Constrained Decoding

Pradeep Dasigi
Carnegie Mellon University / Allen Institute for Artificial Intelligence

Introduction

- Traditional semantic parsers used grammar based parsing algorithms
- Neural semantic parsing has moved towards using encoder decoder models
- Decoders are usually recurrent neural networks that produce sequences
    - But they can produce outputs that are not valid (syntactically or semantically)!

Example from WikiTableQuestions

Athlete	Nation	Olympics	Medals
Gillis Grafström	Sweden (SWE)	1920–1932	4
Evgeni Plushenko	Russia (RUS)	2002–2014	4
Karl Schäfer	Austria (AUT)	1928–1936	2
Katarina Witt	East Germany (GDR)	1984–1988	2
Tenley Albright	United States (USA)	1952-1956	2
Kim Yu-na	South Korea (KOR)	2010–2014	2
Patrick Chan	Canada (CAN)	2014	2

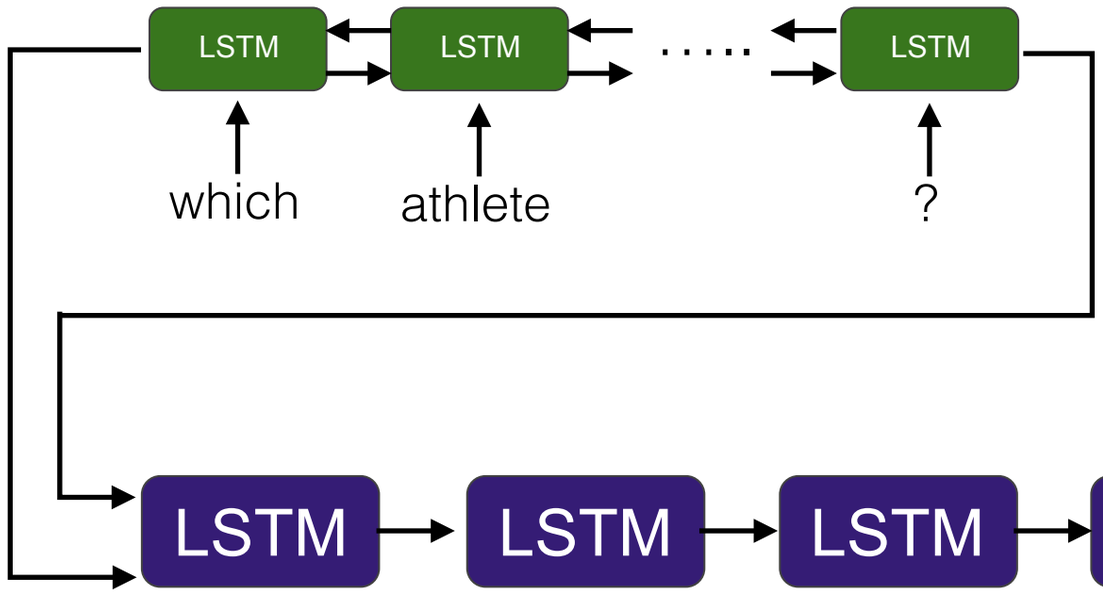
Question:

Which athlete was from South Korea after 2010?

```
((reverse athlete)
 (and
  (nation south_korea)
  (year ((reverse date)
    (>= 2010-mm-dd)))
```

[Pasupat and Liang, 2015]

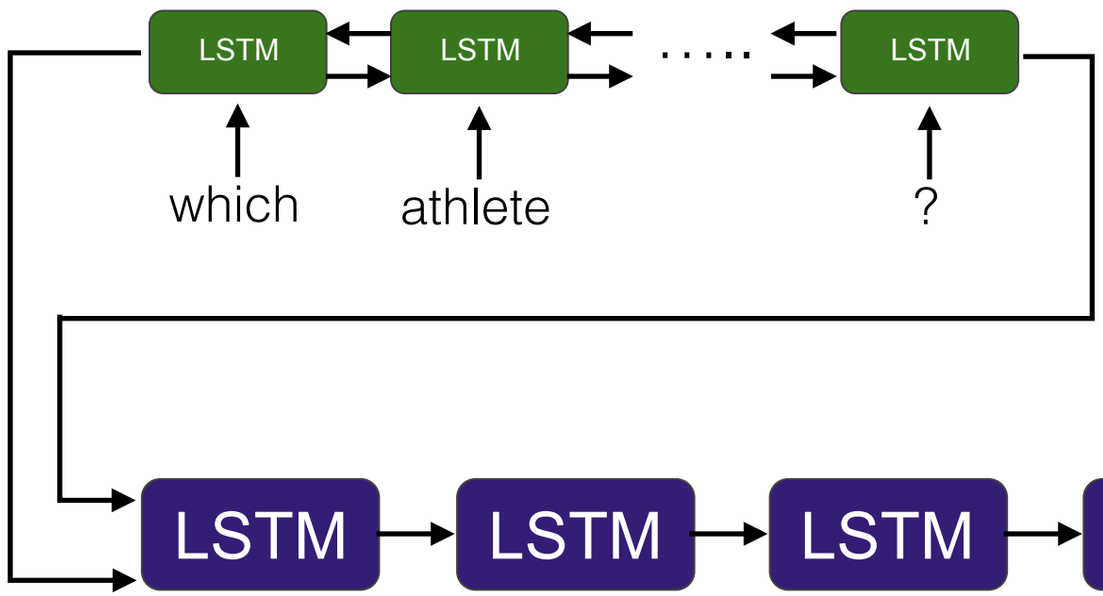
Seq2Seq Output Space



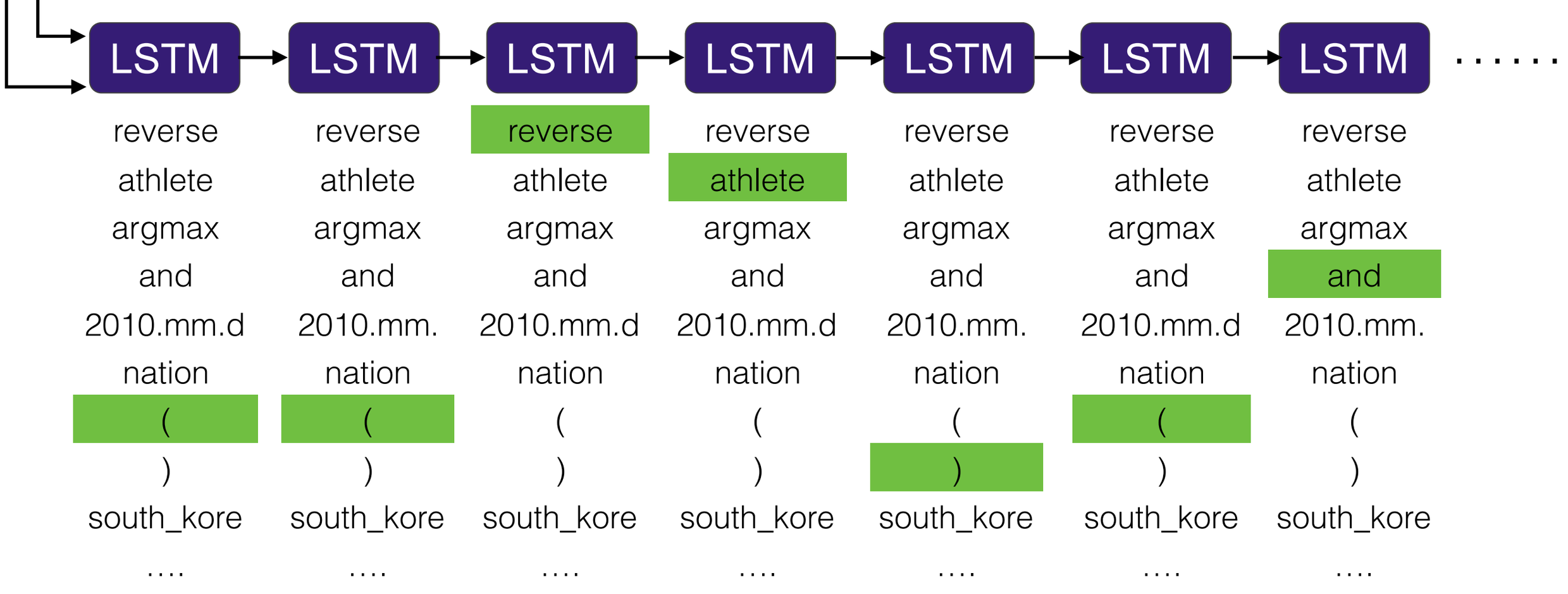
Athlete	Nation	Olympics	Medals
Kim Yu-na	South Korea (KOR)	2010–2014	2
Tenley Albright	United States (USA)	1952-1956	2

reverse	reverse	reverse	reverse	reverse	reverse	reverse
athlete	athlete	athlete	athlete	athlete	athlete	athlete
argmax	argmax	argmax	argmax	argmax	argmax	argmax
and	and	and	and	and	and	and
2010.mm.dd	2010.mm.dd	2010.mm.dd	2010.mm.dd	2010.mm.dd	2010.mm.dd	2010.mm.dd
nation	nation	nation	nation	nation	nation	nation
(	(	(	(	(	(	(
)	)	)	)	)	)	)
south_korea	south_korea	south_korea	south_korea	south_korea	south_korea	south_korea
….	….	….	….	….	….	….

