Inferring Logical Forms From Denotations

Ice Pasupat and Percy Liang
Stanford University

ACL 2016
Monday, August 8, 2016

Semantic Parsing

Utterance

“Where did the last 1st place finish occur?”

World

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

from WikiTableQuestions dataset (Pasupat & Liang, 2015); simplified

Semantic Parsing

Parse utterances into executable logical forms

“Where did the last 1st place finish occur?”

R[Venue].argmax(Position.1st, Index)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Semantic Parsing

Parse utterances into executable logical forms

"Where did the last 1st place finish occur?"

R[Venue].argmax(Position.1st, Index)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Denotation

Semantic Parsing

Parse utterances into executable logical forms

"Where did the last 1st place finish occur?"

R[Venue].argmax(Position.1st, Index)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Semantic Parsing

Parse utterances into executable logical forms

“Where did the last 1st place finish occur?”

R[Venue].argmax(Position.1st, Index)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Semantic Parsing

Parse utterances into executable logical forms

"Where did the last 1st place finish occur?"

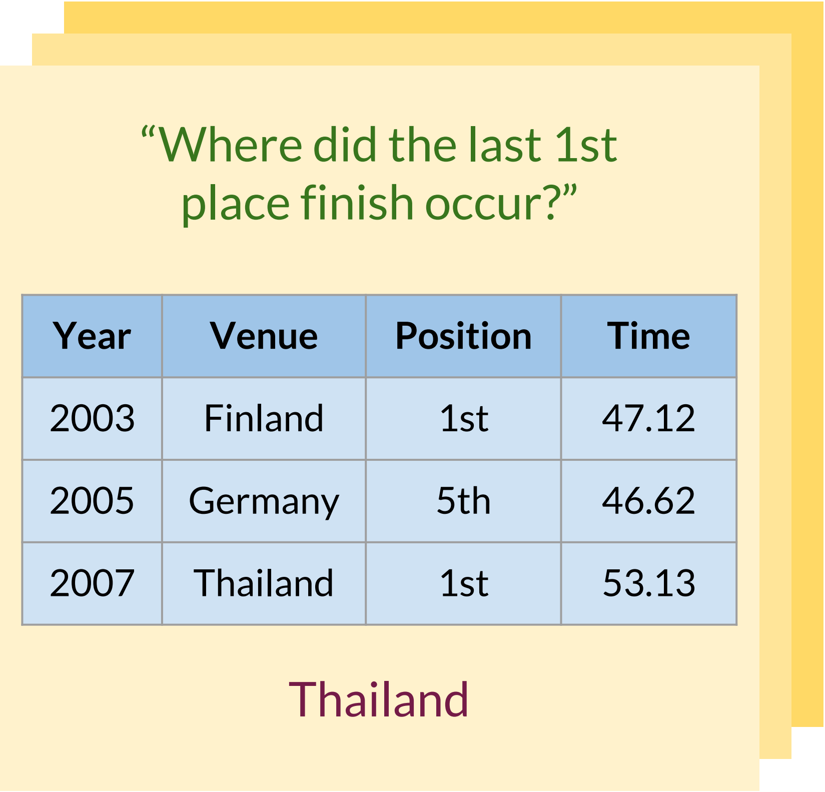
R[Venue].argmax(Position.1st, Index)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Semantic Parsing

Setting: Learn a semantic parser from denotations

Training Data



"Where did the last 1st place finish occur?"

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Thailand

The logical form is latent!

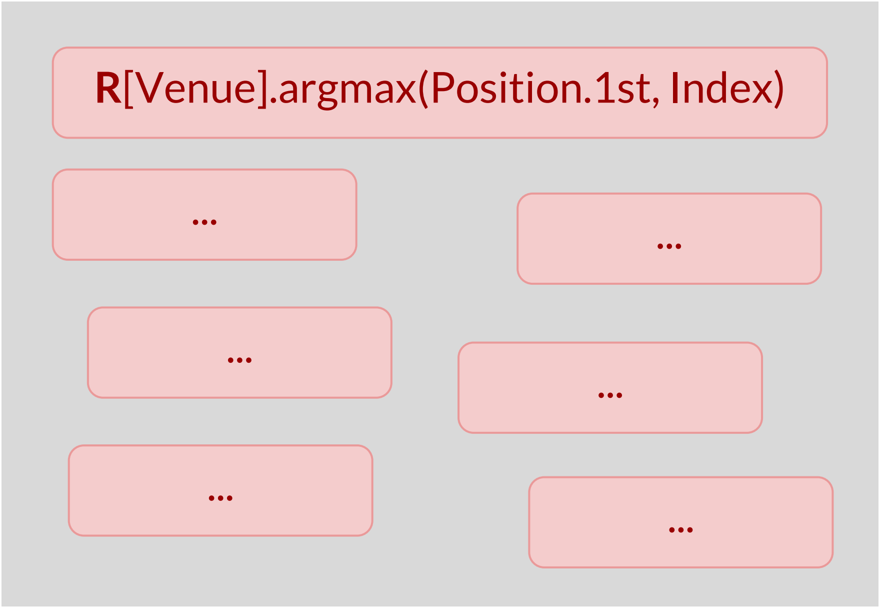
Space of Logical Forms

For each input utterance and world, we can set the space of logical forms that we want the semantic parser to consider

"Where did the last 1st place finish occur?"

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

logical forms



Restricting the Space of Logical Forms

- Restrict which predicates can appear

“… Germany …” → Venue.Germany

“… German …” → ~~Venue.Germany~~

- Only allow certain logical form compositions

$$\mathrm{F}(\mathrm{Values}_1) - \mathrm{F}(\mathrm{Values}_2) \rightarrow \mathrm{Values}$$

Must have a parallel structure

(Pasupat & Liang, 2015)

Restricting the Space of Logical Forms

- Restrict which predicates can appear

“… Germany …” → Venue.Germany

“… German …” → ~~Venue.Germany~~

- Only allow certain logical form compositions

$$F(\text{Values}_1) - F(\text{Values}_2) \rightarrow \text{Values}$$

Must have a parallel structure

+ Easier to learn

– Low coverage

(Pasupat & Liang, 2015)

Expanding the Space of Logical Forms

- Less restriction on predicates

"… Germany …" → Venue.Germany

"… German …" → Venue.Germany

- Very generic logical form composition

(Details in the paper)

$\text{Set}_1 - \text{Set}_2 \rightarrow \text{Set}$

Anything goes!

+ Higher coverage

Expanding the Space of Logical Forms

- Less restriction on predicates

“… Germany …” → Venue.Germany

“… German …” → Venue.Germany

- Very generic logical form composition

(Details in the paper)

$\text{Set}_1 - \text{Set}_2 \rightarrow \text{Set}$

Anything goes!

+ Higher coverage

– Two new challenges …

Space of Logical Forms

A semantic parser defines a distribution on logical forms

Consistent Logical Forms

A logical form is consistent if it executes to the target denotation

During training, a semantic parser learns to maximize the probability of consistent logical forms

all logical forms
consistent

Consistent Logical Forms

Challenge 1: Identifying this region of consistent logical forms during training = finding needles in a haystack

all logical forms
consistent
(actual size)

Spurious Logical Forms

“Where did the last 1st place finish occur?”

✓ **R**[Venue].argmax(Position.1st, Index)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Spurious Logical Forms

Sometimes a consistent logical form is spurious: it gets the correct denotation for a wrong reason

“Where did the last 1st place finish occur?”

⚠ **R**[Venue].argmax(Position.1st, Time)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Spurious Logical Forms

Challenge 2: With the expanded space of logical forms, we get even more spurious logical forms!

"Where did the last 1st place finish occur?"

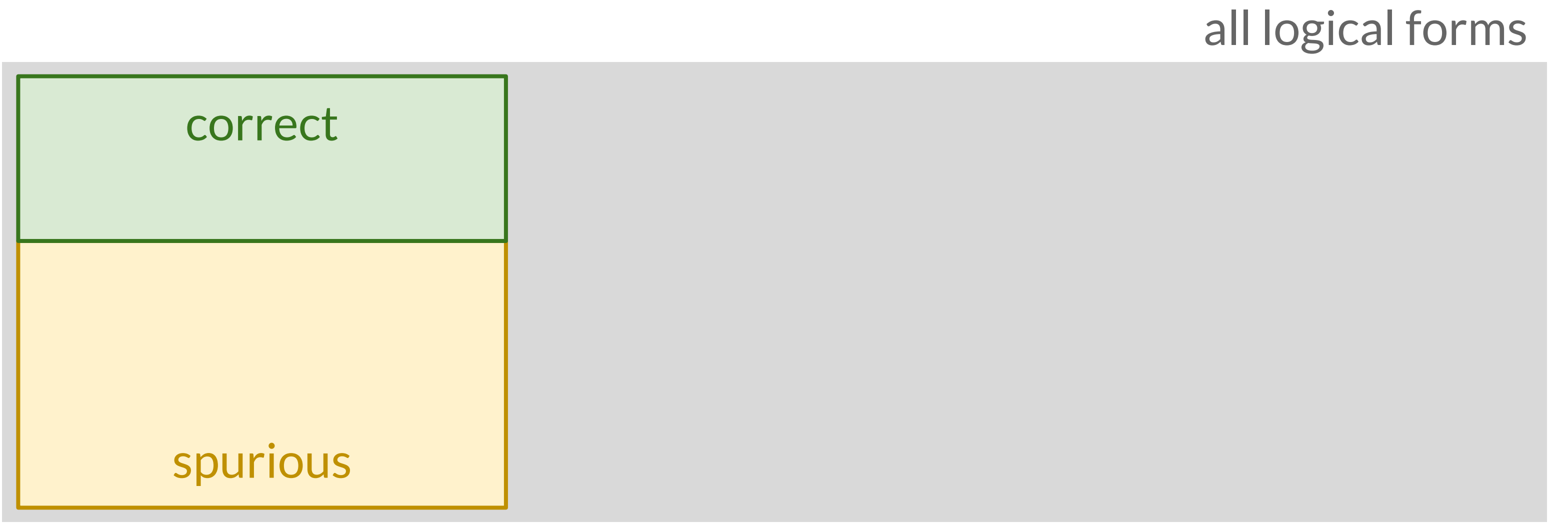
⚠ **R**[Venue].**R**[Next].Year.avg(**R**[Year].Type.Row)

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Spurious Logical Forms

Challenge 2: With the expanded space of logical forms, we get even more spurious logical forms!

- These spurious logical forms can hurt learning since they give misleading signals



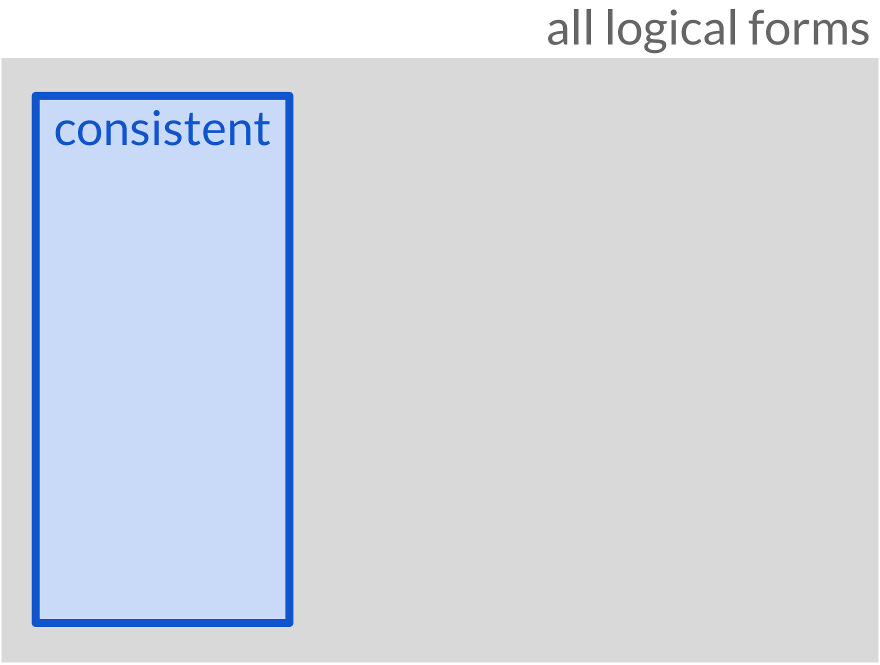
Challenge 1: Enumerate Consistent LFs

Given a training example (an utterance, a world, and the target denotation), find all consistent logical forms

"Where did the last 1st place finish occur?"

