Lecture 8: Speed-up Synthesis with Abstract Semantics

Yu Feng

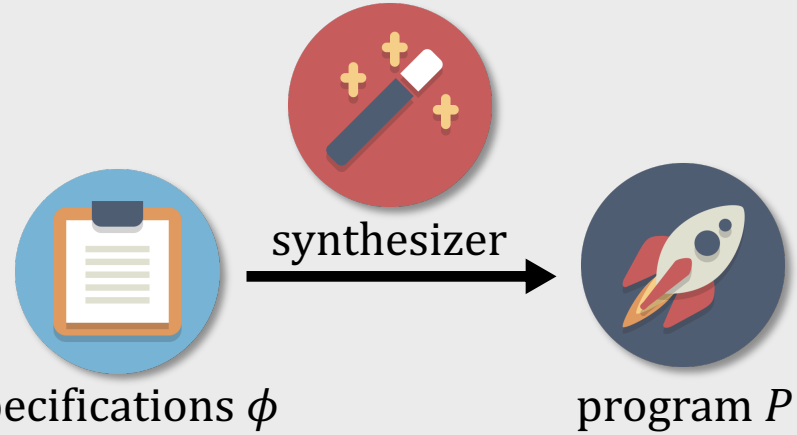
Spring 2021

Program Synthesis in a Nutshell

- Problem Formalization
- A Data Wrangling Example & DSL

Program Synthesis in a Nutshell

Problem Formalization



Find a program P that satisfies specifications ϕ.

Program Synthesis in a Nutshell

Problem Formalization

multi-modal

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

site	coll_date	SP_A_pos	SP_A_neg	SP_B_pos	SP_B_neg
site1	27/10/2007	0	0	1	0
site2	27/10/2007	0	0	0	1
site3	28/10/2007	2	0	0	0

examples

occurs(`unite`) ∧
occurs(`group_by`) ∧
hasChild(`group_by`, `unite`) ∧
... logical constraints

I need to reformat the data so that there is just one row per site visit (i.e. in a given site name and date combo) with columns for total found by species and the fish status (i.e. speciesA_pos, SpeciesA_neg, Sp_B_pos.. etc).

natural languages

...

specifications ϕ

synthesizer

program P

Find a program P that satisfies specifications ϕ.

Program Synthesis in a Nutshell

Problem Formalization

multi-modal

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

site	coll_date	SP_A_pos	SP_A_neg	SP_B_pos	SP_B_neg
site1	27/10/2007	0	0	1	0
site2	27/10/2007	0	0	0	1
site3	28/10/2007	2	0	0	0

examples

occurs(`unite`) ∧
occurs(`group_by`) ∧
hasChild(`group_by`, `unite`) ∧
... logical constraints

I need to reformat the data so that there is just one row per site visit (i.e. in a given site name and date combo) with columns for total found by species and the fish status (i.e. speciesA_pos, SpeciesA_neg, Sp_B_pos.. etc).

natural languages

...

multi-paradigm

neural deductive ...

synthesizer

specifications ϕ → program P

Find a program P that satisfies specifications ϕ.

Program Synthesis in a Nutshell

A Running Example from StackOverflow[1]

[Title] r script to count columns within dataset

[Example]

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

site	cat	sts
site1	SpeciesB_positive	1
site2	SpeciesB_negative	1
site3	SpeciesA_positive	2

[Description]

I need to reformat the data so that there is just one row per site visit (i.e. in a given site name and date combo) with columns for total found by species and the fish status (i.e. speciesA_pos, SpeciesA_neg, Sp_B_pos.. etc).

figured I need to sum within site. My thoughts were to use split/apply/aggregate/for loops etc but tried various combinations and not getting anywhere. apologies I'm not familiar with R. any comments appreciated!

