Computing Minimalism

Sandiway Fong
Departments of Linguistics and Computer Science
University of Arizona

this work is in part jointly developed with Jason Ginsburg, U. of Aizu, Japan

25th Annual CUNY Conference on Human Sentence Processing. March 14-16, 2012.

Minimal Computation (MC)

Chomsky (2011)

- language is a computational system:
  - Generative Procedure (GP)
  - Parsing Procedure (PP)
- *we should consider computational efficiency,* but…
- when it comes to Minimal Computation (MC):
  - *conflicts between GP and PP are always resolved in favor of GP*

(12) S' {{that book}, S} = {{that book}, {John, {read, {that, book}}}

In (12) there are two *copies* of the object {that, book}. S', with the two copies, has the right form for CI (conceptual-intentional) but not of course for SM (sensory-motor)—first because linearization is required and second because the hierarchically less prominent copy is not pronounced. The latter property follows at once from MC: at least the most prominent copy must be pronounced, or there will be no evidence that topicalization took place at all, but computation is reduced if at most one copy is pronounced—massively reduced in nontrivial cases.[8]

Minimal Computation (MC)

- Chomsky (2011)

also follows from MC. The inadequacy is severe. Anyone who has worked on parsing programs knows that a major difficulty is posed by "filler-gap" problems: given *who* in (10), the problem for parsing/perception is to locate the gap where it receives its interpretation in the argument structure of the sentence, not a trivial matter in general. These problems would largely be obviated if all copies were pronounced, violating MC.

In brief, where there is a conflict between communicative and computational efficiency, the latter seems to win, hands down. It appears that Aristotle's dictum should be reversed: language is not sound with meaning but meaning with sound, a very different matter. Externalization by the SM system appears to be a secondary property of language.

.. explore idea that the Generative Procedure (GP) and Parsing Procedure (PP) share architecture

PP can help GP minimize computation

What constitutes an implementation?

- Minimalism:
  - (Chomsky 2001) *and later papers*
- Components of the theory:
  1. Recursive Merge (internal/external)
  2. Probe/Goal search (active/inactive)
  3. Agreement (value Case, interpretable/ uninterpretable features)
  4. Phases (*limits on search*)

What constitutes an implementation?

assume, are raising constructions and their exceptional-Case-marking (ECM) counterparts, as shown schematically in (4a), where β is the matrix clause, α is an infinitival with YP a verbal phrase (the case most relevant here), and P is the probe: T with a raising verb (case (4b)), *v* with an ECM transitive verb (case (4c)).[10]

(4) a. $[_\beta$ P $[_\alpha$ [Subj [H YP]]]]

b. i. there are likely to be awarded several prizes

ii. several prizes are likely to be awarded

c. i. we expect there to be awarded several prizes

ii. we expect several prizes to be awarded

The Case/agreement properties of Subj in (4a), and its overt location, are determined by properties of the matrix probe P, not internally to α. α is a TP with defective head T_{def}, which is unable to determine Case/agreement but has an EPP-feature, overtly manifested in (4c). Raising-ECM parallels give good reason to believe that the EPP-feature is manifested in (4b) as well, by trace of the matrix subject; preference for Merge over (more complex) Move gives a plausible reason for the surface distinction between [Spec, T_{def}] in (4b) and in (4c) (see MI). In (4bi) and (4ci), the EPP-feature of T_{def} is satisfied by Merge of expletive; in (4bii) and (4cii), by raising of the direct object.

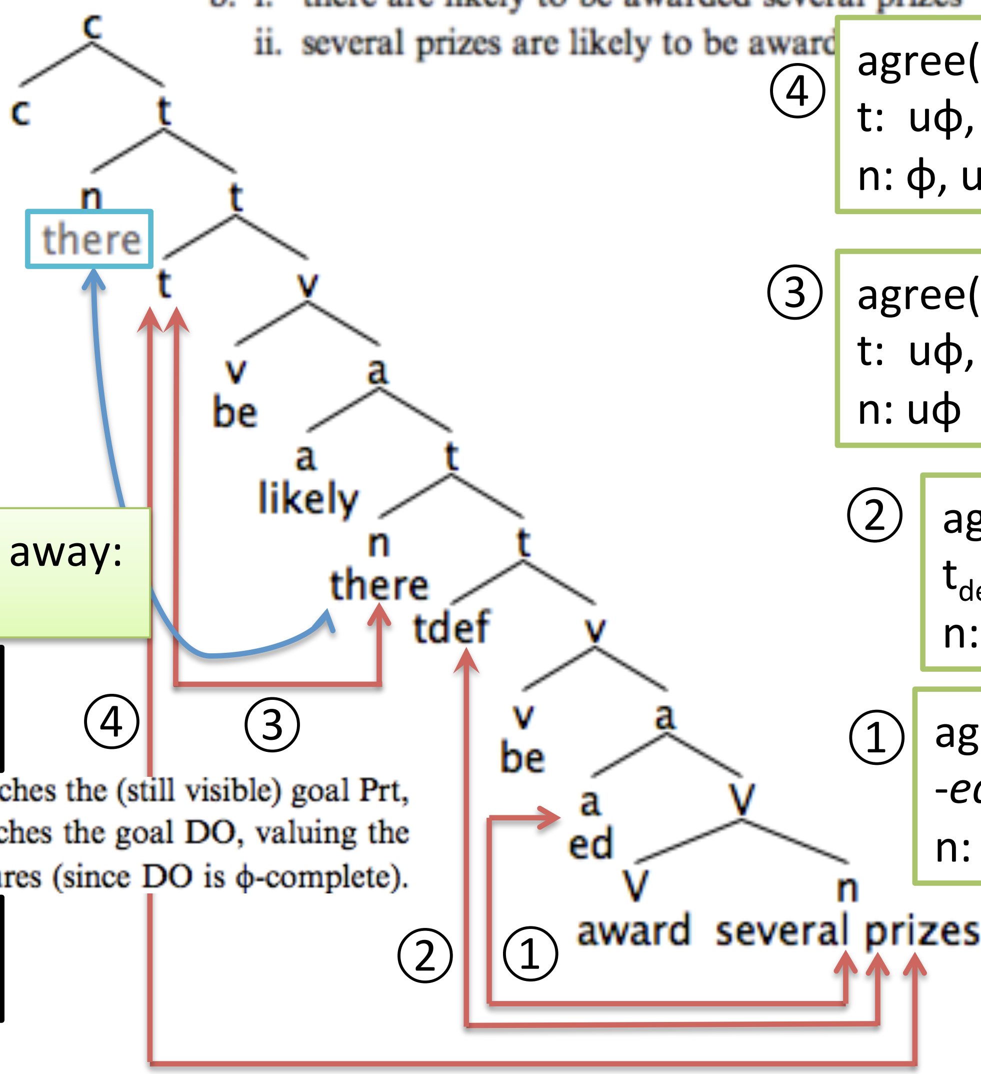
excerpt from Derivation by Phase

Grammar formalisms e.g. Stabler (1998), (2011) (Lecomte & Retoré) 2001

What constitutes an implementation?

1. theta merge V & N
2. merge PRT & V
3. merge v & A
4. merge T & v
5. merge expl & T
6. merge A & Tdef
7. merge v & A
8. merge T & v
9. move to spec-T
10. merge C & T

(4) a. [$_\beta$ P [$_\alpha$ [Subj [H YP]]]]
b. i. there are likely to be awarded several prizes
ii. several prizes are likely to be award



④ agree(t,n)
t: uϕ, Nominative
n: ϕ, uCase

③ agree(t,n)
t: uϕ, Nominative
n: uϕ

② agree(t_{def},n)
t_{def}: uϕ
n: ϕ, uCase

① agree(a,n)
-*ed*: uϕ, uCase
n: ϕ, uCase

implementation optimizes this away:
uCase can be unified during agree

*actually, should need agree(t,a) as well to value Case on PRT

cussed. At the next stage, the probe T/*v* matches the (still visible) goal Prt, valuing its Case feature; and the probe matches the goal DO, valuing the Case feature of DO as well as its own features (since DO is ϕ-complete).

one probe, multiple goals:
matrix T agrees with 3 goals

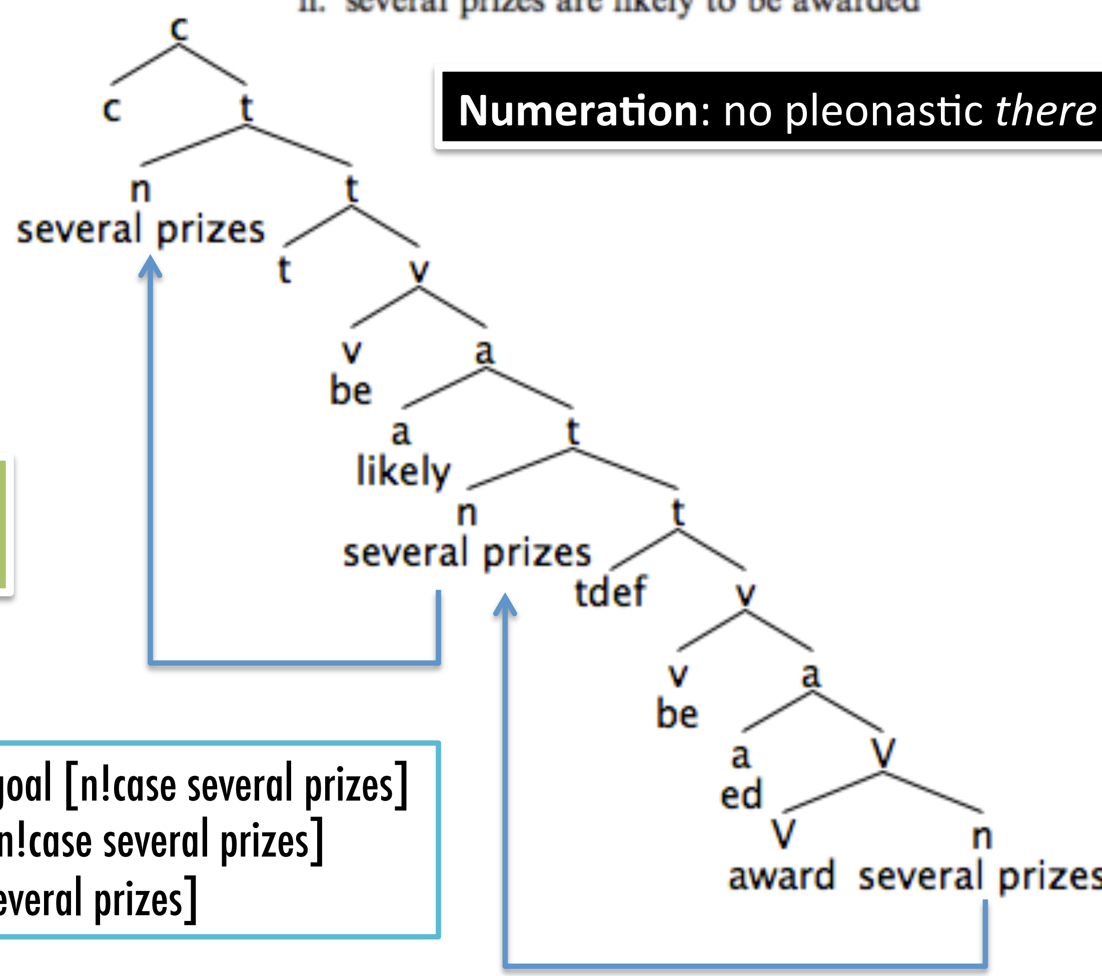
What constitutes an implementation?