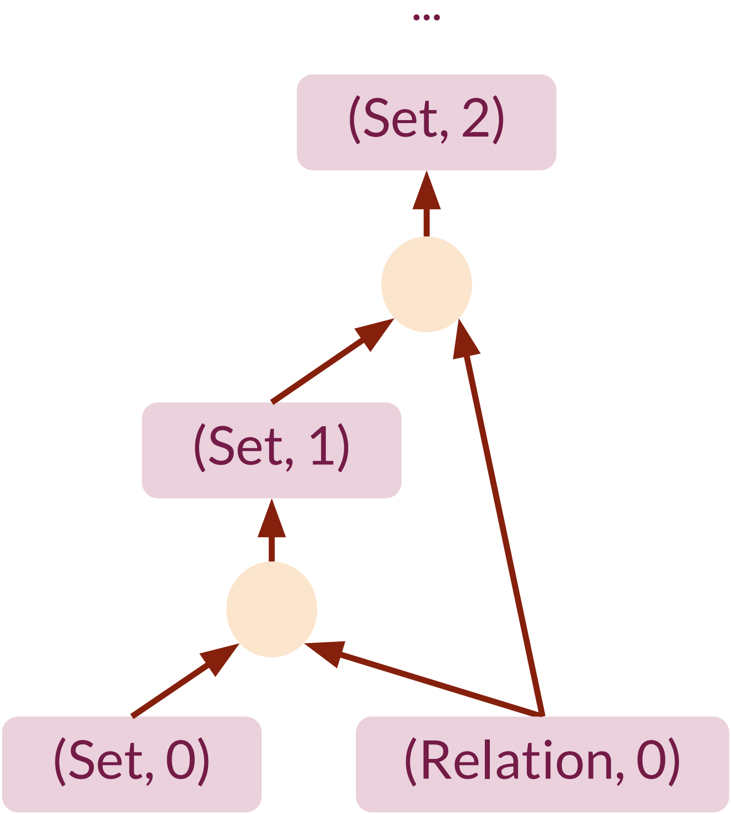
Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Thailand



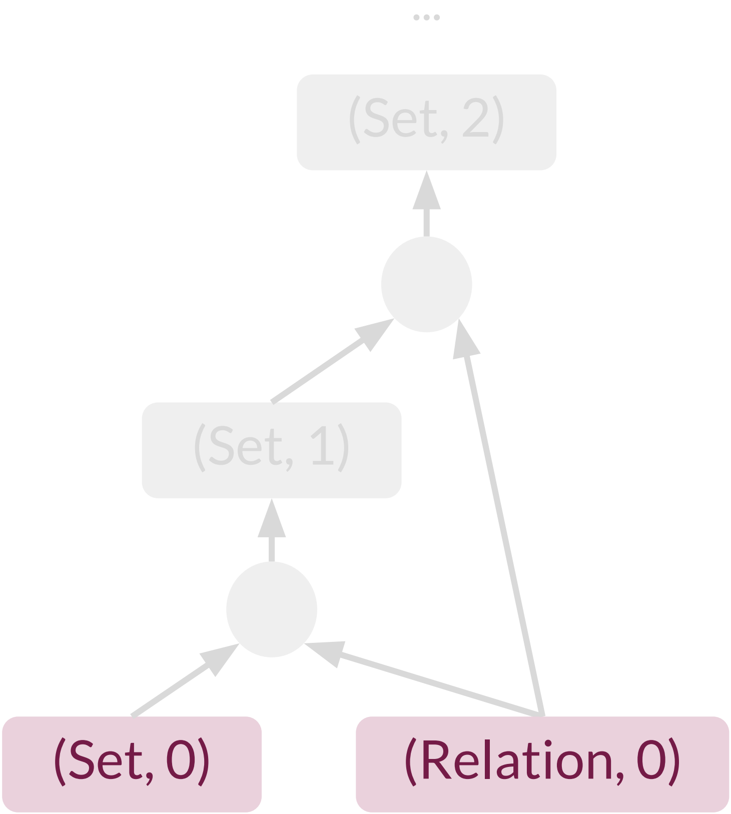
Review: Beam Search

Idea: Compose logical forms of increasing sizes then keep the consistent final logical forms



Review: Beam Search

A cell (*c*, *s*) contains logical forms with the same category *c* and "size" *s*



...

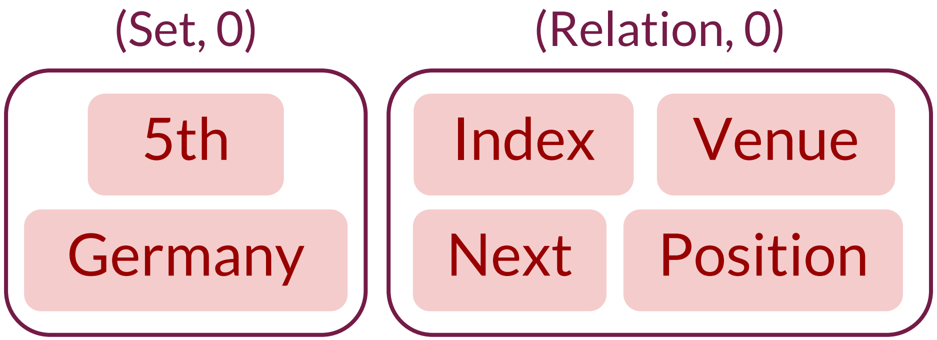
(Set, 2)

(Set, 1)

(Set, 0)

(Relation, 0)

[Similar to cell (*c*, *i*, *j*) in CYK algorithm for syntactic parsing]



(Set, 0)

5th

Germany

(Relation, 0)

Index

Venue

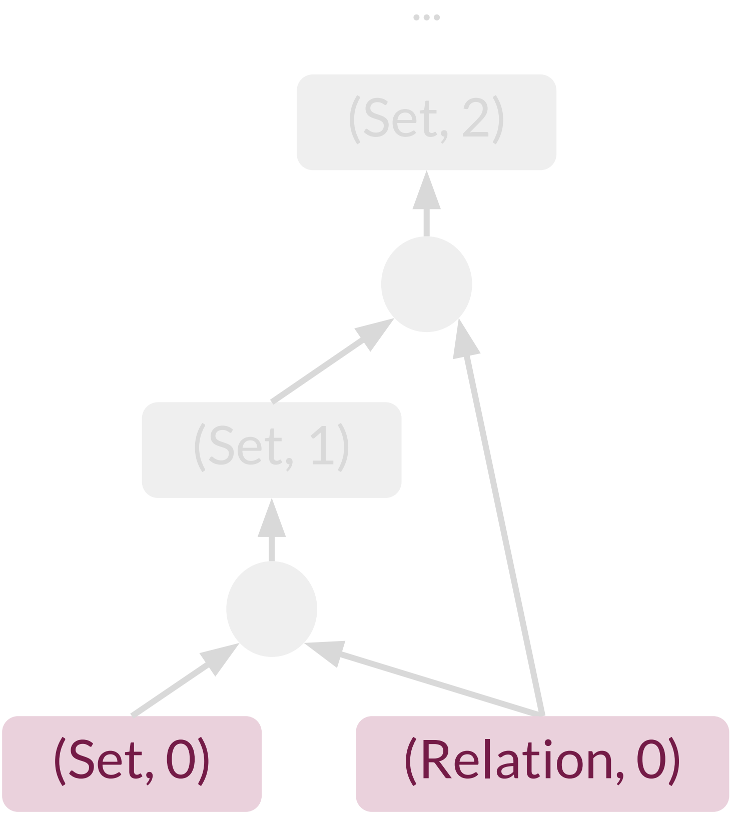
Next

Position

Review: Beam Search

Start from base predicates (size = 0)

For the sake of illustration, assume any cell / column can become a base predicate



(Set, 0)

5th

Germany

(Relation, 0)