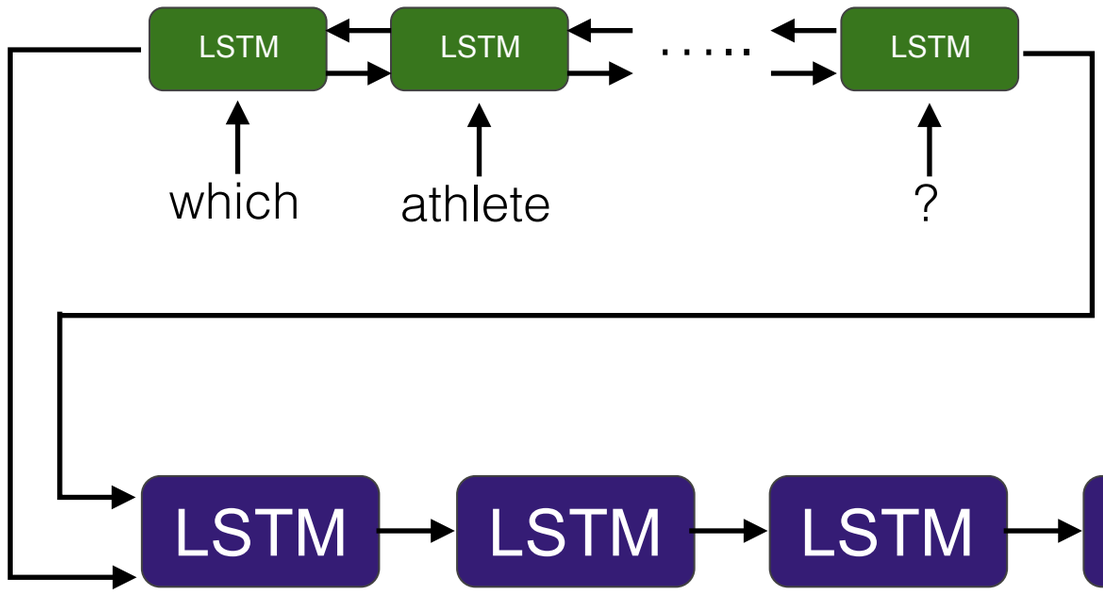
Seq2Seq Output Space



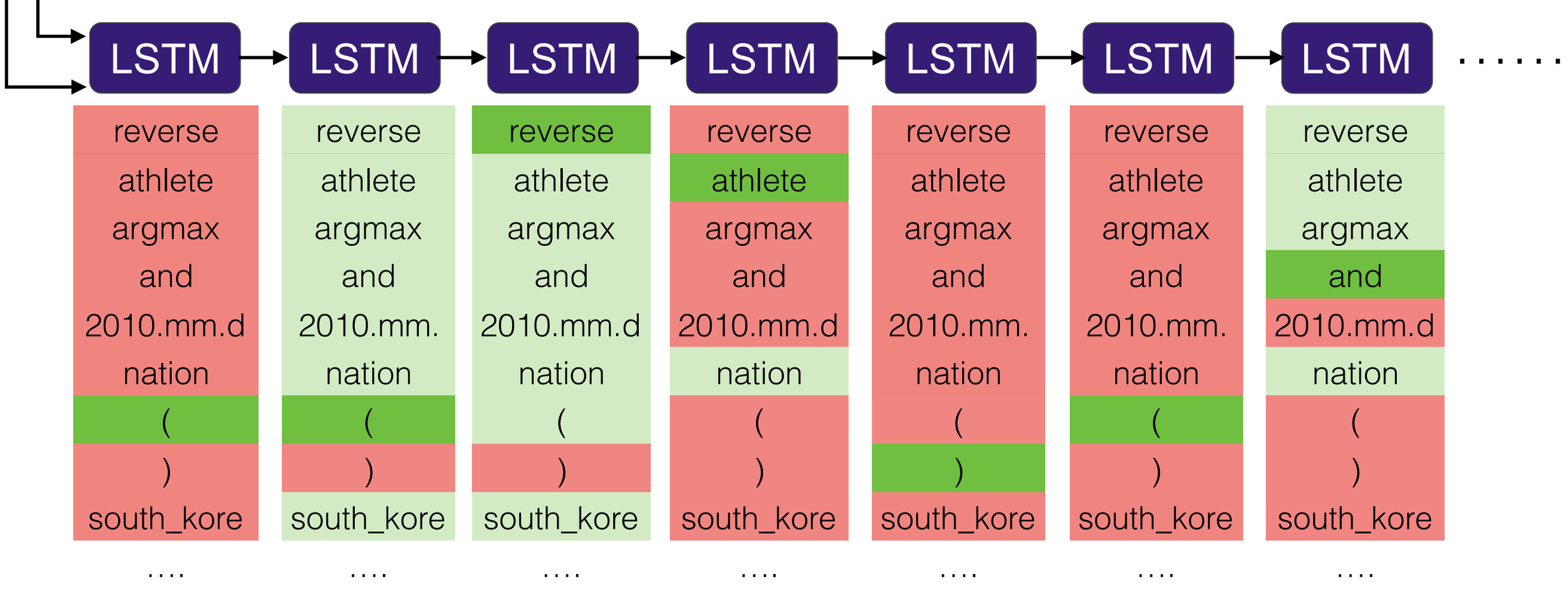
Athlete	Nation	Olympics	Medals
Kim Yu-na	South Korea (KOR)	2010–2014	2
Tenley Albright	United States (USA)	1952-1956	2



Seq2Seq Output Space



Athlete	Nation	Olympics	Medals
Kim Yu-na	South Korea (KOR)	2010–2014	2
Tenley Albright	United States (USA)	1952-1956	2



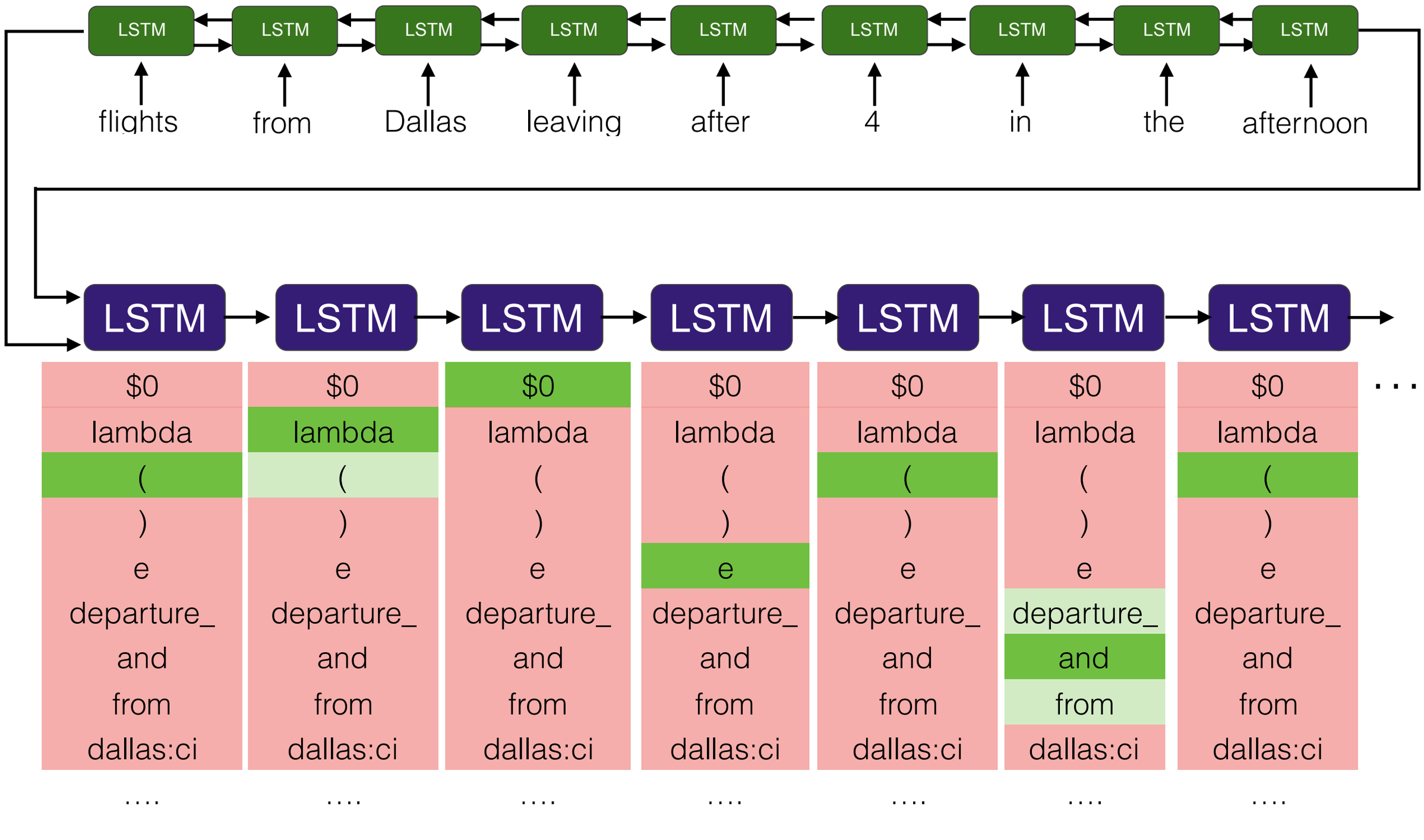
Example from ATIS

flights from Dallas leaving after 4 in the afternoon

```
(lambda $0 e
  (and
    (>(departure_time $0) 1600:ti)
    (from $0 dallas:ci)))
```

[Hemphill et al., 1990; Dahl et al., 1994]

Seq2Seq Output Space



Constrained Decoding

- Constrain the output space to selections that matter
- **Inference:** Avoid invalid parses
- **Training:** Do not waste modeling power in distinguishing invalid parses from valid ones!

Token-based Decoding:

The output space is tokens, but they are constrained to be relevant at each time step.

Grammar-based Decoding:

The output space is production rules, and a grammar defines the constraints.

Constrained Decoding

- Constrain the output space to selections that matter
- **Inference:** Avoid invalid parses
- **Training:** Do not waste modeling power in distinguishing invalid parses from valid ones!

Token-based Decoding

Dong and Lapata. 2016. Language to Logical Form with Neural Attention. In ACL.

Dong and Lapata. 2018. Coarse-to-Fine Decoding for Neural Semantic Parsing. In ACL.

Goldman, Latcinnik, Naveh, Globerson and Berant. 2018. Weakly-supervised Semantic Parsing with Abstract Examples. In ACL.

Grammar-based Decoding:

Xiao, Dymetman, and Gardent. 2016. Sequence-based Structured Prediction for Semantic Parsing. In ACL.

Yin and Neubig. 2017. A Syntactic Neural Model for General Purpose Code Generation. In ACL.