[1] Example adapted from https://stackoverflow.com/questions/39369502/r-script-to-reshape-and-count-columns-within-dataset

Program Synthesis in a Nutshell

A Running DSL for Data Wrangling[1]

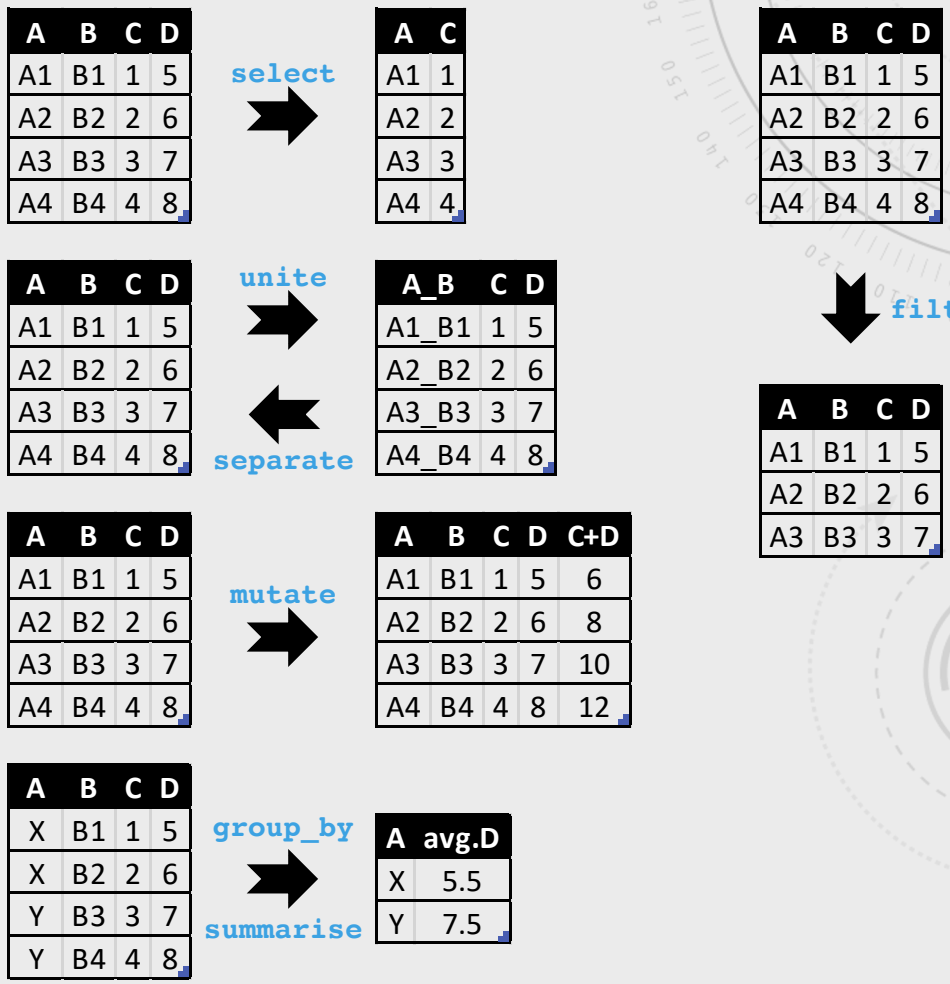
$t \rightarrow x_i$ (input table)

$\texttt{select}(t, \vec{c}_{arg})$	(column projection)
$\texttt{unite}(t, c_{tgt}, \vec{c}_{arg})$	(column merging)
$\texttt{separate}(t, \vec{c}_{tgt}, c_{arg})$	(column splitting)
$\texttt{mutate}(t, c_{tgt}, op, \vec{c}_{arg})$	(column arithmetic)
$\texttt{group\_by}(t, \vec{c}_{arg})$	(row grouping)
$\texttt{summarise}(t, c_{tgt}, a, \vec{c}_{arg})$	(row aggregation)
$\texttt{filter}(t, f, \vec{c}_{arg})$	(row filtering)

$op \rightarrow + \mid - \mid \times \mid \div$

$a \rightarrow \texttt{min} \mid \texttt{max} \mid \texttt{sum} \mid \texttt{count} \mid \texttt{avg}$

x_i: the i-th input table
t: table
c, $\vec{c}$: column(s) of table
op: arithmetic operation
a: aggregation function
f: higher-order boolean function