Index

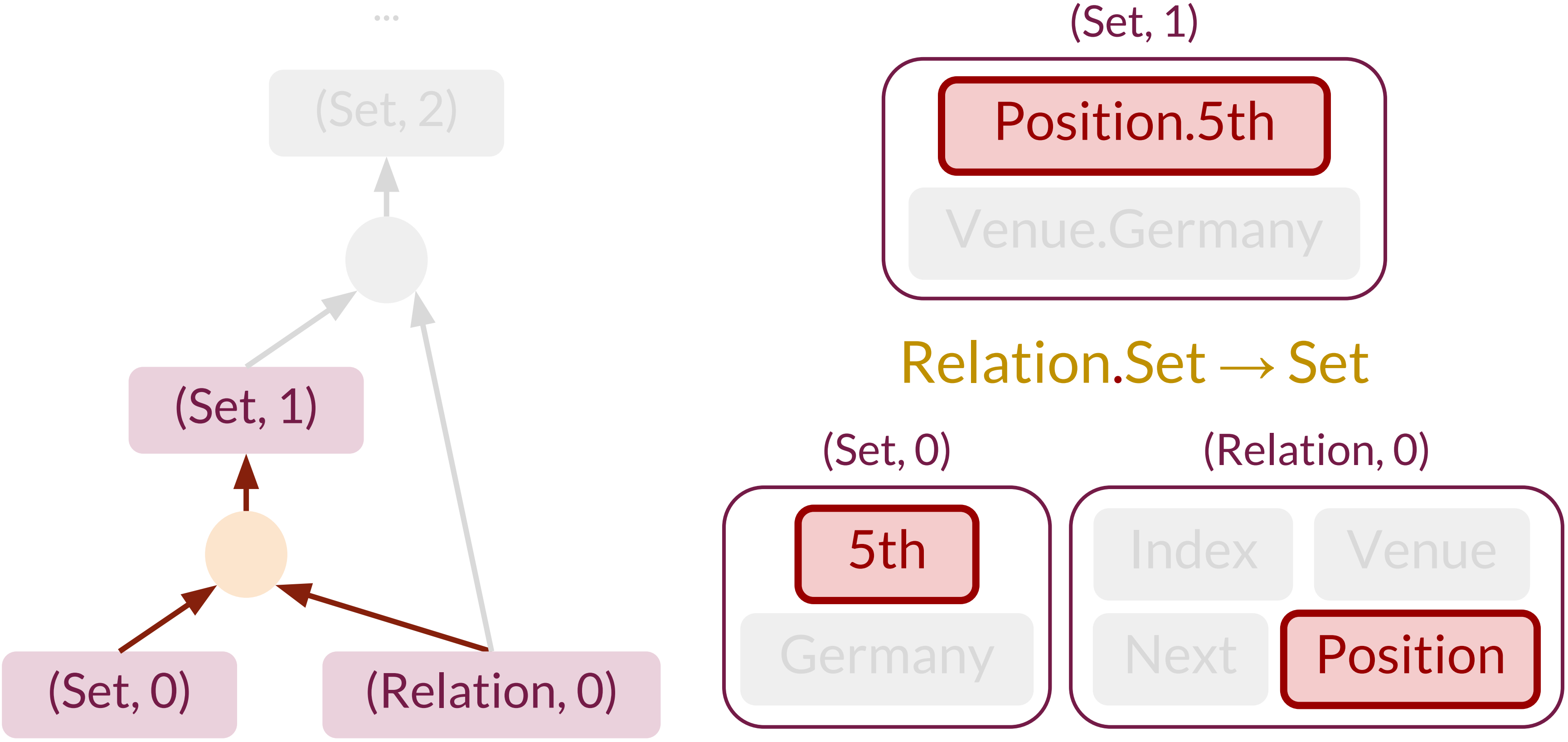
Venue

Next

Position

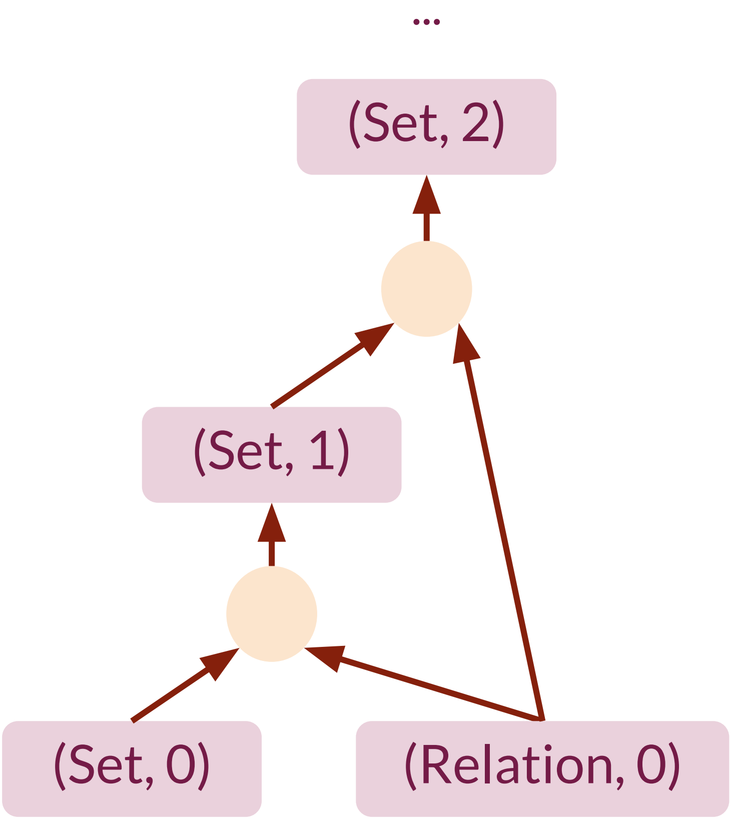
Review: Beam Search

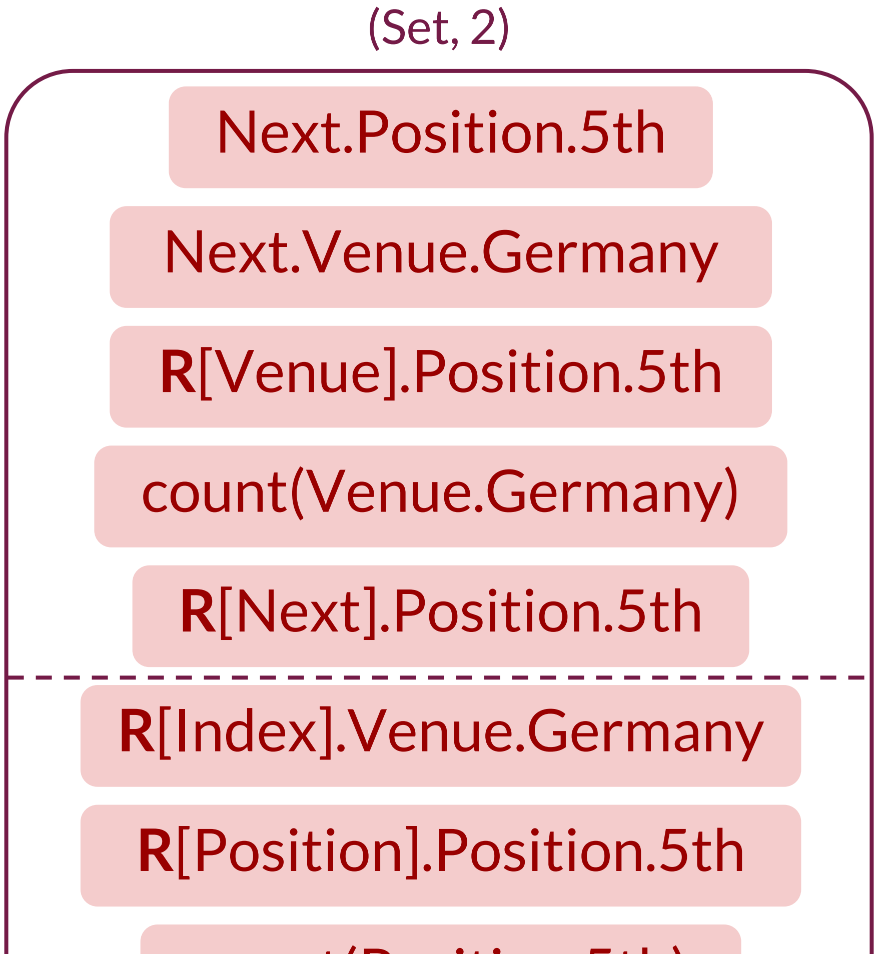
Start from base predicates (size = 0) and compose partial logical forms of increasing sizes



Review: Beam Search

To control the search space, we prune cells to a fixed beam size



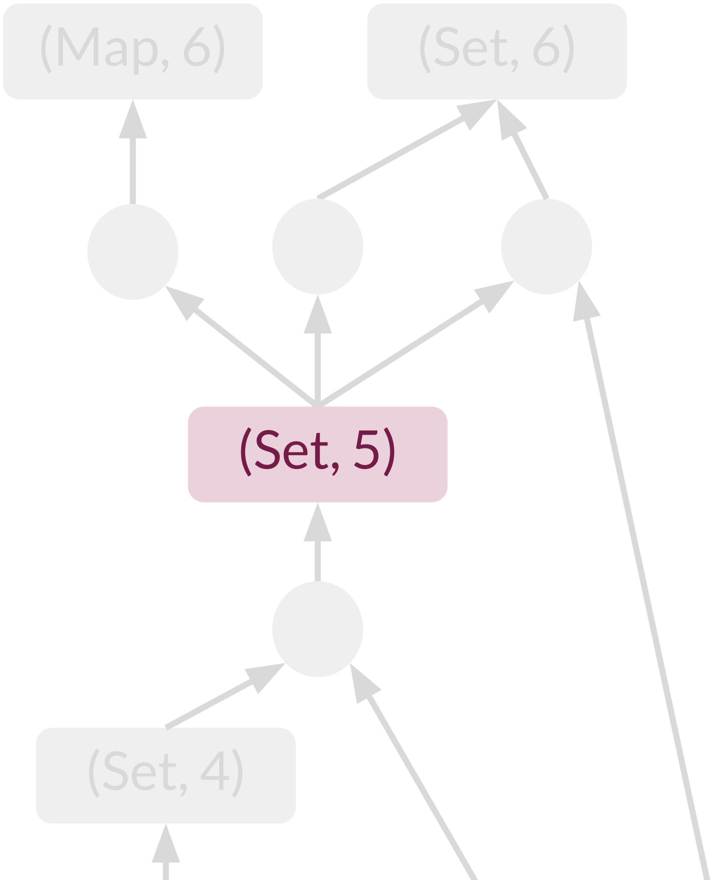

(Set, 2)

- Next.Position.5th
- Next.Venue.Germany
- **R**[Venue].Position.5th
- count(Venue.Germany)
- **R**[Next].Position.5th

- **R**[Index].Venue.Germany
- **R**[Position].Position.5th
- count(Position.5th)

Review: Beam Search

Finally, collect complete logical forms that execute to the target denotation



(Set, 5)

R[Venue].Index.min(R[Year].Position.1st)

R[Venue].argmax(Position.1st, Time)

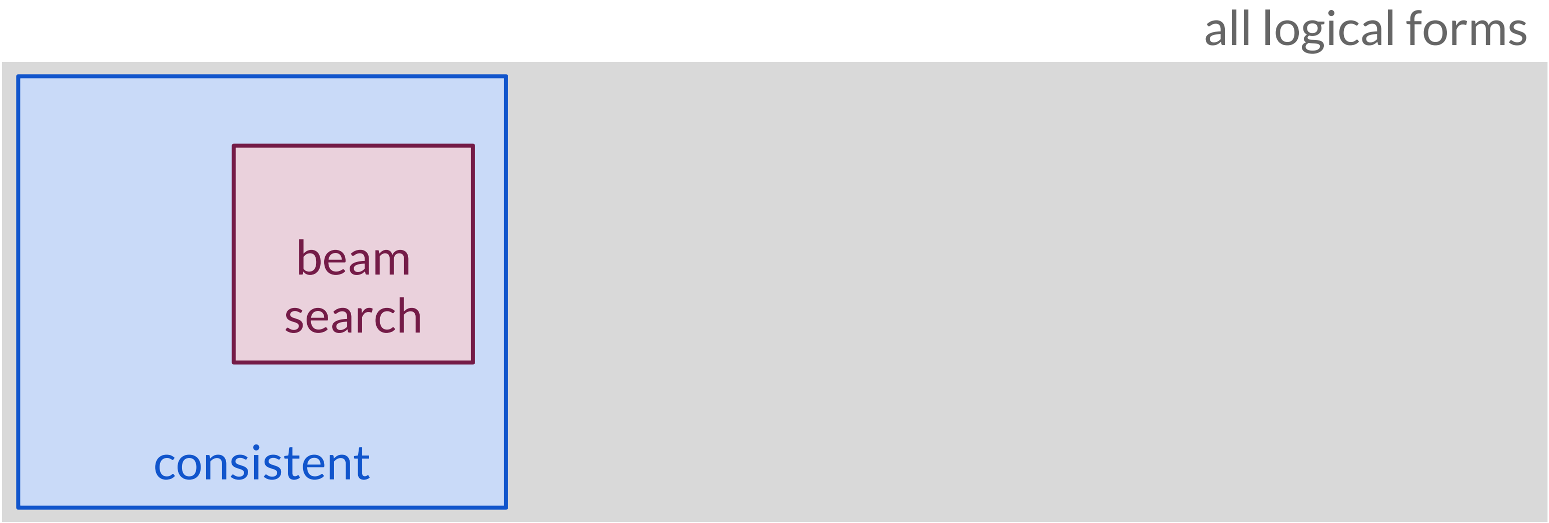
R[Year].argmin(Position.1st, Time)

count(Venue.Germany) - count(…)

R[Venue].argmax(Position.1st, Index)

Review: Beam Search

Only generates a partial list of logical forms
→ Misses many consistent logical forms



Better Way to Control Search Space

Observation: Many logical forms execute to the same denotation

Index.1

Position.5th

Venue.Germany

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

all execute to $\{r_1\}$

Better Way to Control Search Space

If we only care about denotations, these logical forms are interchangeable

R[Time]. Index.1

R[Time]. Position.5th

R[Time]. Venue.Germany

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

all execute to {46.62}

Better Way to Control Search Space

So if we collapse them into one “meta” logical form, the search space will be reduced