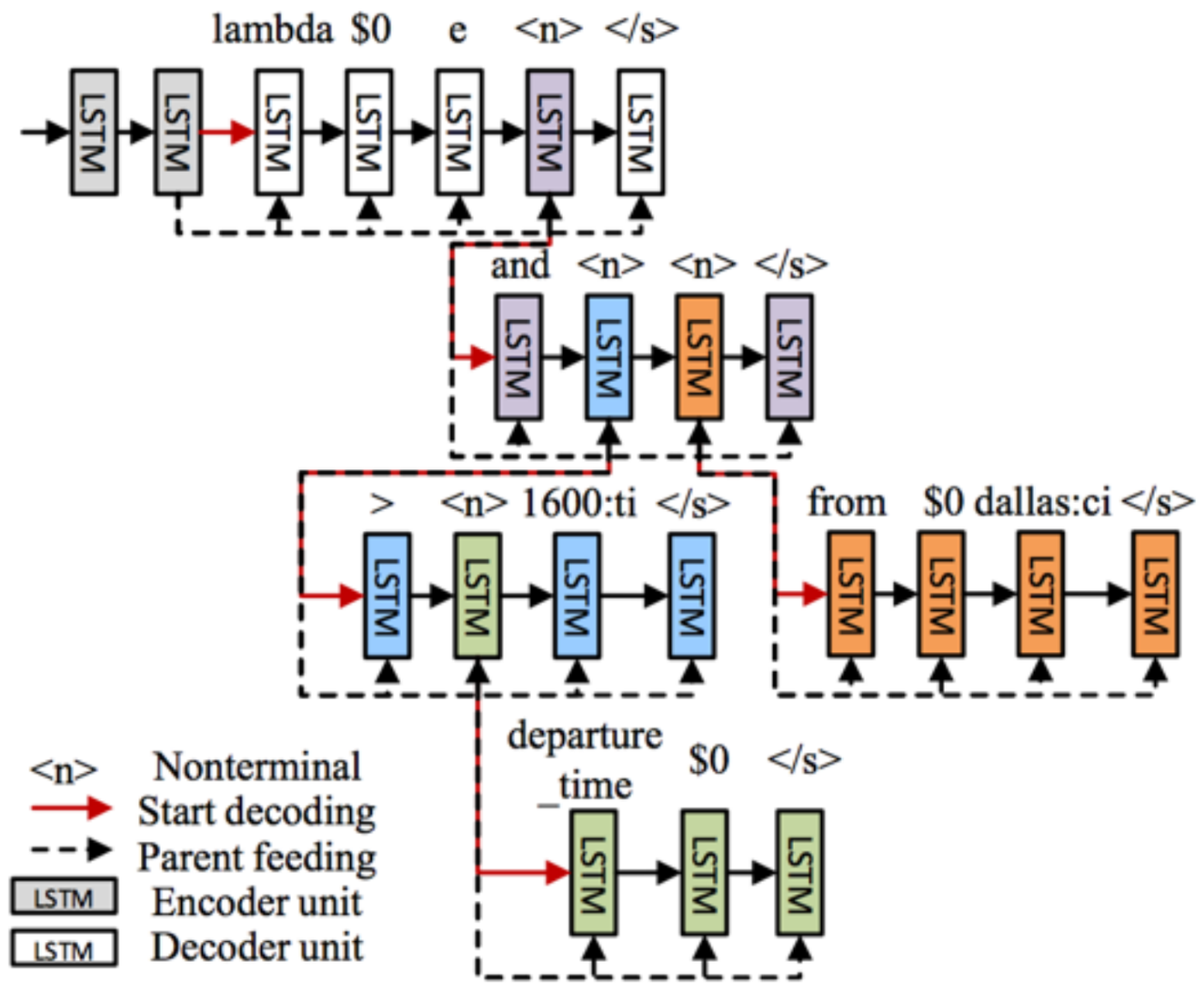
Krishnamurthy, Dasigi, and Gardner. 2017. Neural Semantic Parsing with Type Constraints for Semi-Structured Tables. In EMNLP.

Token-based Constrained Decoding

Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon

```
(lambda $0 e
  (and
    (> (departure_time $0) 1600:ti)
    (from $0 dallas:ci)))
```

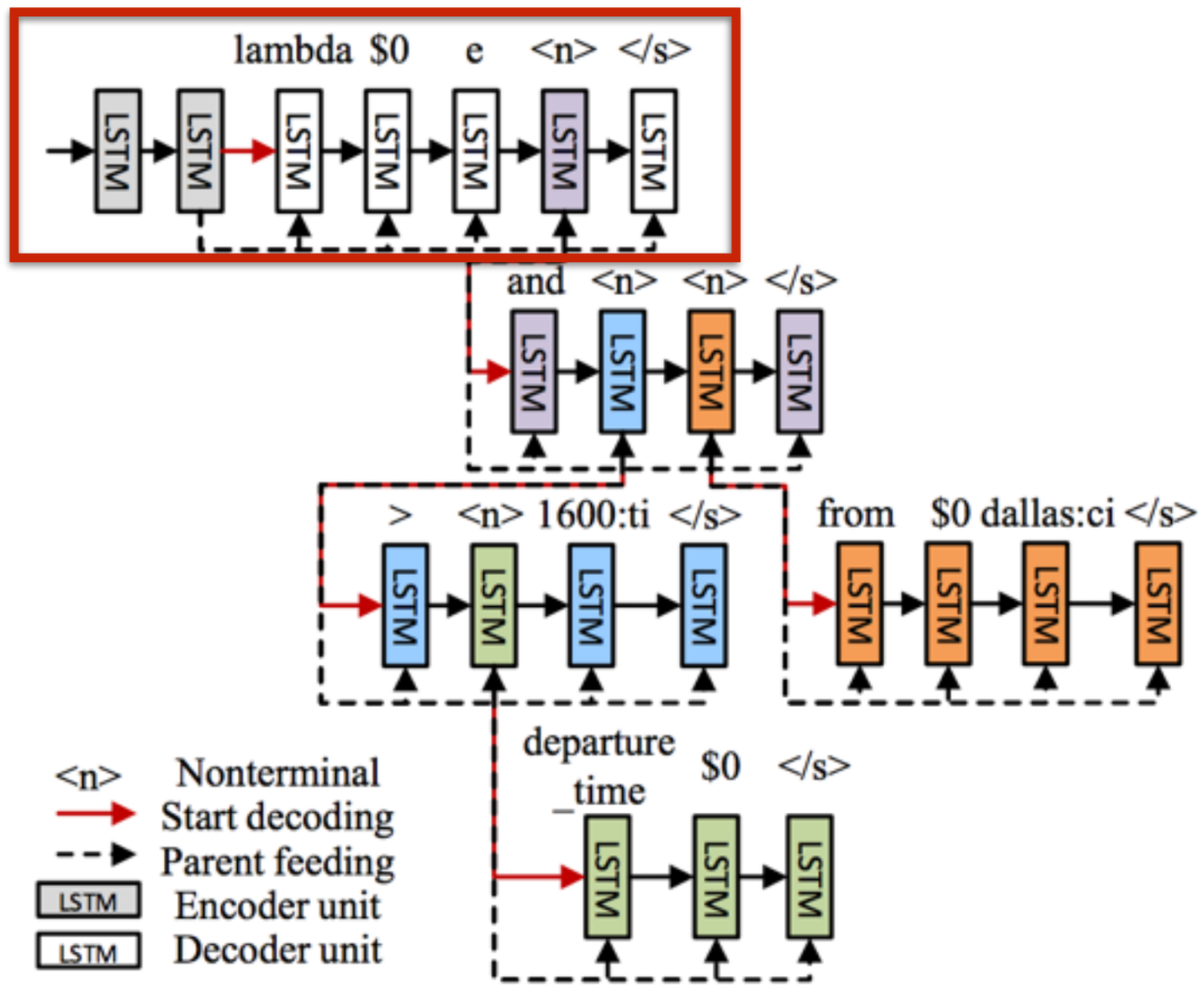


[Dong and Lapata, 2016]

Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon

(lambda $0 e **<n>**)

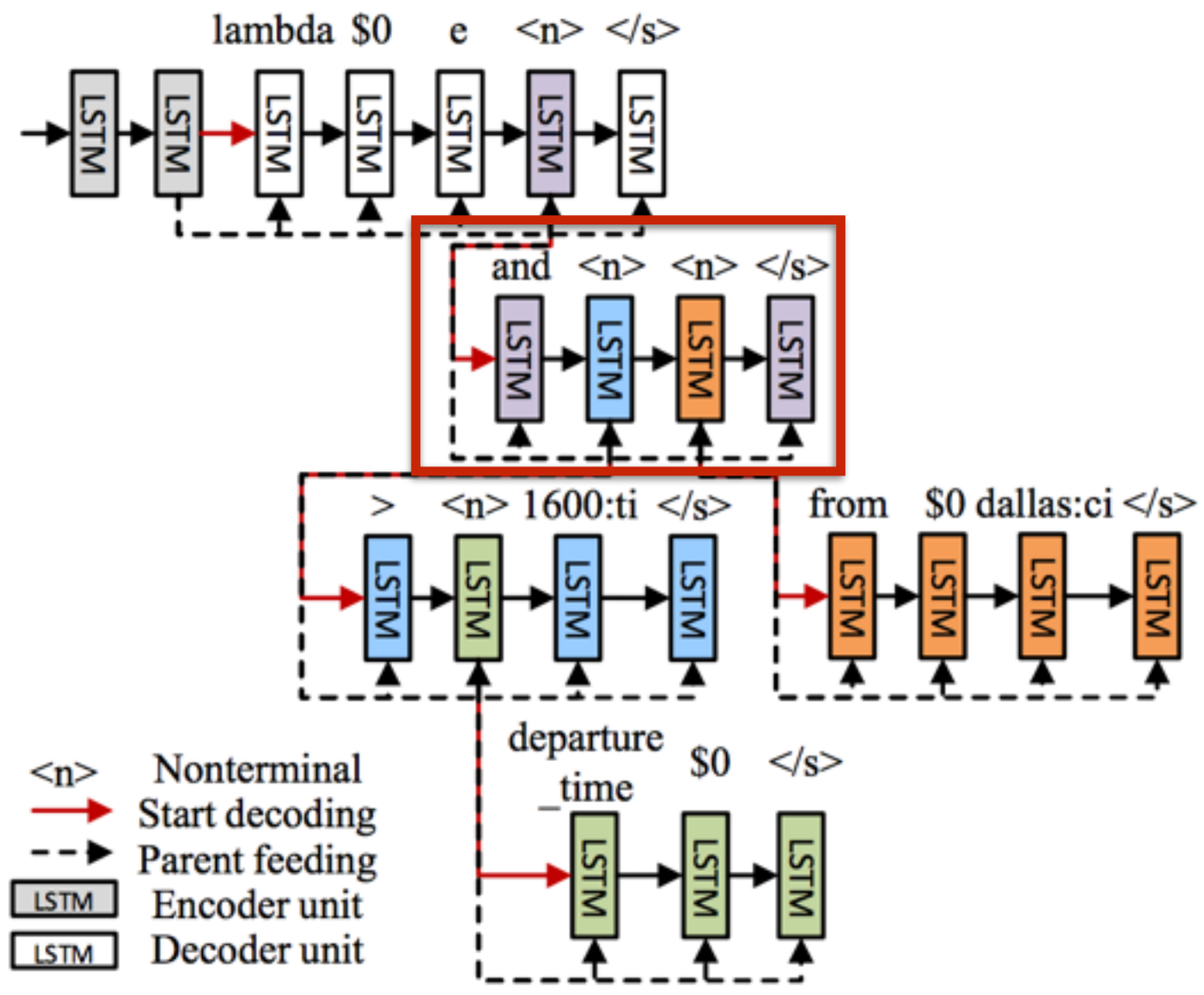


[Dong and Lapata, 2016]

Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon

(lambda $0 e
  (and **<n>** <n>))

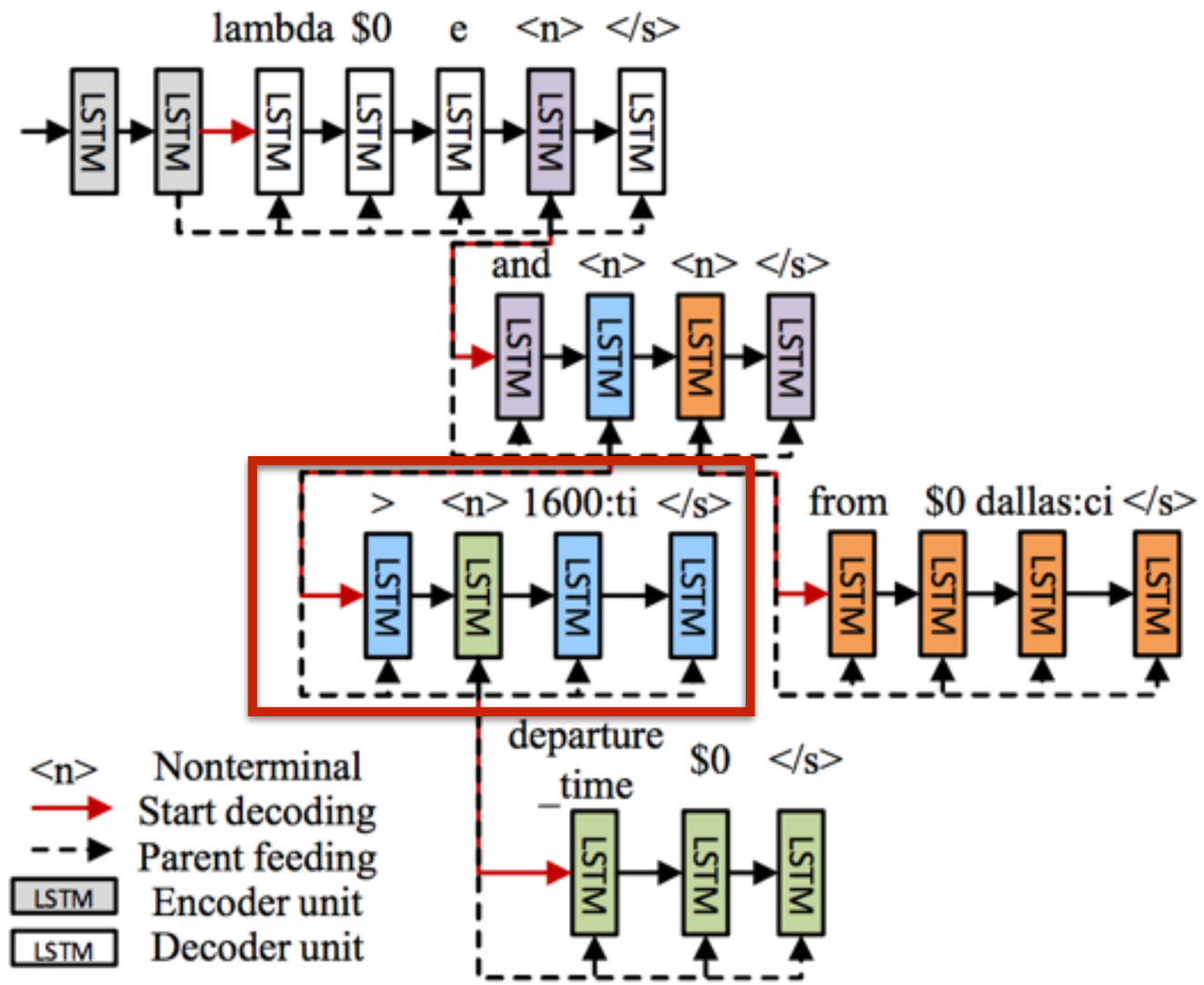


[Dong and Lapata, 2016]

Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon

```
(lambda $0 e
  (and
    (> <n> 1600:ti)
   <n>))
```

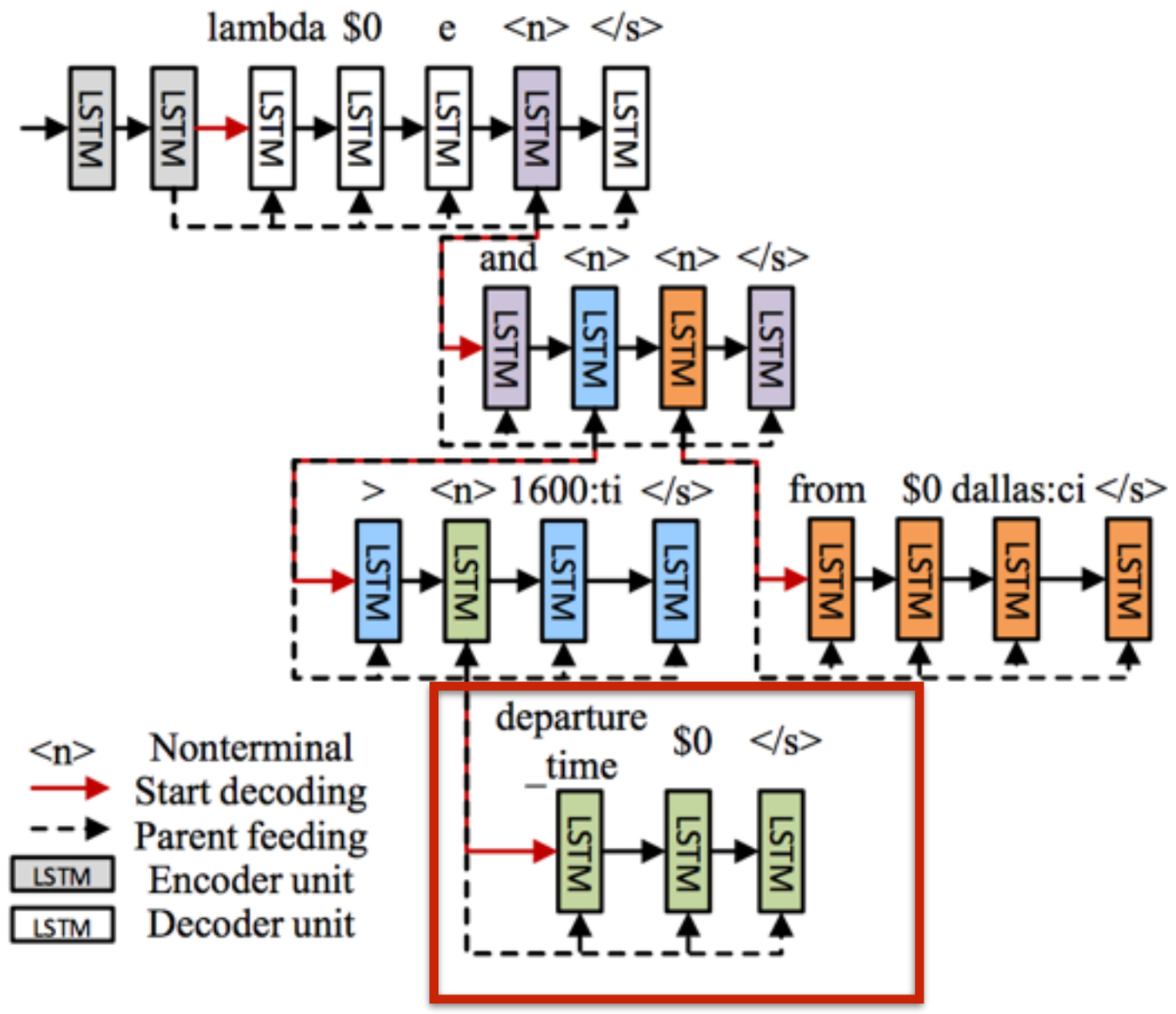


[Dong and Lapata, 2016]

Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon

```
(lambda $0 e
  (and
   (> (departure_time $0) 1600:ti)
   <n>))
```

