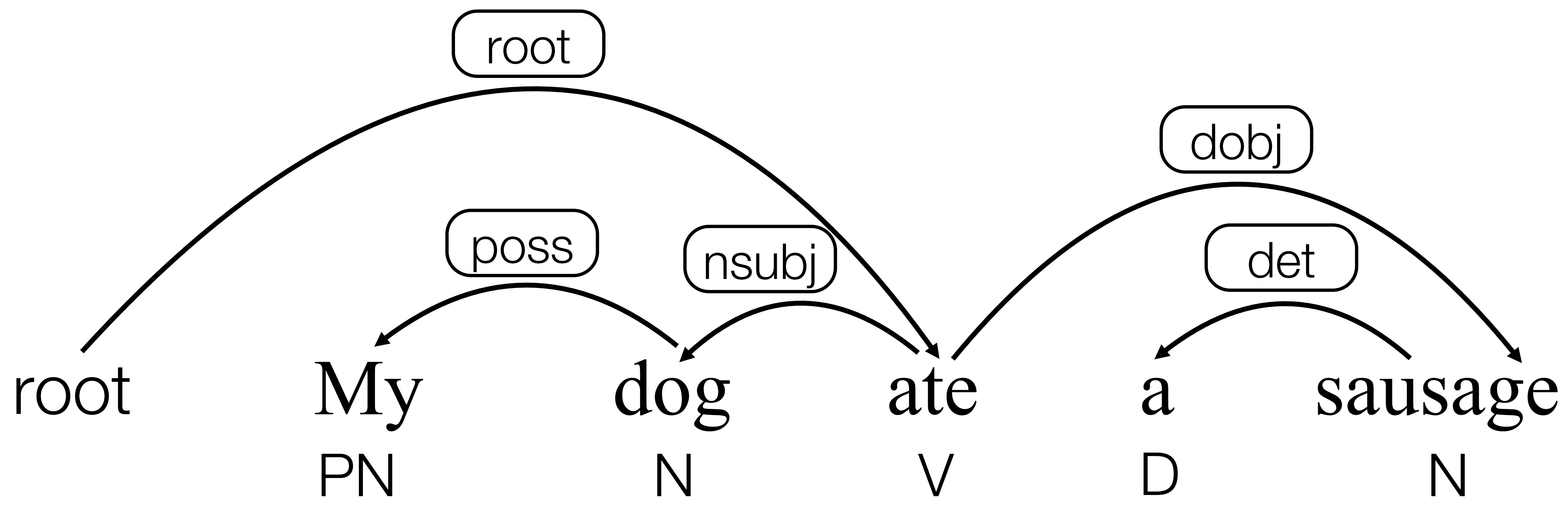
(Some slides adapted from Lori Levin, Noah Smith, and J&M)

Recap: Parsing

- The process of predicting **syntactic representations**
- Different types of syntactic representations are possible, for example:

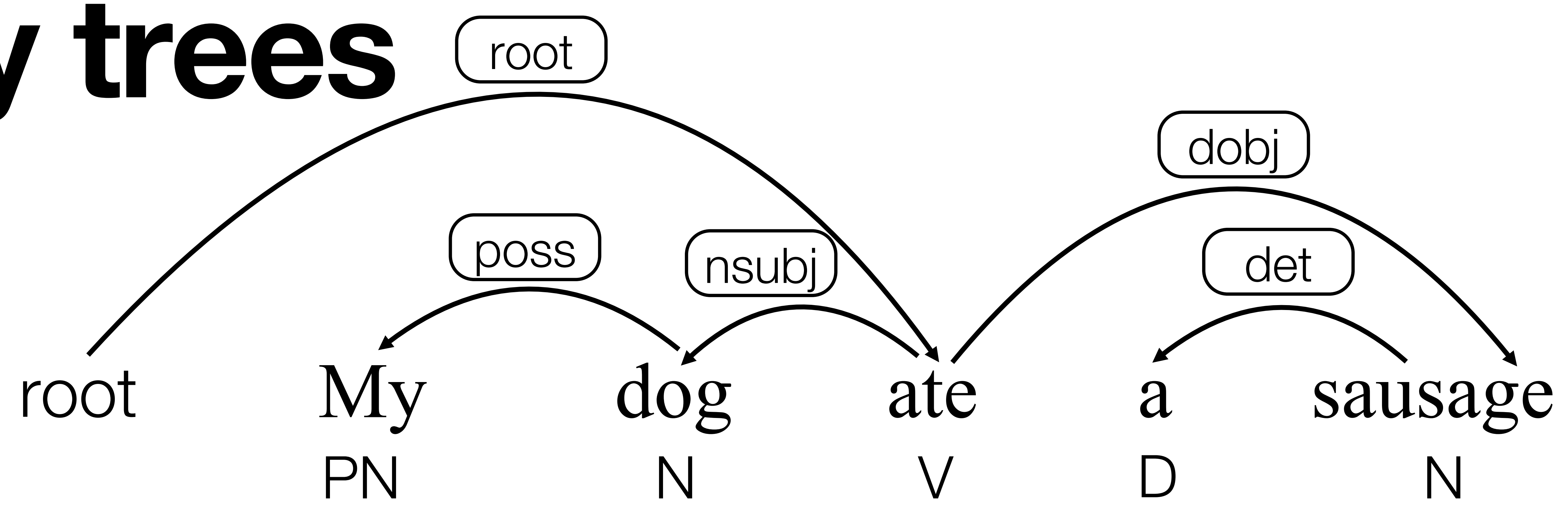


constituency (aka phrase-structure) tree



dependency tree

Recap: Dependency trees



- Nodes are words (along with part-of-speech tags)
- Directed arcs encode syntactic dependencies between words
- Labels are types of relations between words
  - **poss**: possessive
  - **dobj**: direct object
  - **nsubj**: (noun) subject
  - **det**: determiner

Dependency Parsers

Two main approaches:

- Transition-based dependency parsing
- Graph-based dependency parsing

Transition-based dependency parsing

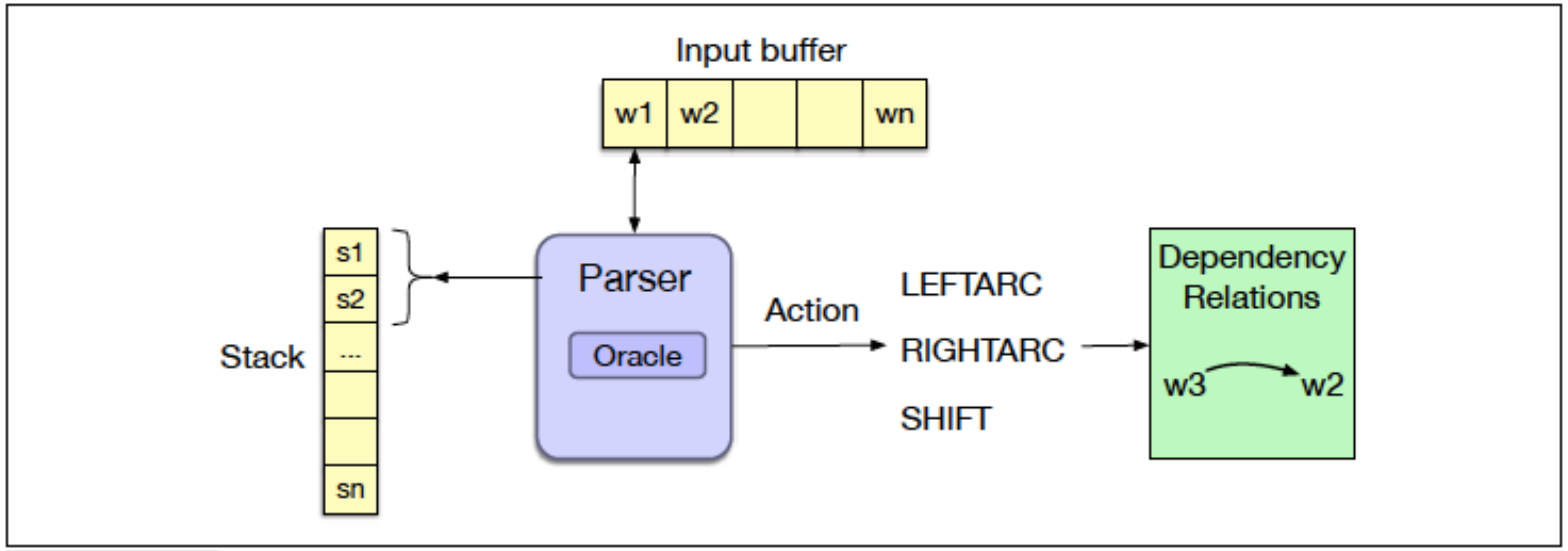
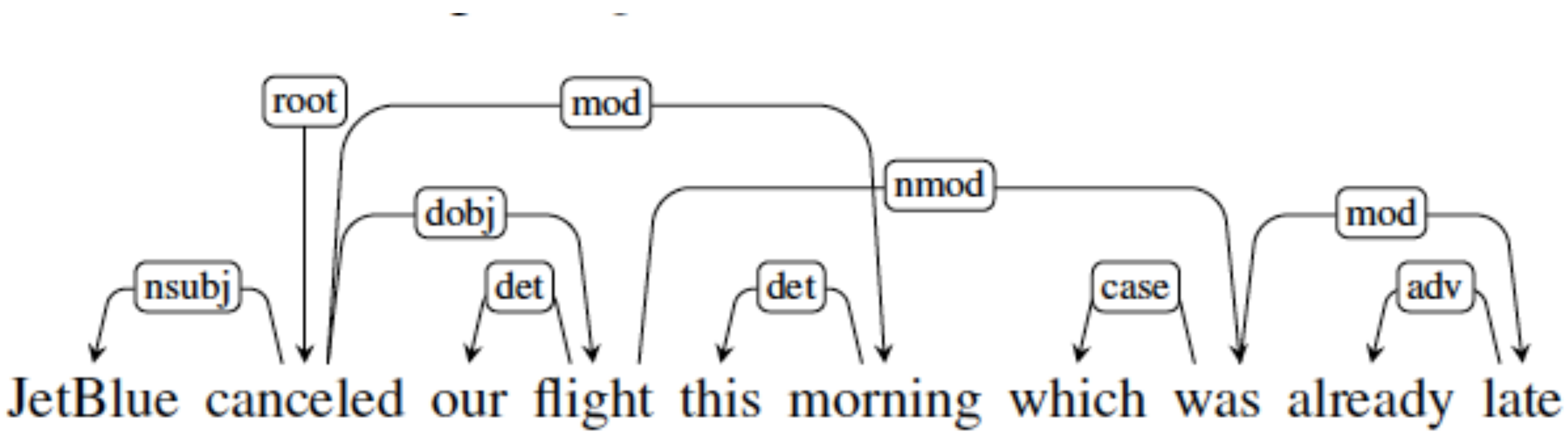


Figure 14.5 Basic transition-based parser. The parser examines the top two elements of the stack and selects an action by consulting an oracle that examines the current configuration.

- Oracle is learned from a treebank
- Complexity is linear in length of sentence
- Cannot produce non-projective parse trees

Non-projective parse tree



(14.3)

- Note crossing edges
- Not that common in English, but very common in free word order languages

Graph-based dependency parsing

- Score every edge in fully-connected graph
- Find maximum spanning tree starting at ROOT

- Scoring is learned from a treebank
- Not linear time, but can produce non-projective parse trees