(4) a. [$_\beta$ P [$_\alpha$ [Subj [H YP]]]]
b. i. there are likely to be awarded several prizes
ii. several prizes are likely to be awarded

1. theta merge V & N
2. merge PRT & V
3. merge v & A
4. merge T & v
5. move to spec–T
6. merge A & Tdef
7. merge v & A
8. merge T & v
9. move to spec–T
10. merge C & T

Numeration: no pleonastic *there*

same number of Merge steps as before



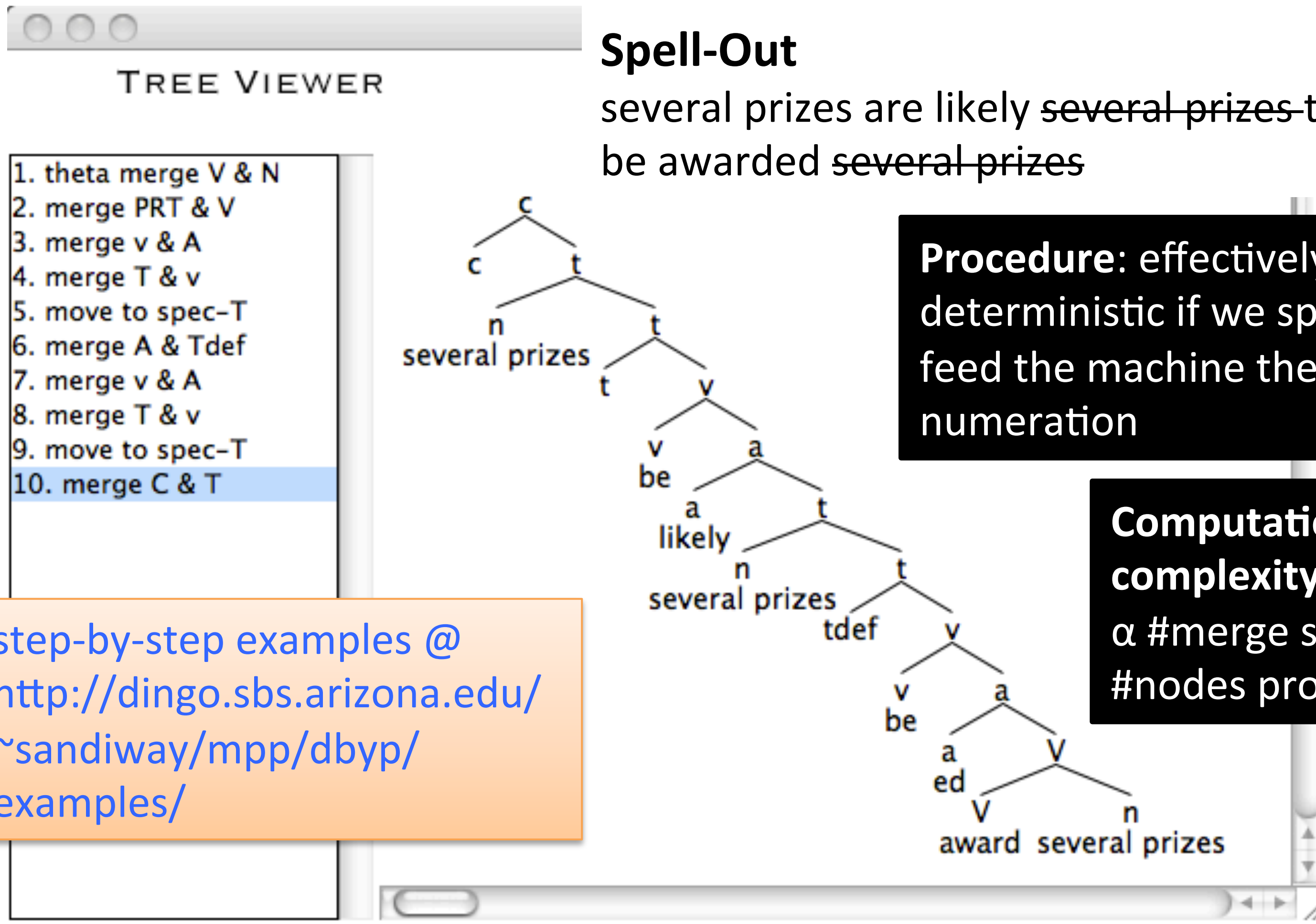
probe [a!case ed] agrees with goal [n!case several prizes]
probe [tdef] agrees with goal [n!case several prizes]
probe [t] agrees with goal [n several prizes]

Computation steps

- Generative Procedure (GP)

Spell-Out

several prizes are likely ~~several prizes~~ to be awarded ~~several prizes~~



Procedure: effectively deterministic if we spoon feed the machine the numeration

Computational complexity?
α #merge steps + β #nodes probed + …

step-by-step examples @ http://dingo.sbs.arizona.edu/~sandiway/mpp/dbyp/examples/

Minimal Computation

- overriding condition of **Minimal Computation** (MC) means design for the **Generative Procedure** (GP):
  - *"whenever efficiency of design and communication conflicts: design wins"*
- no **Parsing Procedure** (PP) is specified
  - parsing need not be efficient
- **Example**:
  - several prizes are likely ~~several prizes~~ to be awarded ~~several prizes~~

suggest consequences for cognitive architecture. Keeping to the principle of MC, the GP yields forms that are appropriate for semantic interpretation at CI but not for production and perception at sensory-motor (SM), though this SM inadequacy also follows from MC. The inadequacy is severe. Anyone who has worked on parsing programs knows that a major difficulty is posed by "filler-gap" problems: given *who* in (10), the problem for parsing/perception is to locate the gap where it receives its interpretation in the argument structure of the sentence, not a trivial matter in general. These problems would largely be obviated if all copies were pronounced, violating MC.

Chomsky (2011)

Design Minimalism

Generative Procedure (GP)

- Design minimalism:
  - good for complexity, but what about data?
- Limits on derivation

  conservation of syntactic objects (SO)
  - e.g. only copies, can't create indices
- Problem for modules

  e.g. **Binding theory** (BT)

  assuming we still want to have a syntactic BT …
  - no Free Indexation
  - no theory-internal levels of representations
  - no Binding Conditions A, B and C

Growing the implementation

Binding theory (*joint work with Jason Ginsburg, U. of Aizu, Japan*)

- Idea:
  - **doubling constituent** (DC), adapted from Kayne (2002)
  - e.g. [he John] ⇨ uCase, iϕ, theta Merge

    [$_{TP}$ John thinks [$_{TP}$ [he John] [$_{v}$ is [$_{AP}$ smart [he John]]]]]

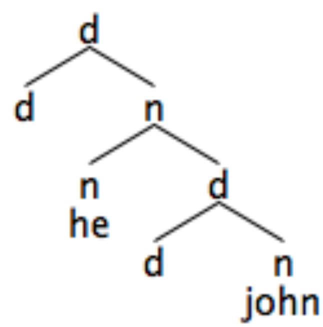
    John thinks ~~John~~ he is smart ~~John~~ ~~he~~

  - **Phases**: derive distributional differences between pronouns and anaphors

related work:
Zwart (2002)
Heinat (2003)

John$_i$ thinks he$_i$ is smart

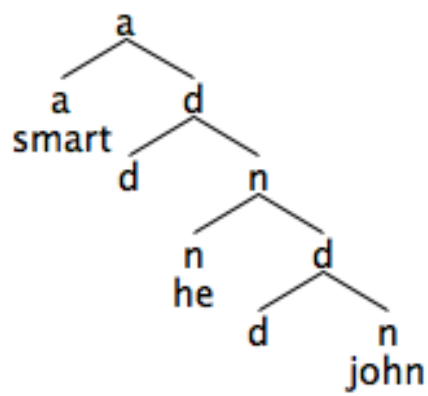
1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec–T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec–T
14. merge C & T



scoreboard	*he*	*John*
lacks	theta, Case	theta, Case
licensed		

John$_i$ thinks he$_i$ is smart

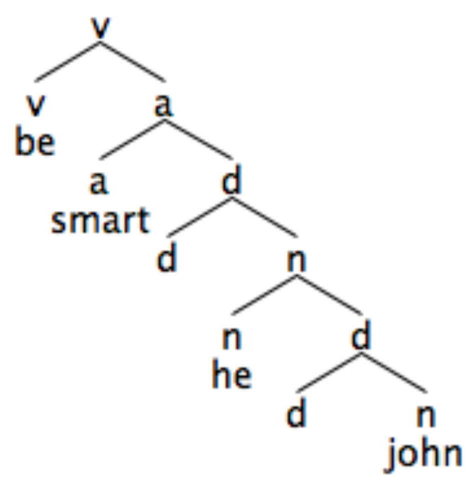
1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec-T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec-T
14. merge C & T



scoreboard	*he*	*John*
lacks	Case	theta, Case
licensed	theta	

John$_i$ thinks he$_i$ is smart

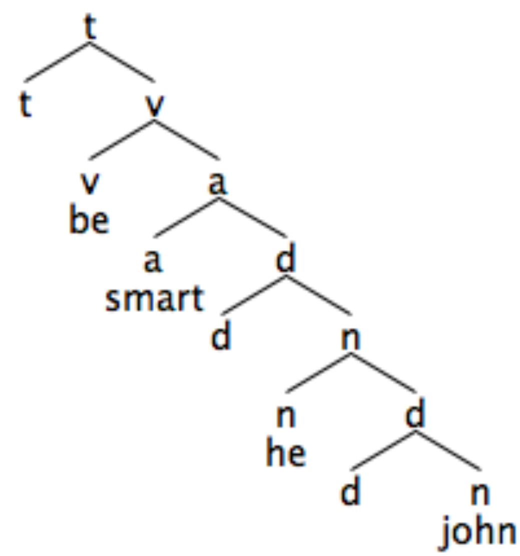
1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec–T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec–T
14. merge C & T



scoreboard	*he*	*John*
lacks	Case	theta, Case
licensed	theta	

John$_i$ thinks he$_i$ is smart

1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec-T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec-T
14. merge C & T