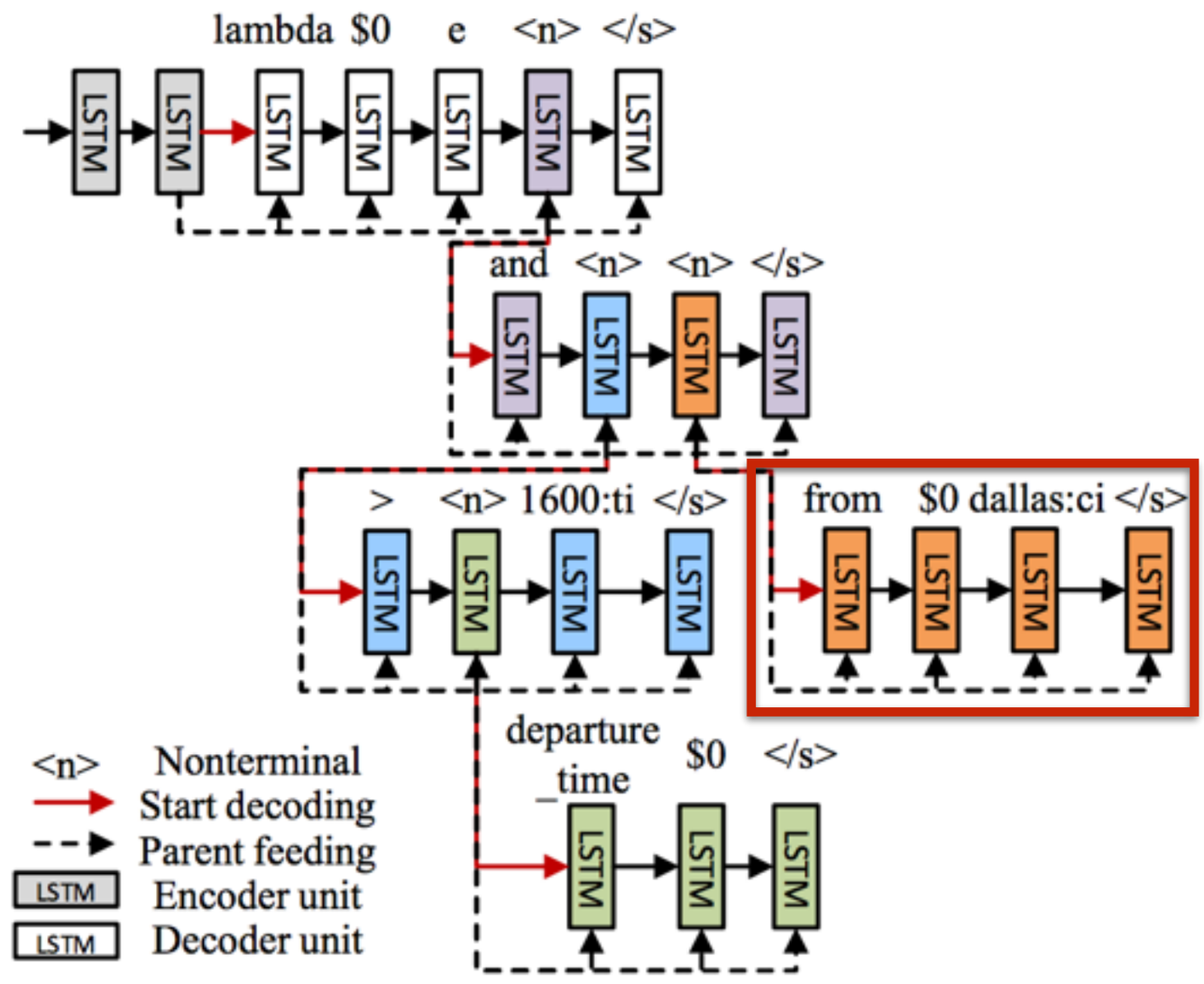


[Dong and Lapata, 2016]

Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon

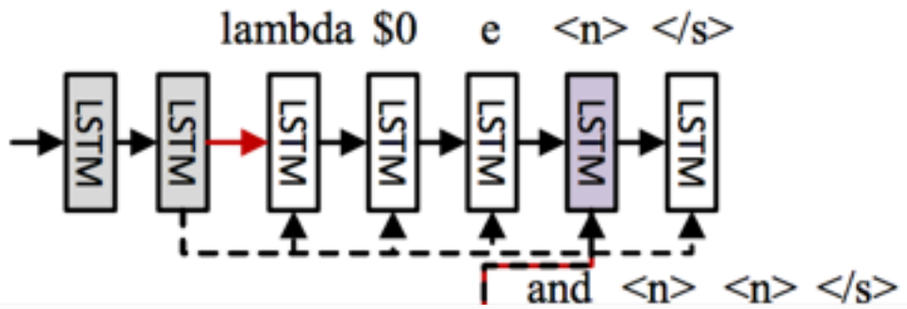
```
(lambda $0 e
  (and
    (> (departure_time $0) 1600:ti)
    (from $0 dallas:ci)))
```



[Dong and Lapata, 2016]