Graph-based dependency parsing

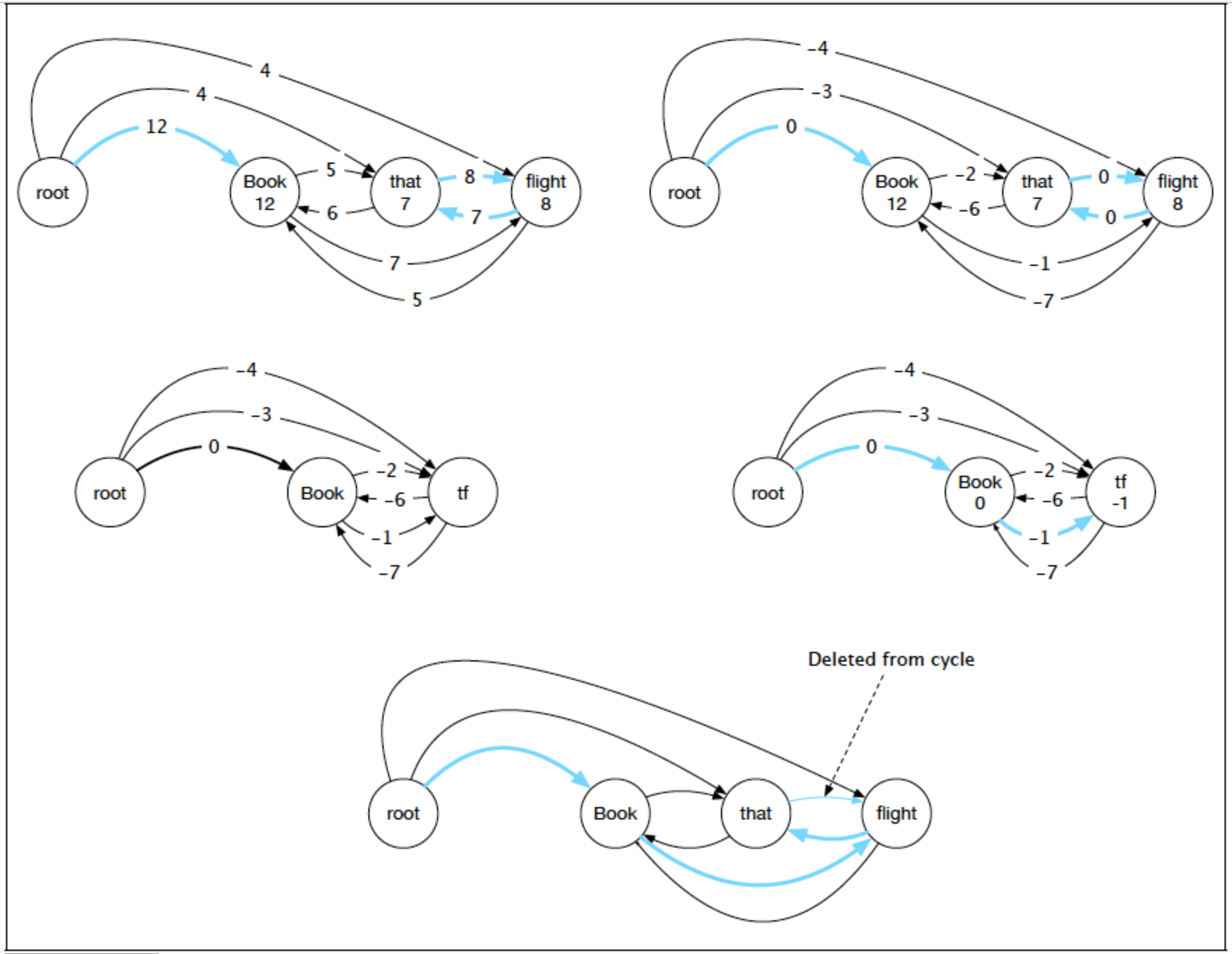


Figure 14.14 Chu-Liu-Edmonds graph-based example for *Book that flight*

Recap: Chomsky Hierarchy

- Type 3: Finite State Machines/Regular Expressions/Regular Grammars
  - $A \rightarrow Bw$ or $A \rightarrow w$
- Type 2: Push Down Automata/Context Free Grammars
  - $A \rightarrow \gamma$ where γ is any sequence of terminals/non-terminals
- Type 1: Linear-Bounded Automata/Context Sensitive Grammars
  - $\alpha A \beta \rightarrow \alpha \gamma \beta$ where γ is not empty
- Type 0: Turing Machines/Unrestricted Grammars
  - $aAb \rightarrow aab$ but $bAb \rightarrow bb$

Recap: Mildly Context-Sensitive Grammars

- We really like CFGs, but are they in fact expressive enough to capture all human grammar?
- Many approaches start with a “CF backbone”, and add registers, equations, or hacks, that are ***not*** CF.
- Several non-hack extensions (CCG, TAG, etc.) turn out to be weakly equivalent!
  - “Mildly context sensitive”
    - So CSFs get even less respect…
    - And so much for the Chomsky Hierarchy being such a big deal

Feature structures
and
Verb Subcategorization Frames

Review: Inflectional Morphology and syntactic agreement

- Morphology is the study of the internal structure of words.
  - **Derivational morphology.** How new words are created from existing words.
    - *[grace]*
    - *[[grace]ful]*
    - *[un[grace]ful]]*
  - **Inflectional morphology.** How **features** relevant to the **syntactic context** of a word are marked on that word.
    - This example illustrates number (singular and plural) and tense (present and past).
    - Green indicates irregular.   Blue indicates zero marking of inflection.   Red indicates regular inflection.
    - This student walks.
    - These students walk.
    - These students walked.
  - ***Compounding.*** Creating new words by combining existing words
    - With or without spaces: surfboard, golf ball, blackboard

Review: Features, morphology, FSTs:

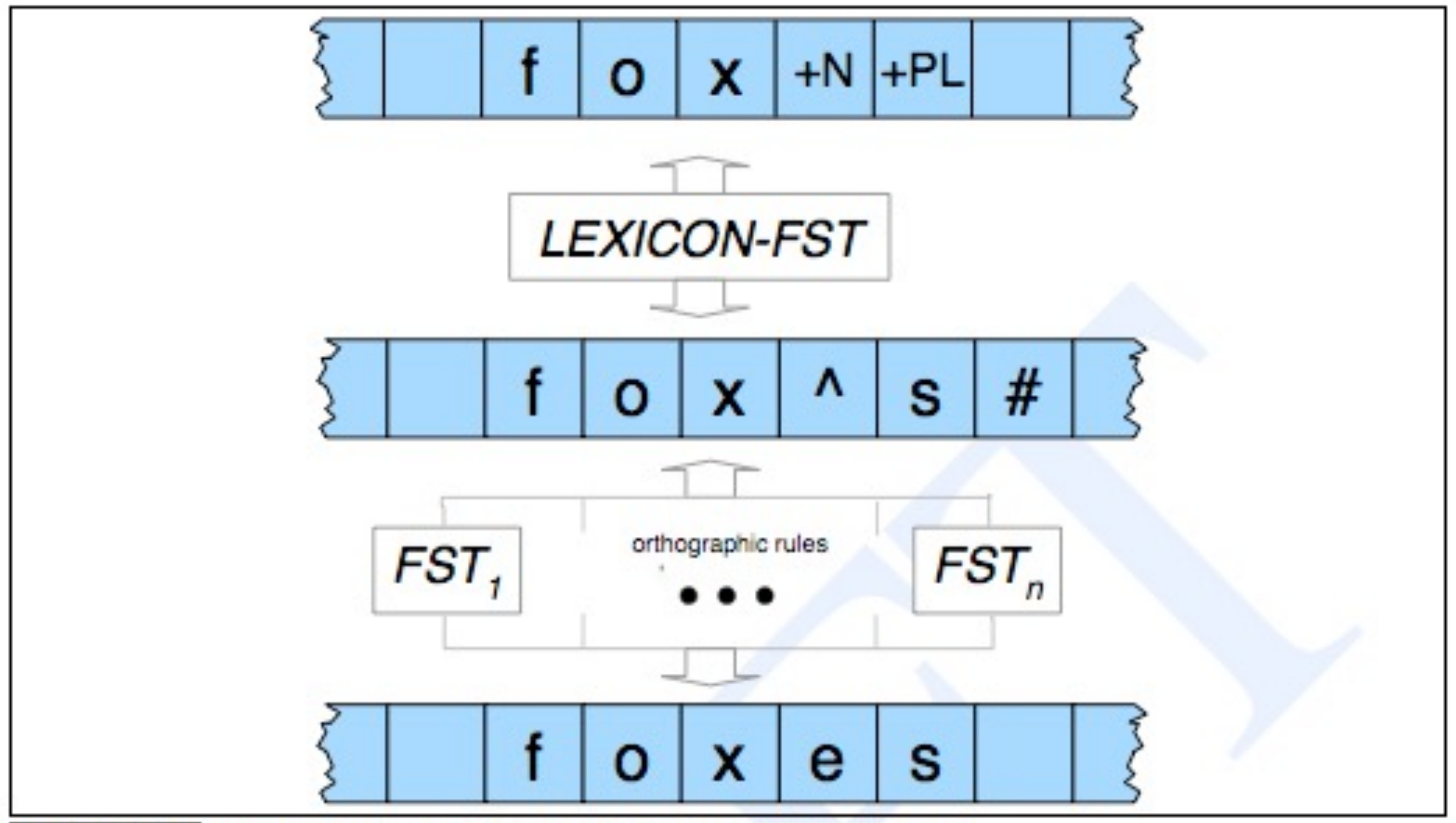
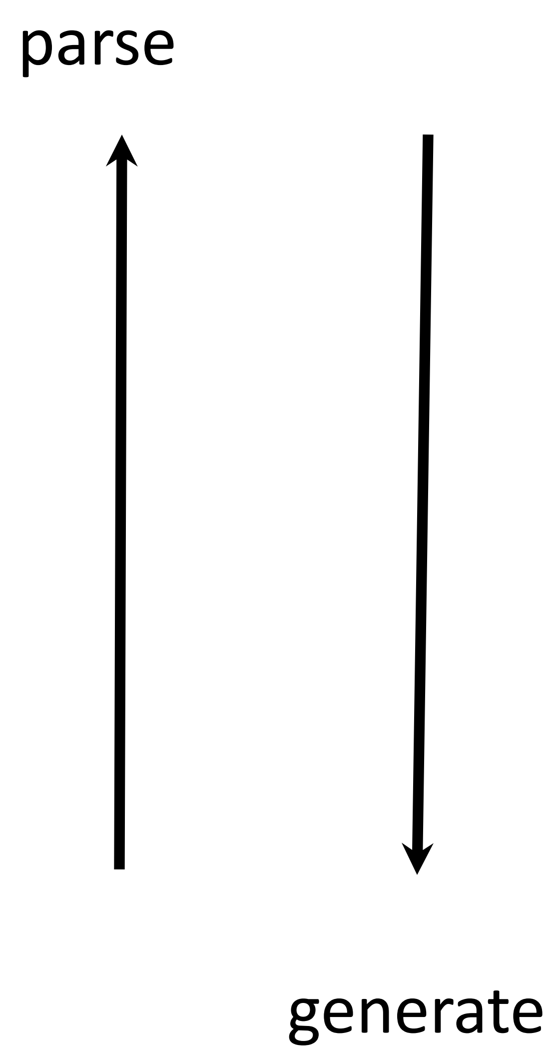