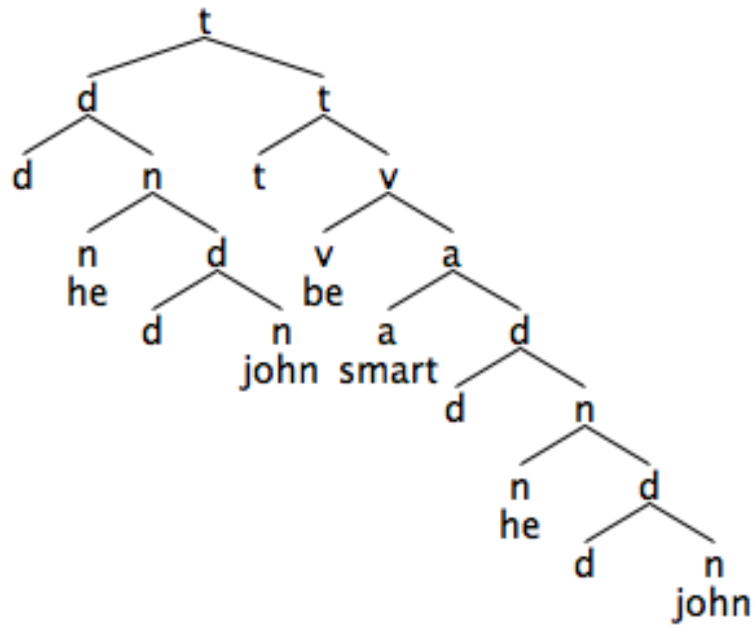


Probe [t] agrees with goal [d[d][n[n he][d!case!arg[d][n john]]]]

scoreboard	*he*	*John*
lacks		theta, Case
licensed	theta, Case	

John$_i$ thinks he$_i$ is smart

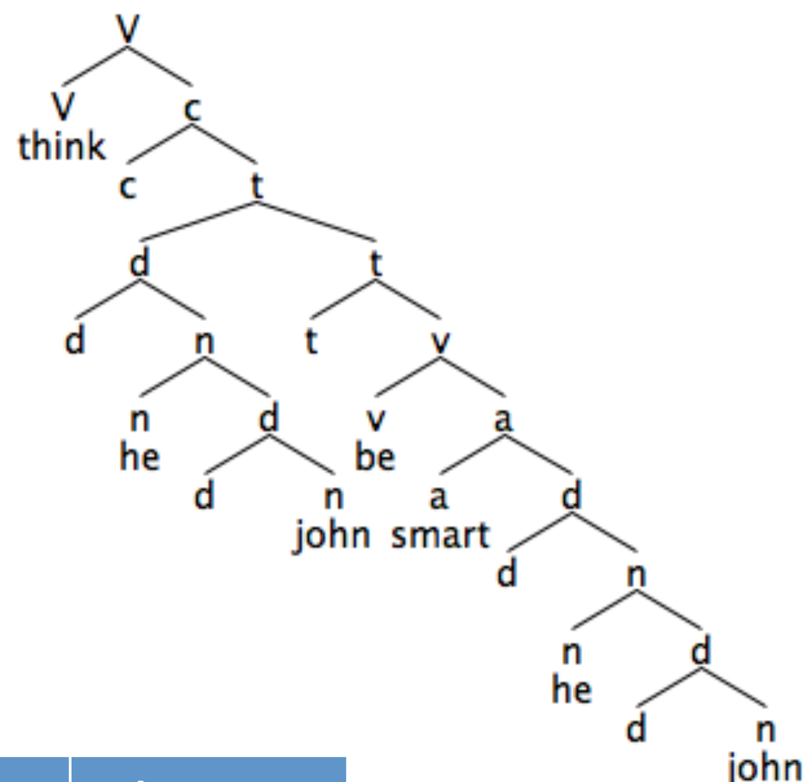
1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec-T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec-T
14. merge C & T



scoreboard	*he*	*John*
lacks		theta, Case
licensed	theta, Case	

John$_i$ thinks he$_i$ is smart

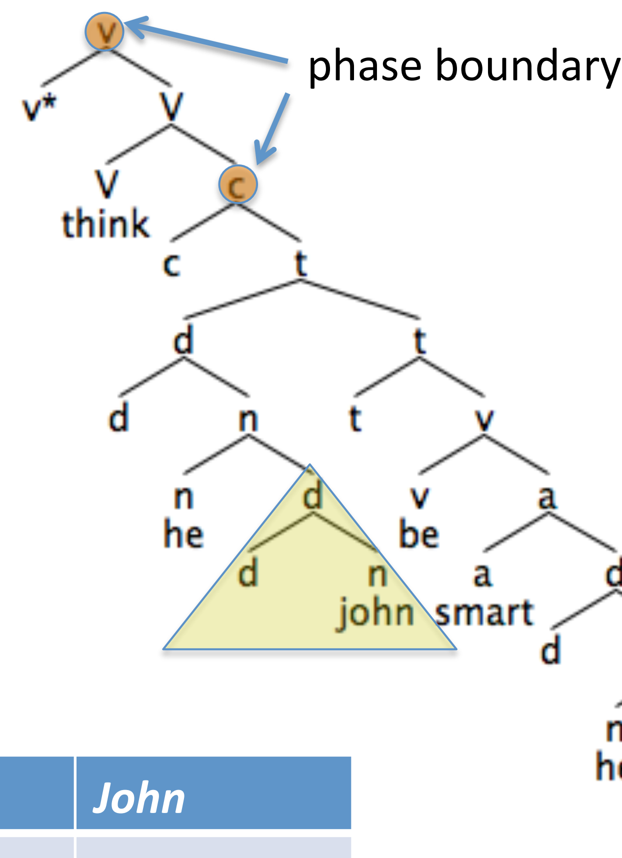
1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec–T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec–T
14. merge C & T



scoreboard	*he*	*John*
lacks		theta, Case
licensed	theta, Case	

John$_i$ thinks he$_i$ is smart

1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec-T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec-T
14. merge C & T

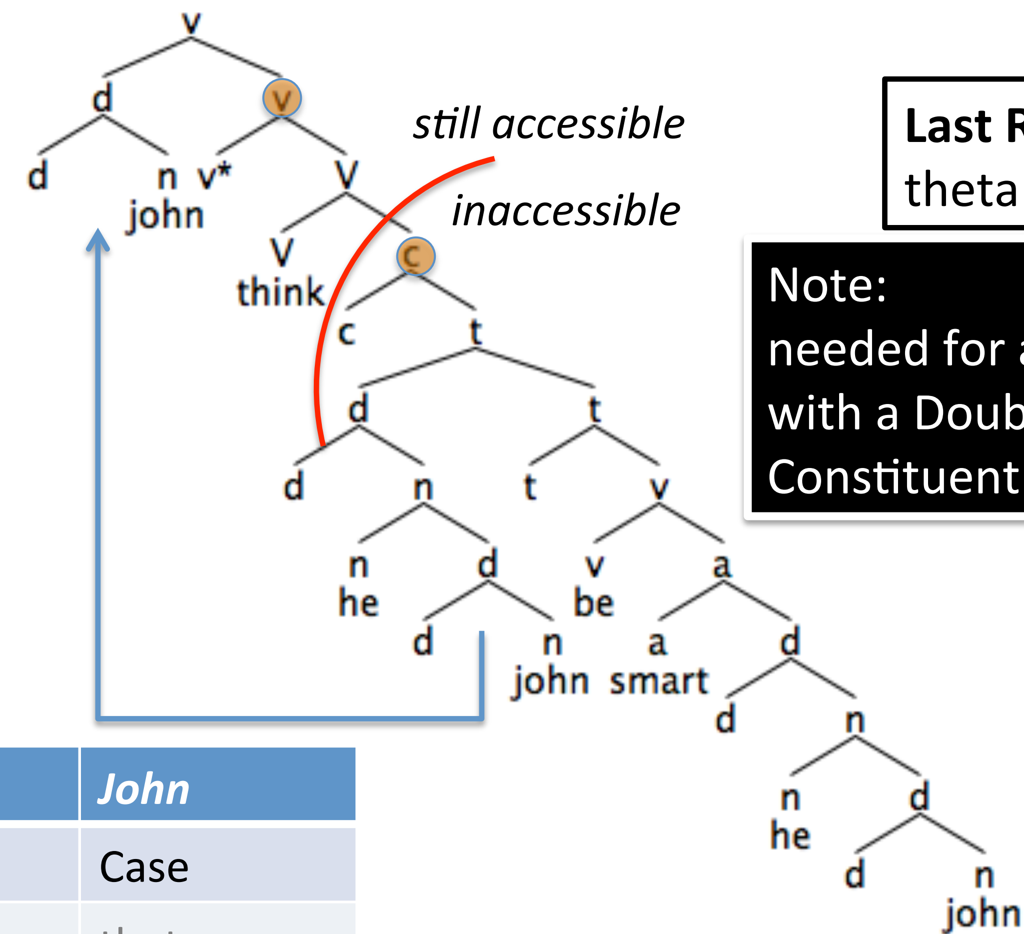


Essential idea:
probe-goal search has limited range
Suppose r-expr with unvalued uninterpretable feature at the limit of search is allowed to undergo theta Merge …
obviously, timing is critical

scoreboard	*he*	*John*
lacks		theta, Case
licensed	theta, Case	

John$_i$ thinks he$_i$ is smart

1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec–T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec–T
14. merge C & T