[1] DSL adapted from Wang. C. et al. Visualization by Example. POPL'20

Program Synthesis in a Nutshell

A Running Example from StackOverflow

[Example]

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

unite →

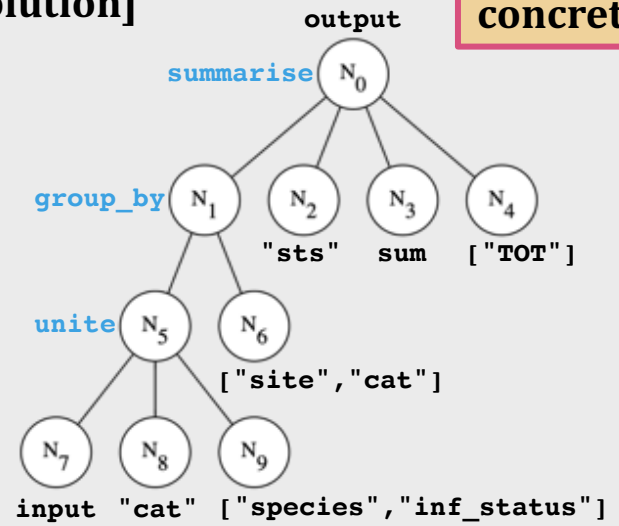
sample_ID	site	coll_date	cat	TOT
382870	site1	27/10/2007	SpeciesB_positive	1
382872	site2	27/10/2007	SpeciesB_negative	1
487405	site3	28/10/2007	SpeciesA_positive	1
487405	site3	28/10/2007	SpeciesA_positive	1

group_by → summarise

site	cat	sts
site1	SpeciesB_positive	1
site2	SpeciesB_negative	1
site3	SpeciesA_positive	2

[Solution]

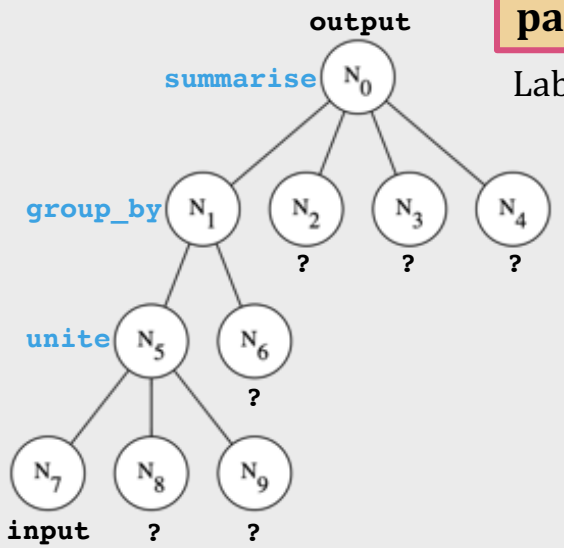
concrete program




```
T0 = unite( input, "cat", ["species", "inf_status"] )
T1 = group_by( T0, ["site", "cat"] )
output = summarise( T1, "sts", sum, ["TOT"] )
```

partial program / sketch

Labels of some AST nodes are yet to be determined.



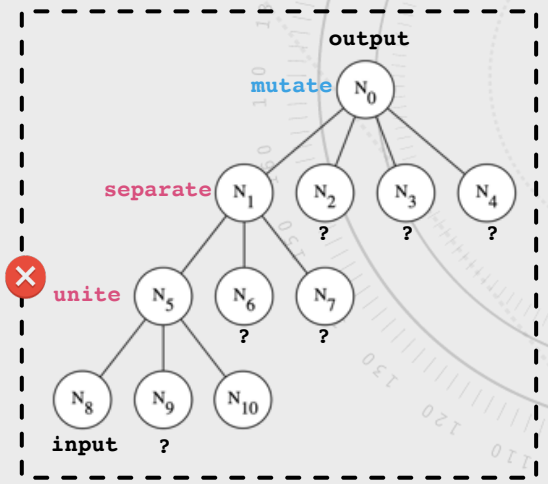

```
T0 = unite( input, ?, ? )
T1 = group_by( T0, ? )
output = summarise( T1, ?, ?, ? )
```