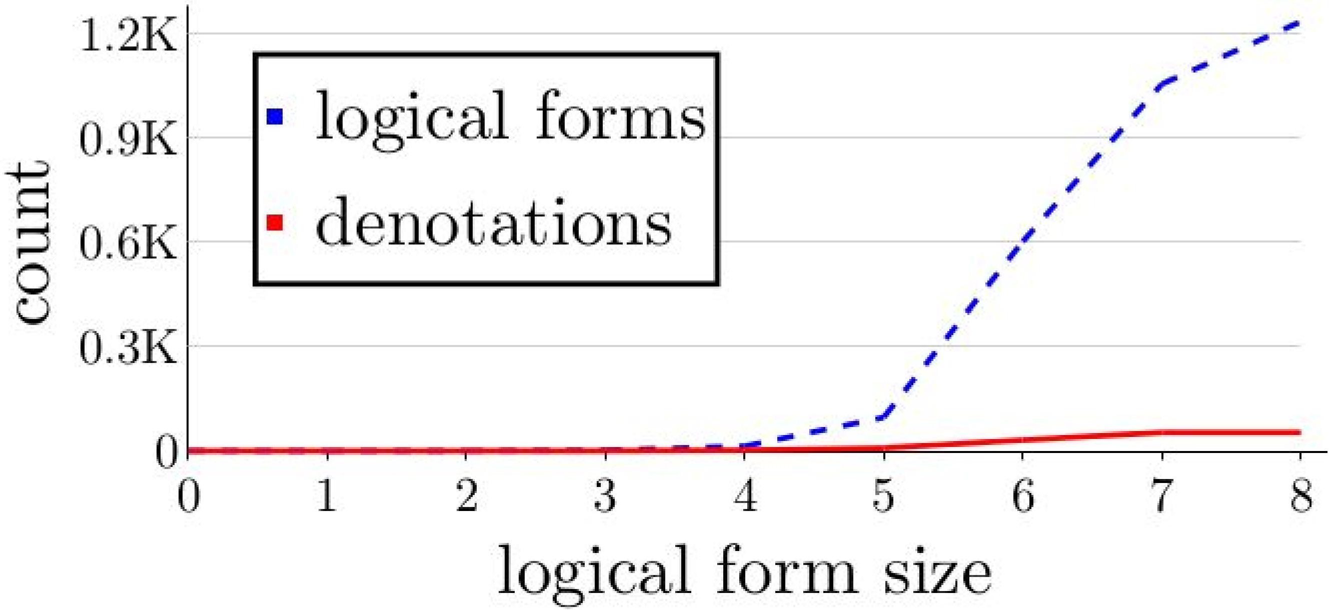
R[Time]. $\{r_1\}$

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

executes to
{46.62}

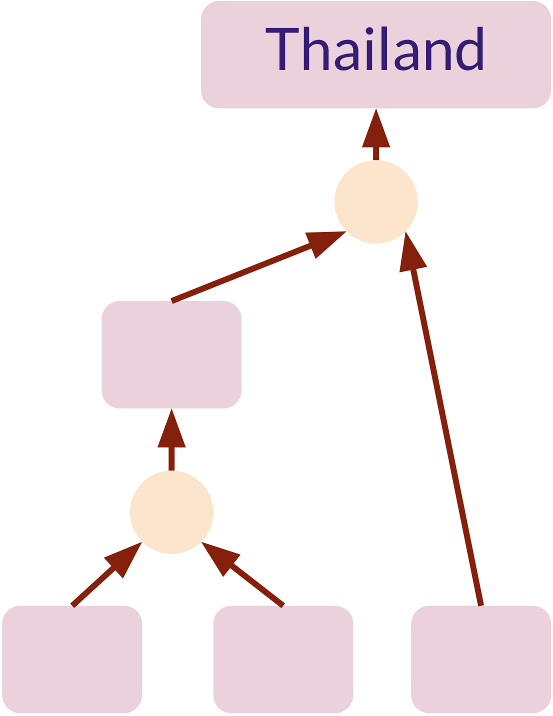
Better Way to Control Search Space

So if we collapse them into one "meta" logical form, the search space will be reduced a lot!

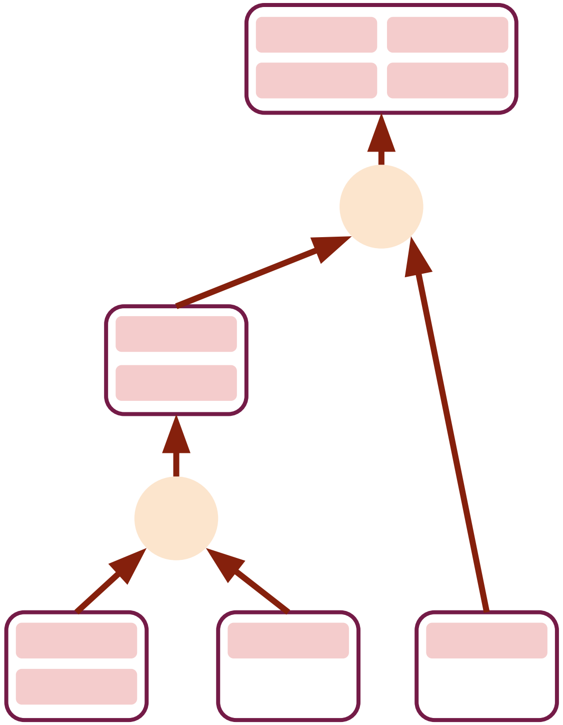


Dynamic Programming on Denotations

Step 1: Build a parse chart leading to the target denotation



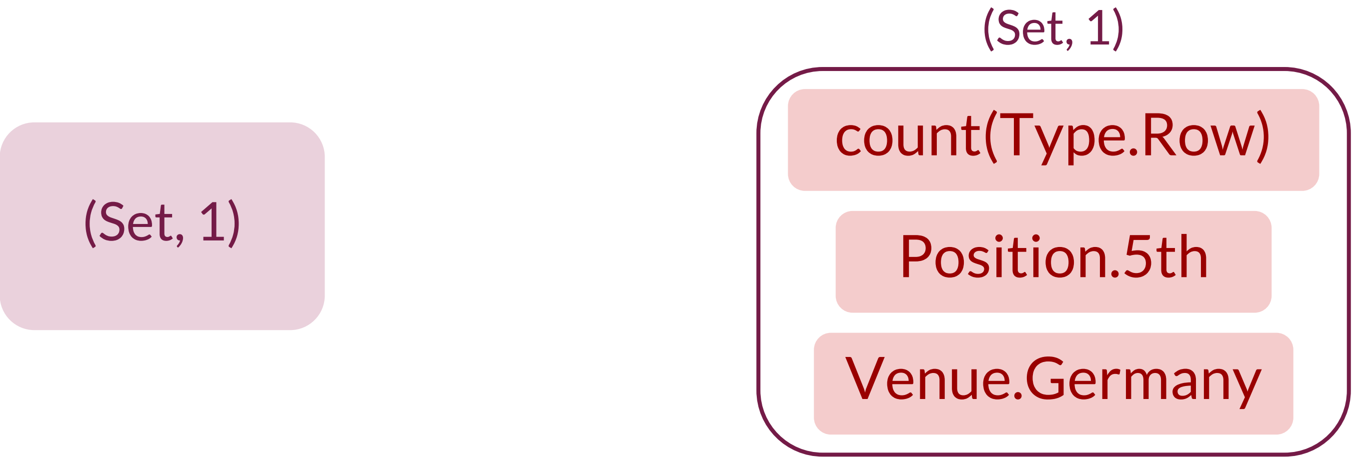
Step 2: Retrieve actual logical forms



Dynamic Programming on Denotations

Step 1: Build a parse chart to the target denotation

- Group logical forms based on denotations
- Each cell becomes (category, size, denotation)



(Set, 1)

(Set, 1)

count(Type.Row)

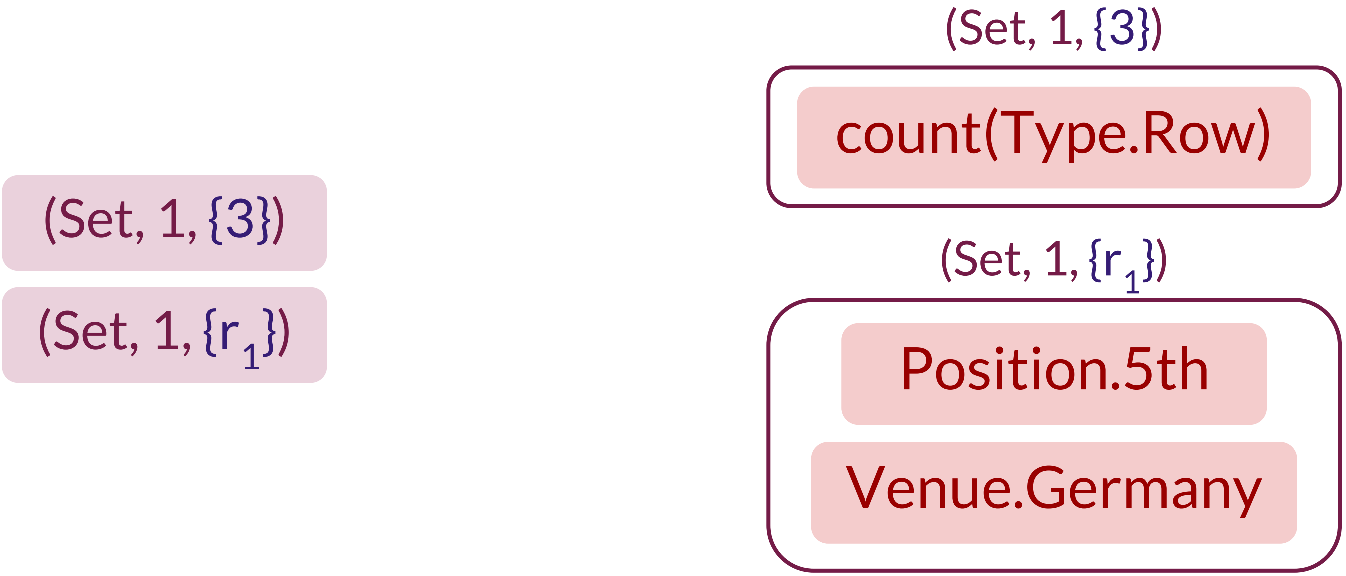
Position.5th

Venue.Germany

Dynamic Programming on Denotations

Step 1: Build a parse chart to the target denotation

- Group logical forms based on denotations
- Each cell becomes (category, size, denotation)



(Set, 1, {3})

(Set, 1, {r_1})

(Set, 1, {3})

count(Type.Row)

(Set, 1, {r_1})

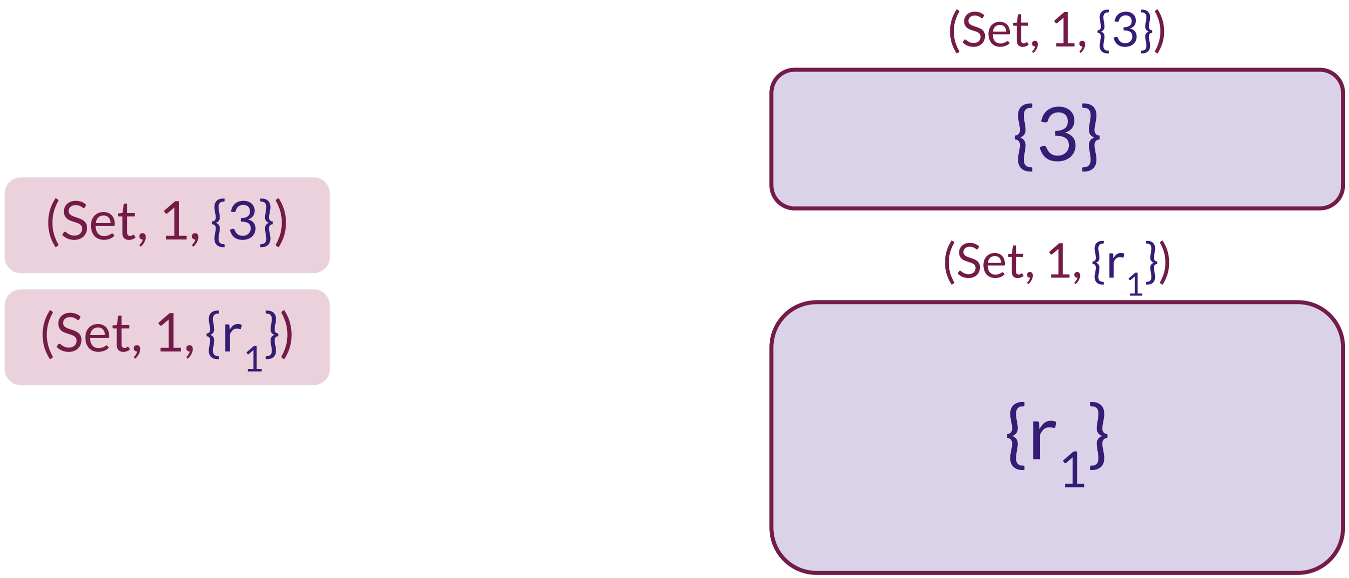
Position.5th

Venue.Germany

Dynamic Programming on Denotations

Step 1: Build a parse chart to the target denotation

- Since we only care about denotations, we can collapse logical forms in each cell