Figure 3.19 Generating or parsing with FST lexicon and rules



Linguistic features

- (Linguistic "features" vs. ML "features".)
- Human languages usually include *agreement* constraints; in English, e.g., subject/verb
  - I often swim
  - **He** often swim**s**
  - They often swim
- *Could* have a separate category for each minor type: N1s, N1p, …, N3s, N3p, …
  - *Each* with its own set of grammar rules!

A day without features…

- NP1s → Det-s N1s
- NP1p → Det-p N1p

  …
- NP3s → Det-s N3s
- NP3p → Det-p N3p

  …
- S1s → NP1s VP1s
- S1p → NP1p VP1p
- S3s → NP3s VP3s
- S3p → NP3p VP3p

Linguistic features

- *Could* have a separate category for each minor type: N1s, N1p, … , N3s, N3p, …
  – *Each* with its own set of grammar rules!
- Much better: represent these regularities using independent ***features****:* number, gender, person, …
- Features are typically introduced by lexicon; checked and propagated by ***constraint equations*** attached to grammar rules

Feature Structures (FSs)

Having multiple orthogonal features with values leads naturally to ***Feature Structures***:

[Det
[root: *a*]
[number: sg]]

A feature structure's values can in turn be FSs:

[NP
[agreement: [[number: sg]
[person: 3rd]]]]

Feature Path: <NP agreement person>

Adding constraints to CFG rules

- S → NP VP

  <NP number> = <VP number>

- NP → Det Nominal

  <NP head> = <Nominal head>

  <Det head agree> = <Nominal head agree>

FSs from lexicon, constrs. from rules

Lexicon entry:

[Det
[root: *a*]
[number: sg]]

Rule with constraints:

NP → Det Nominal

<NP number> = <Det number>

<NP number> = <Nominal number>

- Combine to get result:

[NP [Det
[root: *a*]
[number: sg]]
[Nominal [number: sg] …]
[number: sg]]

Similar issue with VP types

Another place where grammar rules could explode:

Jack laughed

VP → Verb *for many **specific** verbs*

Jack found a key

VP → Verb NP *for many **specific** verbs*

Jack gave Sue the paper

VP → Verb NP NP *for many **specific** verbs*

Verb Subcategorization

Verbs have sets of allowed args. Could have many sets of VP rules.
Instead, have a SUBCAT feature, marking sets of allowed arguments:

- +none -- Jack laughed
- +np -- Jack found a key
- +np+np -- Jack gave Sue the paper
- +vp:inf -- Jack wants to fly
- +np+vp:inf -- Jack told the man to go
- +vp:ing -- Jack keeps hoping for the best
- +np+vp:ing -- Jack caught Sam looking at his desk
- +np+vp:base -- Jack watched Sam look at his desk
- +np+pp:to -- Jack gave the key to the man
- +pp:loc -- Jack is at the store
- +np+pp:loc -- Jack put the box in the corner
- +pp:mot -- Jack went to the store
- +np+pp:mot -- Jack took the hat to the party
- +adjp -- Jack is happy
- +np+adjp -- Jack kept the dinner hot
- +sthat -- Jack believed that the world was flat
- +sfor -- Jack hoped for the man to win a prize

50-100 possible ***frames*** for English; a single verb can have several.
(Notation from James Allen "Natural Language Understanding")

Verb frames are ***not*** totally semantic

- It does seem to be partly lexical:

  John wants to fly

  John likes to fly

  John likes flying

  *John wants flying

- Can vary with dialect:

  *??*The car needs washed *(only in Pittsburghese?!)*

Frames for “ask”

(in J+M notation)

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]

Adding transitivity constraint

- S → NP VP

  <NP number> = <VP number>

- NP → Det Nominal

  <NP head> = <Nominal head>

  <Det head agree> = <Nominal head agree>

- VP → Verb NP

  <VP head> = <Verb head>

  <VP head subcat> = +np *(which means transitive)*

Applying a verb subcat feature

Lexicon entry:

[Verb
[root: *found*]
[head: find]
[subcat: +np]]

Rule with constraints:

VP → Verb NP

<VP head> = <Verb head>

<VP head subcat> = +np

- Combine to get result:

[VP [Verb
[root: *found*]
[head: find]
[subcat: +np]]
[NP …]
[head: find [subcat: +np]]]]

Relation to LFG constraint notation

- VP → Verb NP

  <VP head> = <Verb head>

  <VP head subcat> = +np

  from JM book is the same as the LFG expression

- VP → Verb NP

  (↑ head) = (↓ head)

  (↑ head subcat) = +np

Unification

- Merging FSs (and failing if not possible) is often done through ***Unification***
- Simple FS examples:

  [number sg] ⊔ [number sg] = [number sg]

  [number sg] ⊔ [number pl] FAILS

  [number sg] ⊔ [number []] = [number sg]

  [number sg] ⊔ [person 3rd] = [number sg,
  person 3rd]

New kind of “=” sign

- Already had two meanings in programming:
  - “:=“ means “***make*** the left be equal to the right”
  - “==” means “the left and right ***happen to be*** equal”

- Now, a third meaning:
  - ⊔ “=” means “make the left and the right ***be the same thing*** (from now on)” (and ***fail*** if not possible)
    - (Like Lisp **EQ**.)

Seems tricky. Why bother?

