Parallel Computation in a Free Merge World

Sandiway Fong
University of Arizona
with Jason Ginsburg
Osaka Kyoiku University

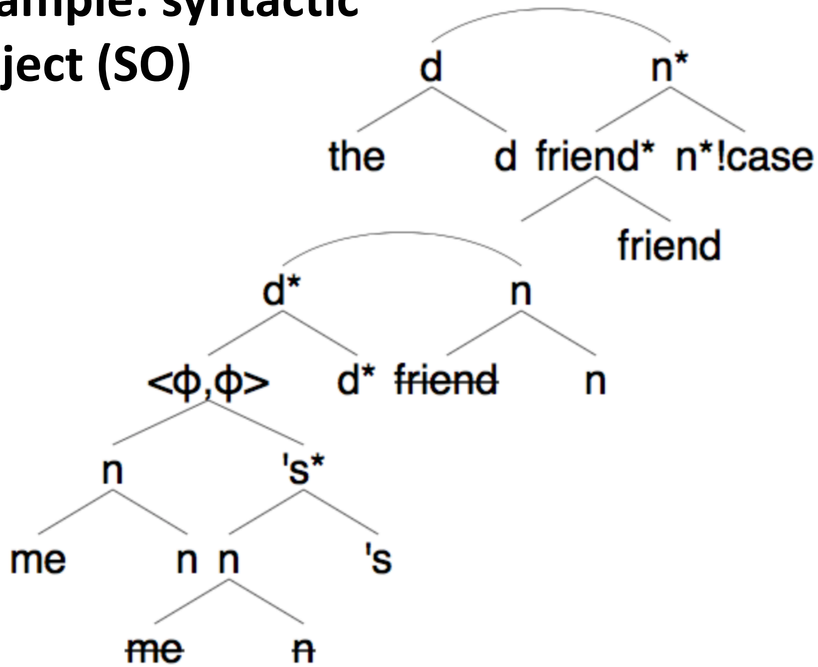
Acknowledgement: Dr. Nobuyoshi Asai, U. of Aizu, Japan for the test platform

Contents

Linguistic Framework and Combinatorics

Phrase Structure Computation

Example: syntactic object (SO)



the friend of mine

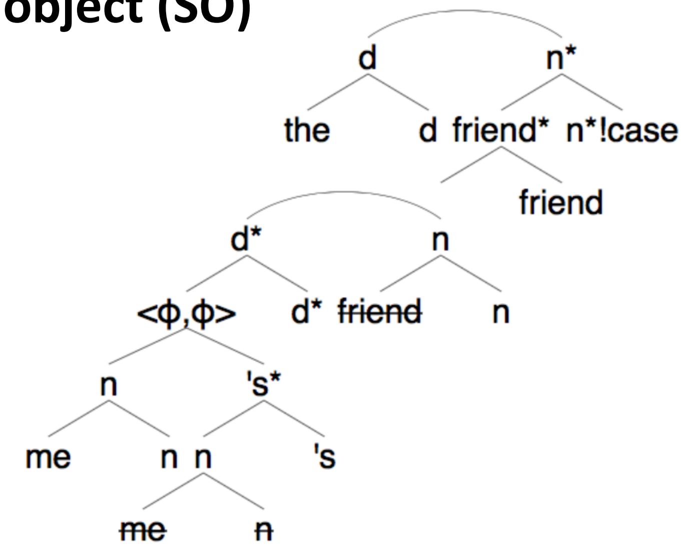
[Joint work with Jason Ginsburg]
Theoretical basis:

- Chomsky (2007) and Oishi (2015)
- nominal and determiner phrase structure (n*/d*-root) parallels verbal phrase structure (v*-root)

- Pair Merge (PM) analysis:
  - <Determiner, Noun>
    - forced by non-head determiner
  - cf. *{{d, root}, {n, root}}
    - (unlabeled Set Merge (SM))
- Relabeling: Cecchetto & Donati (2015)
  - my friend = {me, {‘s, friend}}
  - friend of mine = {**friend**, {me, {‘s, ~~friend~~}}}

Phrase Structure Computation

Example: syntactic object (SO)



the friend of mine

- Example input: (a list of heads):
  [friend, n, [me, n,'s, d*], n*,[the, d]]
- Combinatorial Task: recursively apply operations:
  1. External Set Merge (ESM): form $\{H, \alpha\}$; SO: α, Input: [H,..]
  2. Internal Set Merge (ISM(β)): form $\{\beta, \alpha\}$; SO: α and $\beta \subset^{+} \alpha$
  3. External Pair Merge (EPM): form $\langle H, \alpha \rangle$; SO: α, Input: [H,..]
  4. Internal Pair Merge (IPM): form $\langle \alpha, \beta \rangle$; SO: α and $\beta \subset^{+} \alpha$

  with constraints such as:
  1. $*\langle \beta[!F], \alpha \rangle$ where !F = unvalued feature F
  2. $*\mathrm{ISM}(\beta_i)\ \mathrm{ISM}(\beta_i)$; i.e. can't ISM same β_i twice, etc.
- ☜Example output:
  <{the, d}, {{friend, <{{{me, n}, {{me, n}, 's}}, d*}, {friend, n}>}, n*}>
- Example of questions answered by computation:

A. is this the shortest derivation? **YES**

B. are there other possible derivations? **YES**, only longer ones…

Manually Guided Derivation…

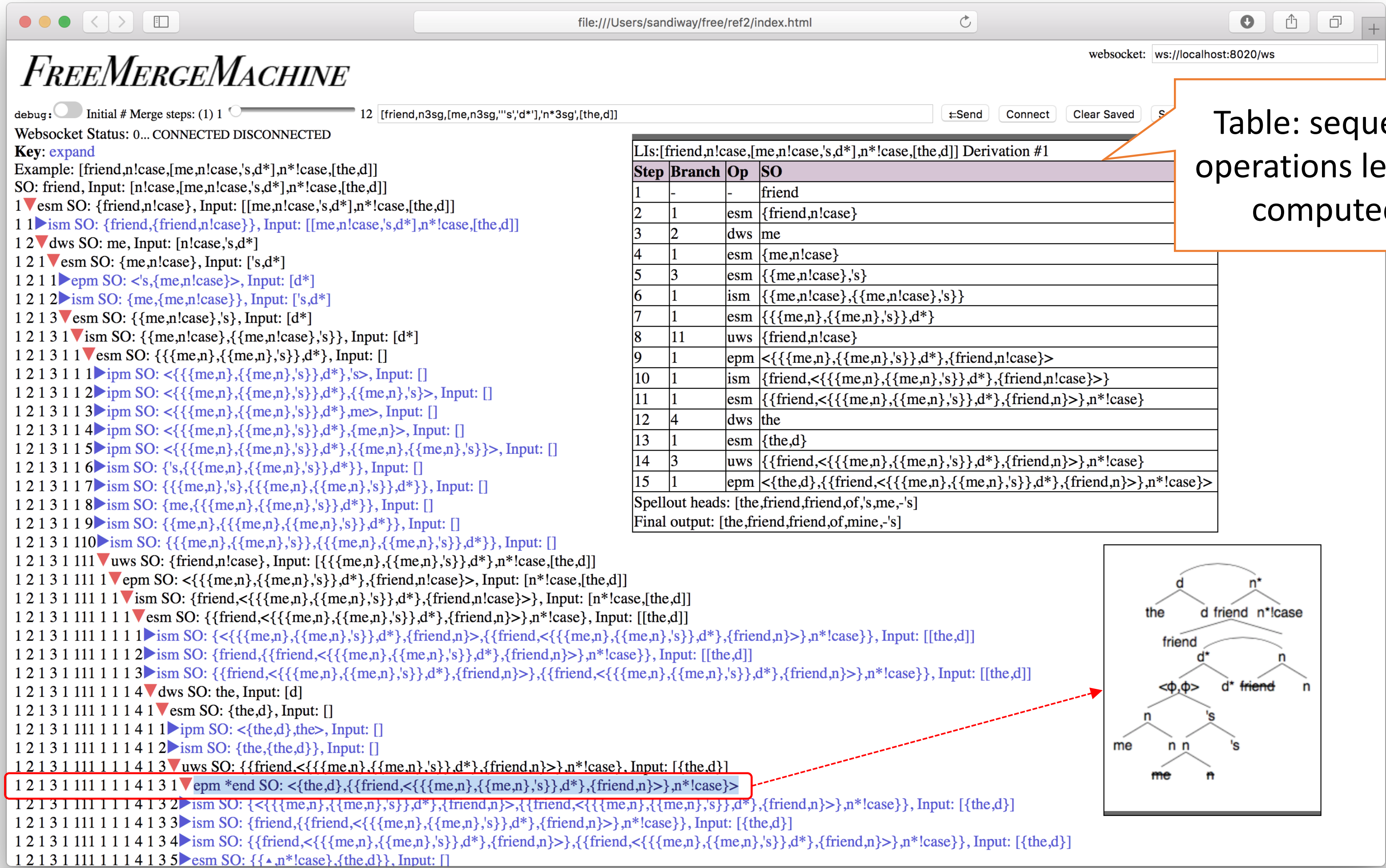
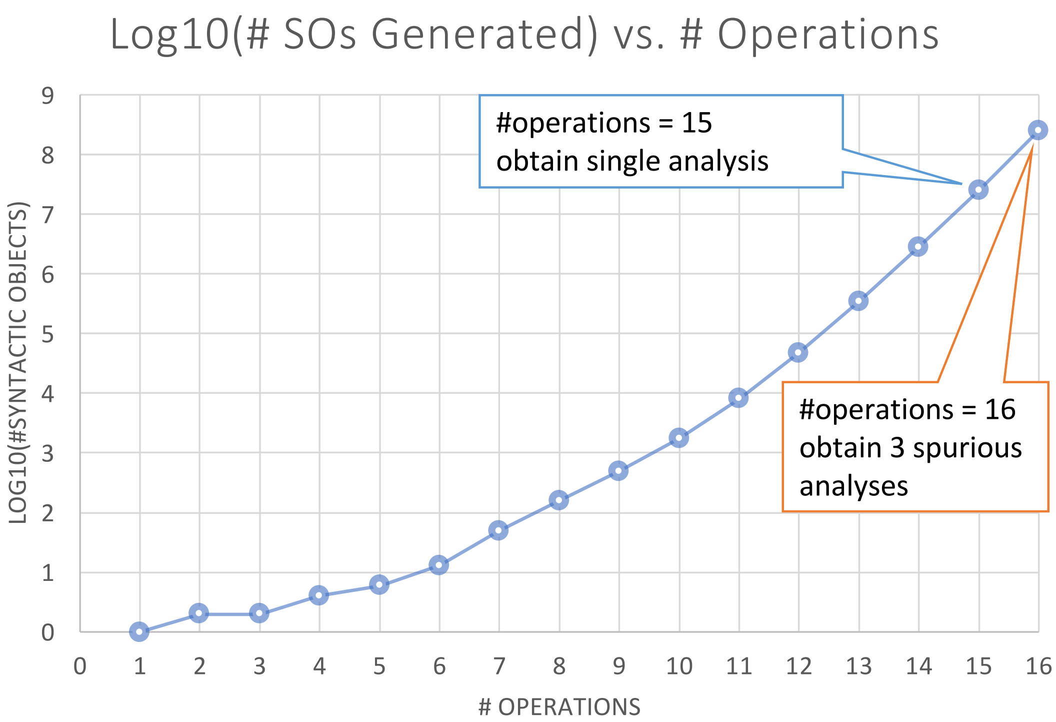