(Set, 1, {3})

(Set, 1, $\{r_1\}$)

(Set, 1, {3})

{3}

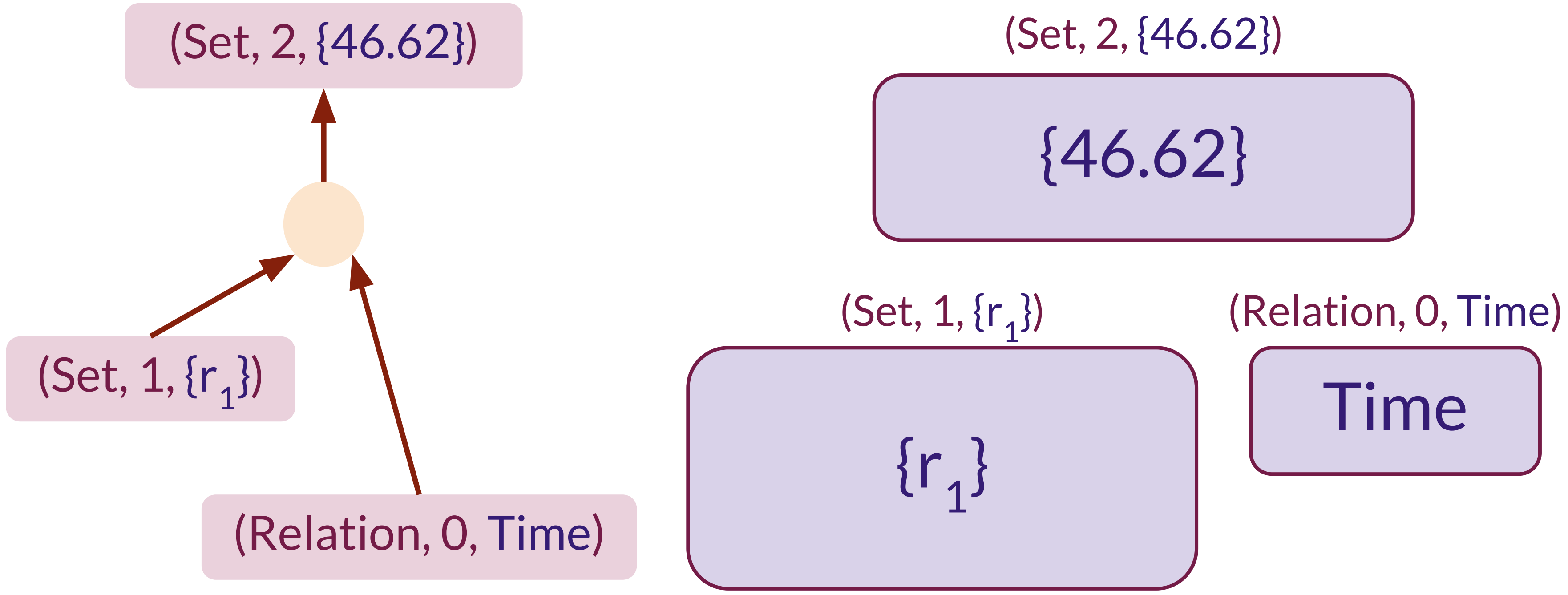
(Set, 1, $\{r_1\}$)

$\{r_1\}$

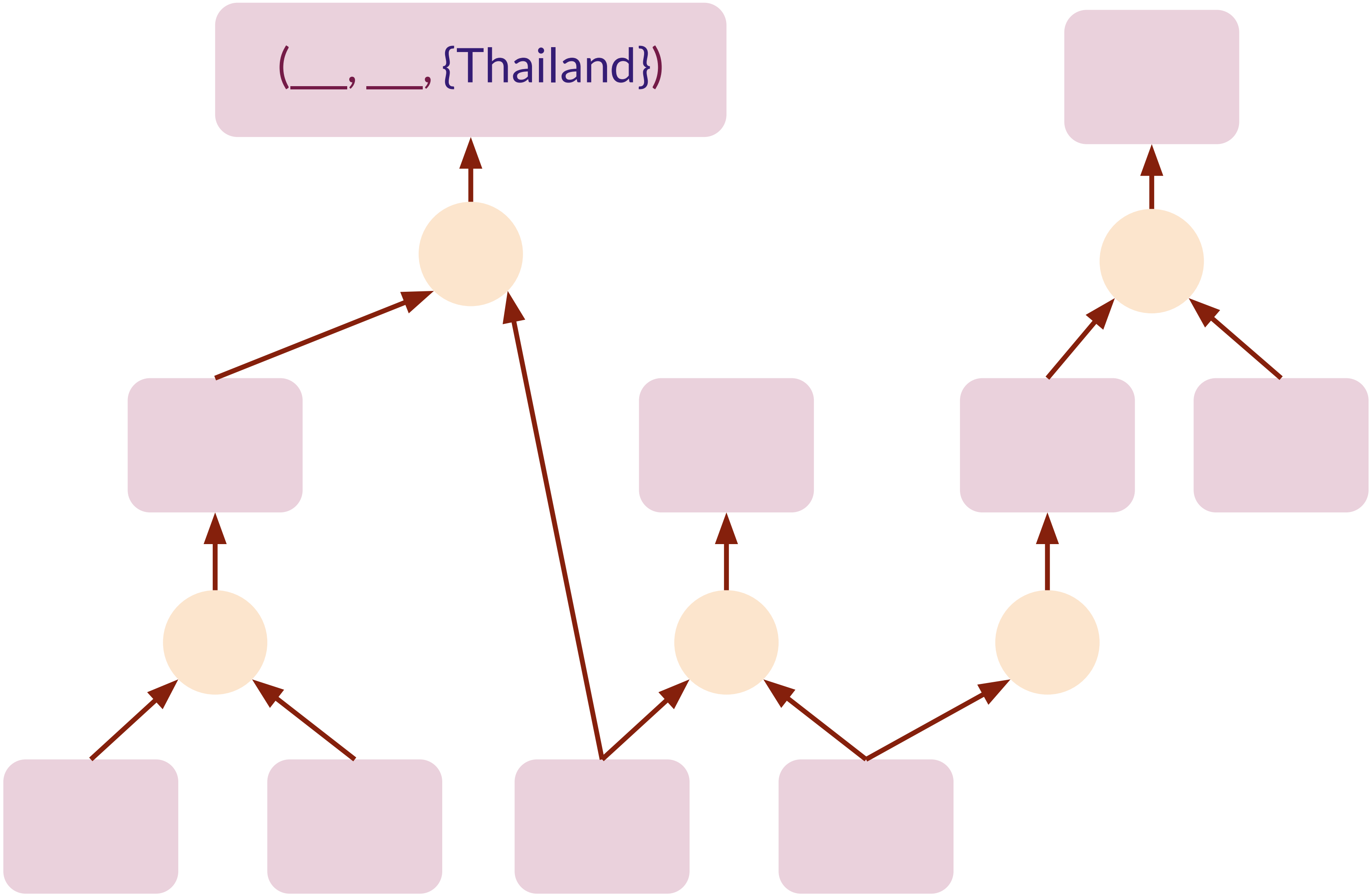
Dynamic Programming on Denotations

Step 1: Build a parse chart to the target denotation

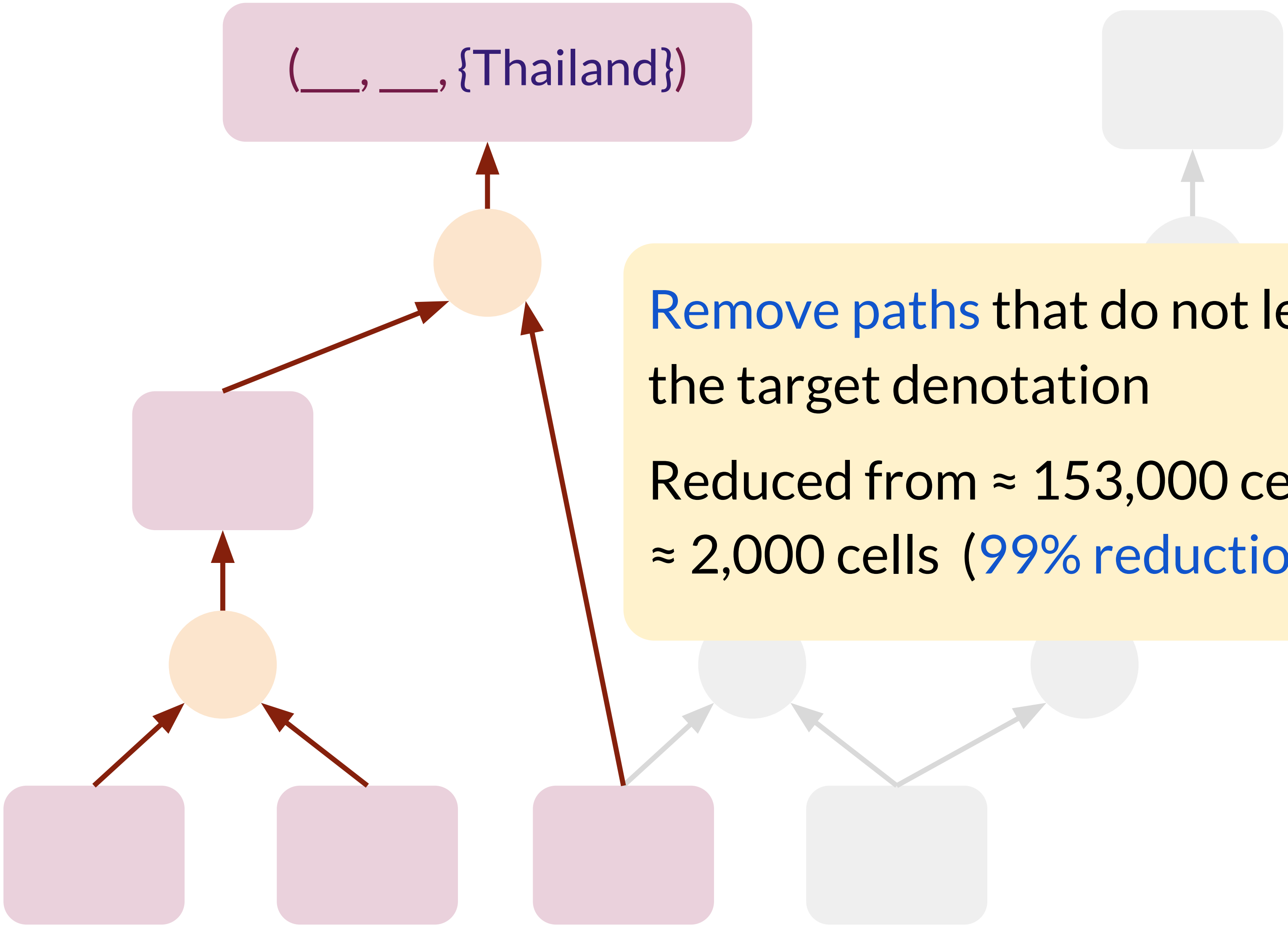
- Since we only care about denotations, we can collapse logical forms in each cell