- Unification allows the systems that use it to handle many complex phenomena in “simple” elegant ways:
  - There seems to be a dog in the yard.
  - There seem to be dogs in the yard
- Unification makes this work smoothly.
  - Make the Subjects of the clauses EQ:

    <VP subj> = <VP COMP subj>

    [VP [subj: (1)] [COMP [subj: (1)]]]

(Ask Lori Levin for LFG details.)

Complexity

- Unification parsing is “quite expensive”.
  - NP-Complete in some versions!
- So maybe *too* powerful?

    (like GoTo or Call-by-Name?)

  - Add restrictions to make it tractable:
    - Tomita’s Pseudo-unification (Tomabechi too)
    - Gerald Penn work on tractable HPSG: ALE

Semantic roles

and PropBank and FrameNet

- Before we talk about semantic roles, we need to talk about semantics (meaning).

Key Challenge of Meaning

- We actually ***say*** very little - much more is left unsaid, because it’s assumed to be widely known.

- Examples:
  - Reading newspaper stories
  - Using restaurant menus
  - Learning to use a new piece of software

Meaning Representation Languages

- Symbolic representation that does two jobs:
  - Conveys the meaning of a **sentence**
  - Represents (some part of) the **world**
- We're assuming a very literal, context-independent, inference-free version of meaning!
  - Semantics vs. linguists' "pragmatics"
  - "Meaning representation" vs some philosophers' use of the term "semantics".
- For now we'll use **first-order logic**. Also called First-Order Predicate Calculus. Logical form.

Representing NL meaning

- Fortunately, there has been a lot of work on this (since Aristotle, at least)
  - Panini in India too
- Especially, ***formal mathematical logic*** since 1850s (!), starting with George Boole etc.
  - Wanted to replace NL proofs with something more formal
- Deep connections to set theory

Model-Theoretic Semantics

- **Model:** a simplified representation of (some part of) the world: **sets** of objects, properties, relations (**domain**).
- **Non-logical vocabulary:** like variable and function names
  - Each element **denotes** (maps to) a well-defined part of the model. *("Grounding".)*
  - Such a mapping is called an **interpretation**
- **Logical vocabulary:** used to *compose* larger meanings
  - like reserved words in programming languages
  - or function words in grammar

A Model

- **Domain**: Noah, Karen, Rebecca, Frederick, Green Mango, Casbah, Udipi, Thai, Mediterranean, Indian
- **Properties**: Green Mango and Udipi are crowded; Casbah is expensive
- **Relations**: Karen likes Green Mango, Frederick likes Casbah, everyone likes Udipi, Green Mango serves Thai, Casbah serves Mediterranean, and Udipi serves Indian

- n, k, r, f, g, c, u, t, m, i
- Crowded = {g, u}
- Expensive = {c}
- Likes = {(k, g), (f, c), (n, u), (k, u), (r, u), (f, u)}
- Serves = {(g, t), (c, m), (u, i)}

Some English

- *Karen likes Green Mango and Frederick likes Casbah.*
- *Noah and Rebecca like the same restaurants.*
- *Noah likes expensive restaurants.*
- *Not everybody likes Green Mango.*

- What we want is to be able to represent these statements in a way that lets us compare them to our model.
- **Truth-conditional semantics**: need operators and their meanings, given a particular model.

First-Order Logic

- **Terms** refer to elements of the domain: **constants**, **functions**, and **variables**
  - Noah, SpouseOf(Karen), X
- **Predicates** are used to refer to sets and relations; predicate applied to a term is a **Proposition**
  - Expensive(Casbah)
  - Serves(Casbah, Mediterranean)
- Logical connectives (**operators**):

  $\land$ (and), $\lor$ (or), $\neg$ (not), $\Rightarrow$ (implies), ...

- **Quantifiers** ...

Logical operators: truth tables

A	B	A ∧ B	A ∨ B	A ⇒ B
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	1

- Only really need ∧ and ¬

  "A ∨ B" is "(¬A) ∧ (¬B)"

  "A ⇒ B" is "¬ (A ∧ ¬ B)" or "¬A ∨ B"

Quantifiers in FOL

- Two ways to use variables:
  - refer to one anonymous object from the domain (**existential**; ∃; “there exists”)
  - refer to all objects in the domain (**universal**; ∀; “for all”)

- *A restaurant near CMU serves Indian food* ∃x Restaurant(x) ∧ Near(x, CMU) ∧ Serves(x, Indian)
- *All expensive restaurants are far from campus* ∀x Restaurant(x) ∧ Expensive(x) ⇒ ¬Near(x, CMU)

Inference

- Big idea: extend the knowledge base, or check some proposition against the knowledge base.
- **Forward chaining** with modus ponens: given α and $\alpha \Rightarrow \beta$, we know β.
- **Backward chaining** takes a query β and looks for propositions α and $\alpha \Rightarrow \beta$ that would prove β.
  - Not the same as backward reasoning (*abduction*).
  - Used by Prolog
- Both are sound, neither is complete by itself.

Inference example

- Starting with these facts:

  $Restaurant(Udipi)$

  $\forall x\ Restaurant(x) \Rightarrow Likes(Noah, x)$

- We can "turn a crank" and get this *new* fact:

  $Likes(Noah, Udipi)$

FOL: Meta-theory

- Well-defined set-theoretic semantics
- **Sound:** can't prove false things
- **Complete:** can prove everything that logically follows from a set of axioms (e.g., with "resolution theorem prover")

- Well-behaved, well-understood
- Mission accomplished?

FOL: But there are also “Issues”

- “Meanings” of sentences are ***truth values***.
- *Extensional* semantics (vs. *Intensional*); Closed World issue
- Only ***first-order*** (no quantifying over *predicates* [which the book does without comment!]).
- Not very good for ***“fluents”*** (time-varying things, real-valued quantities, etc.). Heard of Zeno?
- Brittle: ***anything*** follows from *any* contradiction(!)
- ***Goedel incompleteness***: “This statement has no proof”!

FOL: But there are also "Issues"

- "Meanings" of sentences are ***truth values***.
- *Extensional* semantics (vs. *Intensional*); Closed World issue
- Only ***first-order*** (no quantifying over *predicates* [which the book does without comment]).
- Not very good for ***"fluents"*** (time-varying things, real-valued quantities, etc.)
- Brittle: ***anything*** follows from *any* contradiction(!)
- ***Goedel incompleteness***: "This statement has no proof"!
  - (Finite axiom sets are incomplete w.r.t. the real world.)

- **So:** Most systems use the FOL **descriptive** apparatus (with extensions) but not its **inference** mechanisms.

Lots More To Say About MRLs!

- See chapter 17 for more about:
  - Representing events and states in FOL
  - Dealing with optional arguments (e.g., “eat”)
  - Representing time
  - Non-FOL approaches to meaning
- Interest in this topic (in NLP) waned during the 1990s and early 2000s.
  - It has come back, with the rise of semi-structured databases like Wikipedia.