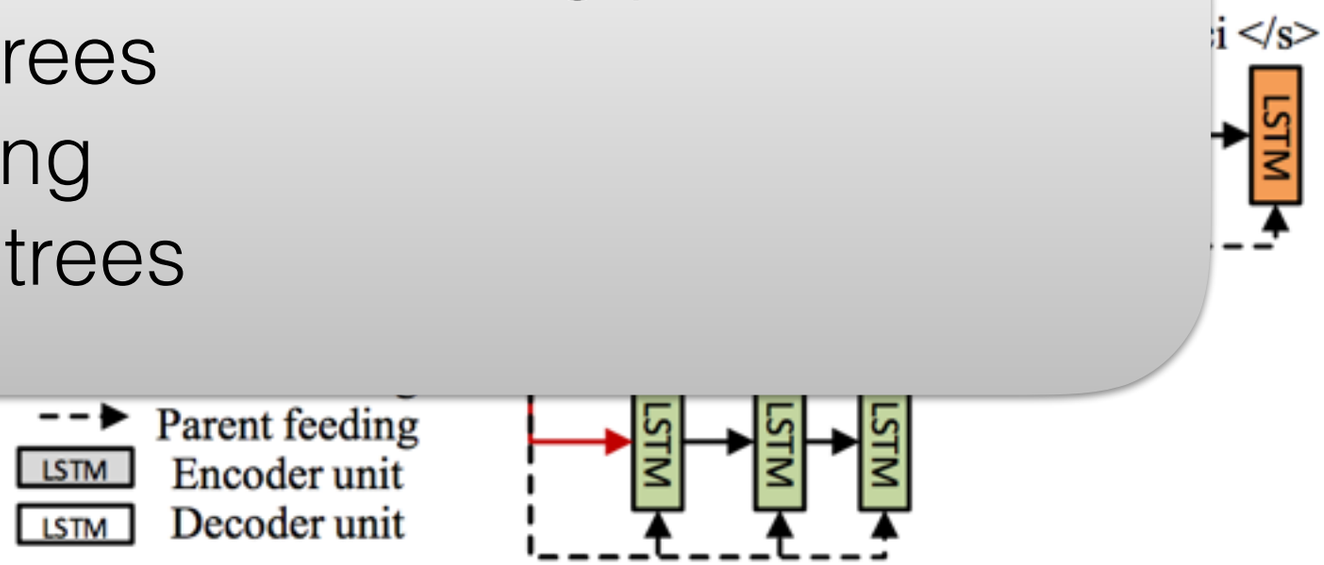
Constraining output structure: Seq2Tree

Flights from Dallas leaving after 4 in the afternoon



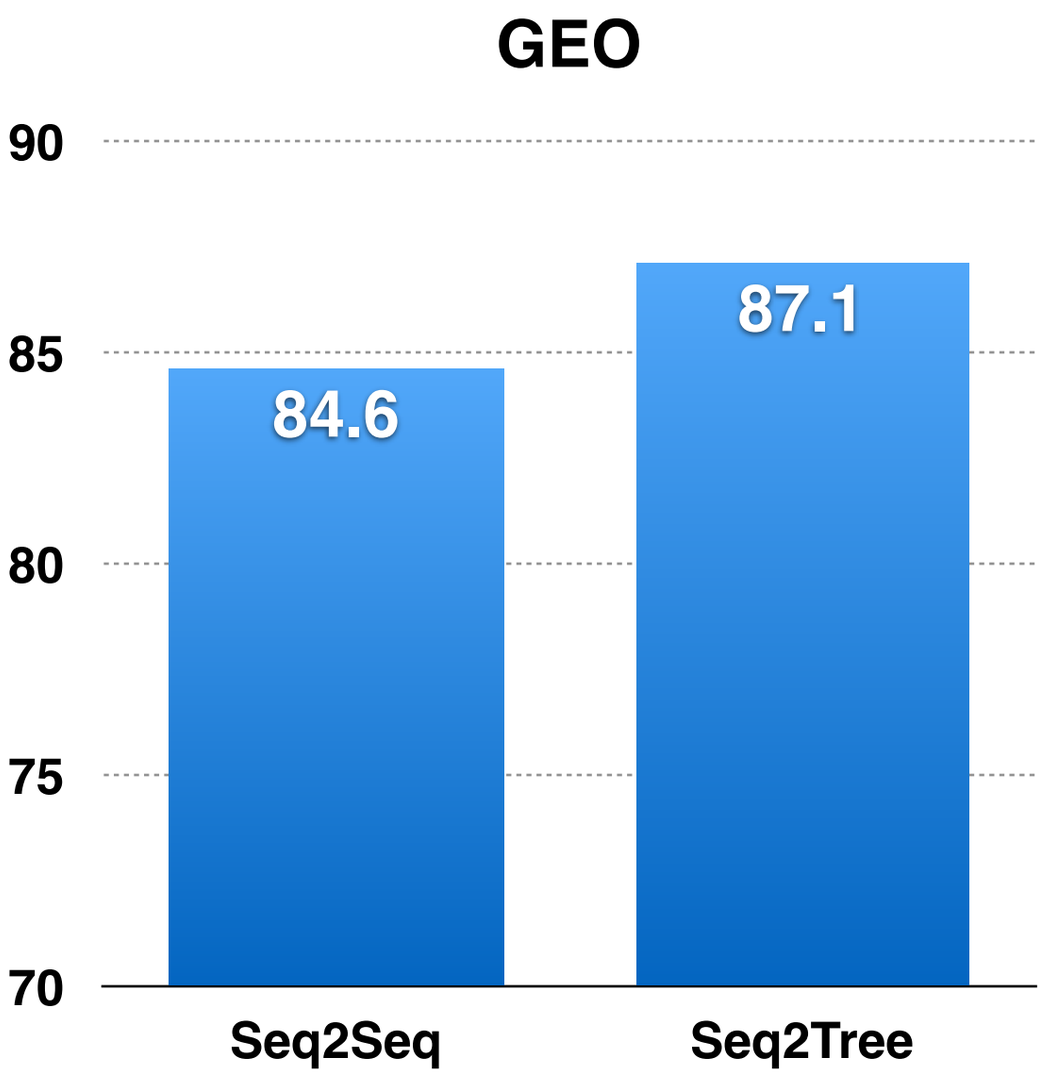
(lambda $0 e
(and
(> (depa
(from $(

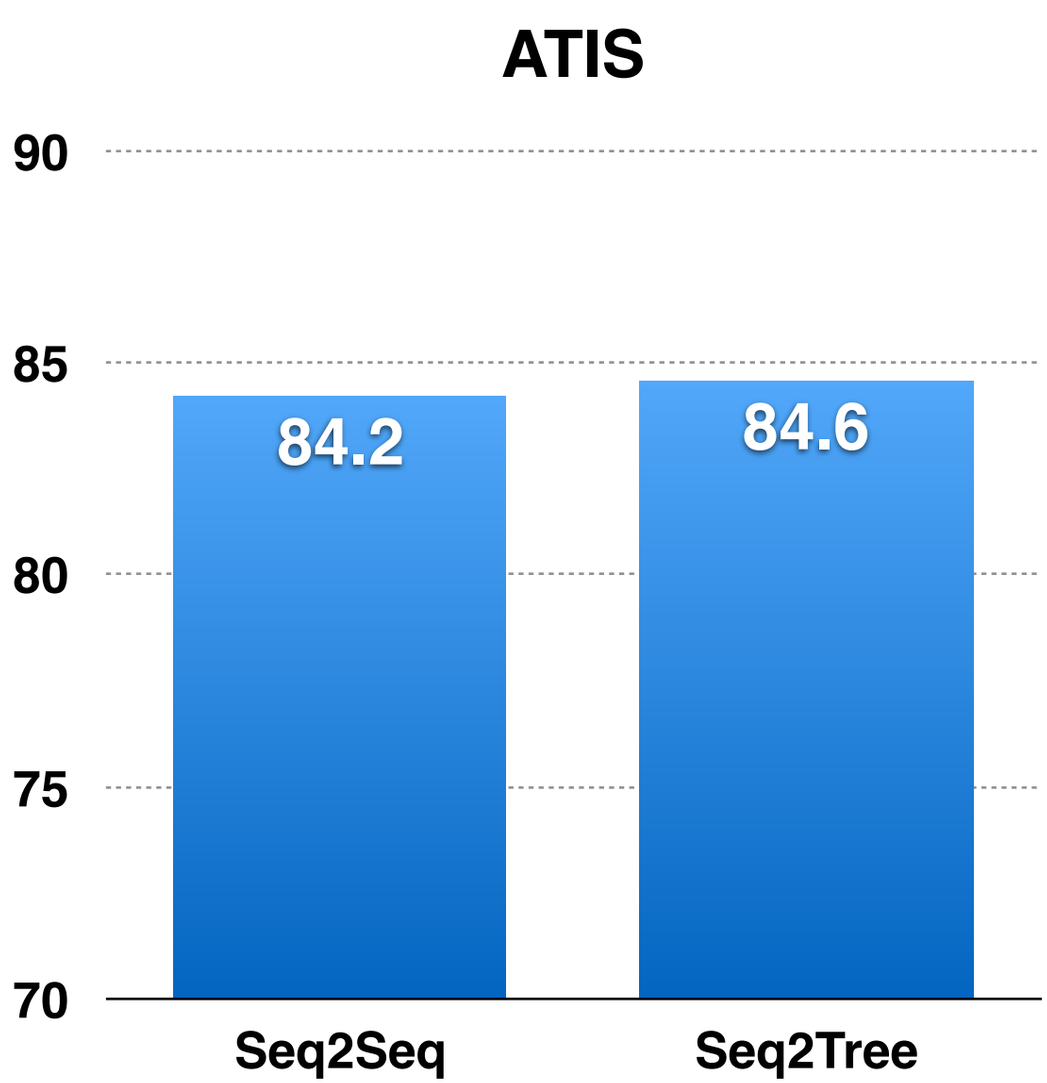
- ☑ Need not explicitly model matching parentheses
- ☑ Syntactically valid trees
- ☑ Allows parent feeding
- ❌ Semantically valid trees



[Dong and Lapata, 2016]

Empirical Comparison with Seq2Seq on GEO and ATIS

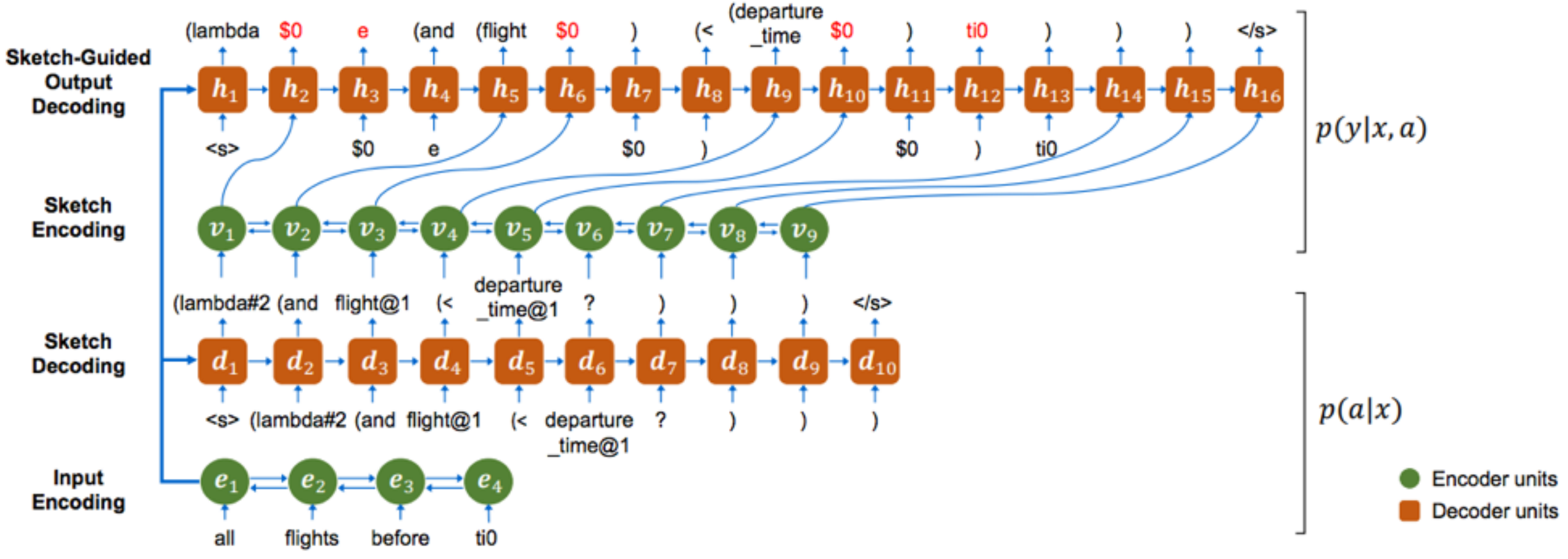




[Dong and Lapata, 2016]

Sketch-Constrained Seq2Tree

- Decoding in two steps:
  1. Decoder 1: Rough sketch conditioned on encoder output
  2. Decoder 2: Finer output constrained by the sketch, conditioned on the outputs of decoder 1 and encoder



[Dong and Lapata, 2018]

Grammar-based Constrained Decoding

Constraining output structure and types

Athlete	Nation	Olympics	Medals
Gillis Grafström	Sweden (SWE)	1920–1932	4
Kim Yu-na	South Korea (KOR)	2010–2014	2
Patrick Chan	Canada (CAN)	2014	2

```
((reverse athlete)
 (and
  (nation south_korea)
  (year ((reverse date)
  (>= 2010-mm-dd)))
```

Basic Types:
Row (**r**); Cell (**c**) (kim_yu_na; 2014; …); Number (**n**); Date (**d**)

Complex Types:
Column (**<c,r>**): athlete; nation; olympics; medals
Binary row operations (**<r,<r,r>>**): and; or
Reverse column operation (**<<c,r>,<r,c>>**): reverse
….

[Krishnamurthy, Dasigi and Gardner, 2017]

Note on the notation of types

Athlete	Nation	Olympi	Medals
Gillis	Sweden	1920–	4
Kim Yu-	South	2010–	2
Patrick	Canada	2014	2

- Complex types
  - Example: column: cell → row **<c,r>**
  - Concrete example: (**nation** south_korea)

- Currying for functions with multiple arguments
  - Example: binary row operator: row, row → row
    Rewritten as: row → (row → row) **<r,<r,r>>**
  - Concrete example:
    (**and** (nation south_korea) (medals 4))

- Higher order functions
  - Example: reverse: (cell → row) → (row → cell) **<<c,r>,<r,c>>**
  - Concrete example:
    ((**reverse** athlete) (and (nation south_korea) (medals 4))

Constraining output structure and types

Athlete	Nation	Olympics	Medals
Gillis Grafström	Sweden (SWE)	1920–1932	4
Kim Yu-na	South Korea (KOR)	2010–2014	2
Patrick Chan	Canada (CAN)	2014	2

```
((reverse athlete)
 (and
  (nation south_korea)
  (year ((reverse date)
  (>= 2010-mm-dd)))
```