Dynamic Programming on Denotations



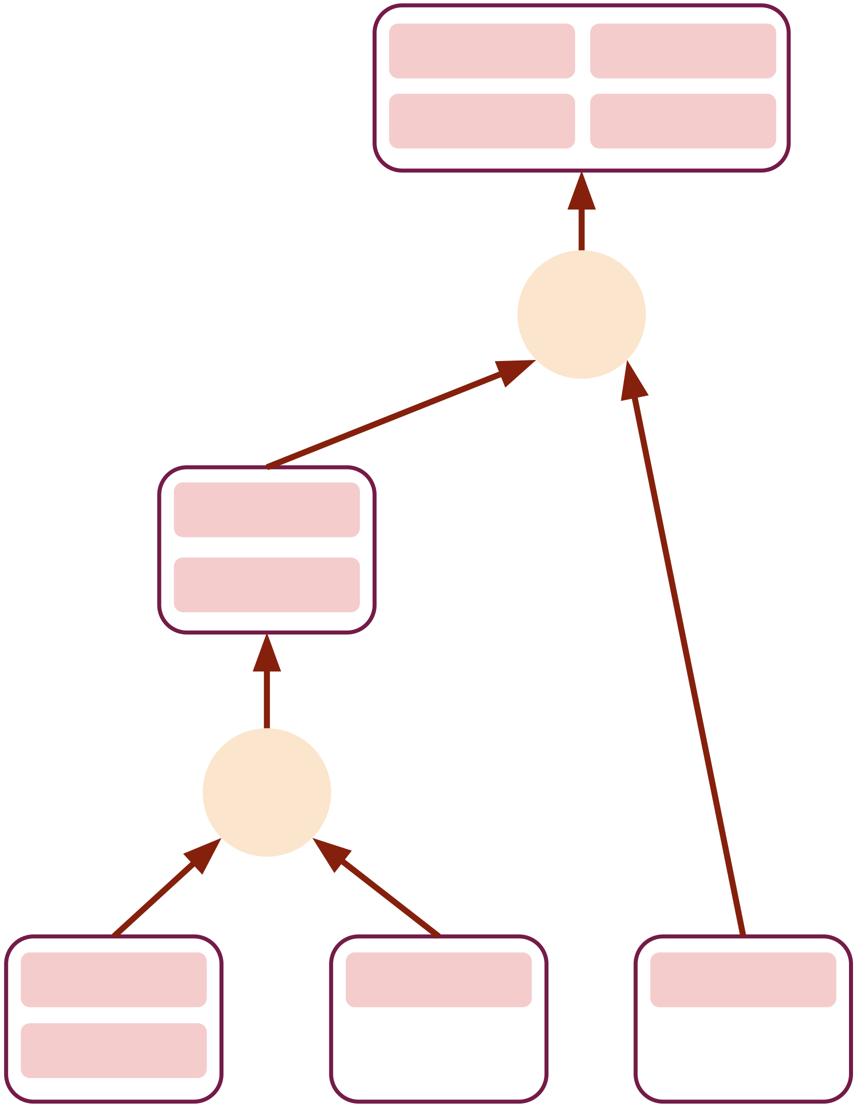
Dynamic Programming on Denotations



Remove paths that do not lead to the target denotation

Reduced from ≈ 153,000 cells to ≈ 2,000 cells (99% reduction!)

Dynamic Programming on Denotations

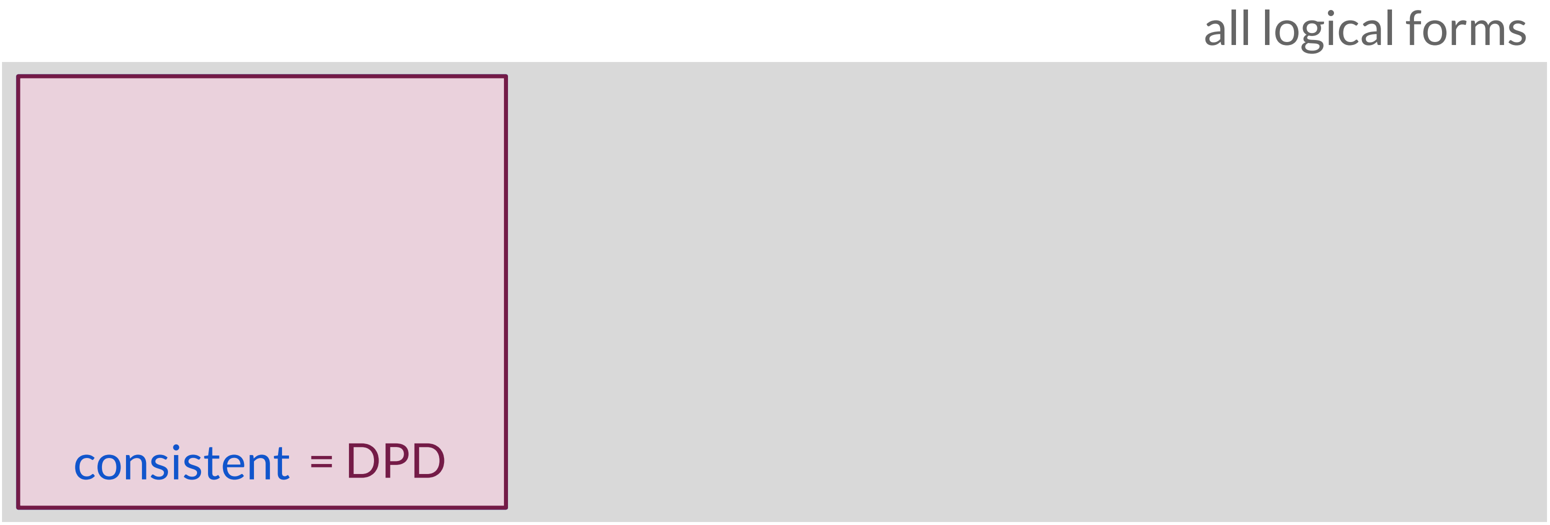


Step 2: To get the actual logical forms, uncollapse the “meta” logical forms by following the backpointers

Dynamic Programming on Denotations

We have eliminated a lot of cells, so it is possible to exhaustively enumerate logical forms along the remaining paths up to a certain logical form size

(in most cases)



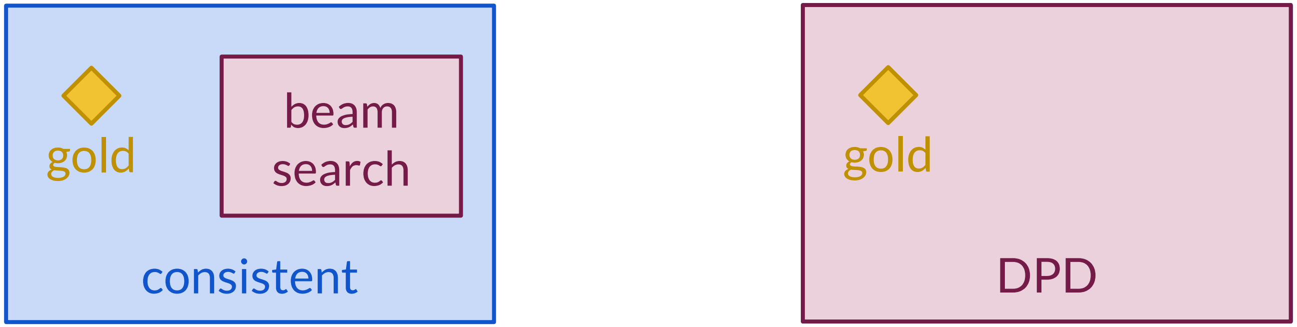
all logical forms

consistent = DPD

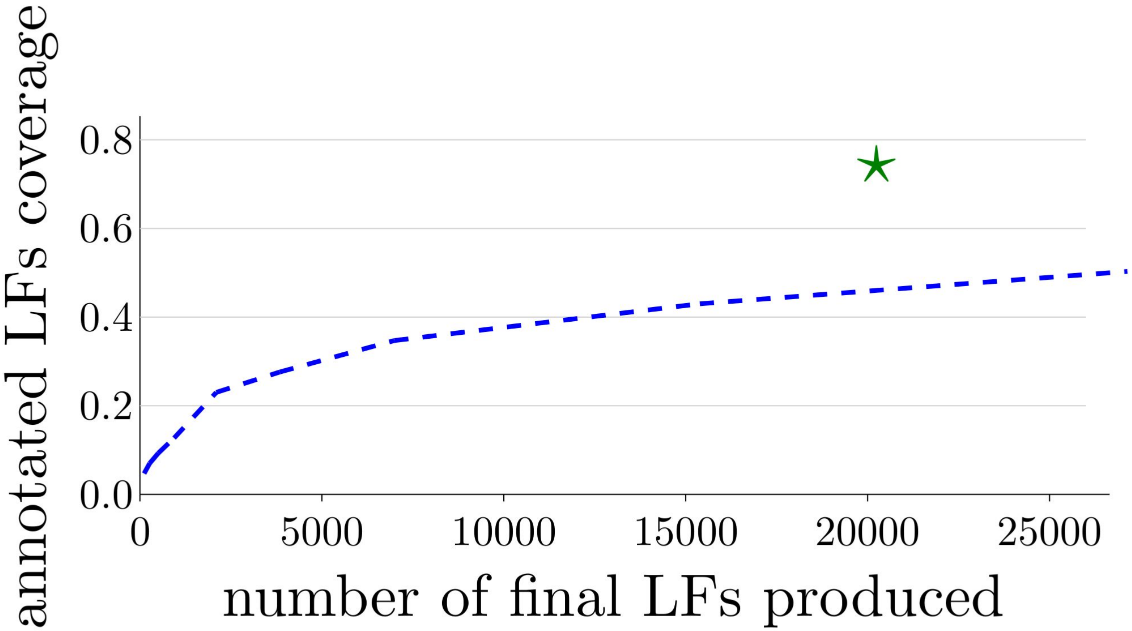
Dynamic Programming on Denotations

Experiment: For each of 300 examples:

- Annotate the example with a gold logical form (consistent and correct)
- Test whether the algorithm can generate the gold logical form