Table: sequence of operations leading to computed SO

Combinatorics for example†

† naïve version

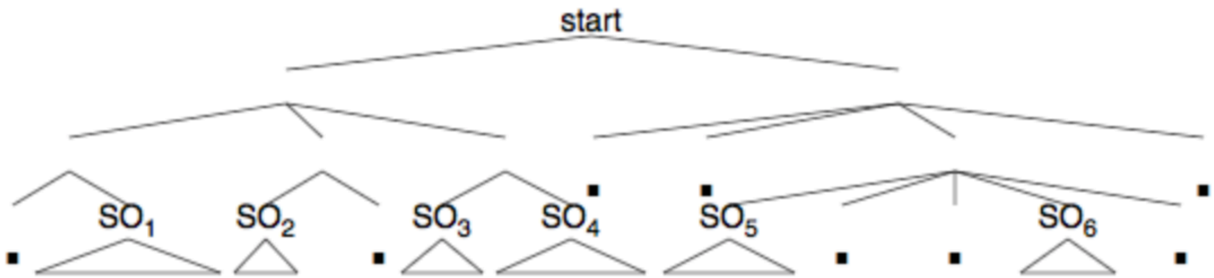


- **logscale** y-axis:
  - e.g. 6 = 10^6 = million
  - 15 operations deep:
    - 25 million SOs generated
    - 1 convergent SO
      - (see previous slide)
  - 16 operations deep:
    - 250 million SOs
    - 3 spurious SOs
      - (see next slides)

Parallelizing the Framework and Results

Two stages of parallel processing

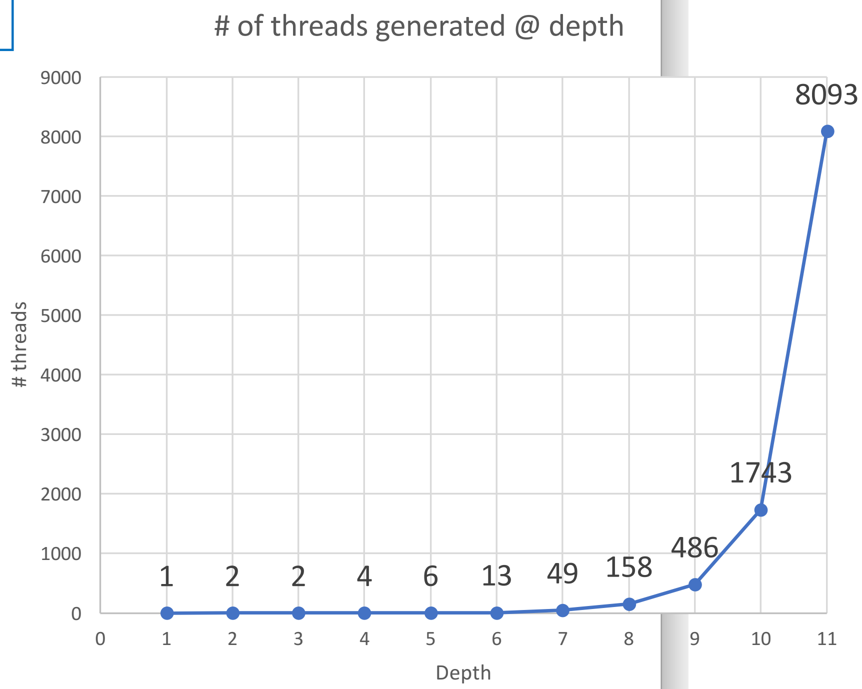
Example:



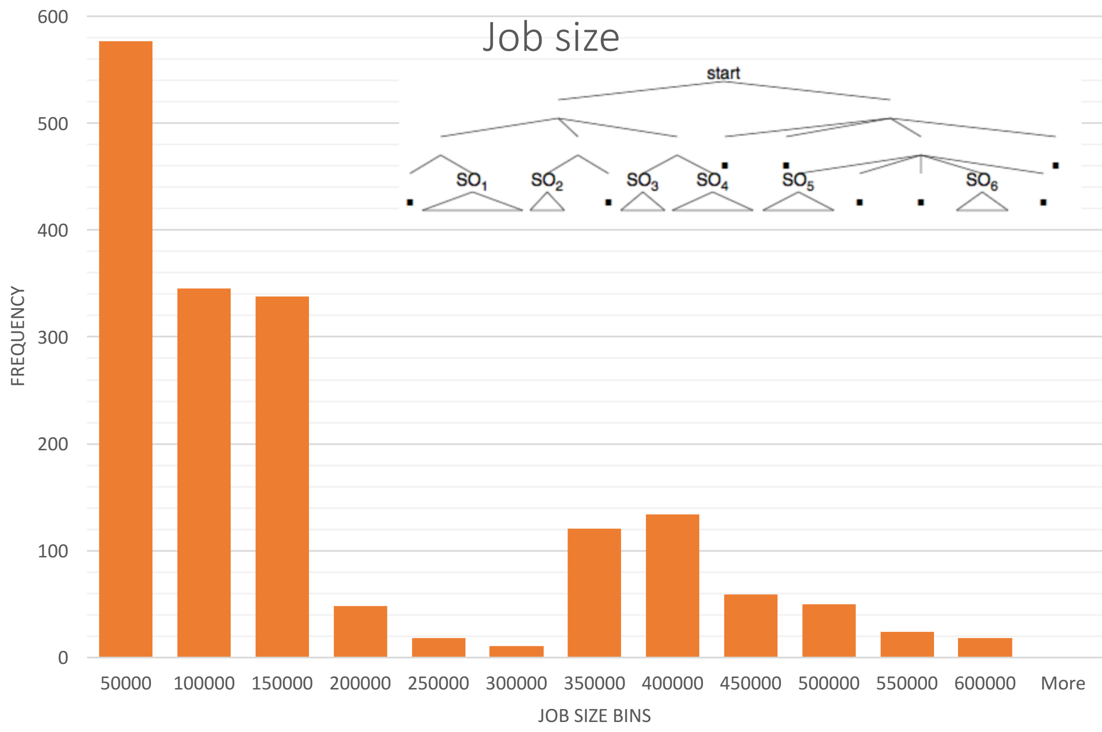
- Stage 1: breadth-first derivation tree search (BFS):
  - $SO_1..SO_6$ are incomplete SOs that can be expanded further
  - ■ represent dead-ends
  - go as deep as necessary to generate the number of starter SOs needed
  - **example**: going 10 deep nets us 1743 SOs

Step 1: Expose threads

Breadth-first search *n* deep:



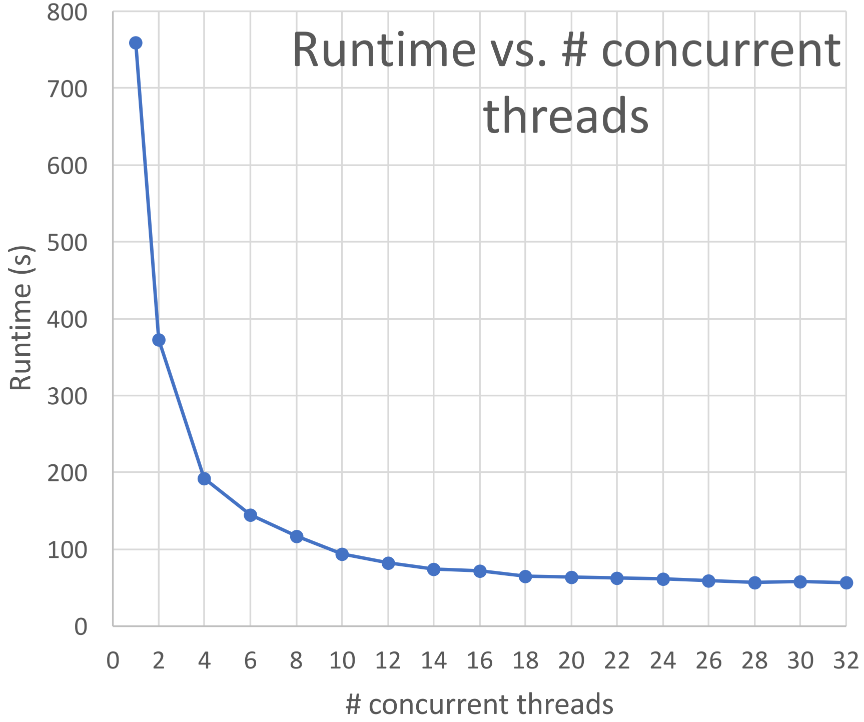
Step 2: Run threads in parallel



our concern: load balancing

- threads binned by job size
- 1743 jobs (threads)
  - produced by initial BFS to 10 operations deep
  - each job (go 6 deep)
- 10x range in job size observed:
  - 50,000 SOs to 600,000 SOs
- 72% of jobs small:
  - belong to the 3 smallest bins, i.e. 0-150,000 SOs

Parallel Speedup

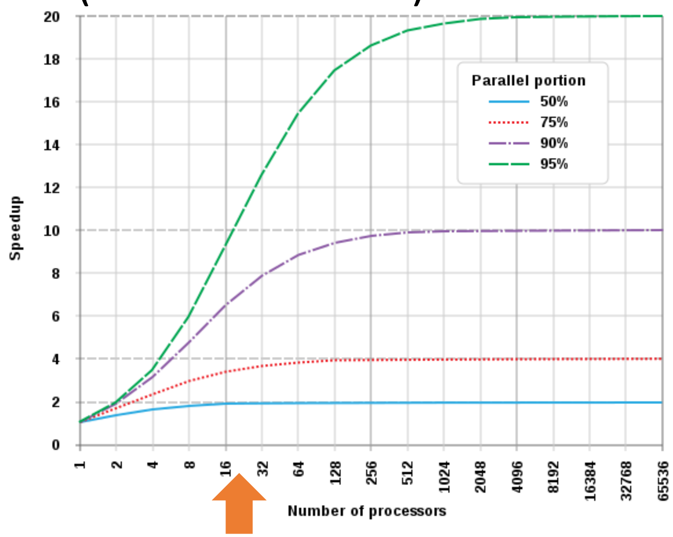


16 CPUs: with HTT, 32 logical cores

- Runtime:
  - Single thread: 759 (secs)
  - 32 threads: 57 (secs)

13x speedup observed

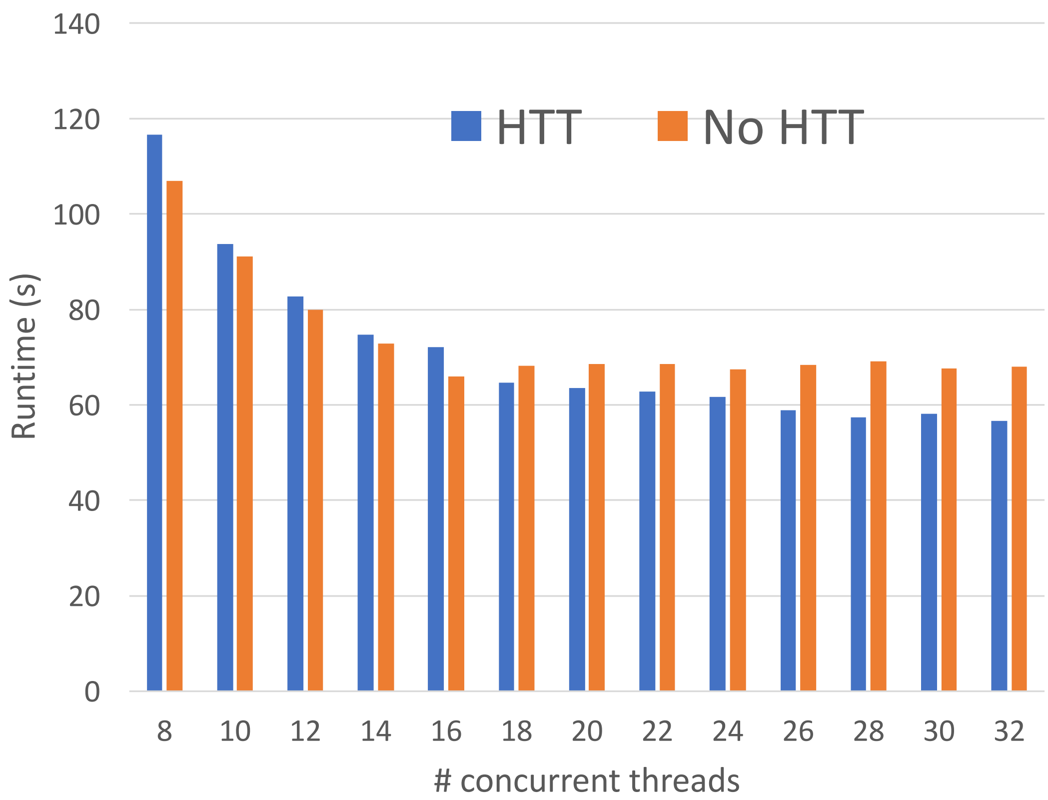
- Speedup: Amdahl's Law
  - (theoretical limit)



Source: wikipedia

we are *somewhere* here

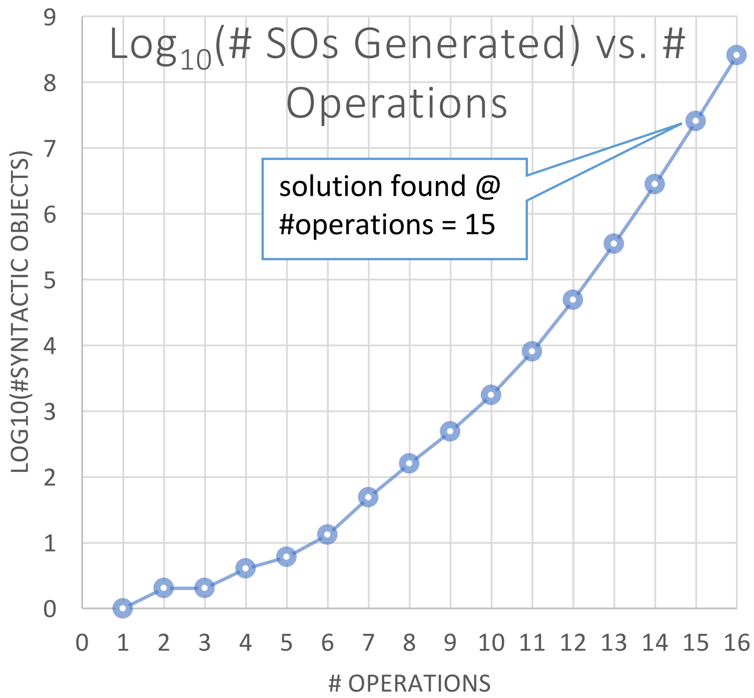
Is Hyper Threading (HTT) useful?