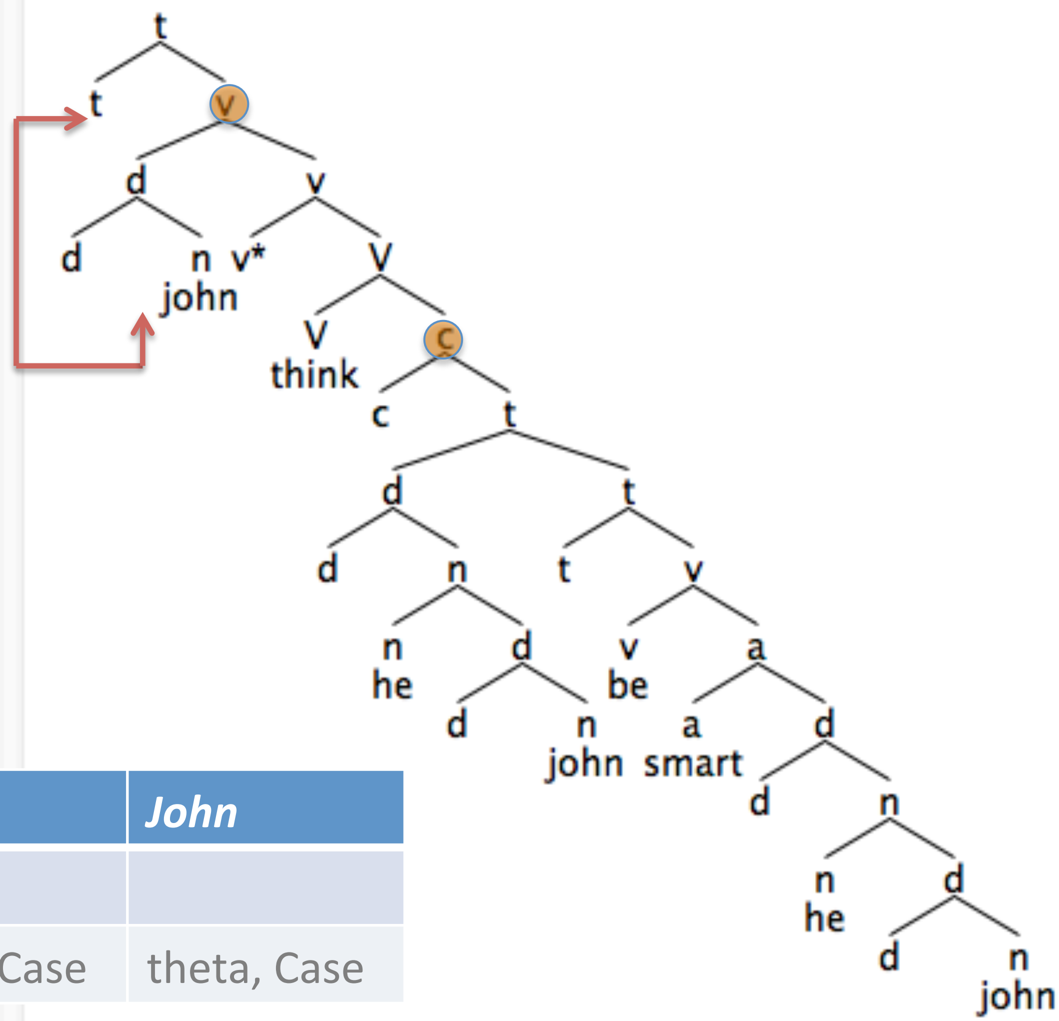
Last Resort (LR): theta Merge

Note:
needed for all theories with a Doubling Constituent (DC)

scoreboard	*he*	*John*
lacks		Case
licensed	theta, Case	theta

John$_i$ thinks he$_i$ is smart

1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec–T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec–T
14. merge C & T

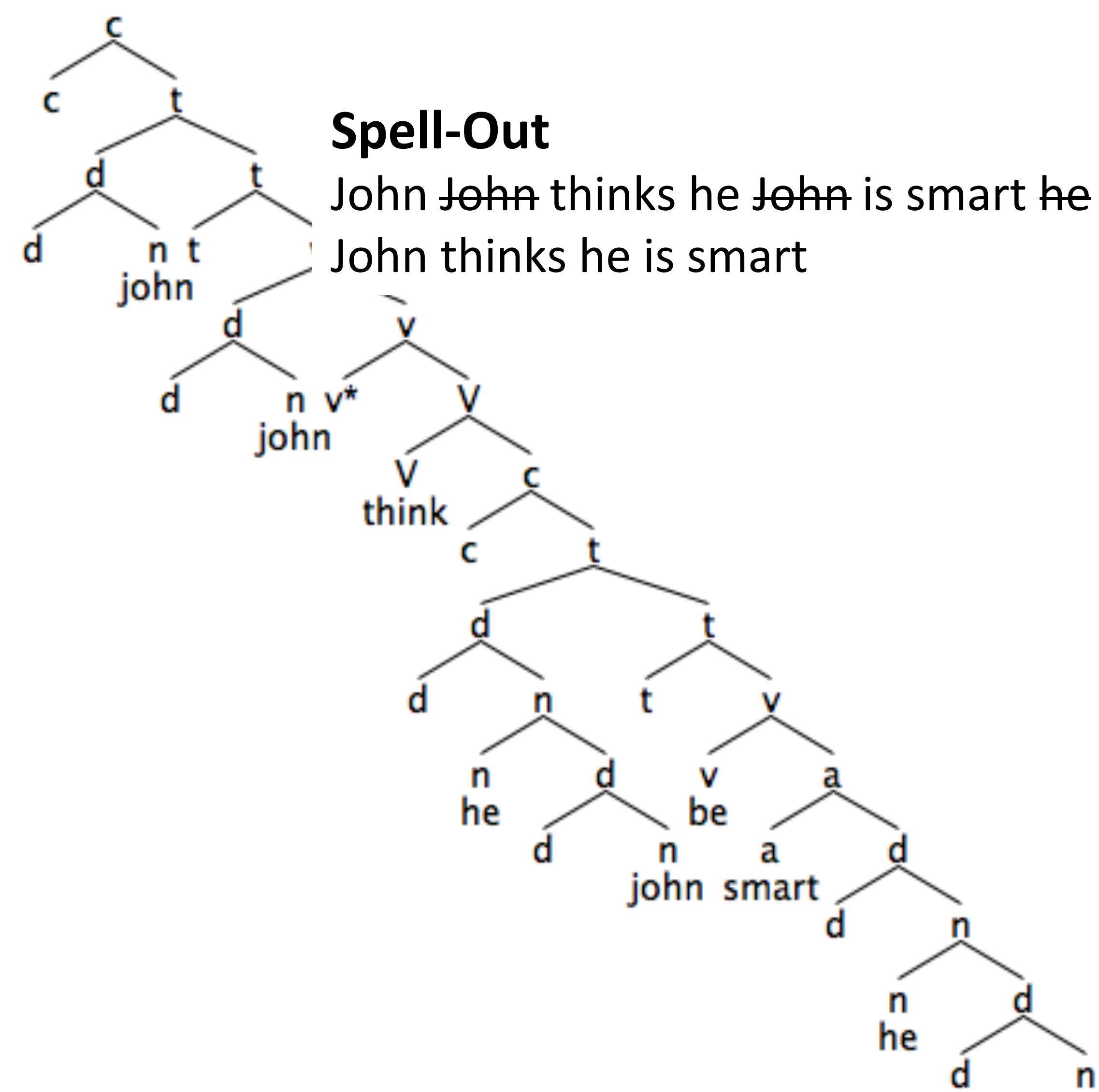


scoreboard	*he*	*John*
lacks		
licensed	theta, Case	theta, Case

Probe [t] agrees with goal [d[d][n john]]

John$_i$ thinks he$_i$ is smart

1. merge D & N
2. merge N & D
3. merge D & N
4. theta merge A & N
5. merge v & A
6. merge T & v
7. move to spec–T
8. merge C & T
9. merge V & C
10. merge v* & V
11. LR move to v*
12. merge T & v
13. move to spec–T
14. merge C & T



Spell-Out

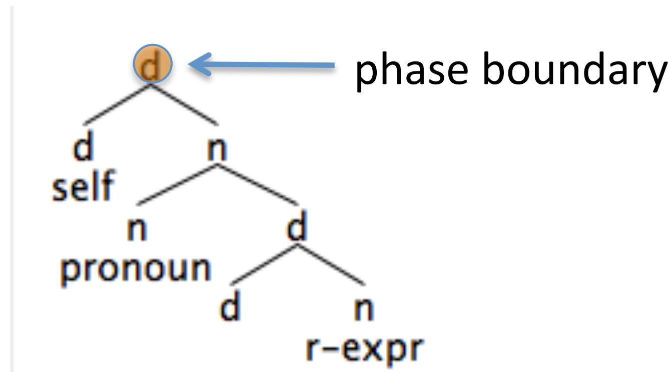
John ~~John~~ thinks he ~~John~~ is smart ~~he~~ ~~John~~

John thinks he is smart

Data

- Analysis of classic BT data:
  - *John$_i$ praises him$_i$
  - John$_i$ praises himself$_i$
  - John$_i$ thinks he$_i$ is smart
  - *He$_i$ thinks John$_i$ is smart
  - *John$_i$ thinks himself$_i$ is smart
  - *John$_i$ thinks that Mary likes himself$_i$
  - John$_i$ considers himself$_i$ to be intelligent
  - *John$_i$ considers him$_i$ to be intelligent
  - John$_i$ likes his$_i$ dog
  - *John$_i$ likes himself$_i$'s dog
  - ?*Hannah$_i$ found a picture of her$_i$
  - Hannah$_i$ found a picture of herself$_i$
  - ?*Hannah found Peter$_i$'s picture of him$_i$
  - Hannah found Peter$_i$'s picture of himself$_i$
  - Hannah$_i$ found Peter's picture of her$_i$
  - Hannah$_i$ found Peter's picture of herself$_i$
  - *etc.*

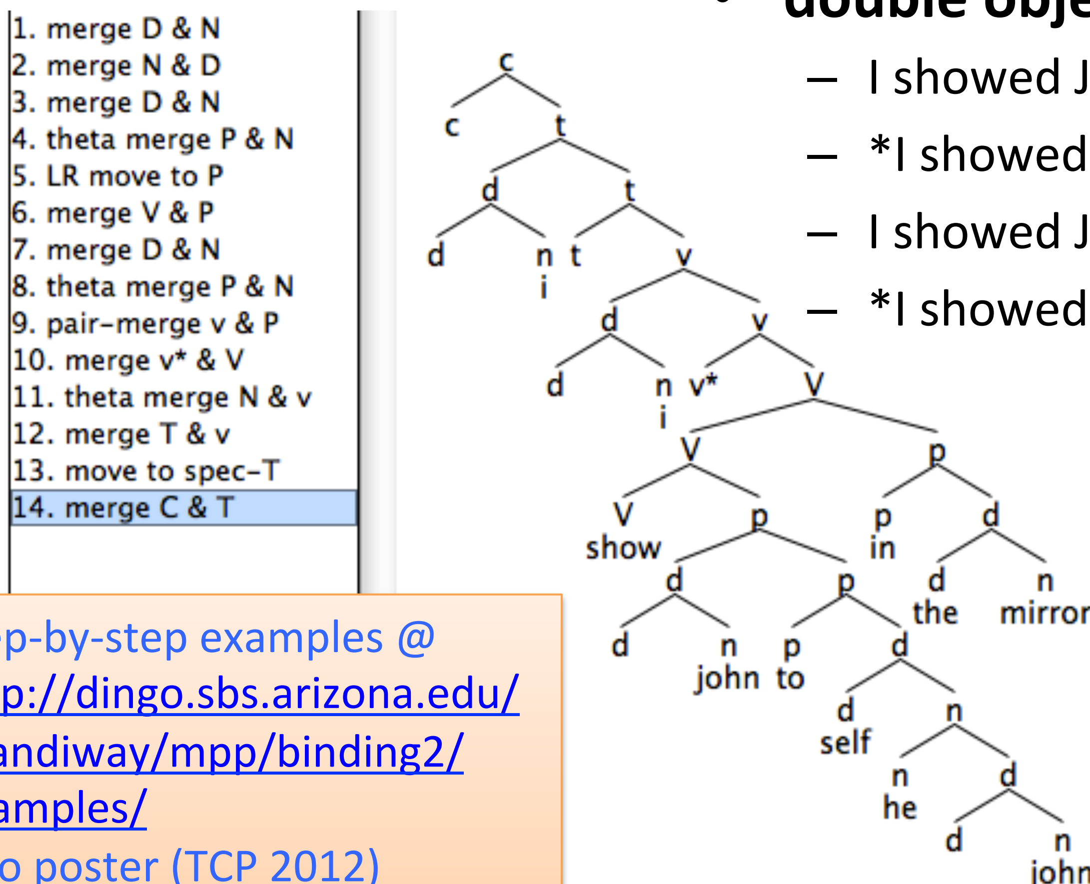
assume the DC for anaphors is different to begin with
[$_D$ [$_D$ self][$_N$ he John]]



step-by-step examples @ http://dingo.sbs.arizona.edu/~sandiway/mpp/binding/examples/
also paper