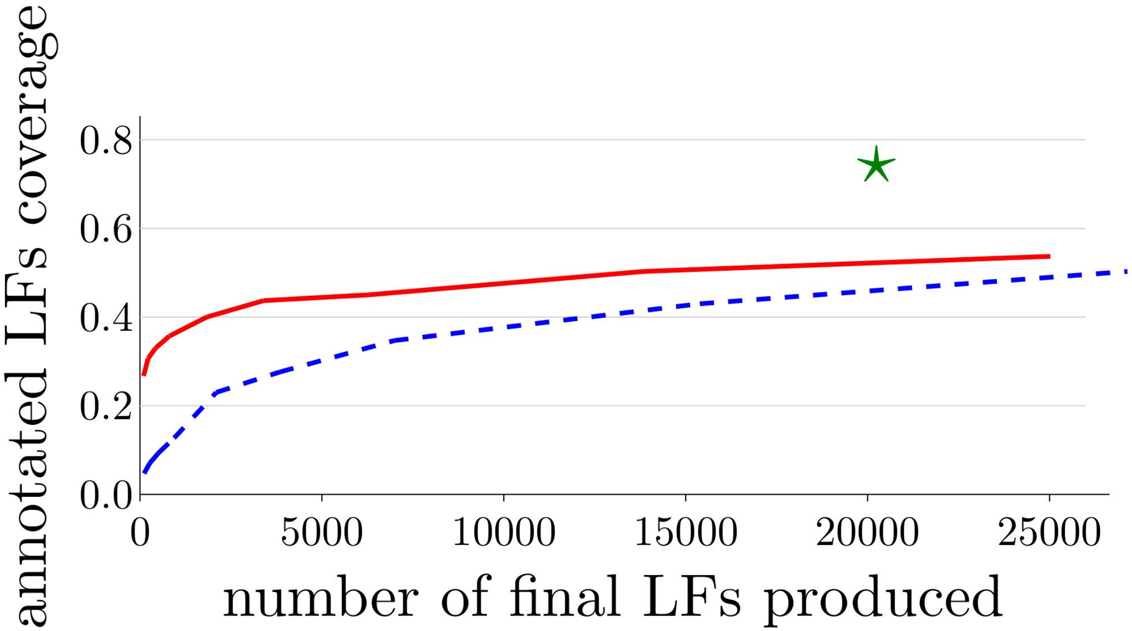
Dynamic Programming on Denotations



★ Dynamic programming on denotations

■ Uninitialized beam search

Dynamic Programming on Denotations



★ Dynamic programming on denotations

🟦 Uninitialized beam search

🟥 Initialized beam search

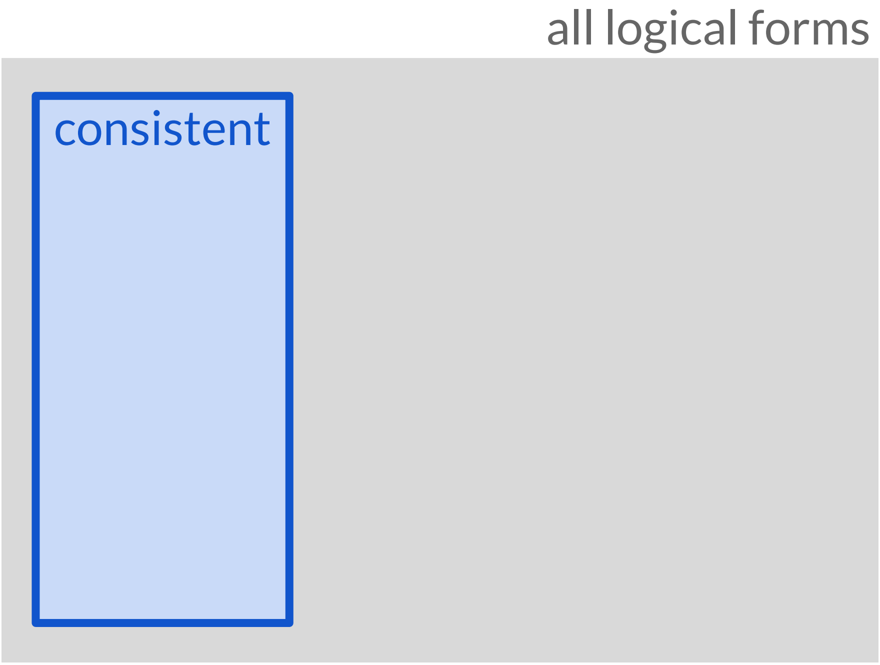
Challenge 1: Enumerate Consistent LFs

Given a training example (an utterance, a world, and the target denotation), find all consistent logical forms

"Where did the last 1st place finish occur?"

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Thailand



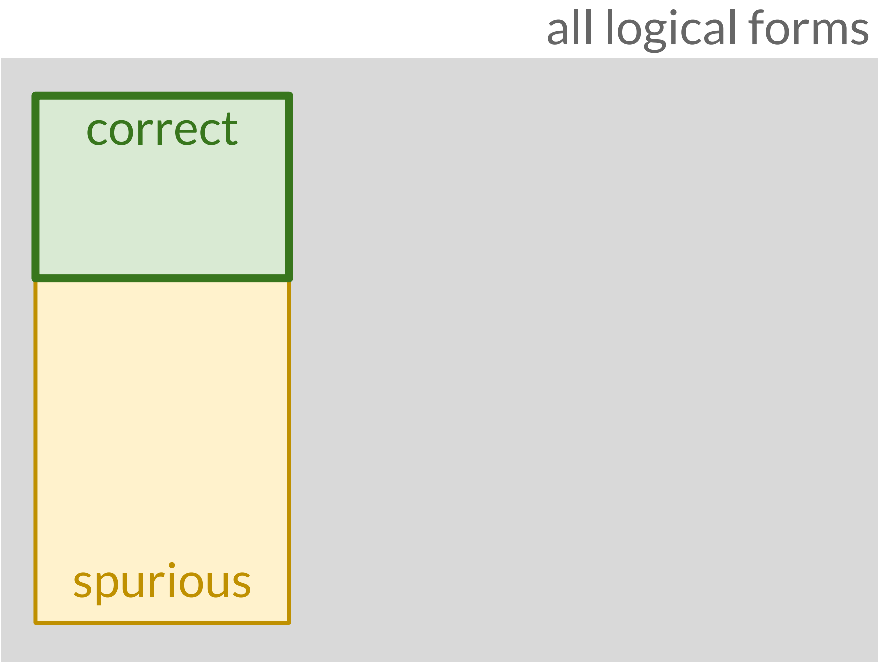
Challenge 2: Prune Spurious LFs

Given the set of consistent logical forms (Task 1), prune out spurious logical forms

“Where did the last 1st place finish occur?”

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

Thailand



Challenge 2: Prune Spurious LFs

Given the set of consistent logical forms (Task 1), prune out spurious logical forms

“Where did the last 1st place finish occur?”

Year	Venue	Position	Time
2003	Finland	1st	47.12
2005	Germany	5th	46.62
2007	Thailand	1st	53.13

✓	**R**[Venue].argmax(Position.1st, Index)	Thailand
⚠	**R**[Venue].argmax(Position.1st, Time)	Thailand

Fictitious Worlds

Intuition: Correct logical forms should give a correct denotation when the world slightly changes

"Where did the last 1st place finish occur?"

Keep sorted columns sorted

Year	Venue	Position	Time
2003	Thailand	1st	53.13
2005	Finland	1st	47.12
2007	Germany	5th	46.62

Resample cells in other columns

✓ **R**[Venue].argmax(Position.1st, Index) — Finland

⚠ **R**[Venue].argmax(Position.1st, Time) — Thailand

Fictitious Worlds

Generate fictitious worlds and execute the logical forms on them

	w
z_1	Thailand
z_2	Thailand
z_3	Thailand
z_4	Thailand
z_5	Thailand

Fictitious Worlds

Generate fictitious worlds and execute the logical forms on them

	w	w_1
z_1	Thailand	
z_2	Thailand	
z_3	Thailand	
z_4	Thailand	
z_5	Thailand	

Fictitious Worlds

Generate fictitious worlds and execute the logical forms on them

	w	w_1
z_1	Thailand	Finland
z_2	Thailand	Thailand
z_3	Thailand	Finland
z_4	Thailand	Finland
z_5	Thailand	Germany

Fictitious Worlds

We can ask humans to answer the question based on the fictitious worlds

	w	w_1
z_1	Thailand	Finland
z_2	Thailand	Thailand
z_3	Thailand	Finland
z_4	Thailand	Finland
z_5	Thailand	Germany
Human	Thailand	Finland

Fictitious Worlds

We can ask humans to answer the question based on the fictitious worlds

	w	w_1
z_1	Thailand	Finland
⚠✖	Thailand	Thailand
z_3	Thailand	Finland
z_4	Thailand	Finland
⚠✖	Thailand	Germany
Human	Thailand	Finland

Fictitious Worlds

We can ask humans to answer the question based on the fictitious worlds

	w	w_1	w_2
z_1	Thailand	Finland	Germany
⚠ ✖ z_2	Thailand	Thailand	Germany
⚠ ✖ z_3	Thailand	Finland	Thailand
z_4	Thailand	Finland	Germany
⚠ ✖ z_5	Thailand	Germany	Finland
Human	Thailand	Finland	Germany

Fictitious Worlds

We can ask humans to answer the question based on the fictitious worlds

	w	w_1	w_2
✓ z_1	Thailand	Finland	Germany
⚠ ✗	Thailand	Thailand	Germany
⚠ ✗	Thailand	Finland	Thailand
✓ z_4	Thailand	Finland	Germany
⚠ ✗	Thailand	Germany	Finland
Human	Thailand	Finland	Germany

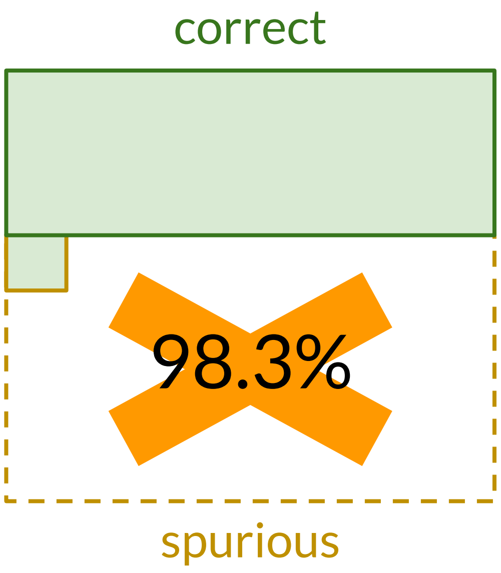
Fictitious Worlds

- Similar to test cases for programs
- In practice:
  - Generate 30 fictitious worlds
  - Choose a subset of 5 worlds that maximizes the expected information gained from the workers' answers (Details in the paper)
  - Ask crowd workers to answer the question based on the chosen worlds

Fictitious Worlds

Results with "Ideal Worker" (emulate a human by executing the gold logical form)

- Ruled out 98.3% of spurious logical forms

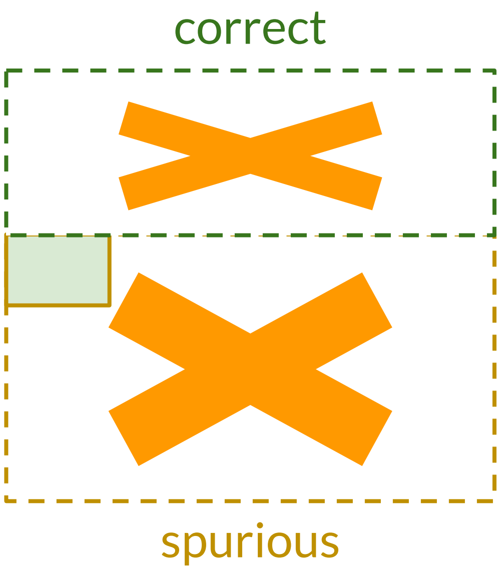


Fictitious Worlds

Results with Actual Workers (Mechanical Turk)

- Pruned correct LFs in 20% of examples, many of which are due to semantic confusions