Terminologies

site	cat	sts
site1	SpeciesB_positive	1
site2	SpeciesB_negative	1
site3	SpeciesA_positive	2

```
output.row == 3
output.col == 3
```

synthesizer



A Partial Program

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

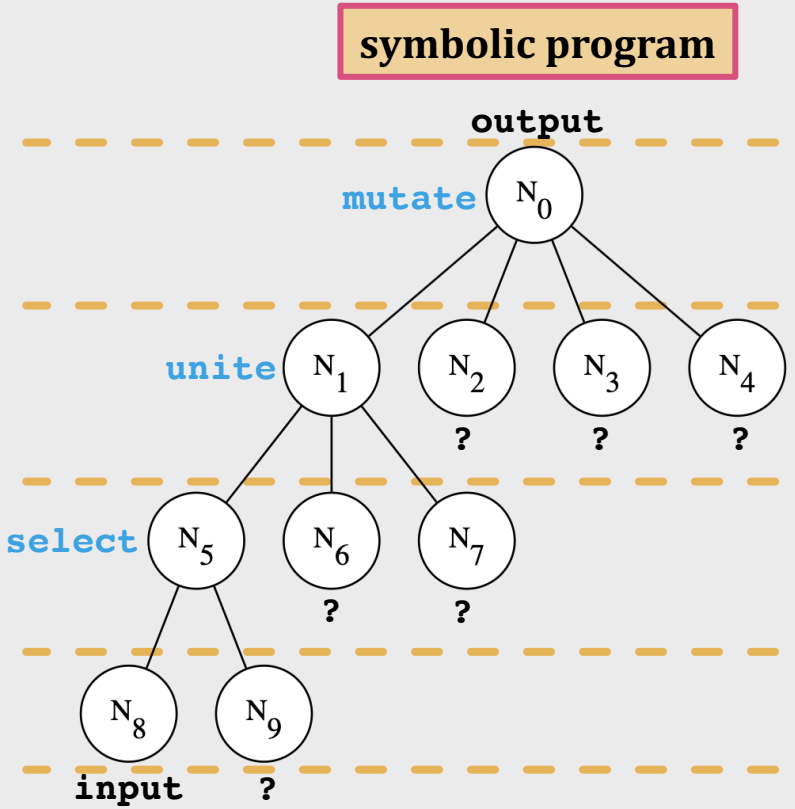
```
input.row == 4
input.col == 6
```

$$
\begin{aligned}
t &\rightarrow x_i \\
&\mid \texttt{select}(t, \vec{c}_{arg}) \\
&\mid \texttt{unite}(t, c_{tgt}, \vec{c}_{arg}) \\
&\mid \texttt{separate}(t, \vec{c}_{tgt}, c_{arg}) \\
&\mid \texttt{mutate}(t, c_{tgt}, op, \vec{c}_{arg}) \\
&\mid \texttt{group\_by}(t, \vec{c}_{arg}) \\
&\mid \texttt{summarise}(t, c_{tgt}, a, \vec{c}_{arg}) \\
&\mid \texttt{filter}(t, f, \vec{c}_{arg}) \\
op &\rightarrow + \mid - \mid \times \mid \div \\
a &\rightarrow \texttt{min} \mid \texttt{max} \mid \texttt{sum} \mid \texttt{count} \mid \texttt{avg}
\end{aligned}
$$

select	out.row==in.row ∧ out.col<=in.col-1
unite	out.row==in.row ∧ out.col==in.col-1
separate	out.row==in.row ∧ out.col==in.col+1
mutate	out.row==in.row ∧ out.col==in.col+1
group_by	out.row==in.row ∧ out.col==in.col
summarise	out.row<=in.row ∧ out.col<=in.col+1
filter	out.row<=in.row-1 ∧ out.col==in.col