More Data

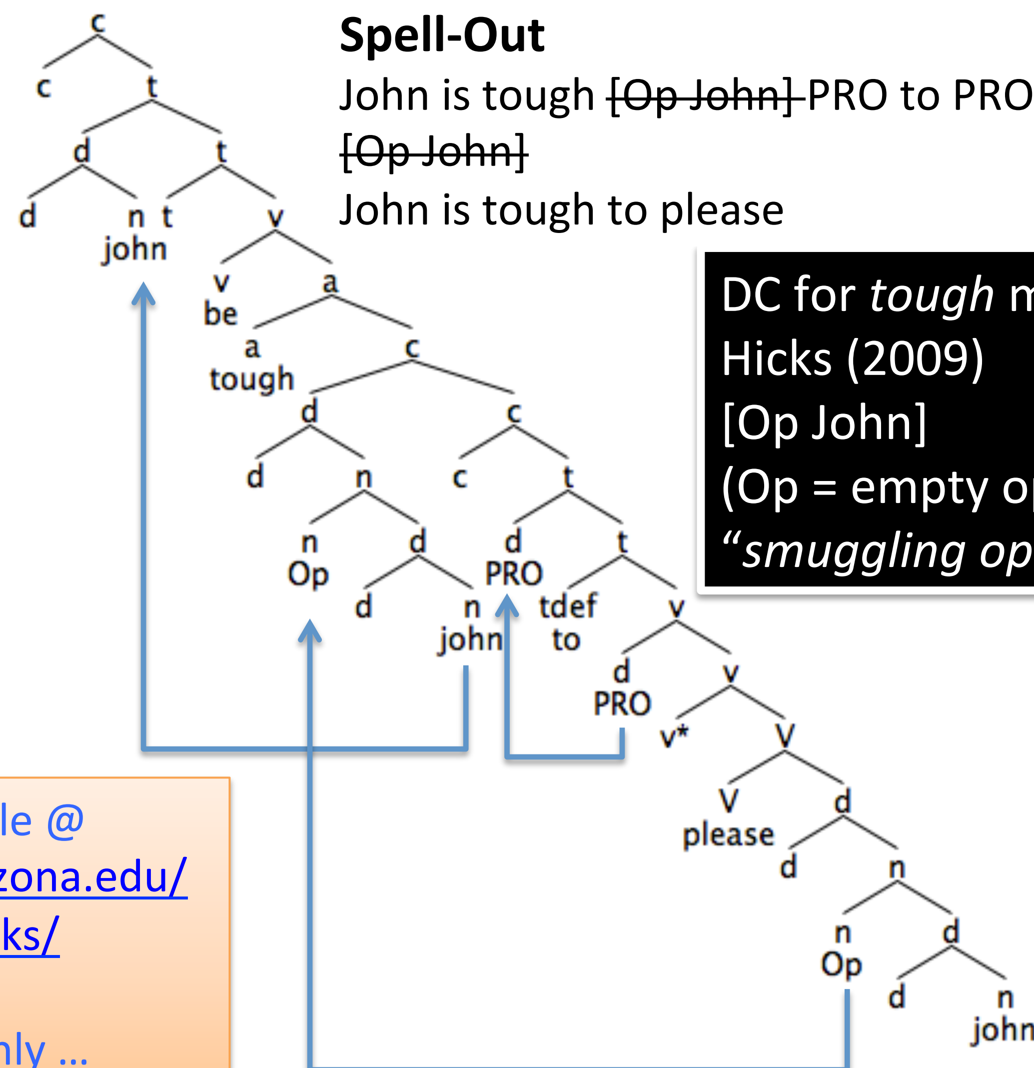
- **double objects**
  - I showed John to himself in the mirror
  - *I showed himself to John in the mirror
  - I showed John himself in the mirror
  - *I showed himself John in the mirror



step-by-step examples @ http://dingo.sbs.arizona.edu/~sandiway/mpp/binding2/examples/
also poster (TCP 2012) http://dingo.sbs.arizona.edu/~sandiway/mpp/TCP2012.pdf

Even More Data

1. merge D & N
2. theta merge N & D
3. merge D & N
4. theta merge V & N
5. merge v* & V
6. theta merge N & v
7. merge T & v
8. move to spec-T
9. merge C & T
10. move to spec-C
11. merge A & C
12. merge v & A
13. merge T & v
14. move to spec-T
15. merge C & T



Spell-Out

John is tough ~~[Op John]~~ PRO to PRO please ~~[Op John]~~

John is tough to please

DC for *tough* movement
Hicks (2009)
[Op John]
(Op = empty operator)
"smuggling operation"

step-by-step example @ http://dingo.sbs.arizona.edu/~sandiway/mpp/hicks/examples/
tentative analysis only …

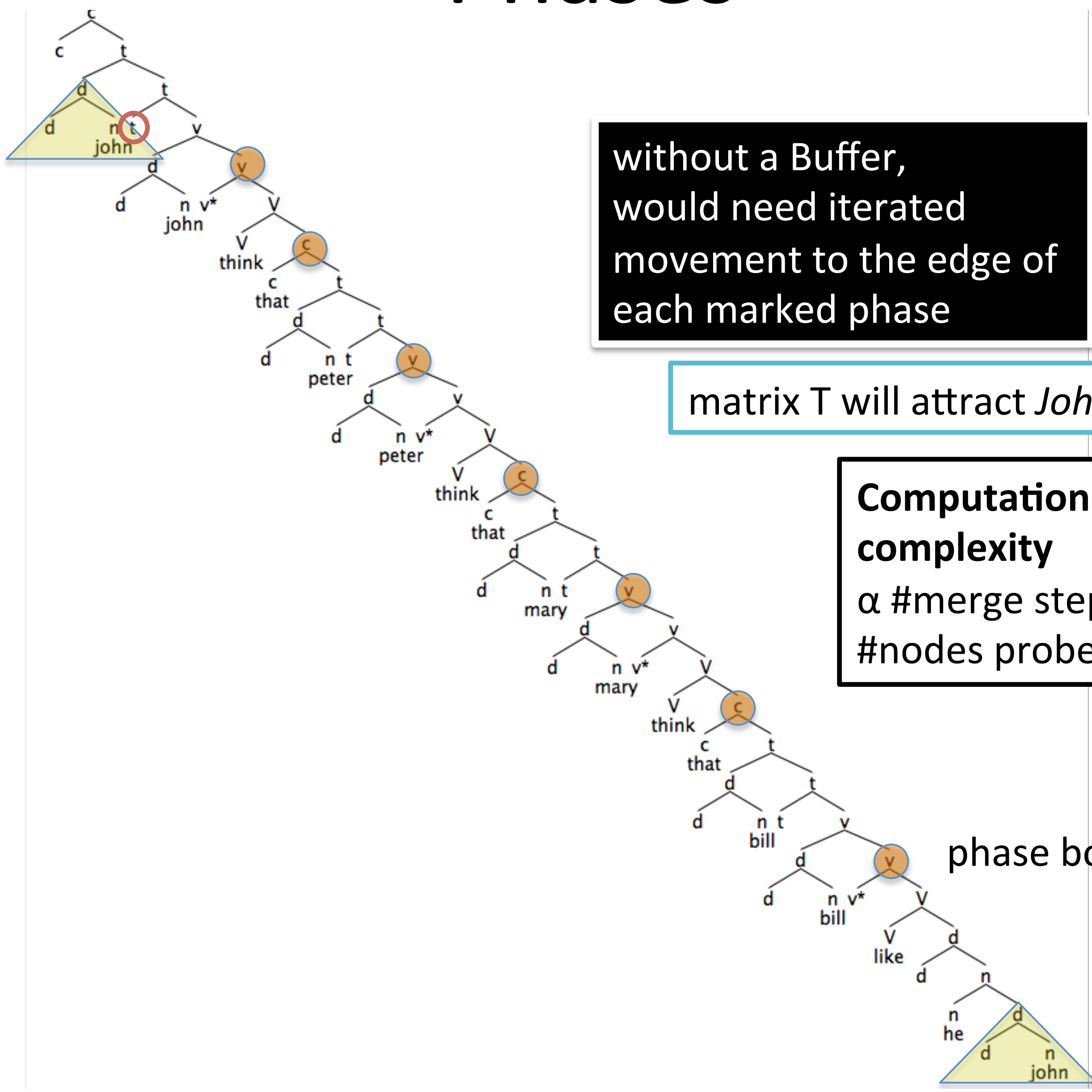
Computational Complexity

Phases

- Long distance:
  - **John**$_i$ thinks that Peter$_{*i}$ thinks that Mary thinks that Bill$_{*i}$ likes **him**$_i$
- Cost of locality:
  - probe-goal search is local
  - iterated movement to the edge of a Phase

coreference possibilities:
[he John] preferred over [he Peter]

Phases



without a Buffer,
would need iterated
movement to the edge of
each marked phase

matrix T will attract *John*

Computational complexity
α #merge steps + β #nodes probed + …

phase boundary

Computational Complexity

Phases

- Long distance:
  - **John**$_i$ thinks that Peter$_{*i}$ thinks that Mary thinks that Bill$_{*i}$ likes **him**$_i$

coreference possibilities:
[he John] preferred over [he Peter]

- Cost of locality:
  - probe-goal search is local
  - iterated movement to the edge of a Phase

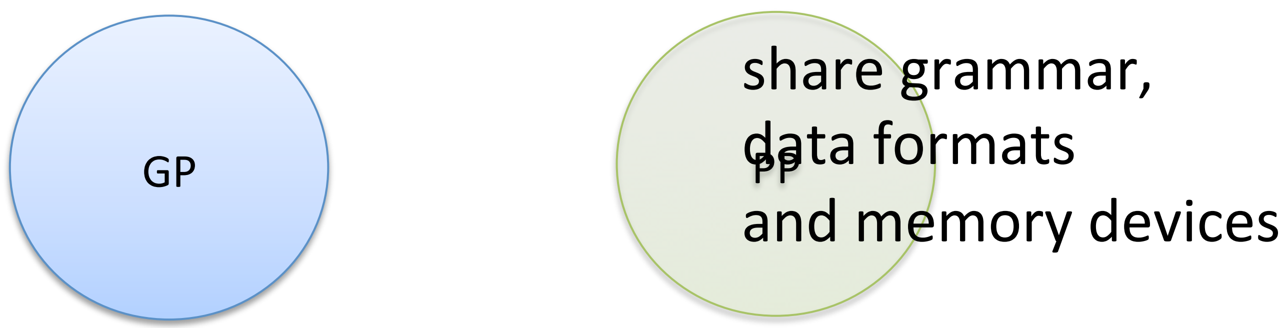
comes with possible complications:
edge feature management
order of goals: “tuck in”

- **Alternate implementation**
  - *eliminate iterated movement of this sort* (smaller trees)
  - use a **Buffer** (for theta Merge)
  - no extra ambiguity
    - **Preference**: external Merge < internal Merge < Buffer

where does this Buffer come from?