Name	Birth	Death
Beethoven	1756	1750
Mozart	1770	1827
Bach	1685	1791

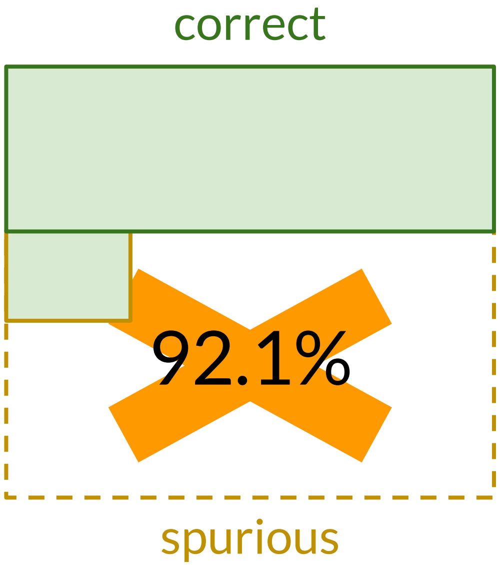


Fictitious Worlds

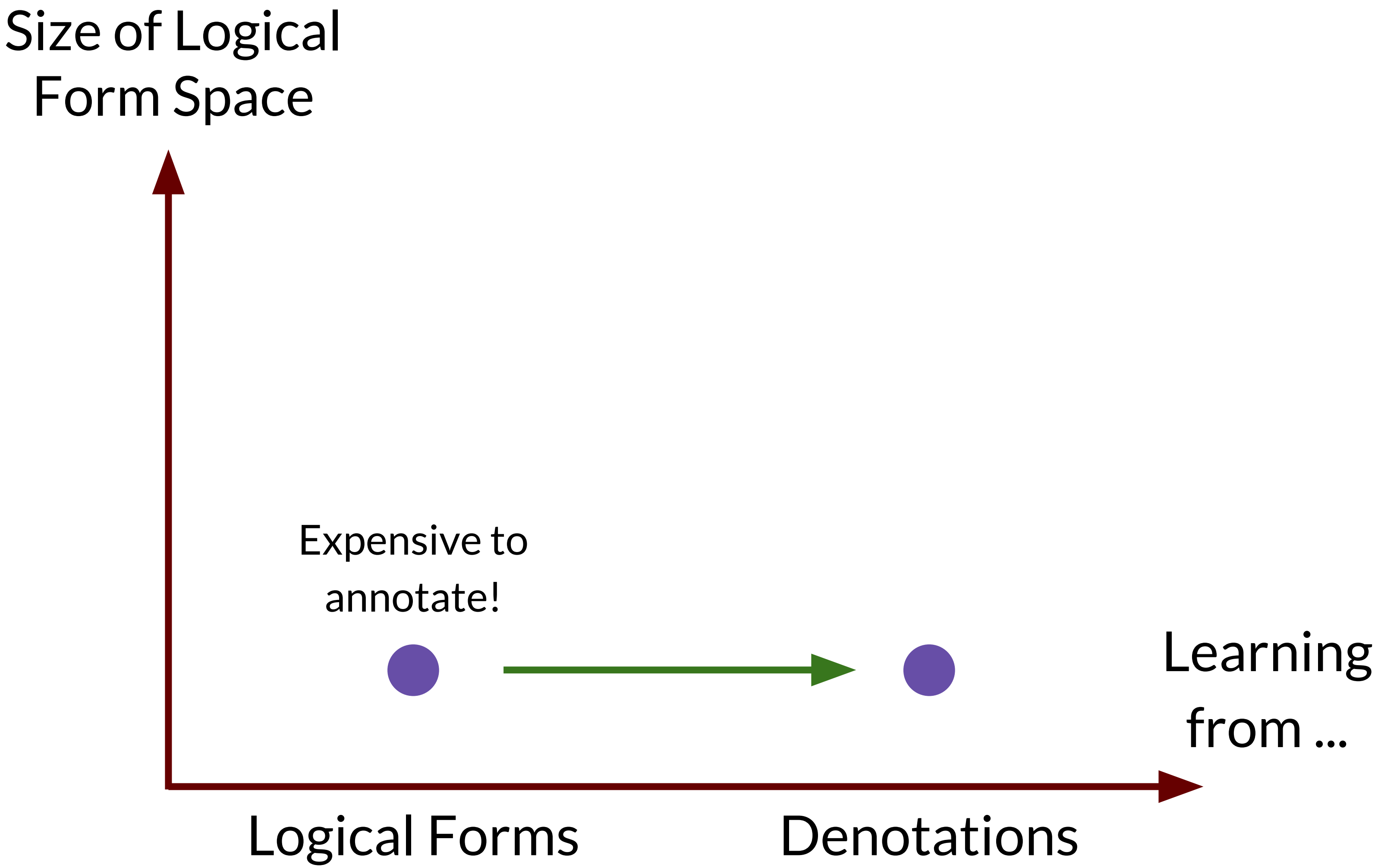
Results with Actual Workers (Mechanical Turk)

- Pruned correct LFs in 20% of examples, many of which are due to semantic confusions
- For other examples, could prune out 92.1% of spurious logical forms

Name	Birth	Death
Beethoven	1770	1750
Mozart	1756	1827
Bach	1685	1791



Summary



Size of Logical Form Space

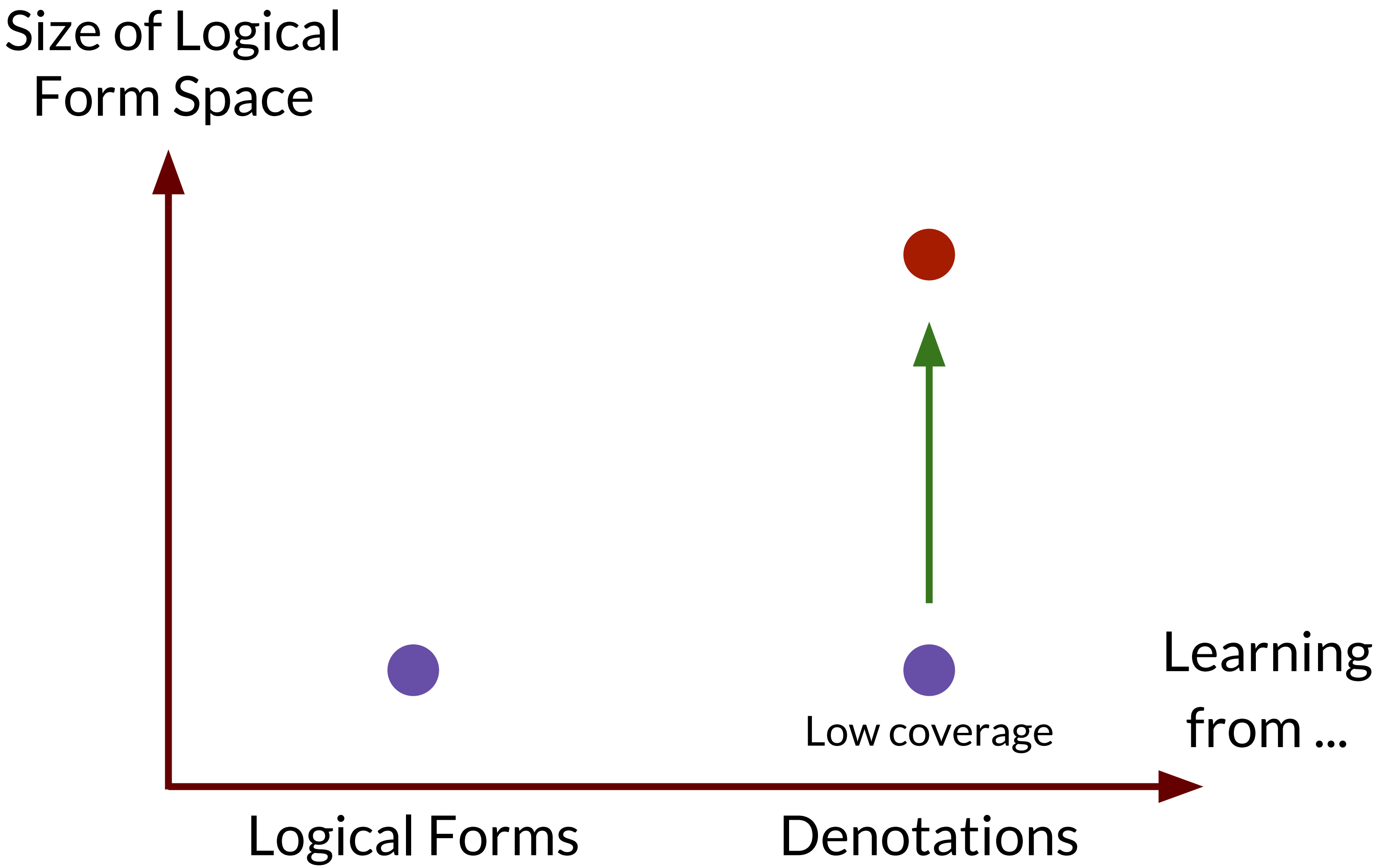
Expensive to annotate!

Learning from ...

Logical Forms

Denotations

Summary



Size of Logical Form Space

Low coverage

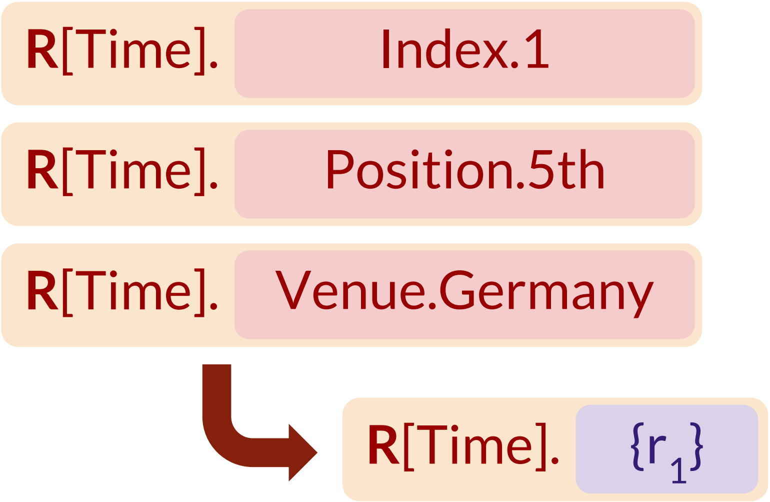
Learning from ...

Logical Forms

Denotations

Summary

Two techniques to handle larger logical form spaces



R[Time]. Index.1

R[Time]. Position.5th

R[Time]. Venue.Germany

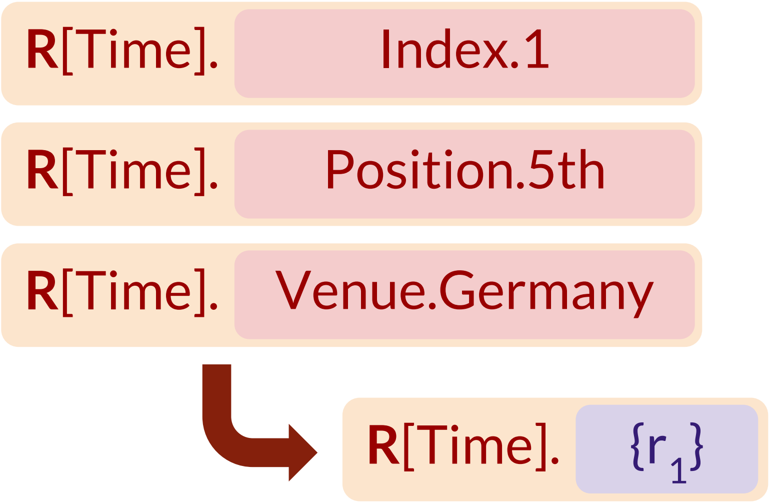
R[Time]. $\{r_1\}$

Dynamic programming on denotations:

Use intermediate denotations to control search space

Summary

Two techniques to handle larger logical form spaces



R[Time]. Index.1

R[Time]. Position.5th

R[Time]. Venue.Germany

R[Time]. $\{r_1\}$

Year	Venue	Position	Time
2003	Thailand	1st	53.13
2005	Finland	1st	47.12
2007	Germany	5th	46.62

✓ argmax on Index	Finland
⚠ argmax on Time	Thailand

Dynamic programming on denotations:
Use intermediate denotations to control search space

Fictitious worlds:
Use denotations on mutated worlds to detect spurious logical forms

Code, data, and reproducible results:

CodaLab

http://tinyurl.com/acl2016-inferring

WikiTableQuestions Dataset:

http://tinyurl.com/wikitablequestions

Thank you!