Semantic roles

and PropBank and FrameNet

Semantic Cases/Thematic Roles

- Developed in late 1960's and 1970's (Fillmore and others)

- Postulate a limited set of abstract semantic relationships between a verb & its arguments: *thematic roles* or *case roles*

- Part of the verb's (predicate's) semantics

Verbs' subcat frames and roles change together

- *John broke the window with a hammer.*
- *The hammer broke the window.*
- *The window broke.*
- *John broke the window when Bill threw him into it.*

Related problem: Mismatch between FOPC and linguistic arguments

- *John broke the window with a hammer.*
  - Broke(j,w,h)
- *The hammer broke the window.*
  - Broke(h,w)
- *The window broke.*
  - Broke(w)

- Relationship between 1st argument and the predicate is implicit, inaccessible to the system

Thematic Role example

- *John broke the window with the hammer*
- *John*: AGENT role
  window: THEME role
  hammer: INSTRUMENT role

- Extend LF notation to explicitly use semantic roles

Thematic Roles

- Is there a precise way to define meaning of AGENT, THEME, etc.?
- By definition:
  - "The AGENT is an instigator of the action described by the sentence."
- Testing via sentence rewrite:
  - *John* ***intentionally*** *broke the window*
  - **The hammer* ***intentionally*** *broke the window*

Thematic Roles [2]

- THEME
  - Describes the primary object undergoing some change or being acted upon
  - For transitive verb X, “what was Xed?”
  - *The gray eagle saw the mouse*
    “What was seen?” (A: the mouse)

- (Also called “PATIENT”)

Can We Generalize?

- **Thematic roles** describe general patterns of participants in generic events.
- This gives us a kind of shallow, partial semantic representation.
- First proposed by Panini, before 400 BC!

Thematic Roles

Role	*Definition*	*Example*
Agent	Volitional causer of the event	**The waiter** spilled the soup.
Force	Non-volitional causer of the event	**The wind** blew the leaves around.
Experiencer		**Mary** has a headache.
Theme	Most directly affected participant	Mary swallowed **the pill**.
Result	End-product of an event	We constructed **a new building**.
Content	Proposition of a propositional event	Mary knows **you hate her**.
Instrument		You shot her with **a pistol**.
Beneficiary		I made **you** a reservation.
Source	Origin of a transferred thing	I flew in from **Pittsburgh**.
Goal	Destination of a transferred thing	Go to **hell**!

Thematic Roles

Dumb joke!

Role	*Definition*	*Example*
Agent	Volitional causer of the event	**The waiter** spilled the soup.
Force	Non-volitional causer of the event	**The wind** blew the leaves around.
Experiencer		**Mary** has a headache.
Theme	Most directly affected participant	Mary swallowed **the pill**.
Result	End-product of an event	We constructed **a new building**.
Content	Proposition of a propositional event	Mary knows **you hate her**.
Instrument		You shot her with **a pistol**.
Beneficiary		I made **you** a reservation.
Source	Origin of a transferred thing	I flew in from **Pittsburgh**.
Goal	Destination of a transferred thing	Go to **hell**!

Patient

Review: Verb Subcategorization

Verbs have sets of allowed args. Could have many sets of VP rules. Instead, have a SUBCAT feature, marking sets of allowed arguments:

- +none -- Jack laughed
- +np -- Jack found a key
- +np+np -- Jack gave Sue the paper
- +vp:inf -- Jack wants to fly
- +np+vp:inf -- Jack told the man to go
- +vp:ing -- Jack keeps hoping for the best
- +np+vp:ing -- Jack caught Sam looking at his desk
- +np+vp:base -- Jack watched Sam look at his desk
- +np+pp:to -- Jack gave the key to the man
- +pp:loc -- Jack is at the store
- +np+pp:loc -- Jack put the box in the corner
- +pp:mot -- Jack went to the store
- +np+pp:mot -- Jack took the hat to the party
- +adjp -- Jack is happy
- +np+adjp -- Jack kept the dinner hot
- +sthat -- Jack believed that the world was flat
- +sfor -- Jack hoped for the man to win a prize

50-100 possible ***frames*** for English; a single verb can have several.

(Notation from James Allen "Natural Language Understanding")

Thematic Grid or Case Frame

- Example: break
  - The child broke the vase. < agent theme >
    subj obj
  - The child broke the vase with a hammer.
    < agent theme instr >
    subj obj PP
  - The hammer broke the vase. < theme instr >
    obj subj
  - The vase broke. < theme >
    subj

Thematic Grid or Case Frame

- Example: break
  - The child broke the vase. < agent theme >

    subj obj
  - The child broke the vase with a hammer.

    < agent theme instr >

    subj obj PP
  - The hammer broke the vase. < theme instr >

    obj subj
  - The vase broke. < theme >

    subj

The Thematic Grid or Case Frame shows

- How many arguments the verb has
- What roles the arguments have
- Where to find each argument
  - For example, you can find the agent in the subject position

Diathesis Alternation:

a change in the number of arguments or the grammatical relations associated with each argument

Sentence		agent	theme	goal	
• Chris gave a book to Dana.	<	agent	theme	goal	>
		subj	obj	PP	
• A book was given to Dana by Chris.	<	agent	theme	goal	>
		PP	subj	PP	
• Chris gave Dana a book.	<	agent	theme	goal	>
		subj	obj2	obj	
• Dana was given a book by Chris.	<	agent	theme	goal	>
		PP	obj	subj	

The Trouble With Thematic Roles

- They are not formally defined.
- Some roles generalize well, but not all.
- General roles are overly general:
  - "*agent verb theme* with *instrument*" and "*instrument verb theme*" ...
    - The cook opened the jar with the new gadget.

      → The new gadget opened the jar.
    - Susan ate the sliced banana with a fork.

      → #The fork ate the sliced banana.

Two Datasets

- Proposition Bank (PropBank): verb-specific thematic roles
- FrameNet: "frame"-specific thematic roles

- These are both lexicons containing case frames/thematic grids for each verb.

Proposition Bank (PropBank)

- A set of **verb-sense-specific** “frames” with informal English glosses describing the roles
- Conventions for labeling optional modifier roles
- Penn Treebank is labeled with those verb-sense-specific semantic roles.

“Agree” in PropBank

- arg0: agreer
- arg1: proposition
- arg2: other entity agreeing

- The group agreed it wouldn’t make an offer.
- Usually John agrees with Mary on everything.

- arg0 is proto-agent, arg1 proto-patient

“Fall (move downward)” in PropBank

- arg1: logical subject, patient, thing falling
- arg2: extent, amount fallen
- arg3: starting point
- arg4: ending point
- argM-loc: medium
- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Cambridge.

FrameNet

- FrameNet is similar, but abstracts from specific verbs, so that semantic **frames** are first-class citizens.
- For example, there is a single frame called **change_position_on_a_scale**.

change_position_on_a_scale

Core Roles	
ATTRIBUTE	The ATTRIBUTE is a scalar property that the ITEM possesses.
DIFFERENCE	The distance by which an ITEM changes its position on the scale.
FINAL_STATE	A description that presents the ITEM's state after the change in the ATTRIBUTE's value as an independent predication.
FINAL_VALUE	The position on the scale where the Item ends up.
INITIAL_STATE	A description that presents the ITEM's state before the change in the ATTRIBUTE's value as an independent predication.
INITIAL_VALUE	The initial position on the scale from which the ITEM moves away.
ITEM	The entity that has a position on the scale.
VALUE_RANGE	A portion of the scale, typically identified by its end points, along which the values of the ATTRIBUTE fluctuate.
Some Non-Core Roles	
DURATION	The length of time over which the change takes place.
SPEED	The rate of change of the VALUE.
GROUP	The GROUP in which an ITEM changes the value of an ATTRIBUTE in a specified way.