Abstract Semantics of Components

Pruning via Deduction[1]




```
N5 = select( ?, ? )
N1 = unite( N5, ?, ? )
N0 = mutate( N1, ?, ?, ? )
```

site	cat	sts
site1	SpeciesB_positive	1
site2	SpeciesB_negative	1
site3	SpeciesA_positive	2

```
output.row == 3
output.col == 3
```

```
output.row == N0.row
output.col == N0.col
```

satisfiable? → NO

mutate

```
N0.row == 4
N0.col <= 6
```

unite

```
N1.row == 4
N1.col <= 4
```

select

```
N5.row == 4
N5.col <= 5
```

input

```
N8.row == 4
N8.col == 6
```

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

```
input.row == 4
input.col == 6
```

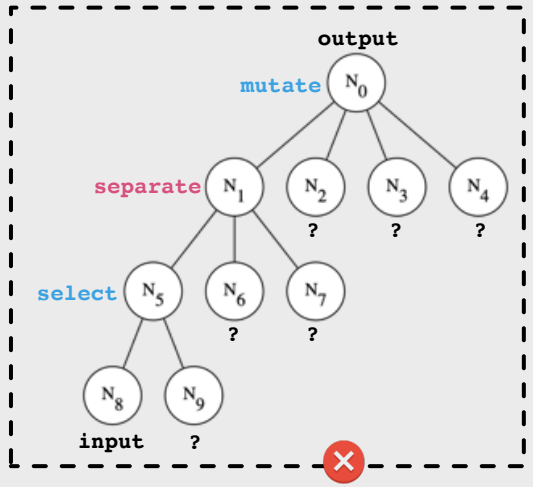
[1] Feng, Y. et al.. Component-based Synthesis of Table Consolidation and Transformation Tasks from Examples. PLDI'17

Learning from Past Mistakes

Equivalent Modulo Conflict (EMC)[1]

select	out.row==in.row ∧ out.col<=in.col-1
unite	out.row==in.row ∧ out.col==in.col-1
separate	out.row==in.row ∧ out.col==in.col+1
mutate	out.row==in.row ∧ out.col==in.col+1
group_by	out.row==in.row ∧ out.col==in.col
summarise	out.row<=in.row ∧ out.col<=in.col+1
filter	out.row<=in.row-1 ∧ out.col==in.col

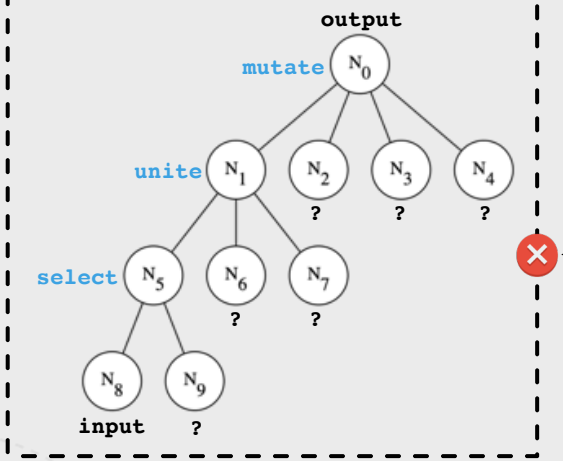
Abstract Semantics of Components



site	cat	sts
site1	SpeciesB_positive	1
site2	SpeciesB_negative	1
site3	SpeciesA_positive	2

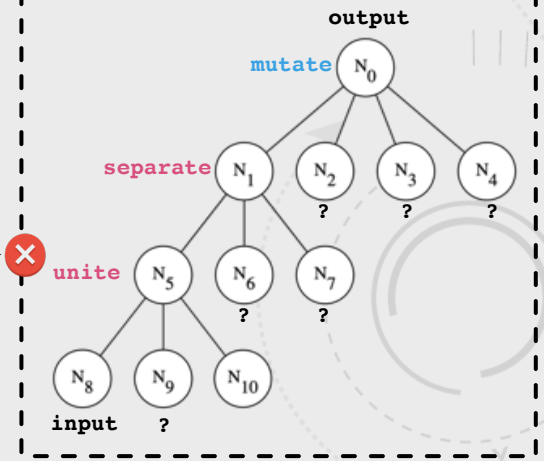
```
output.row == 3
output.col == 3
```

proposed program



Reasoning via Abstract Semantics

input.row==4 ∧ input.col==6 ∧	(input example)
N8.row==input.row ∧ N8.col==input.col ∧	(input alignment)
N5.row==N8.row ∧ N5.col<=N8.col-1 ∧	(`select` semantics)
N1.row==N5.row ∧ N1.col==N5.col-1 ∧	(`unite` semantics)
N0.row==N1.row ∧ N0.col==N1.col+1 ∧	(`mutate` semantics)
output.row==N0.row ∧ output.col==N0.col ∧	(output alignment)
output.row==3 ∧ output.col==3	(output example)