- Hyper Threading Technology:
  - each core has two sets of registers
  - hide memory latency
- Test platform:
  - Intel Xeon E5-2687W HTT-capable (2U), 128GB RAM
  - total of 16 cores (32 logical cores)
- region 8-32 threads:
  - shortest overall runtimes are all achieved by with HTT
- region 18-32 threads:
  - averages about 11.4% improvement over no HTT
- region 8-16 threads:
  - no HTT is 5.5% better

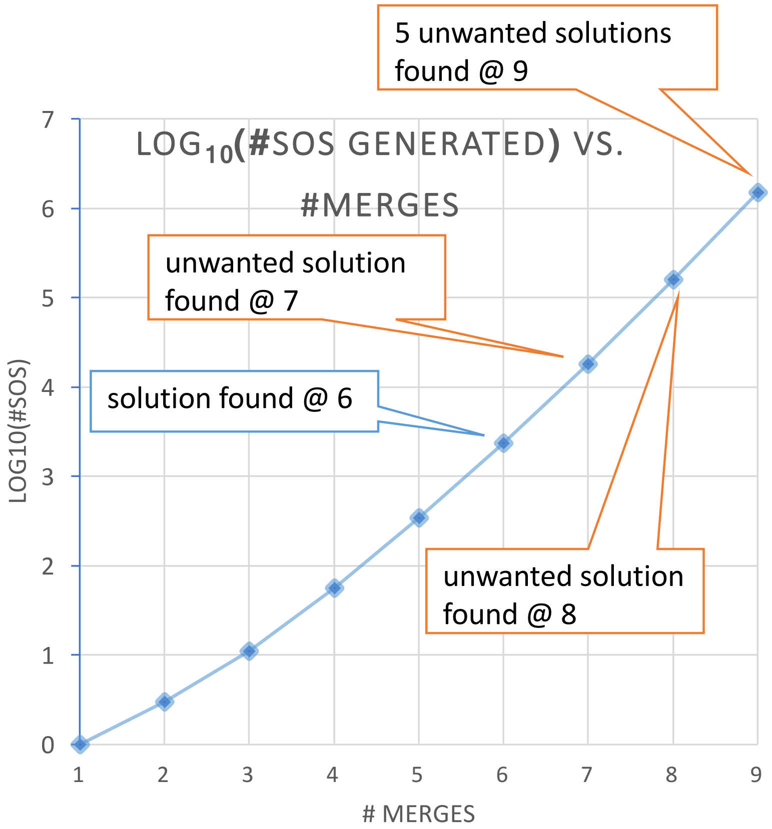
With Workspace (WS) Precomputation

- Results shown earlier, e.g. 57 (secs), were actually computed on a non-naïve model
- region 10^7–10^8 SOs (“**wall**”):
  - too much for the test platform: approx. 4.5 hours CPU time
- Non-naïve model:
  - pre-compute sub-Workspace (WS) SOs
  - # operations required reduced
  - free Merge then is substantially easier
    “walk back from the wall”



Workspace (WS) Precomputation

- **Example**:
  instead of
  [friend, n, **[me, n,'s, d*]**, n*, **[the, d]**]
  actually compute with
  [friend, n, **{{{me, n}, {{me, n}, 's}}, d*}**, n*, **{the, d}**]
  - i.e. use pre-computed mappings:
  1. [me, n,'s, d*] ⟼ {{{me, n}, {{me, n}, 's}}, d*}
  2. [the, d] ⟼ {the, d}
- **Results**:
  - Depth 6: #SOs: 2,324; 1 solution
  - Depth 7: #SOs: 18,202; 2 solutions, *etc*..

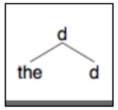


Workspace (WS) Precomputation:

debug: Initial # Merge steps: (2) 1 12 [friend,n3sg,@dP2,'n*3sg',@dP1] ↕Send Connect Clear Saved Save Output

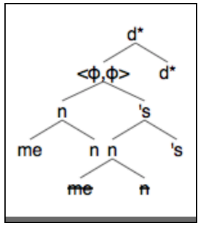
Manually Guided Derivation…

Example (@dP1): [the,d]
SO: the, Input: [d]
1 esm ✓ SO: {the,d}

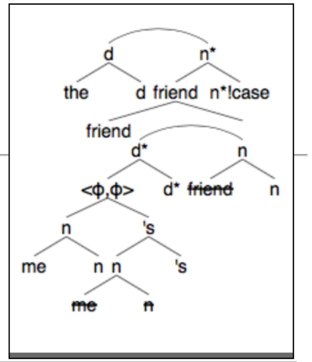


LIs:[the,d] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	the
2	1	esm	{the,d}
Spellout heads: [the]			
Final output: [the]			

Example (@dP2): [me,n!case,'s,d*]
SO: me, Input: [n!case,'s,d*]
1 esm SO: {me,n!case}, Input: ['s,d*]
1 1 epm SO: <'s,{me,n!case}>, Input: [d*]
1 1 1 ism SO: {me,<'s,{me,n!case}>}, Input: [d*]
1 2 ism SO: {me,{me,n!case}}, Input: ['s,d*]
1 3 esm SO: {{me,n!case},'s}, Input: [d*]
1 3 1 ism SO: {{me,n!case},{{me,n!case},'s}}, Input: [d*]
1 3 1 1 esm ✓ SO: {{{me,n},{{me,n},'s}},d*}
1 3 2 esm *unlabeled SO: {{{me,n},'s},d*}, Input: []



LIs:[me,n!case,'s,d*] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	me
2	1	esm	{me,n!case}
3	3	esm	{{me,n!case},'s}
4	1	ism	{{me,n!case},{{me,n!case},'s}}
5	1	esm	{{{me,n},{{me,n},'s}},d*}
Spellout heads: ['s,me]			
Final output: [my]			

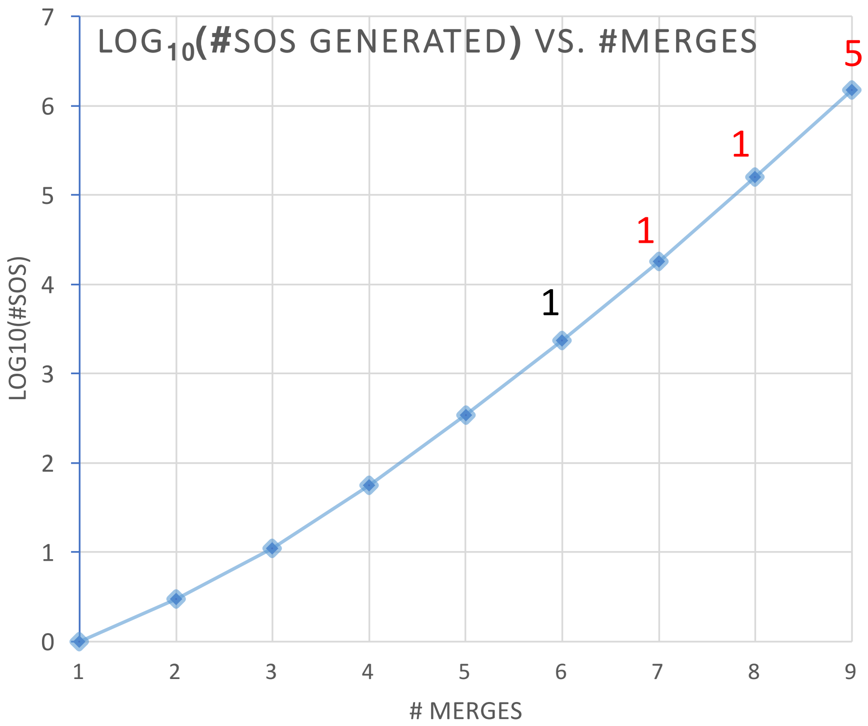


Example: [friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}]
SO: friend, Input: [n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}]
1 esm SO: {friend,n!case}, Input: [{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}]
1 1▼epm SO: <{{{me,n},{{me,n},'s}},d*},{friend,n!case}>, Input: [n*!case,{the,d}]
1 1 1▼ism SO: {friend,<{{{me,n},{{me,n},'s}},d*},{friend,n!case}>}, Input: [n*!case,{the,d}]
1 1 1 1▼esm SO: {{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}, Input: [{the,d}]
1 1 1 1 1▼epm *end SO: <{the,d},{{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}>
1 1 1 1 2▶ism SO: {<{{{me,n},{{me,n},'s}},d*},{friend,n}>,{{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}}, Input: [{the,d}]
1 1 1 1 3▶ism SO: {friend,{{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}}, Input: [{the,d}]
1 1 1 1 4▶ism SO: {{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},{{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}}, Input: [{the,d}]
1 1 1 1 5▶esm SO: {{▴,n*!case},{the,d}}, Input: []
1 2▶ism SO: {friend,{friend,n!case}}, Input: [{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}]
1 3▶esm SO: {{friend,n!case},{{{me,n},{{me,n},'s}},d*}}, Input: [n*!case,{the,d}]

LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	friend
2	1	esm	{friend,n!case}
3	1	epm	<{{{me,n},{{me,n},'s}},d*},{friend,n!case}>
4	1	ism	{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n!case}>}
5	1	esm	{{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}
6	1	epm	<{the,d},{{friend,<{{{me,n},{{me,n},'s}},d*},{friend,n}>},n*!case}>
Spellout heads: [the,friend,of,'s,me]			
Final output: [the,friend,of,mine]			

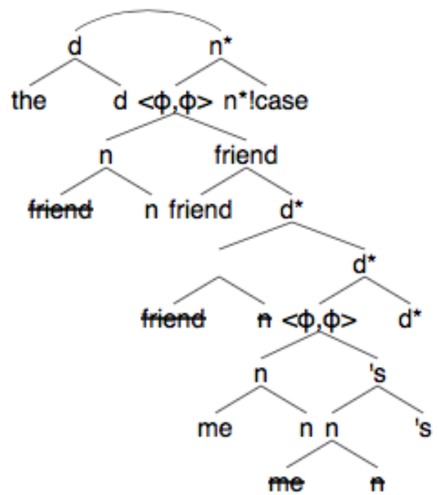
Depth = 6

Improve the Framework



- Parallel processing allows us to discover 5 extra analyzes at depth 9 (out of $\approx 10^6$ SOs) 10x quicker…