Generative and Parsing Procedures

- Idea:
  - Generative Procedure (GP) and Parsing Procedure (PP) are not separate devices

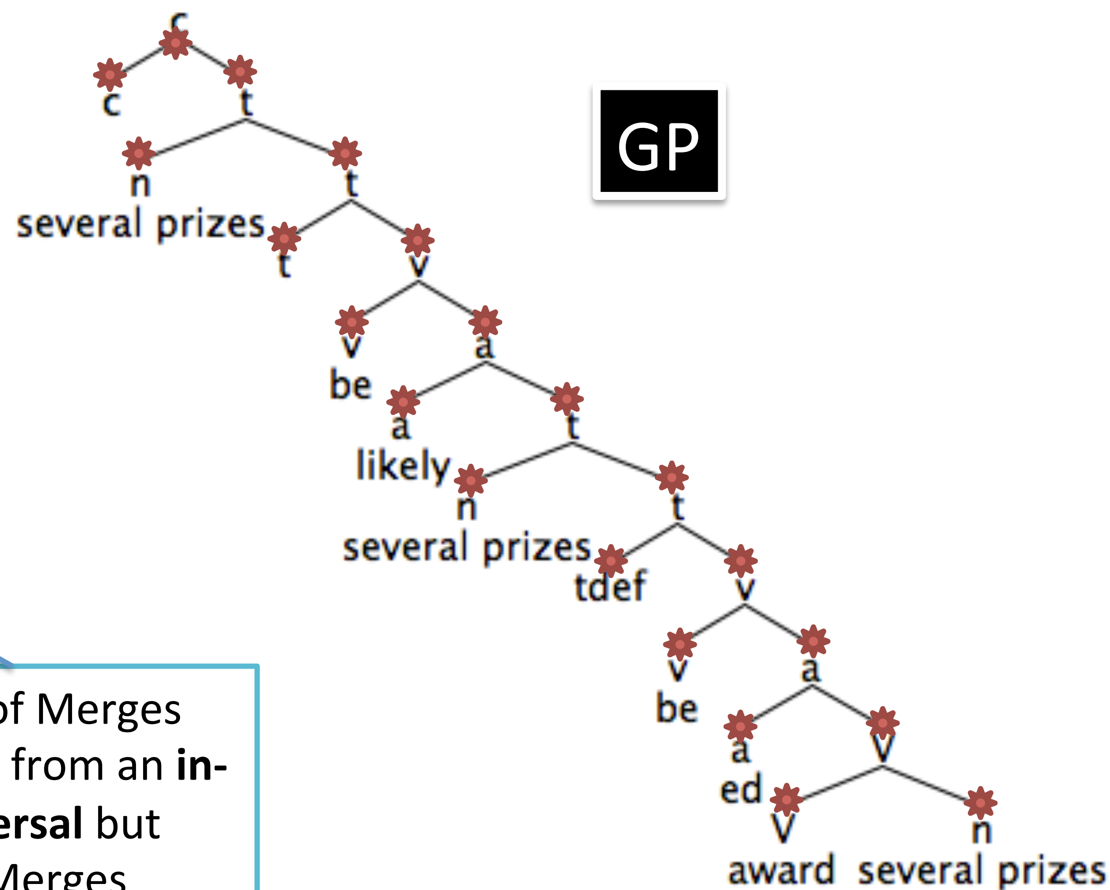


  - *but GP has priority over PP* (Minimal Computation)

Generative and Parsing Procedures

1. theta merge V & N
2. merge PRT & V
3. merge v & A
4. merge T & v
5. move to spec-T
6. merge A & Tdef
7. merge v & A
8. merge T & v
9. move to spec-T
10. merge C & T

sequence of Merges can be had from an **in-order traversal** but reporting Merges on the way back up…

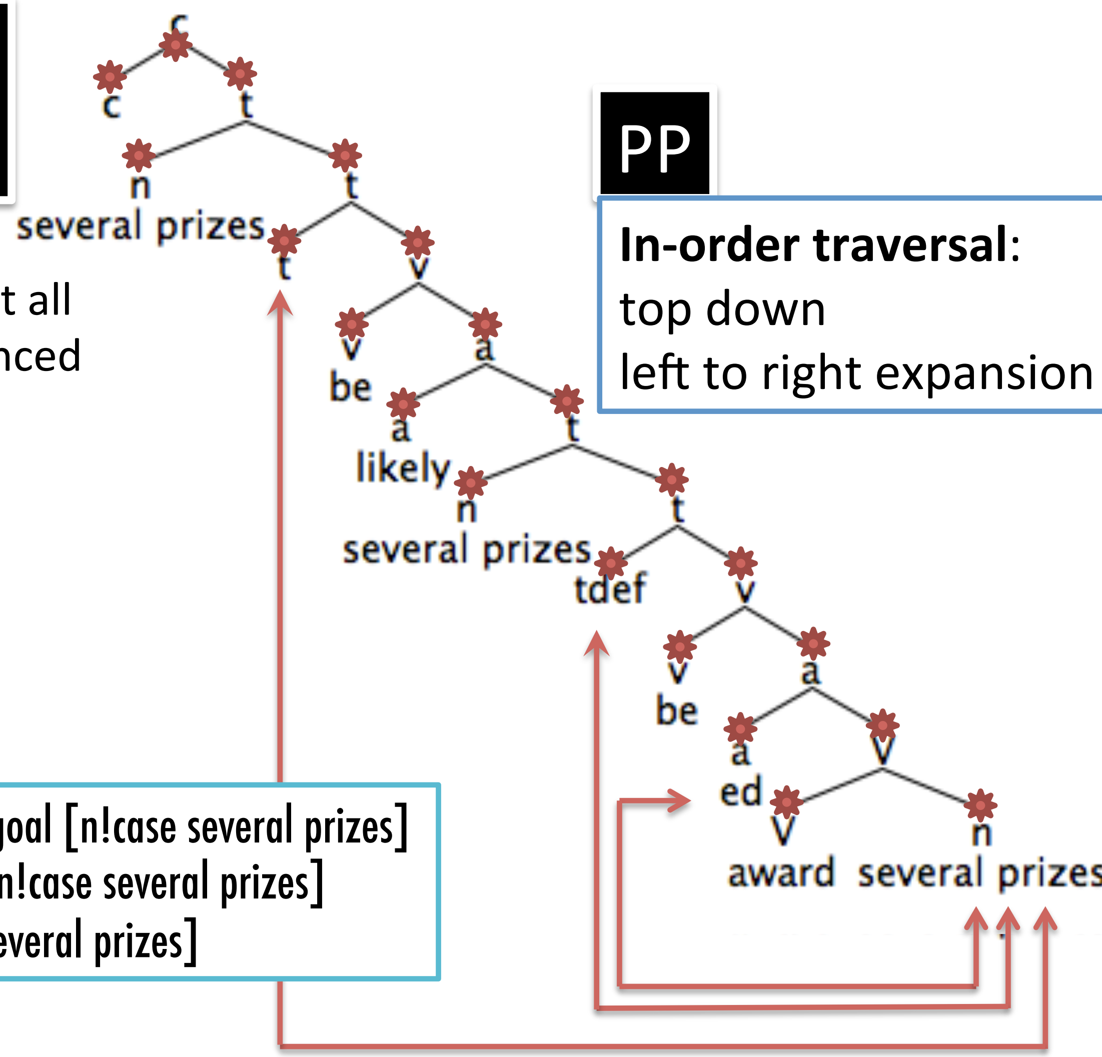


GP

Generative and Parsing Procedures

Two problems peculiar to parsing that must be dealt with

1. displacement without all copies being pronounced
2. empty heads (t,v and others)



PP

In-order traversal:
top down
left to right expansion

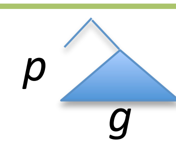
probe [a!case ed] agrees with goal [n!case several prizes]
probe [tdef] agrees with goal [n!case several prizes]
probe [t] agrees with goal [n several prizes]

Parsing Procedure (PP)

- **Implementation summary**
  1. left-to-right, incremental expansion of the covering grammar:
     - no need to recode GP

     same traversal
     GP: report Merges on way back up
     PP: lazy evaluation (*freeze*)

  2. probe-goal: GP, PP *nearly the same*
     - **timing issues**: active/inactive goals

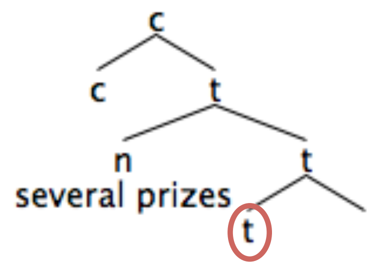
     c-command domain

  3. displacement: must put copies in gap positions
     - need some memory device: Buffer

       GP: Buffer = internal Merge
     - potential ambiguity (fill from input or Buffer)
  4. empty heads: e.g T/T_{def}, v/v*
     - PP (ambiguity), GP (no ambiguity, part of numeration)

Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]



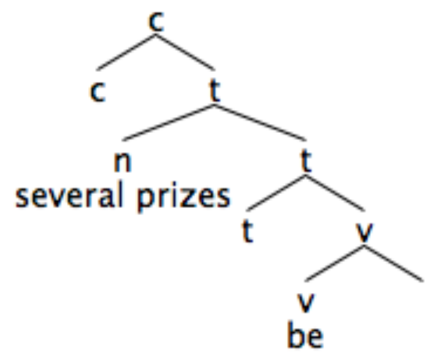
Buffer [[n!case several prizes]]