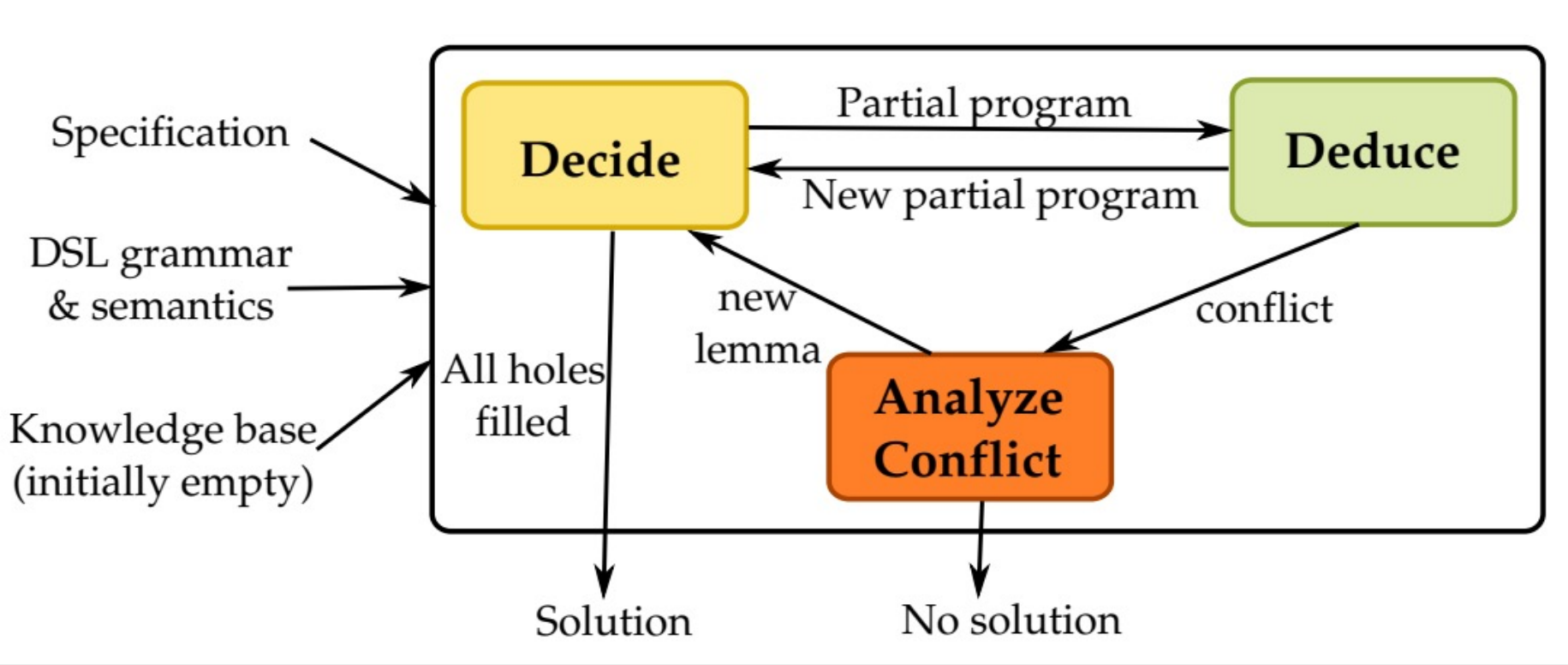


sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

```
input.row == 4
input.col == 6
```