Extra Analyses Uncovered: Depth 7 & 8

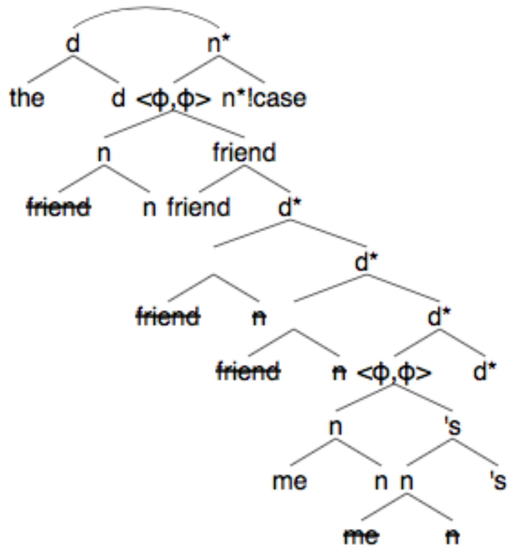


Depth = 7

Analysis:

Extraneous ISM of {~~friend~~,n} to the edge of friend

<ϕ, ϕ> because {~~friend~~,n} and friend have identical ϕ-features



Depth = 8

LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1

Step	Branch	Op	SO
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	7	ism	{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	5	ism	{{friend,n!case},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	6	esm	{{{friend,n},{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}}},n*!case}
7	1	epm	<{the,d},{{{friend,n},{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}}},n*!case}>

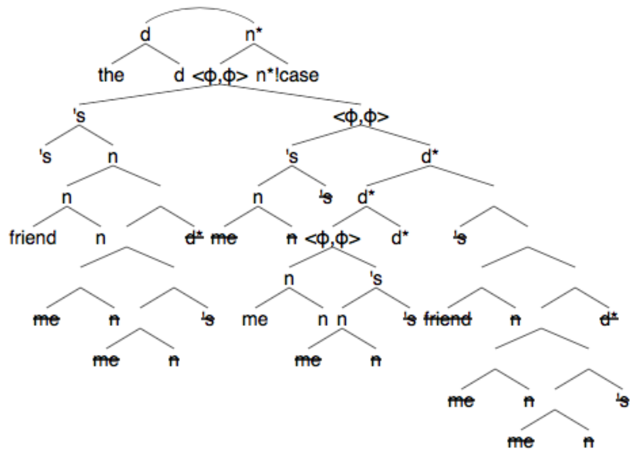
Spellout heads: [the,friend,of,'s,me]
Final output: [the,friend,of,mine]

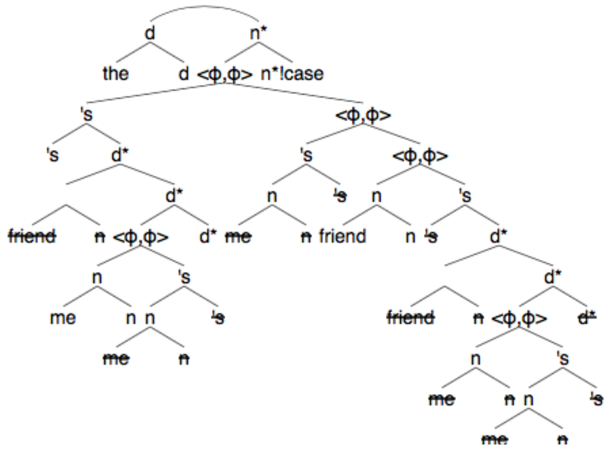
LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1

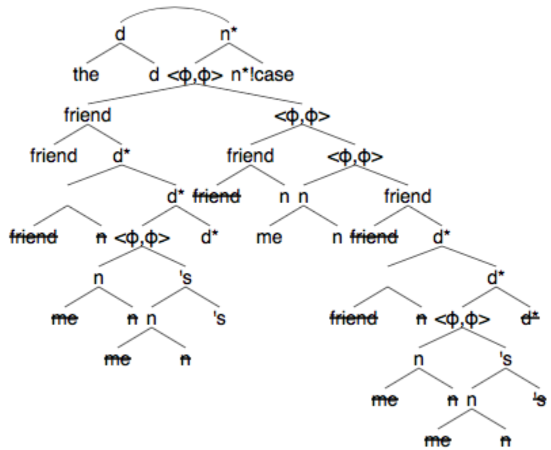
Step	Branch	Op	SO
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	8	ism	{{friend,n!case},{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	7	ism	{friend,{{friend,n!case},{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	6	ism	{{friend,n!case},{friend,{{friend,n!case},{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}
7	6	esm	{{{friend,n},{friend,{{friend,n},{{friend,n},{{{me,n},{{me,n},'s}},d*}}}}},n*!case}
8	1	epm	<{the,d},{{{friend,n},{friend,{{friend,n},{{friend,n},{{{me,n},{{me,n},'s}},d*}}}}},n*!case}>

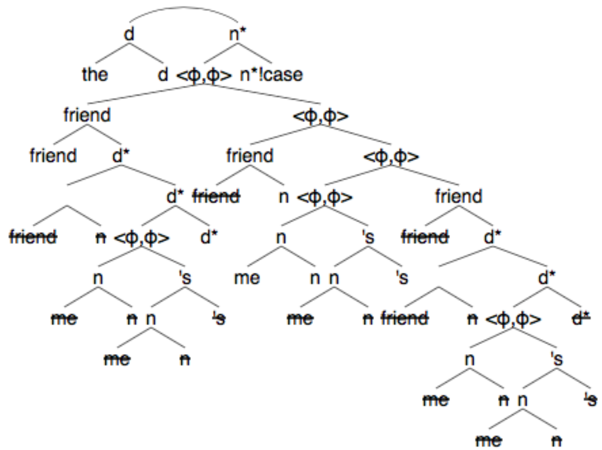
Spellout heads: [the,friend,of,'s,me]
Final output: [the,friend,of,mine]

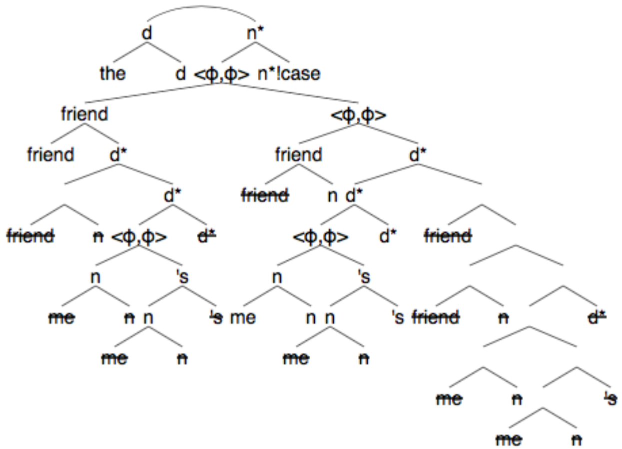
Extra Analyses Uncovered: Depth 9



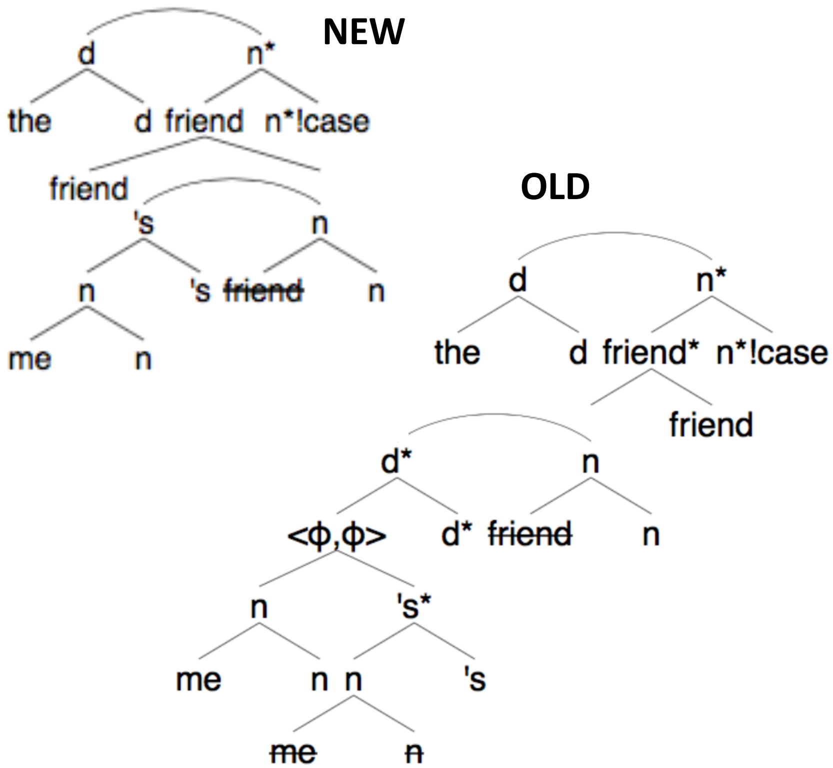








Improve the Framework: Theory Adjustment



[Joint work with Jason Ginsburg]