At this point we know there is a copy of *several prizes* downstream
but where exactly.. we don't yet have enough information

Note:
introduced probe T
its c-command domain doesn't exist yet…
freeze probe-goal search

Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]

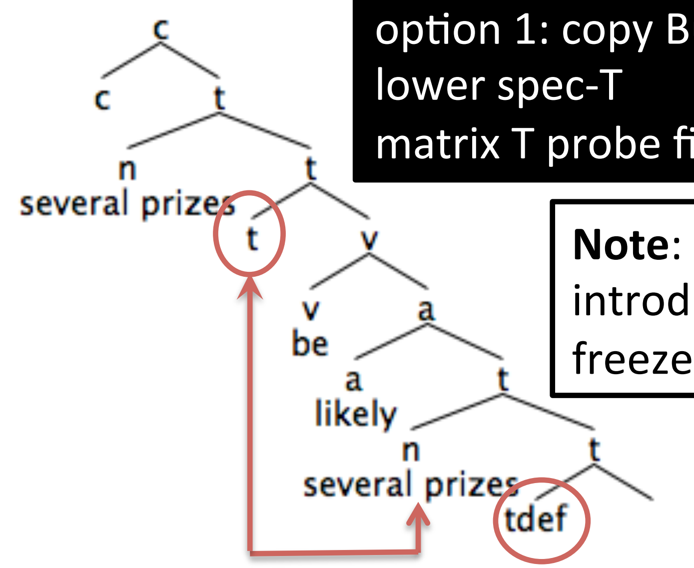


Buffer [[n!case several prizes]]

Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]

option 1: copy Buffer into open
lower spec-T
matrix T probe finds its goal

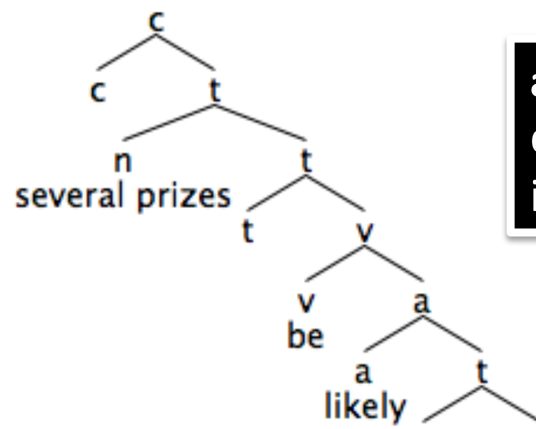


Probe [t] agrees with goal [n several prizes]
Buffer [[n several prizes]]

Note:
introduced probe T_{def}
freeze probe-goal search

Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]

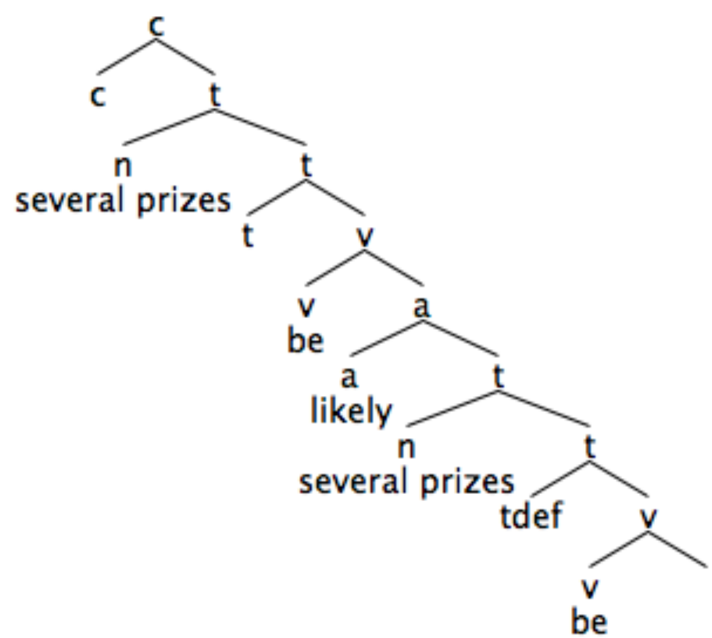


Buffer [[n!case several prizes]]

alternate choice:
don't drop *several prizes* into subject position

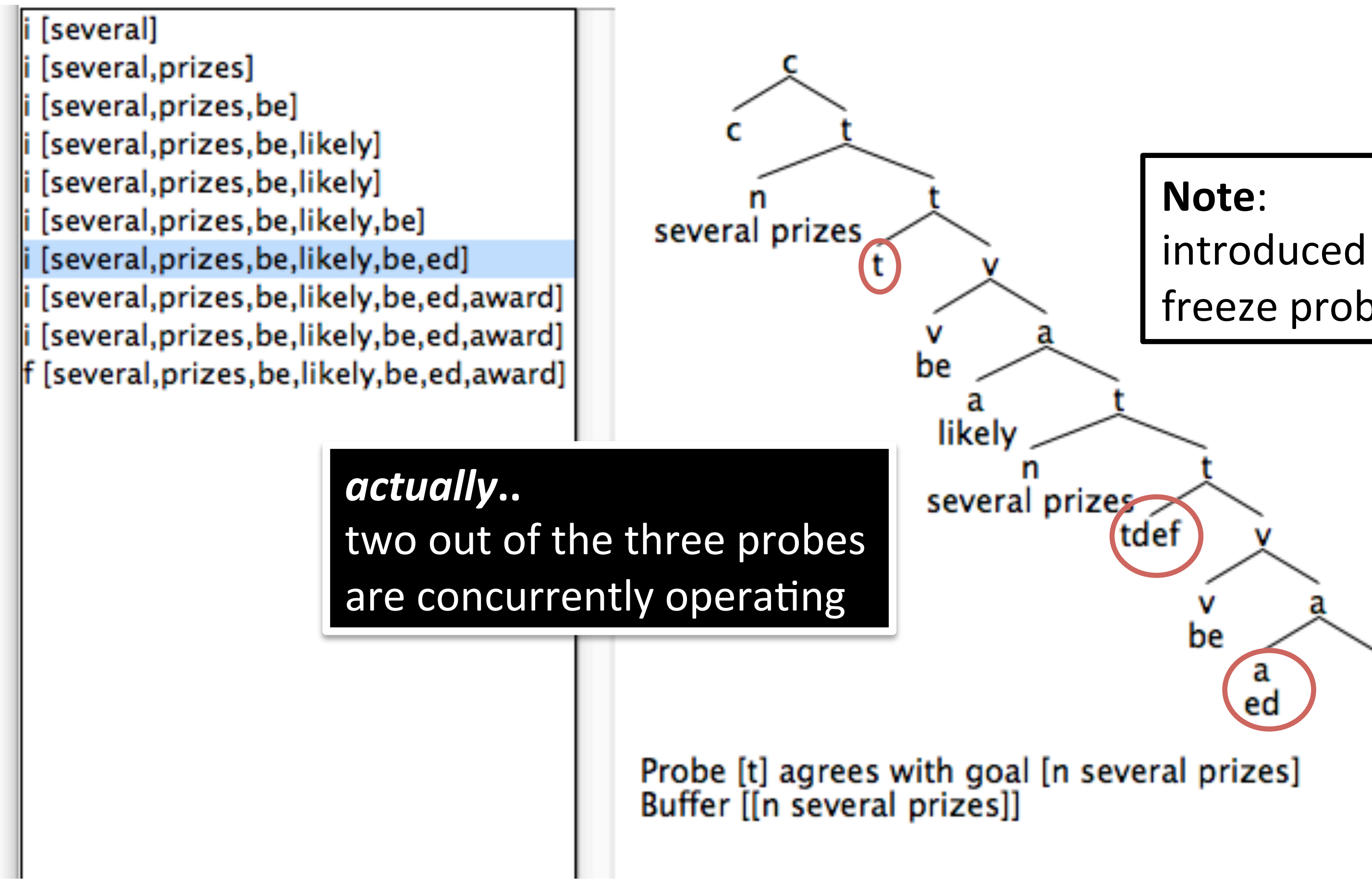
Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]



Probe [t] agrees with goal [n several prizes]
Buffer [[n several prizes]]

Several prizes are likely to be awarded



***actually*..**
two out of the three probes
are concurrently operating

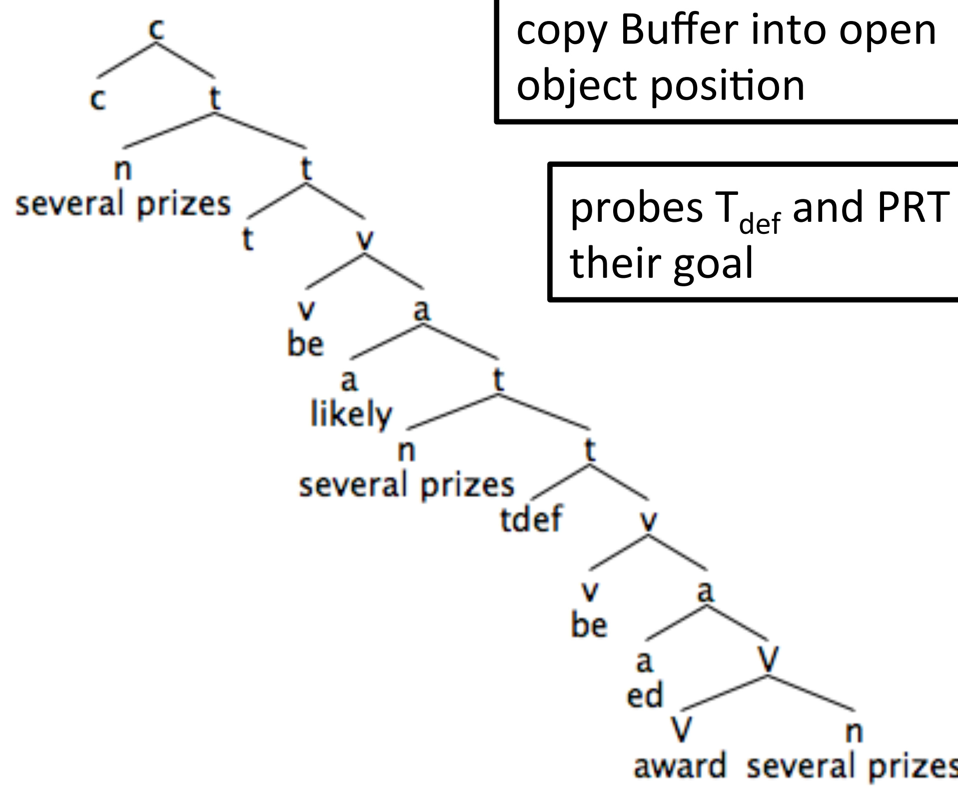
Note:
introduced probe PRT –*ed*
freeze probe-goal search

Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]

copy Buffer into open object position

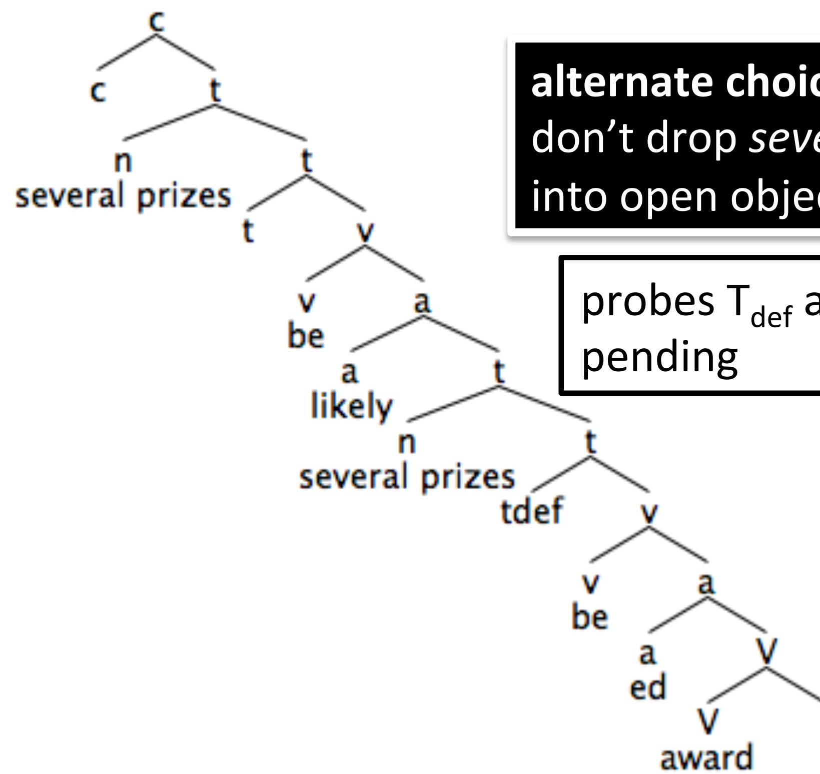
probes T_{def} and PRT find their goal



Probe [t] agrees with goal [n several prizes]
Probe [tdef] agrees with goal [n several prizes]
Probe [a ed] agrees with goal [n several prizes]

Several prizes are likely to be awarded

i [several]
i [several,prizes]
i [several,prizes,be]
i [several,prizes,be,likely]
i [several,prizes,be,likely]
i [several,prizes,be,likely,be]
i [several,prizes,be,likely,be,ed]
i [several,prizes,be,likely,be,ed,award]
i [several,prizes,be,likely,be,ed,award]
f [several,prizes,be,likely,be,ed,award]

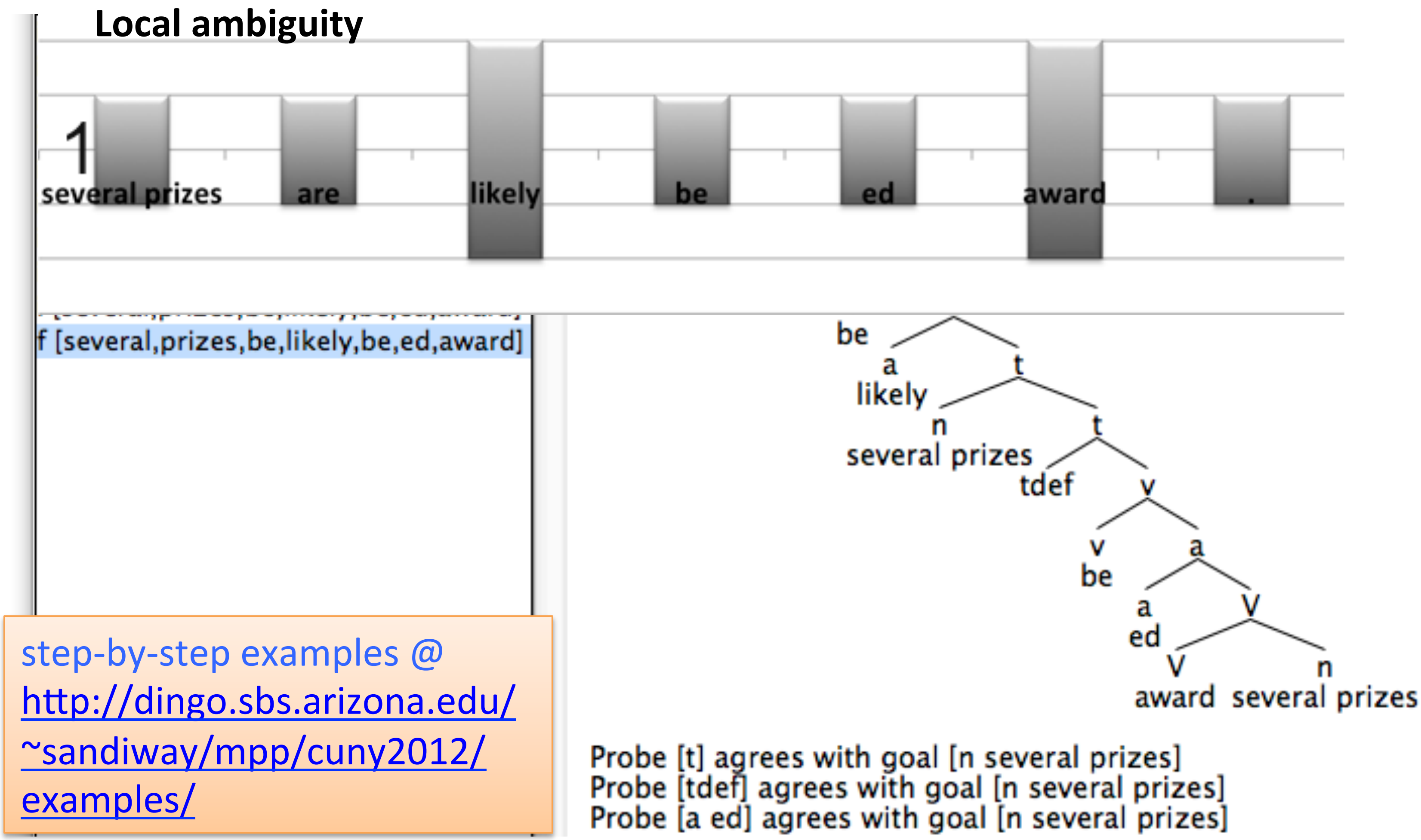


alternate choice:
don't drop *several prizes* into open object position

probes T_{def} and PRT still pending

Probe [t] agrees with goal [n several prizes]
Buffer [[n several prizes]]

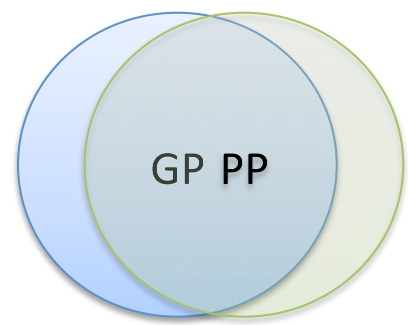
Several prizes are likely to be awarded



step-by-step examples @
http://dingo.sbs.arizona.edu/~sandiway/mpp/cuny2012/examples/

Summary

- **theory**: Chomsky (2001+ …)
- **implement**: merge, agreement, probe-goal search, phases
- GP and PP share architecture:
  - traverse grammar in same way
  - same probe-goal algorithm
  - MC: we need a memory device to handle displacement
  - PP buffer shared

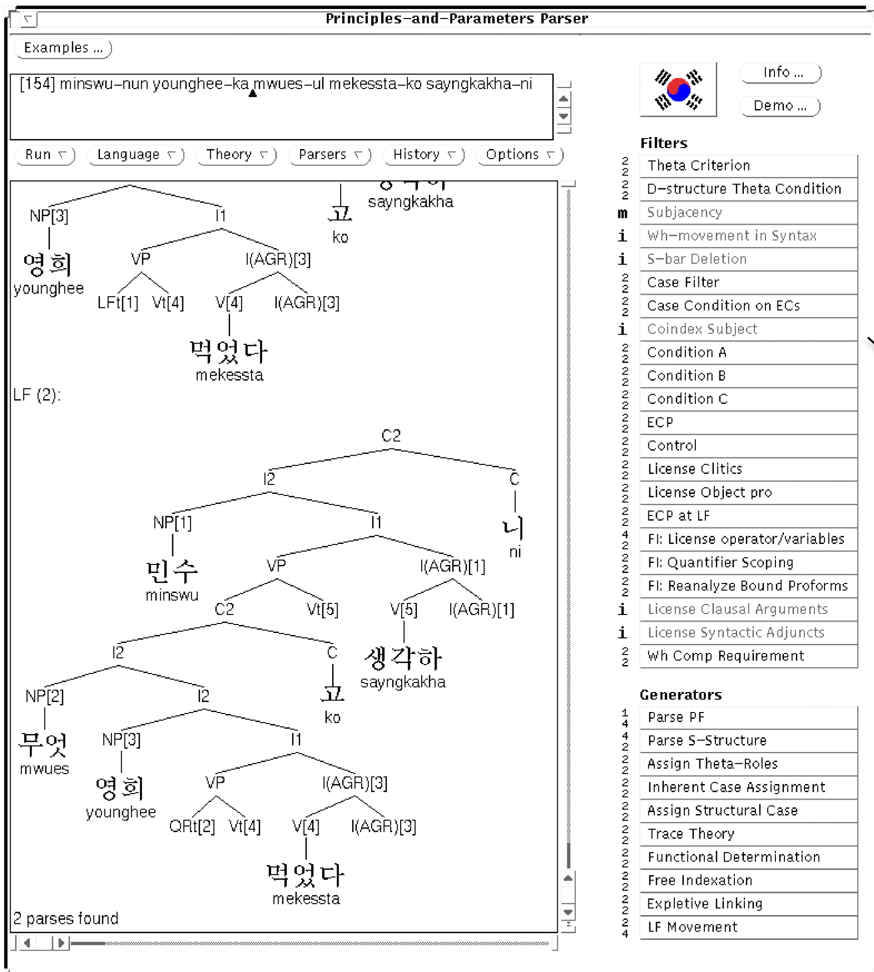


GP: buffer used in DC analysis for pronoun binding

improves GP from the standpoint of MC

Appendix

Principles and Parameters Approach



parser operations corresponding to linguistic principles

PAPPI: Fong (1990)

Principles and Parameters Approach

Generate and test:

33 candidate parses examined by the parser

most are eliminated early by Case/Theta theory sub-system