Oil **rose** in price by 2%
It has **increased** to having them 1 day a month.
Microsoft shares **fell** to 7 5/8.
Colon cancer incidence **fell** by 50% among men.

Many words, not just verbs, share the same frame:

Verbs: advance, climb, decline, decrease, diminish, dip, double, drop, dwindle, edge, explode, fall, fluctuate, gain, grow, increase, jump, move, mushroom, plummet, reach, rise, rocket, shift, skyrocket, slide, soar, swell, swing, triple, tumble
Nouns: decline, decrease, escalation, explosion, fall, fluctuation, gain, growth, hike, increase, rise, shift, tumble
Adverb: increasingly

Conversely, one word has many frames

Example: rise

- **Change-position-on-a-scale**: Oil ROSE in price by two percent.
- **Change-posture**: a protagonist changes the overall position or posture of a body.
  - Source: starting point of the change of posture.
  - Charles ROSE from his armchair.
- **Get-up**: A Protagonist leaves the place where they have slept, their Bed, to begin or resume domestic, professional, or other activities. Getting up is distinct from Waking up, which is concerned only with the transition from the sleeping state to a wakeful state.
  - I ROSE from bed, threw on a pair of camouflage shorts and drove my little Toyota Corolla to a construction clearing a few miles away.
- **Motion-directional**: In this frame a Theme moves in a certain Direction which is often determined by gravity or other natural, physical forces. The Theme is not necessarily a self-mover.
  - The balloon ROSE upward.
- **Sidereal-appearance:** An Astronomical_entity comes into view above the horizon as part of a regular, periodic process of (apparent) motion of theAstronomical_entity across the sky. In the case of the sun, the appearance begins the day.
  - At the time of the new moon, the moon RISES at about the same time the sun rises, and it sets at about the same time the sun sets.

  Each day the sun's RISE offers us a new day.

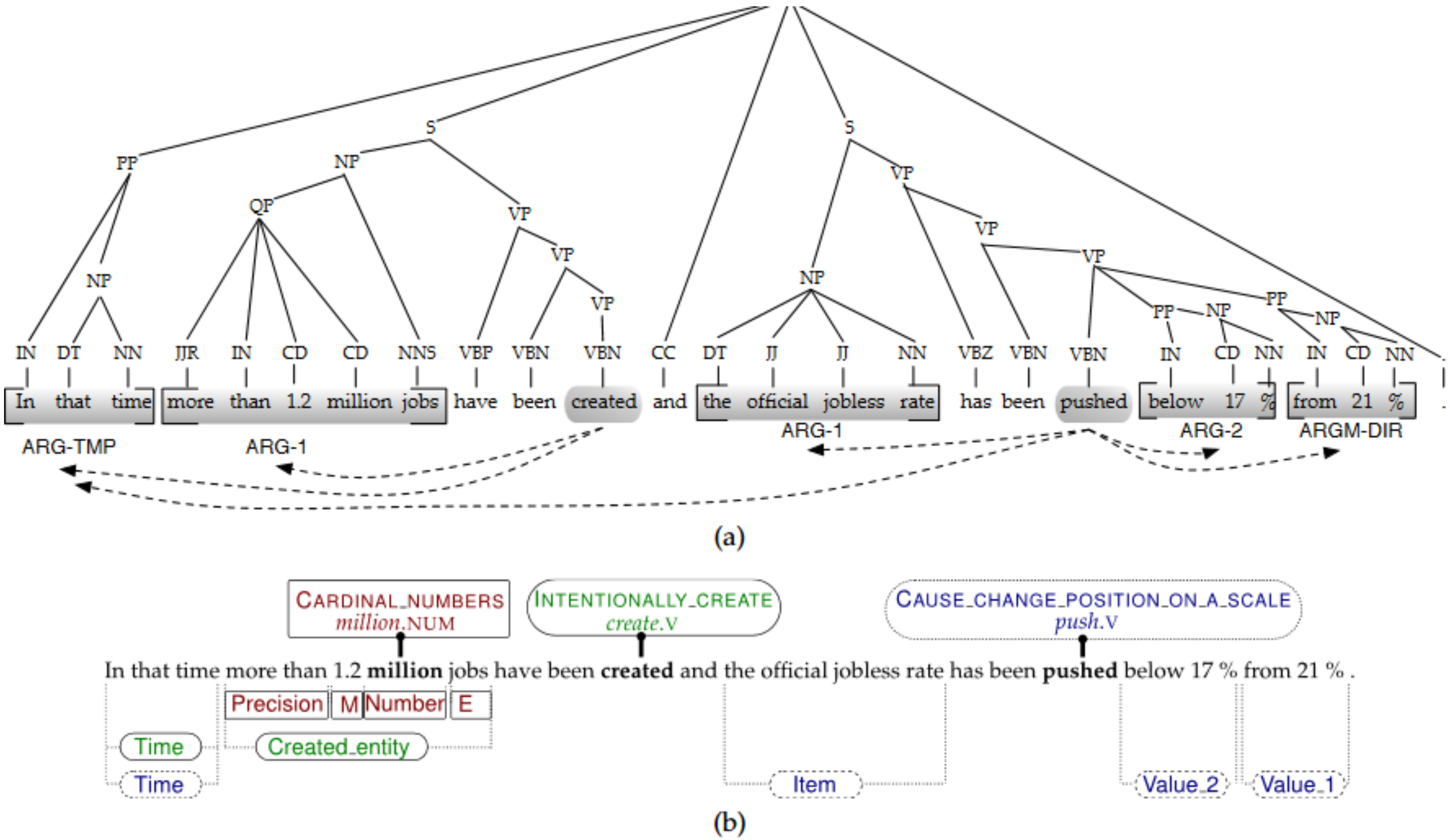
FrameNet

- Frames are not just for verbs!
- **Verbs**: advance, climb, decline, decrease, diminish, dip, double, drop, dwindle, edge, explode, fall, fluctuate, gain, grow, increase, jump, move, mushroom, plummet, reach, rise, rocket, shift, skyrocket, slide, soar, swell, swing, triple, tumble
- **Nouns**: decline, decrease, escalation, explosion, fall, fluctuation, gain, growth, hike, increase, rise, shift, tumble
- **Adverb**: increasingly

FrameNet

- Includes inheritance and causation relationships among frames.
- Examples included, but little fully-annotated corpus data.

PropBank vs FrameNet



SemLink

- It would be really useful if these different resources were interconnected in a useful way.
- SemLink project is (was?) trying to do that
- Unified Verb Index (UVI) connects
  - PropBank
  - VerbNet
  - FrameNet
  - WordNet/OntoNotes

Semantic Role Labeling

- Input: sentence
- Output: for each predicate*, labeled spans identifying each of its arguments.