- block licensing of extraneous analyses
- Previously:
  - all Case valuation done through Agree
  - ‘s analyzed as a pair: root ‘s + d* (categorizer)
- Now:
  - distinguish Inherent from Structural Case
  - Inherent Case does not involve ϕ-features: means <ϕ, ϕ> labeling not available
  - Structural Case involves ϕ-feature valuation, and Nom (or Acc) Case for C/T/ (or v*/R)
  - ‘s analyzed as a single re-categorizing head: i.e. n -> d

Improve the Framework: Theory Adjustment

FreeMergeMachine

debug: Initial # Merge steps: (1) 1 12 [friend,n3sg,@dP2

⇇Send | Connect | Clear Saved | Save Output | Clear Output

Websocket Status: 0... CONNECTED DISCONNECTED

Key: expand

Example: [friend,n!case,{{me,n},'s},n*!case,{the,d}]

SO: friend, Input: [n!case,{{me,n},'s},n*!case,{the,d}]

1▼esm SO: {friend,n!case}, Input: [{{me,n},'s},n*!case,{the,d}]

1 1▼epm SO: <{{me,n},'s},{friend,n!case}>, Input: [n*!case,{the,d}]

1 1 1▼ism SO: {friend,<{{me,n},'s},{friend,n!case}>}, Input: [n*!case,{the,d}]

1 1 1 1▼esm SO: {{friend,<{{me,n},'s},{friend,n}>},n*!case}, Input: [{the,d}]

1 1 1 1 1▼epm *end SO: <{the,d},{{friend,<{{me,n},'s},{friend,n}>},n*!case}>

1 1 1 1 2▶ism SO: {<{{me,n},'s},{friend,n}>,{{friend,<{{me,n},'s},{friend,n}>},n*!case}}, Input: [{the,d}]

1 1 1 1 3▶ism SO: {friend,{{friend,<{{me,n},'s},{friend,n}>},n*!case}}, Input: [{the,d}]

1 1 1 1 4▶ism SO: {{friend,<{{me,n},'s},{friend,n}>},{{friend,<{{me,n},'s},{friend,n}>},n*!case}}, Input: [{the,d}]

1 1 1 1 5▶esm SO: {{▲,n*!case},{the,d}}, Input: []

1 2▶ism SO: {friend,{friend,n!case}}, Input: [{{me,n},'s},n*!case,{the,d}]

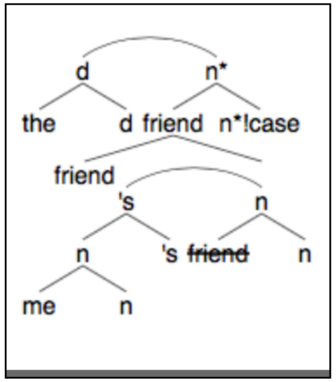
1 3▶esm SO: {{friend,n!case},{{me,n},'s}}, Input: [n*!case,{the,d}]

LIs:[friend,n!case,{{me,n},'s},n*!case,{the,d}] Derivation #1

Step	Branch	Op	SO
1	-	-	friend
2	1	esm	{friend,n!case}
3	1	epm	<{{me,n},'s},{friend,n!case}>
4	1	ism	{friend,<{{me,n},'s},{friend,n!case}>}
5	1	esm	{{friend,<{{me,n},'s},{friend,n}>},n*!case}
6	1	epm	<{the,d},{{friend,<{{me,n},'s},{friend,n}>},n*!case}>

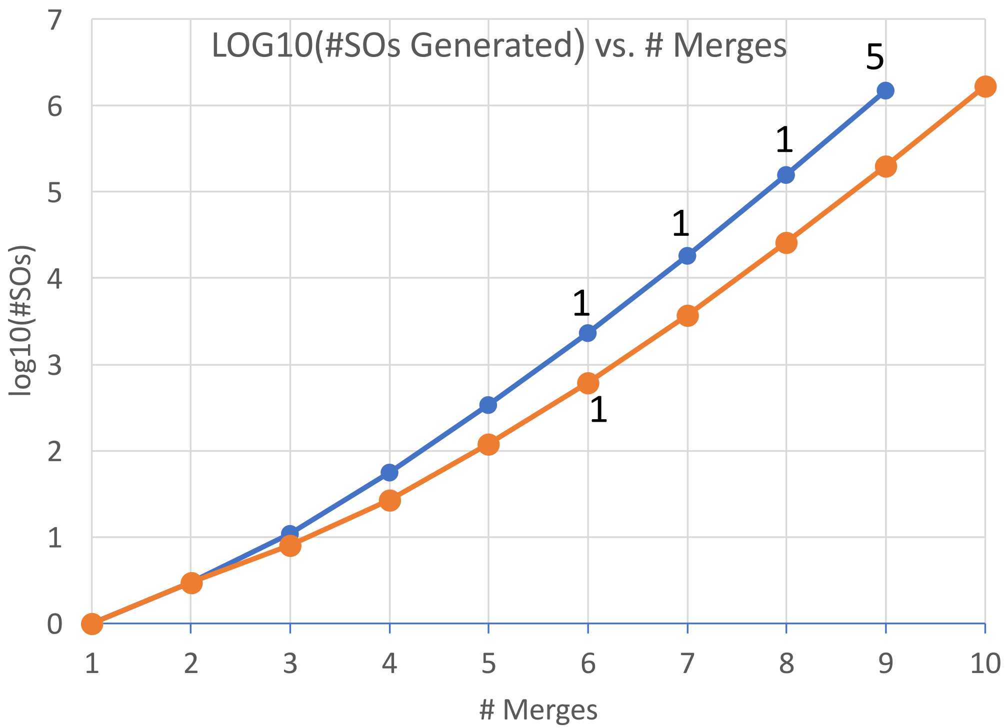
Spellout heads: [the,friend,of,me]

Final output: [the,friend,of,me]



Depth = 6

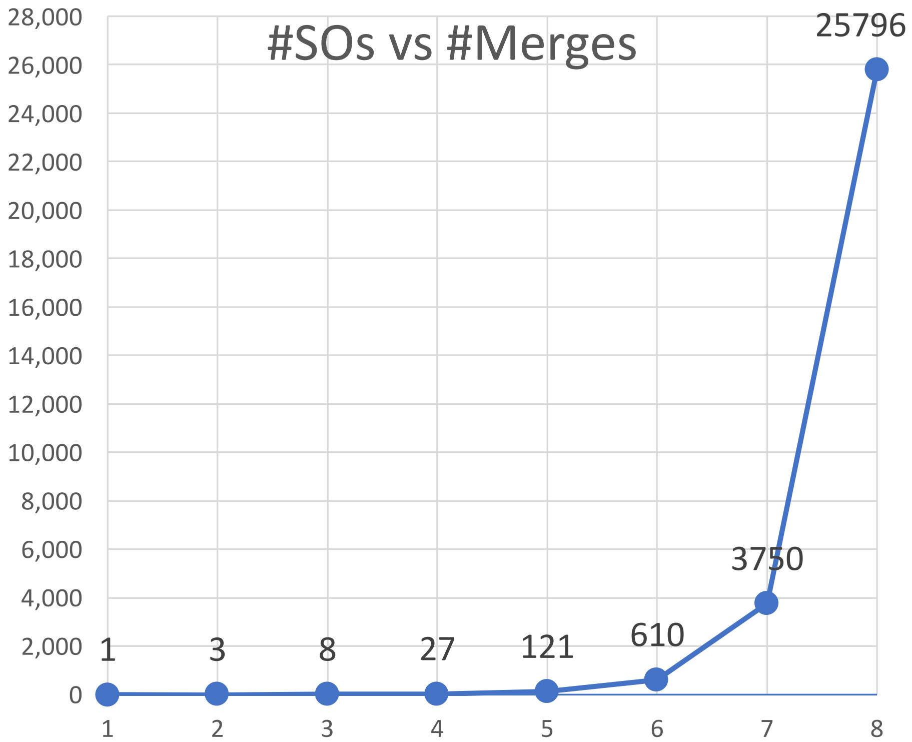
Improve the Framework: Combinatorics



- Orange line: adjusted theory
  - one solution @ 6
  - no extraneous solutions @7-10
  - fewer SOs hypothesized
- Blue line: original theory
  - one solution @ 6
  - one solution @ 7, 8
  - five solutions @ 9