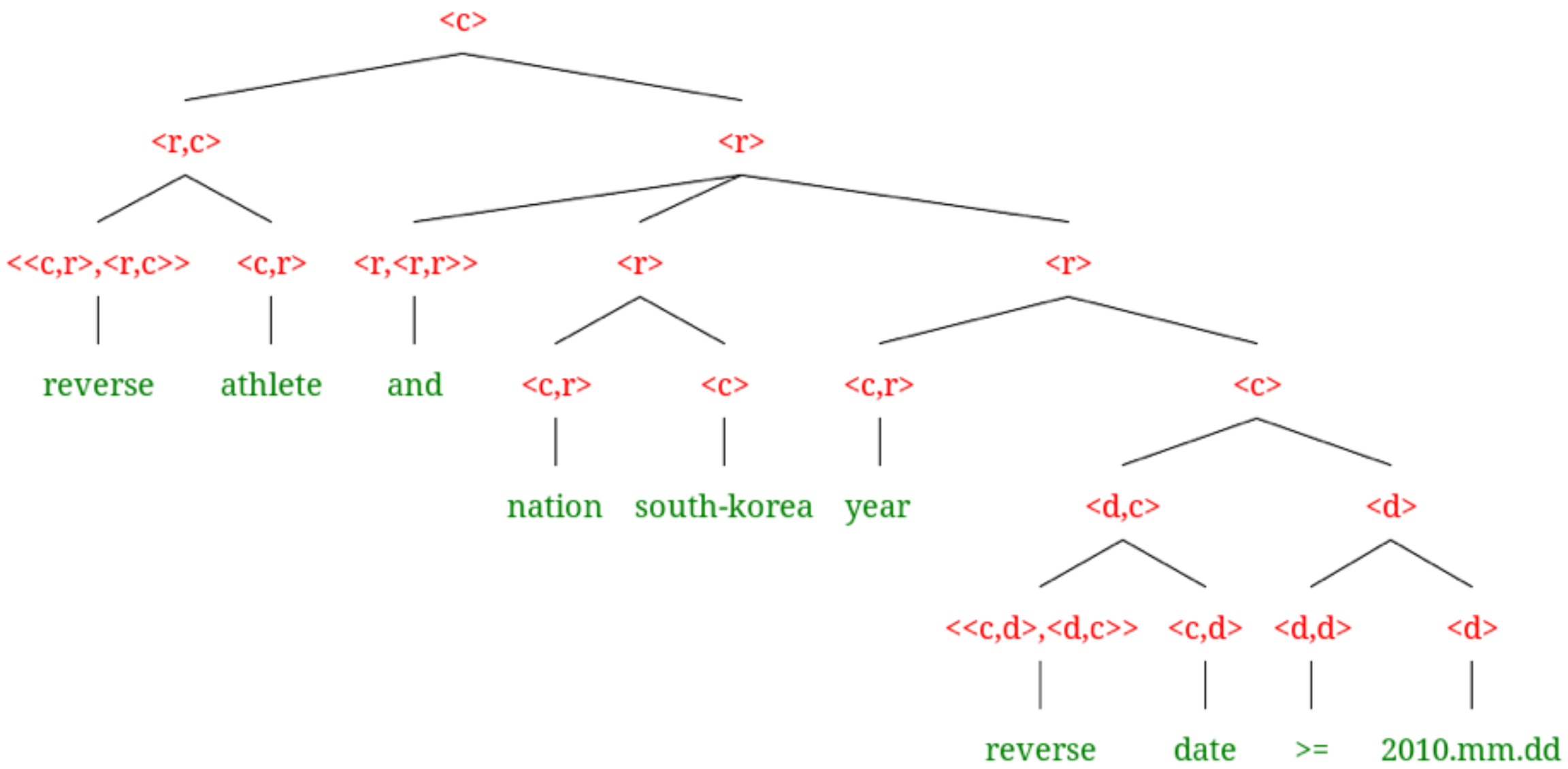
Basic Types:
Row (**r**); Cell (**c**) (kim_yu_na; 2014; …); Number (**n**); Date (**d**)

Complex Types:
Column (**<c,r>**): athlete; nation; olympics; medals
Binary row operations (**<r,<r,r>>**): and; or
Reverse column operation (**<<c,r>,<r,c>>**): reverse
….

[Krishnamurthy, Dasigi and Gardner, 2017]

Constraining output structure and types

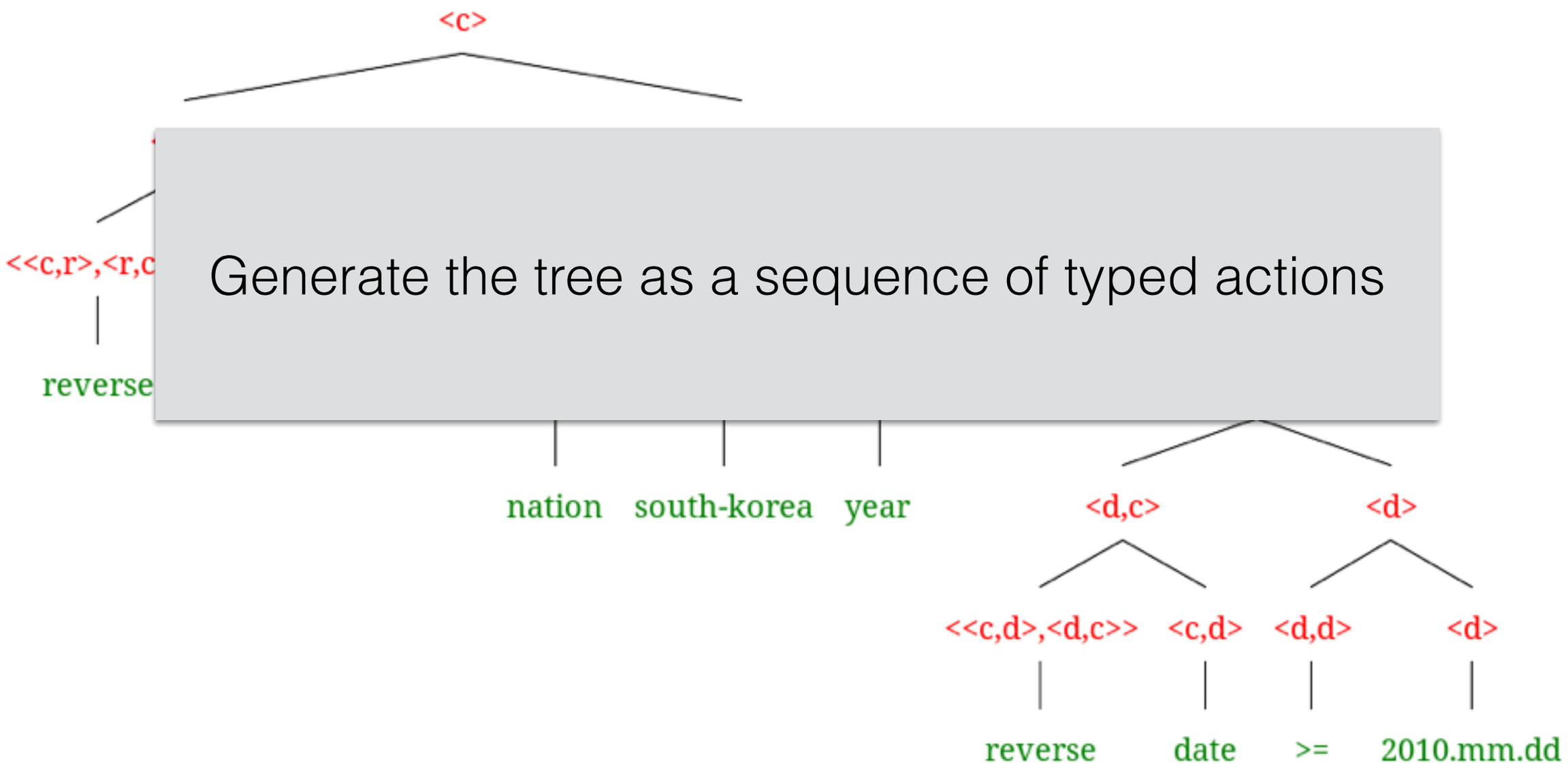
((reverse athlete) (and (nation south_korea) (year ((reverse date) (>= 2010-mm-dd)))



[Krishnamurthy, Dasigi and Gardner, 2017]

Constraining output structure and types

((reverse athlete) (and (nation south_korea) (year ((reverse date) (>= 2010-mm-dd)))



[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

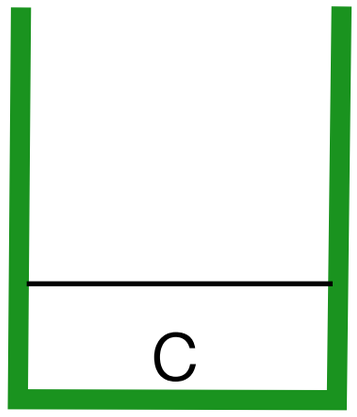
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c

Logical Form

c



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

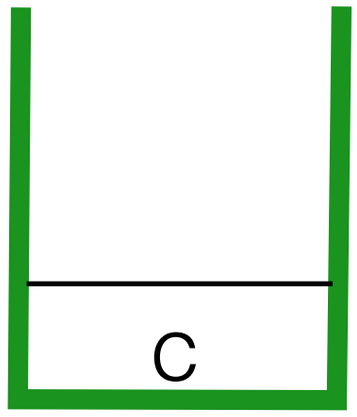
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c

Logical Form

c



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

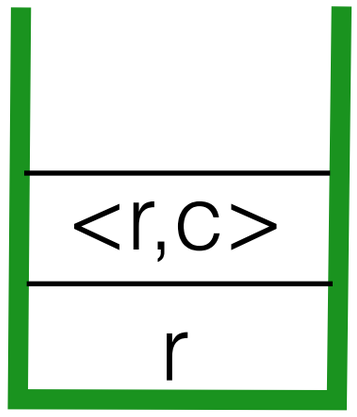
Generated Actions

START → c

c→(<r,c> r)

Logical Form

(<r,c> r)



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

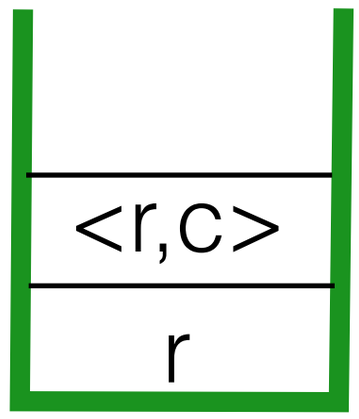
Generated Actions

START → c

c→(<r,c> r)

Logical Form

(**<r,c>** r)



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

Generated Actions

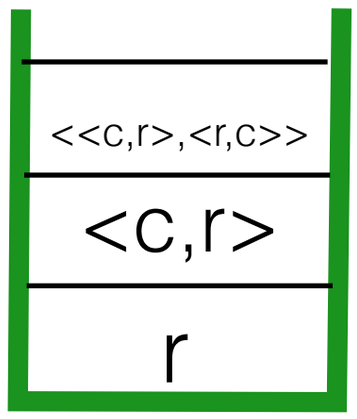
START → c

c→(<r,c> r)

<r,c>→(<<c,r>,<r,c>> <c,r>)

Logical Form

((<<c,r>,<r,c>>, <c,r>) r)



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

Generated Actions

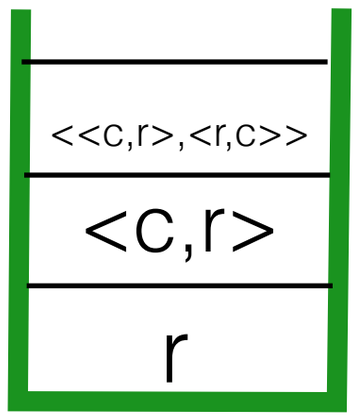
START → c

c→(<r,c> r)