UNSAT Core

Overview of NEO

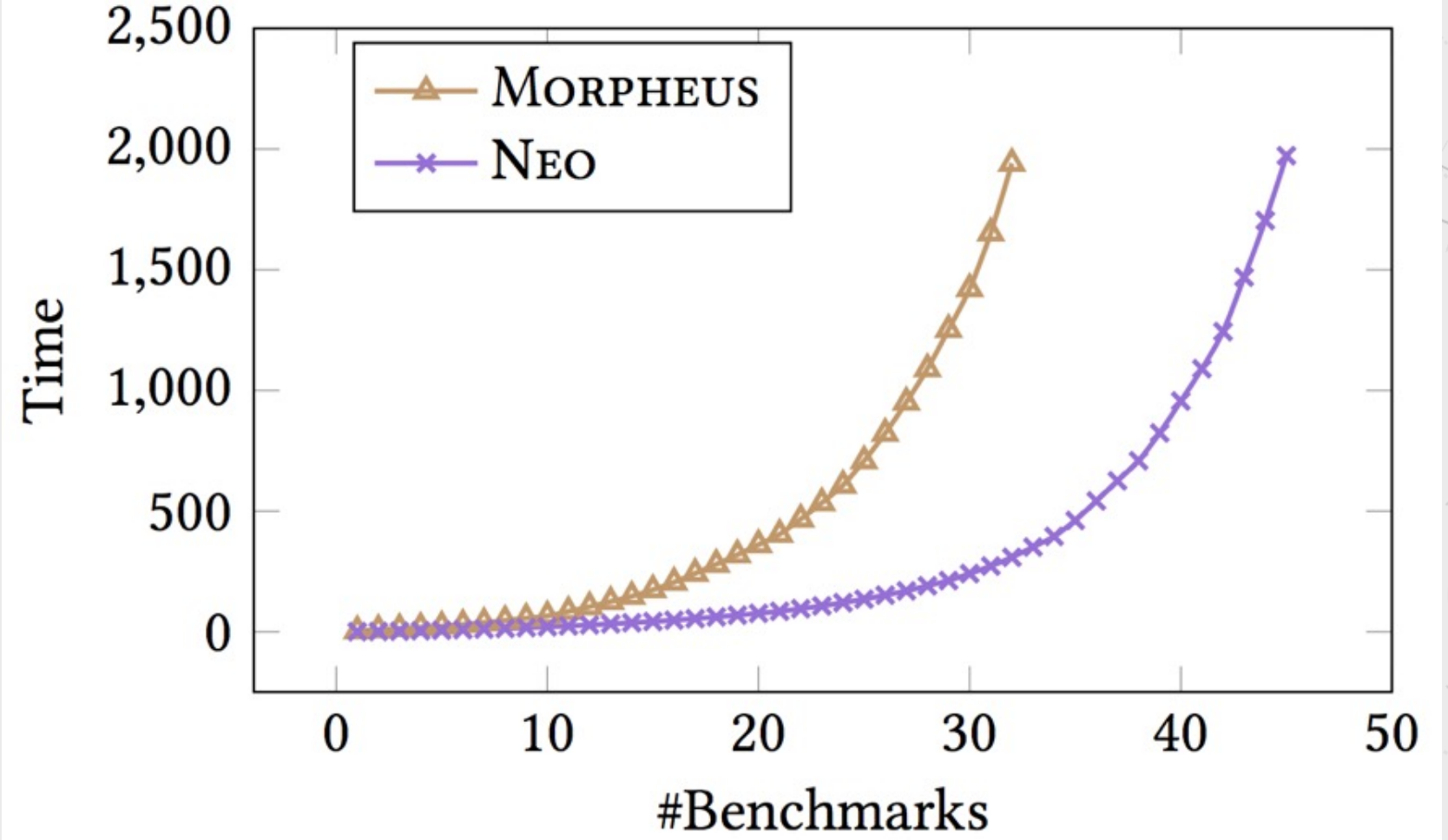


Evaluation Setup

- Experiment Setup
  - Benchmarks: 50 Real-World Challenging Data Wrangling Tasks
  - Uses 2-gram model to guide the search
  - Comparison to MORPHEUS[1]
  - For more evaluation details, please refer to our paper