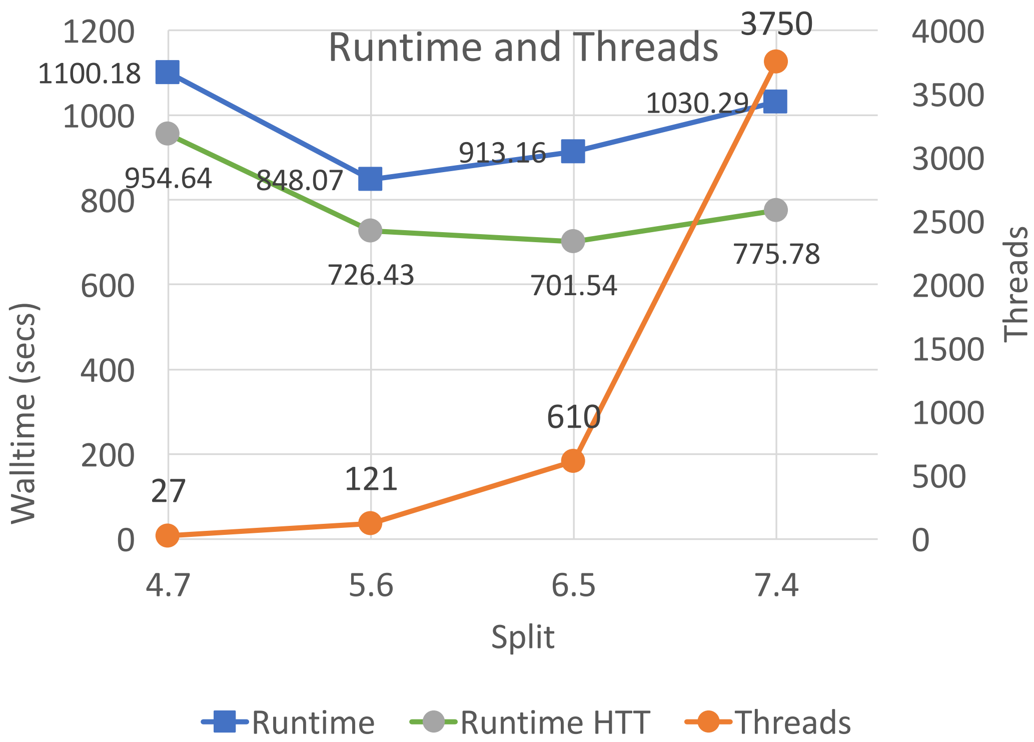
Parallelism: Job size

Parallel Processing Task Size



- **Example**:
  - say we want to search to depth 11 in parallel
  - What is the best way to divvy up the search?
  - We can perform the same search by expressing:
    - 27 threads, each 7 deep
    - 121 threads, each 6 deep
    - 610 threads, each 5 deep
    - 3750 threads, each 4 deep
- Tradeoff:
  - thread overhead vs. load balancing
    - (task size not a constant)
  - RAM wrt. # active threads limits task size

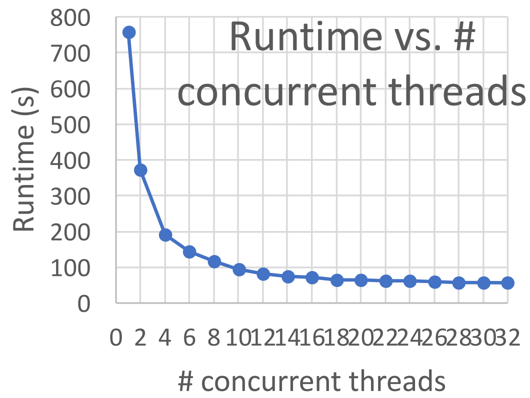
Parallel Processing Task Size: Results



- Conditions:
  - blue line: 16 CPUs used (no HTT); 16 active threads
  - green line: same 16 CPUs + HTT; 32 active threads
  - RAM: 128GB capacity
- Best results:
  - HTT on
  - 610 threads (from 6 deep initially), each job is 5 deep
  - used ≈ 30GB RAM
    - cf. 4.7 split used ≈ 88GB
    - cf. 6.5 split used ≈ 15GB

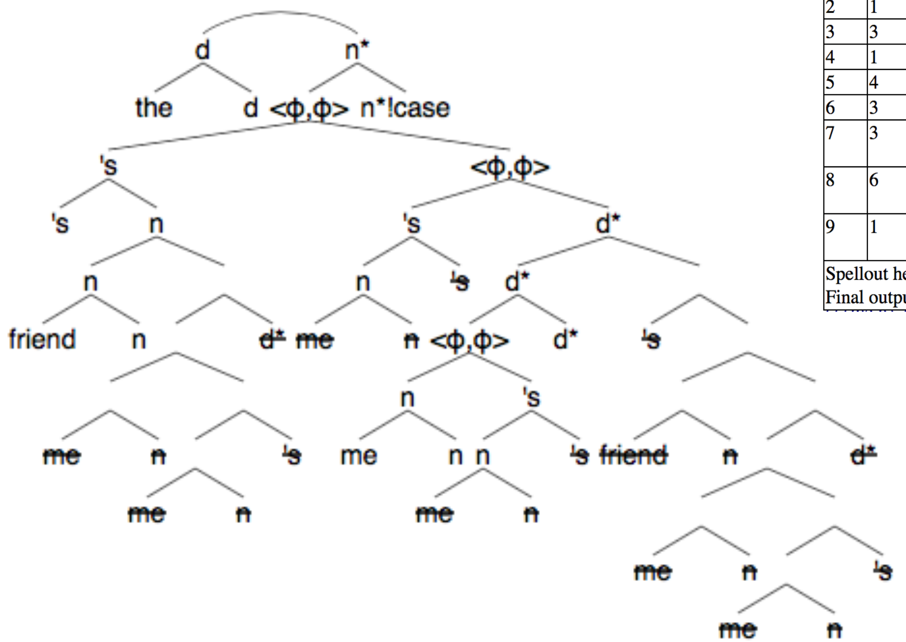
Conclusions

- Application is parallel-friendly
  - search: multiple possible operations
  - speed-up results: 13x on 32 logical cores
- Speed-up allows us to search deeper
  - beyond a basic analysis
- Improve the theory
  - eliminate extraneous analyzes



Appendix

Extra Analyses Uncovered



LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	1	ism	{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	4	ism	{{{{me,n},{{me,n},'s}},d*},{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	3	ism	{{{me,n},'s},{{{{me,n},{{me,n},'s}},d*},{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}
7	3	ism	{{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}},{{{me,n},'s},{{{{me,n},{{me,n},'s}},d*},{'s,{{friend,n!case}, {{{me,n},{{me,n},'s}},d*}}}}}}
8	6	esm	{{{'s,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{{me,n},'s},{{{{me,n},{{me,n},'s}},d*},{'s,{{friend,n},{{{me,n}, {{me,n},'s}},d*}}}}}}},n*!case}
9	1	epm	<{the,d},{{{'s,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{{me,n},'s},{{{{me,n},{{me,n},'s}},d*},{'s,{{friend,n}, {{{me,n},{{me,n},'s}},d*}}}}}}},n*!case}>
Spellout heads: [the,'s,'s,me]			
Final output: [the,'s,'s,me]			

Depth = 9

d*P = {{{me,n},{{me,n},'s}},d*}

ESM {friend,n}

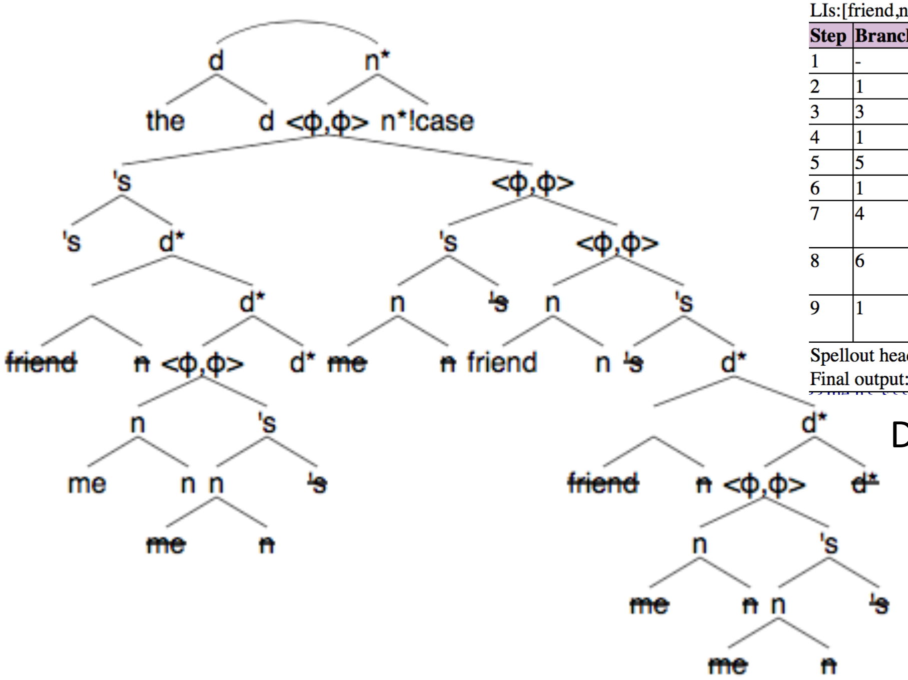
ISM 's (?need a d categorizer)