<r,c>→(<<c,r>,<r,c>> <c,r>)

Logical Form

((**<<c,r>,<r,c>>**, <c,r>) r)



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

Generated Actions

START → c

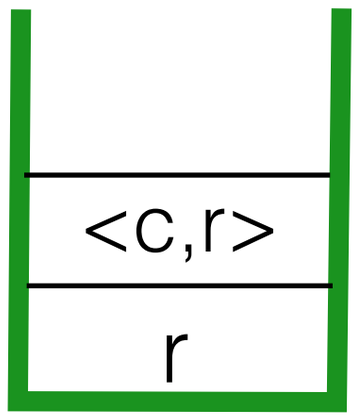
c→(<r,c> r)

<r,c>→(<<c,r>,<r,c>> <c,r>)

<<c,r>,<r,c>>→**reverse**

Logical Form

((reverse **<c,r>**) r)



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

Generated Actions

START → c

c→(<r,c> r)

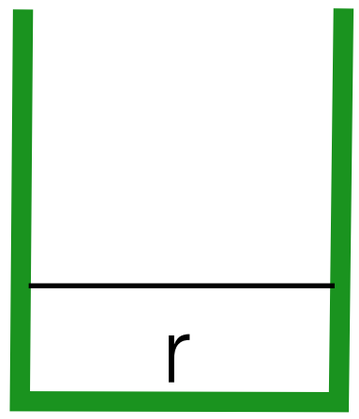
<r,c>→(<<c,r>,<r,c>> <c,r>)

<<c,r>,<r,c>>→**reverse**

<c,r>→**athlete**

Logical Form

((reverse athlete) r)



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

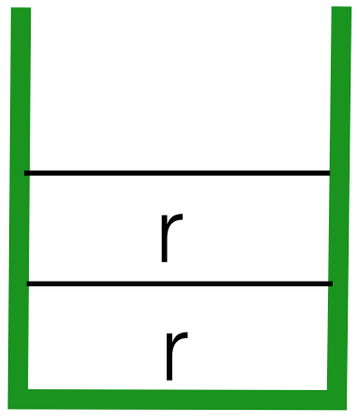
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<r,c>>→**reverse**
<c,r>→**athlete**
r→(<r,<r,r>> r r)
<r,<r,r>>→**and**

Logical Form

((reverse athlete)
 (and **r** r))



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

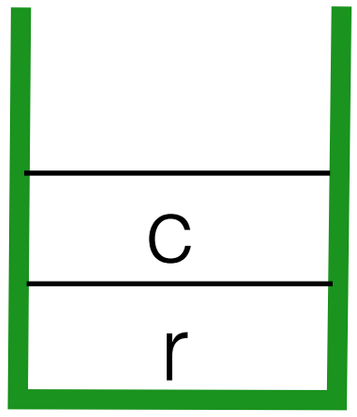
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<r,c>>→**reverse**
<c,r>→**athlete**
r→(<r,<r,r>> r r)
<r,<r,r>>→**and**
r→(<c,r> c)
<c,r>→**nation**

Logical Form

((reverse athlete)
 (and (nation **c**) r))



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

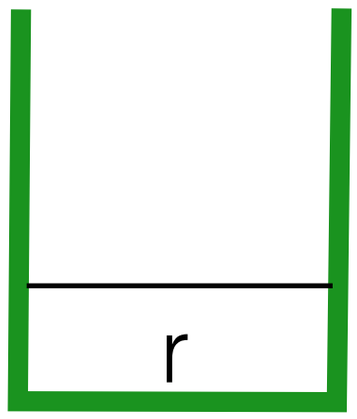
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<r,c>>→**reverse**
<c,r>→**athlete**
r→(<r,<r,r>> r r)
<r,<r,r>>→**and**
r→(<c,r> c)
<c,r>→**nation**
c→**south_korea**

Logical Form

```
((reverse athlete)
 (and (nation south_korea)
      r))
```



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

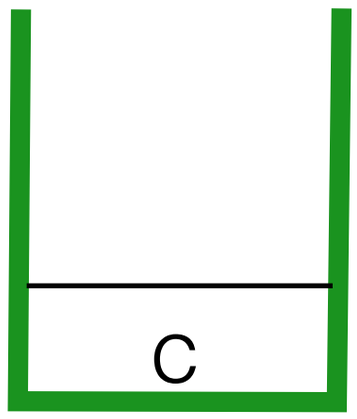
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<r,c>>→**reverse**
<c,r>→**athlete**
r→(<r,<r,r>> r r)
<r,<r,r>>→**and**
r→(<c,r> c)
<c,r>→**nation**
c→**south_korea**
r→(<c,r> c)
<c,r>→**year**

Logical Form

```
((reverse athlete)
 (and (nation south_korea)
      (year c)))
```



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

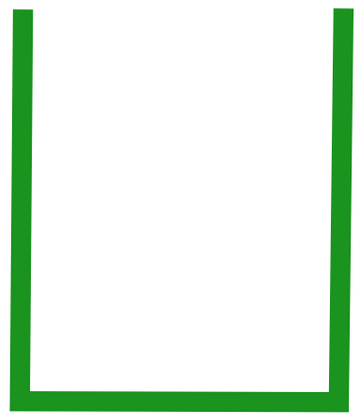
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<r,c>>→**reverse**
<c,r>→**athlete**
r→(<r,<r,r>> r r)
<r,<r,r>>→**and**
r→(<c,r> c)
<c,r>→**nation**
c→**south_korea**
r→(<c,r> c)
<c,r>→**year**
c→(<d,c> d)
<d,c>→(<<c,d>,<d,c>> <c,d>)
<<c,d>,<d,c>>→**reverse**
<c,d>→**date**
d→(**>=** d)
d→**2010.mm.dd**

Logical Form

```
((reverse athlete)
 (and (nation south_korea)
      (year ((reverse date)
             (>= 2010-mm-dd)))
```



Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Grammar-Constrained Decoding

Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<
<c,r>→
r→(<r,<
<r,<r,r>
r→(<c,
<c,r>→
c→**sou**
r→(<c,r>
<c,r>→**year**
c→(<d,c> d)
<d,c>→(<<c,d>,<d,c>> <c,d>)
<<c,d>,<d,c>>→**reverse**
<c,d>→**date**
d→(**>=** d)
d→**2010.mm.dd**

Logical Form

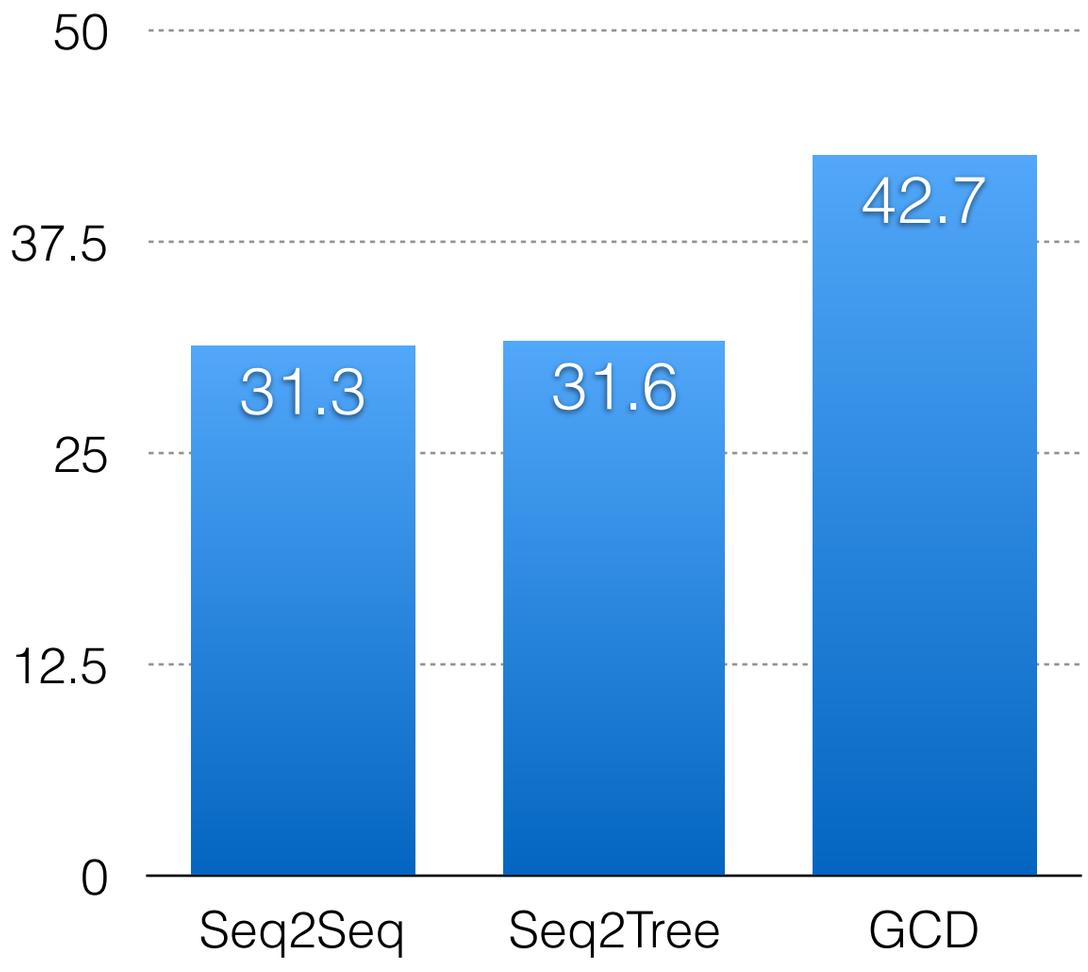
((reverse athlete)
 (and (nation south_korea)
  (year ((reverse date)
d))))

- ☑ Need not explicitly model matching parentheses
- ☑ Syntactically valid trees
- ☑ Semantically valid trees

Non-terminal Stack

[Krishnamurthy, Dasigi and Gardner, 2017]

Empirical Comparison with Seq2Seq and Seq2Tree on WikiTableQuestions



[Krishnamurthy, Dasigi and Gardner, 2017]

Summary

- Constraining output forces decoder to generate only valid outputs
- Impose hard constraints instead of hoping the model would learn them
- Various hard constraints depending on output space
  - Token-level decoding (Seq2Tree, sketches, etc)
  - Grammar-based constraints