Evaluation Results

- Timeout: 5 mins



Caveats: Scalability and Over-fitting

sample_ID	site	coll_date	species	TOT	inf_status
382870	site1	27/10/2007	SpeciesB	1	positive
382872	site2	27/10/2007	SpeciesB	1	negative
487405	site3	28/10/2007	SpeciesA	1	positive
487405	site3	28/10/2007	SpeciesA	1	positive

site	coll_date	SP_A_pos	SP_A_neg	SP_B_pos	SP_B_neg
site1	27/10/2007	0	0	1	0
site2	27/10/2007	0	0	0	1
site3	28/10/2007	2	0	0	0

examples

occurs(`unite`) ∧
occurs(`group_by`) ∧
hasChild(`group_by`, `unite`) ∧
... logical constraints

I need to reformat the data so that there is just one row per site visit (i.e. in a given site name and date combo) with columns for total found by species and the fish status (i.e. speciesA_pos, SpeciesA_neg, Sp_B_pos.. etc).

natural languages

...

multi-paradigm

neural deductive ...

synthesizer

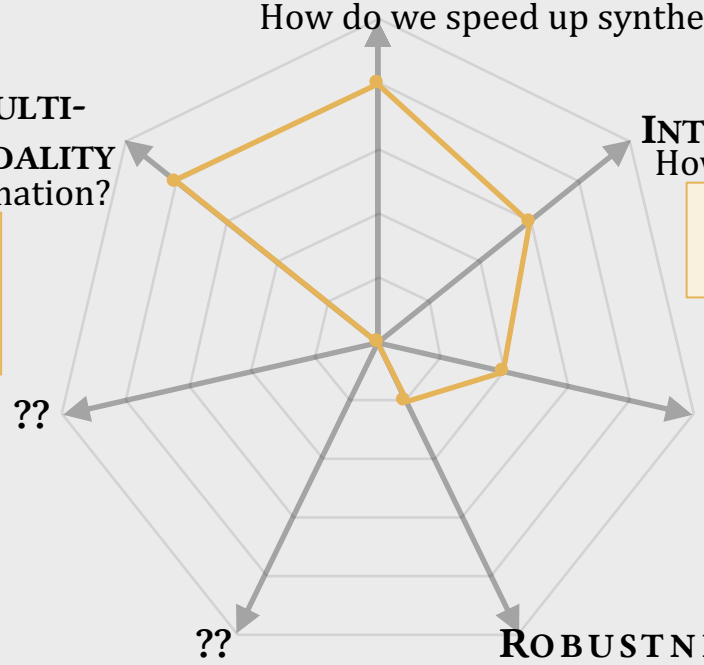
specifications ϕ program P

Find a program P that satisfies specifications ϕ.

Challenges, Conclusions & Future Works

DeepCoder (Balog et al. 2017); Exec (Chen et al. 2018); Neo (Feng et al. 2018); SQLizer (Yaghmazadeh et al. 2018); AutoPandas (Bavishi et al. 2019); Metal (Si et al. 2019); SketchAdapt (Nye et al. 2018); Probe (Barke et al. 2020); Concord (Chen et al. 2020); ...

Scalability
How do we speed up synthesis for given task?



Multi-Modality
How do we process multi-modal information?

Seq2SQL (Zhong et al. 2017); Mars (Chen et al. 2019); Regel (Chen et al. 2020); Viser (Wang et al. 2020); ...

Interactivity
How do we access and utilize extra information from users?

InteractivePROSE (Le et al. 2017); DialSQL (Gur et al. 2018); GIM (Peleg et al. 2020); SampleSy (Ji et al. 2020); ...

Continuality
How do we distill useful knowledge across synthesis?

NELL (Mitchell et al. 2015); Net2Net (Chen et al. 2016); Parishi et al. 2019; ...

Robustness
How do we tolerate specification mistakes/noises during synthesis?

FlashFill (Gulwani 2011); RuleSynth (Singh 2017); Bester (Peleg et al. 2020); ...