- Example:

  $[_{\text{agent}}$ The batter$]$ hit $[_{\text{patient}}$ the ball$]$ $[_{\text{time}}$ yesterday$]$

- Somewhere between syntactic parsing and full-fledged compositional semantics.

*Predicates are sometimes identified in the input, sometimes not.

But wait. How is this different from dependency parsing?

- Semantic role labeling
  - [$_{agent}$ The batter] hit [$_{patient}$ the ball] [$_{time}$ yesterday]
- Dependency parsing
  - [$_{subj}$ The batter] hit [$_{obj}$ the ball] [$_{mod}$ yesterday]

But wait. How is this different from dependency parsing?

- Semantic role labeling
  - [$_{agent}$ The batter] hit [$_{patient}$ the ball] [$_{time}$ yesterday]
- Dependency parsing
  - [$_{subj}$ The batter] hit [$_{obj}$ the ball] [$_{mod}$ yesterday]

➢ These are not the same task.

➢ Semantic role labeling is much harder.

Subject vs agent

- Subject is a grammatical relation
- Agent is a semantic role

- In English, a subject has these properties
  - It comes before the verb
  - If it is a pronoun, it is in nominative case (in a finite clause)
    - I/he/she/we/they hit the ball.
    - *Me/him/her/us/them hit the ball.
  - If the verb is in present tense, it agrees with the subject
    - She/he/it hits the ball.
    - I/we/they hit the ball.
    - *She/he/it hit the ball.
    - *I/we/they hits the ball.
    - I hit the ball.
    - I hit the balls.

Subject vs agent

- In the most typical sentences (for some definition of "typical"), the agent is the subject:
  - The batter hit the ball.
  - Chris opened the door.
  - The teacher gave books to the students.

- Sometimes the agent is not the subject:
  - The ball was hit by the batter.
  - The balls were hit by the batter.

- Sometimes the subject is not the agent:
  - The door opened.
  - The key opened the door.
  - The students were given books.
  - Books were given to the students.

Semantic Role Labeling

- Input: sentence
- Output: segmentation into roles, with labels

- Example from J&M II book:
- [$_{\text{arg0}}$ The Examiner] issued [$_{\text{arg1}}$ a special edition] [$_{\text{argM-tmp}}$ yesterday]

- (In Propbank notation, arg0 is proto-agent, arg1 is proto-patient.)

Semantic Role Labeling: How It Works

- First, parse.

- For each predicate word in the parse:

  For each node in the parse:

    Classify the node with respect to the predicate.

Features for Semantic Role Labeling

- What is the predicate?
- Phrase type of the constituent
- Head word of the constituent, its POS
- Path in the parse tree from the constituent to the predicate
- Active or passive
- Is the phrase before or after the predicate?
- Subcategorization (≈ grammar rule) of the predicate

Feature example

- Example sentence:

[$_{arg0}$ The Examiner] issued [$_{arg1}$ a special edition] [$_{argM\text{-}tmp}$ yesterday]

- Arg0 features:

issued, NP, Examiner, NNP, ***path***, active, before, VP->VBD NP PP

Example *path*

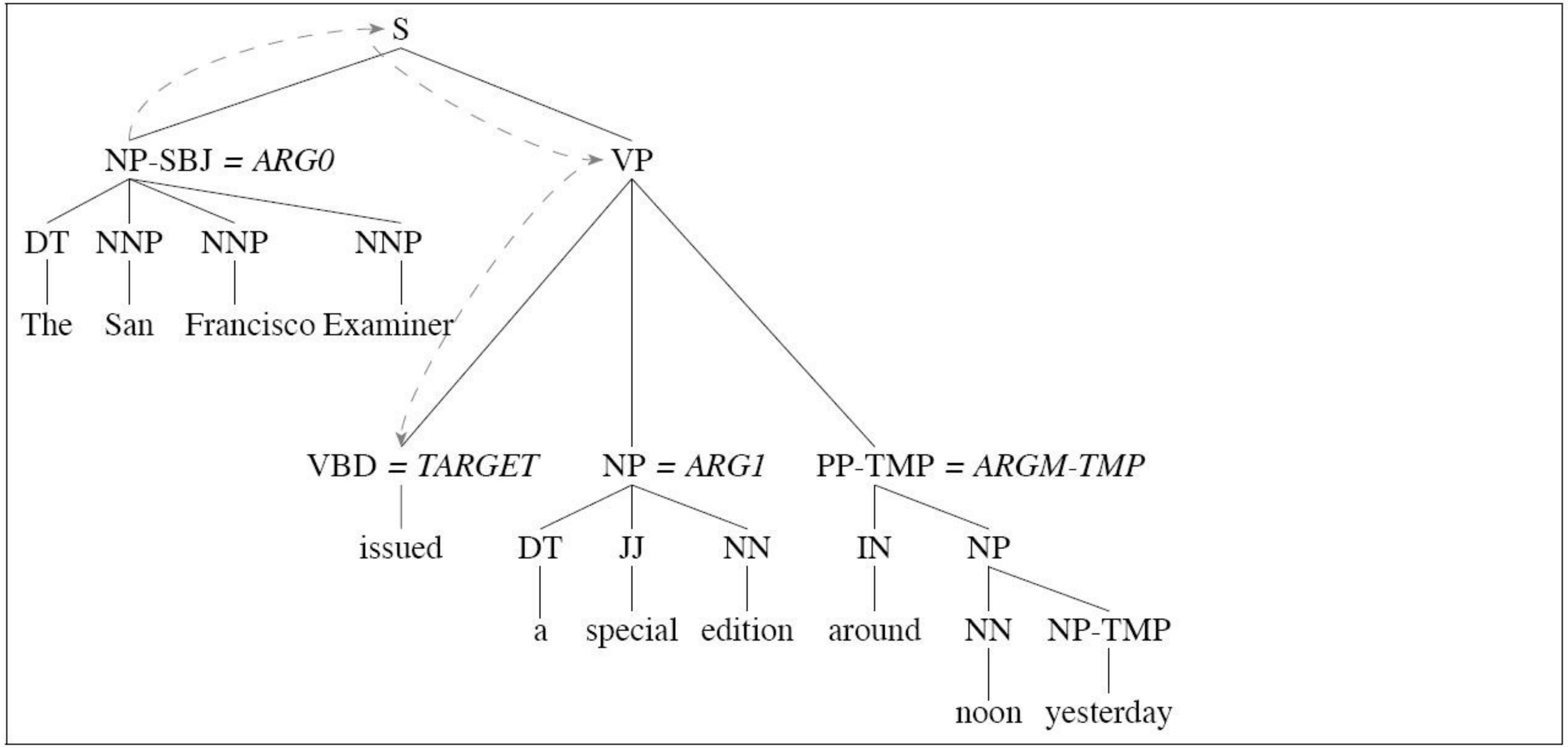


Figure 20.16: Parse tree for a PropBank sentence, showing the PropBank argument labels. The dotted line shows the **path** feature **NP ↑ S ↓ VP ↓ VBD** for **ARG0**, the NP-SBJ constituent *The San Francisco Examiner*