ISM {{{me,n},{{me,n},'s}},d*}

ISM {{me,n},'s}

ISM {'s, {{friend,n},d*P}

Extra Analyses Uncovered



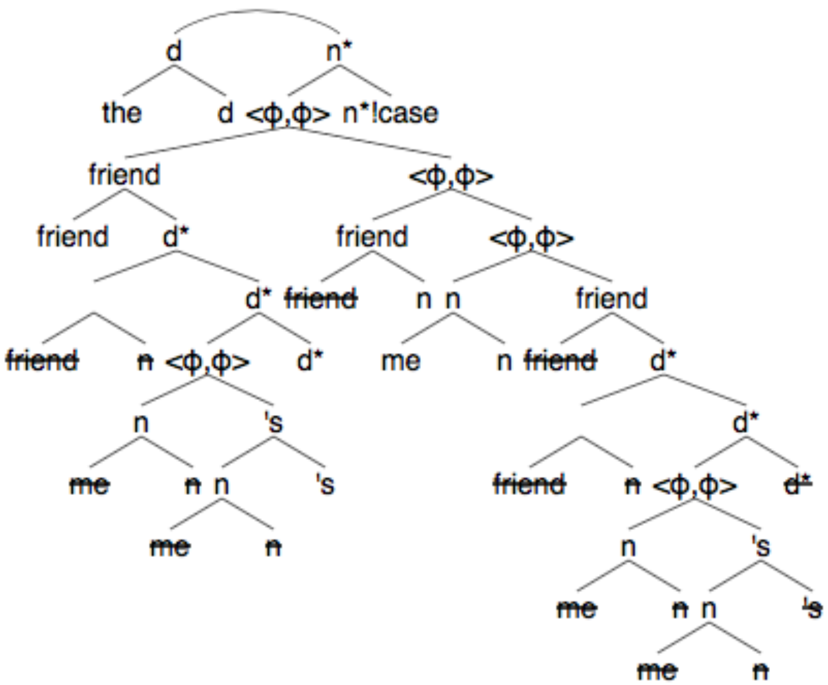
LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1

Step	Branch	Op	SO
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	1	ism	{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	5	ism	{{friend,n!case},{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	1	ism	{{{me,n},'s},{{friend,n!case},{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}
7	4	ism	{{'s,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}},{{{me,n},'s},{{friend,n!case},{'s,{{friend,n!case},{{{me,n}, {{me,n},'s}},d*}}}}}}
8	6	esm	{{{'s,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{{me,n},'s},{{friend,n},{'s,{{friend,n},{{{me,n}, {{me,n},'s}},d*}}}}}},n*!case}
9	1	epm	<{the,d},{{{'s,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{{me,n},'s},{{friend,n},{'s,{{friend,n},{{{me,n}, {{me,n},'s}},d*}}}}}},n*!case}>

Spellout heads: [the,'s,'s,me]
Final output: [the,'s,'s,me]

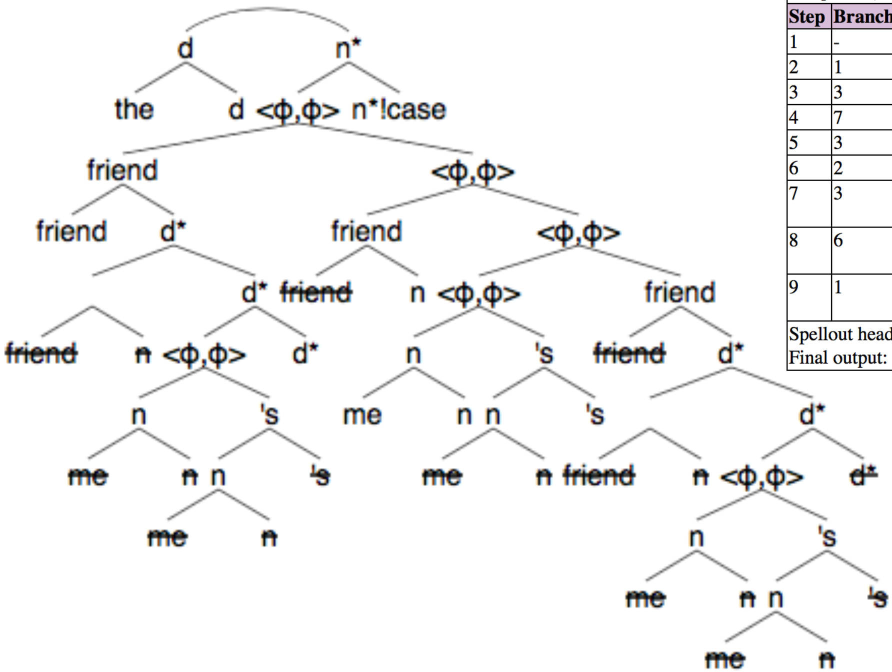
Depth = 9

Extra Analyses Uncovered



LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	7	ism	{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	2	ism	{{me,n},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	4	ism	{{friend,n!case},{{me,n},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}
7	5	ism	{{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}},{{friend,n!case},{{me,n},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}}
8	6	esm	{{{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{friend,n},{{me,n},{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}}}}},n*!case}
9	1	epm	<{the,d},{{{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{friend,n},{{me,n},{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}}}}},n*!case}>
Spellout heads: [the,friend,'s,of,me]			
Final output: [the,friend,of,'s,me]			

Extra Analyses Uncovered



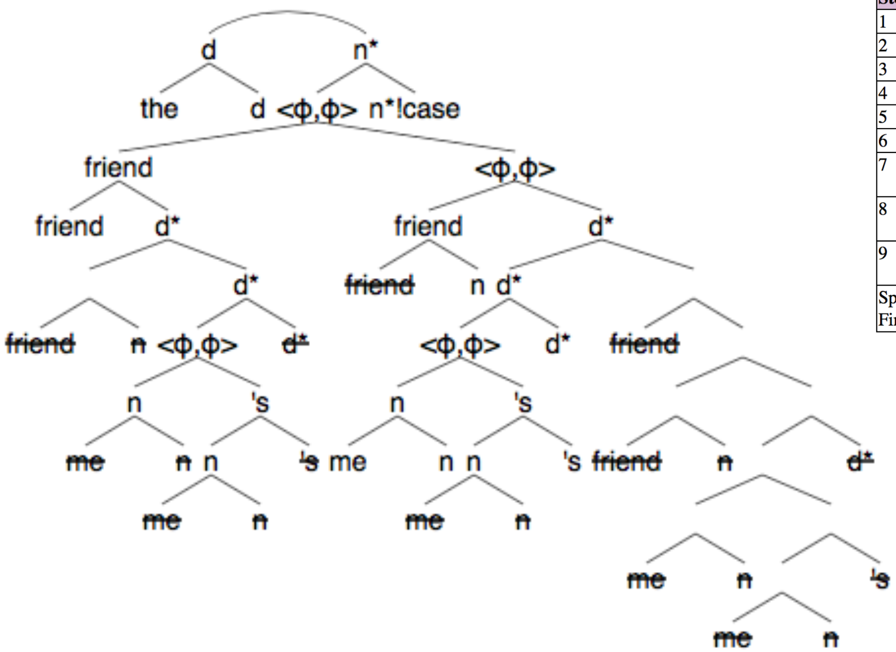
LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1

Step	Branch	Op	SO
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	7	ism	{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	3	ism	{{{me,n},{{me,n},'s}},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	2	ism	{{friend,n!case},{{{me,n},{{me,n},'s}},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}
7	3	ism	{{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}},{{friend,n!case},{{{me,n},{{me,n},'s}},{friend,{{friend,n!case}, {{{me,n},{{me,n},'s}},d*}}}}}}
8	6	esm	{{{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{friend,n},{{{me,n},{{me,n},'s}},{friend,{{friend,n},{{{me,n}, {{me,n},'s}},d*}}}}}},n*!case}
9	1	epm	<{the,d},{{{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{friend,n},{{{me,n},{{me,n},'s}},{friend,{{friend,n},{{{me,n}, {{me,n},'s}},d*}}}}}},n*!case}>

Spellout heads: [the,friend,'s,of,me]
Final output: [the,friend,of,'s,me]

Depth = 9

Extra Analyses Uncovered



LIs:[friend,n!case,{{{me,n},{{me,n},'s}},d*},n*!case,{the,d}] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	friend
2	1	esm	{friend,n!case}
3	3	esm	{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}
4	7	ism	{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}
5	4	ism	{{{{me,n},{{me,n},'s}},d*},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}
6	1	ism	{{friend,n!case},{{{{me,n},{{me,n},'s}},d*},{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}}}}
7	2	ism	{{friend,{{friend,n!case},{{{me,n},{{me,n},'s}},d*}}},{{friend,n!case},{{{{me,n},{{me,n},'s}},d*},{friend,{{friend,n!case}, {{{me,n},{{me,n},'s}},d*}}}}}}
8	6	esm	{{{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{friend,n},{{{{me,n},{{me,n},'s}},d*},{friend,{{friend,n},{{{me,n}, {{me,n},'s}},d*}}}}}},n*!case}
9	1	epm	<{the,d},{{{friend,{{friend,n},{{{me,n},{{me,n},'s}},d*}}},{{friend,n},{{{{me,n},{{me,n},'s}},d*},{friend,{{friend,n}, {{{me,n},{{me,n},'s}},d*}}}}}},n*!case}>
Spellout heads: [the,friend,of,'s,me]			
Final output: [the,friend,of,mine]			

Depth = 9