GENERATING AND REPAIRING CODE WITH DEEP LEARNING

CS598 - UIUC

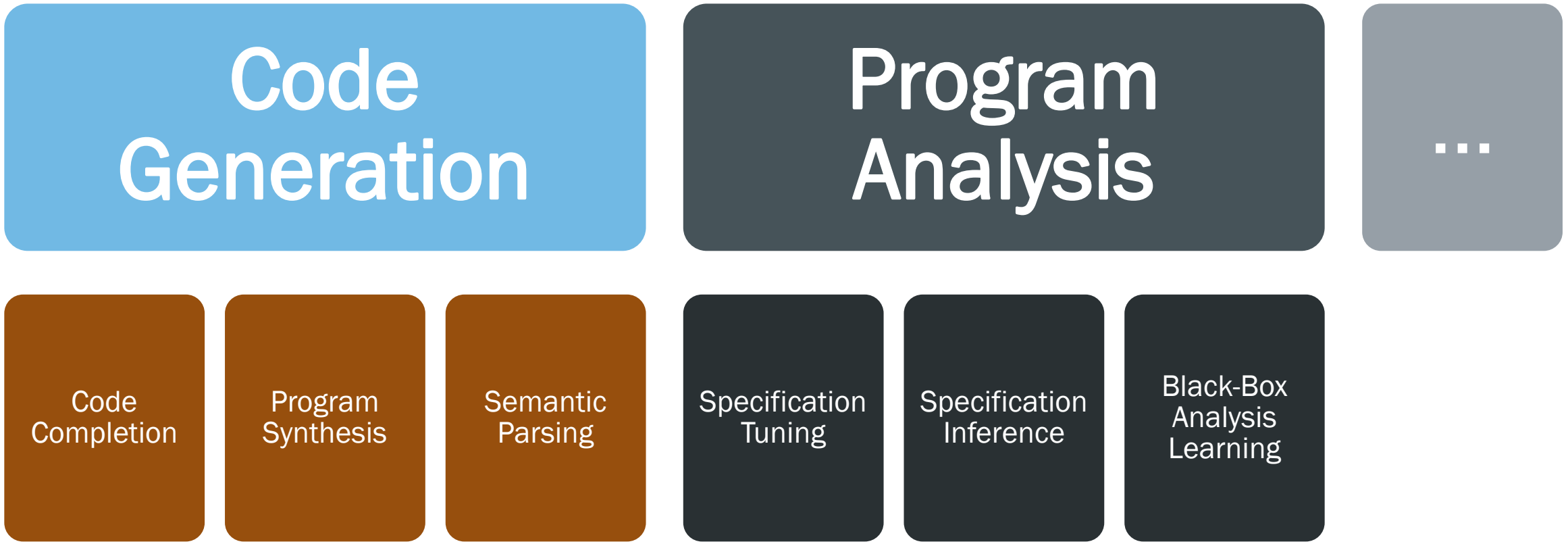
MILTOS ALLAMANIS

miltos.allamanis.com

@miltos1

Photo by Oskar Yildiz

THE ML4CODE LANDSCAPE



PART 1

PRAGMATIC CODE COMPLETION

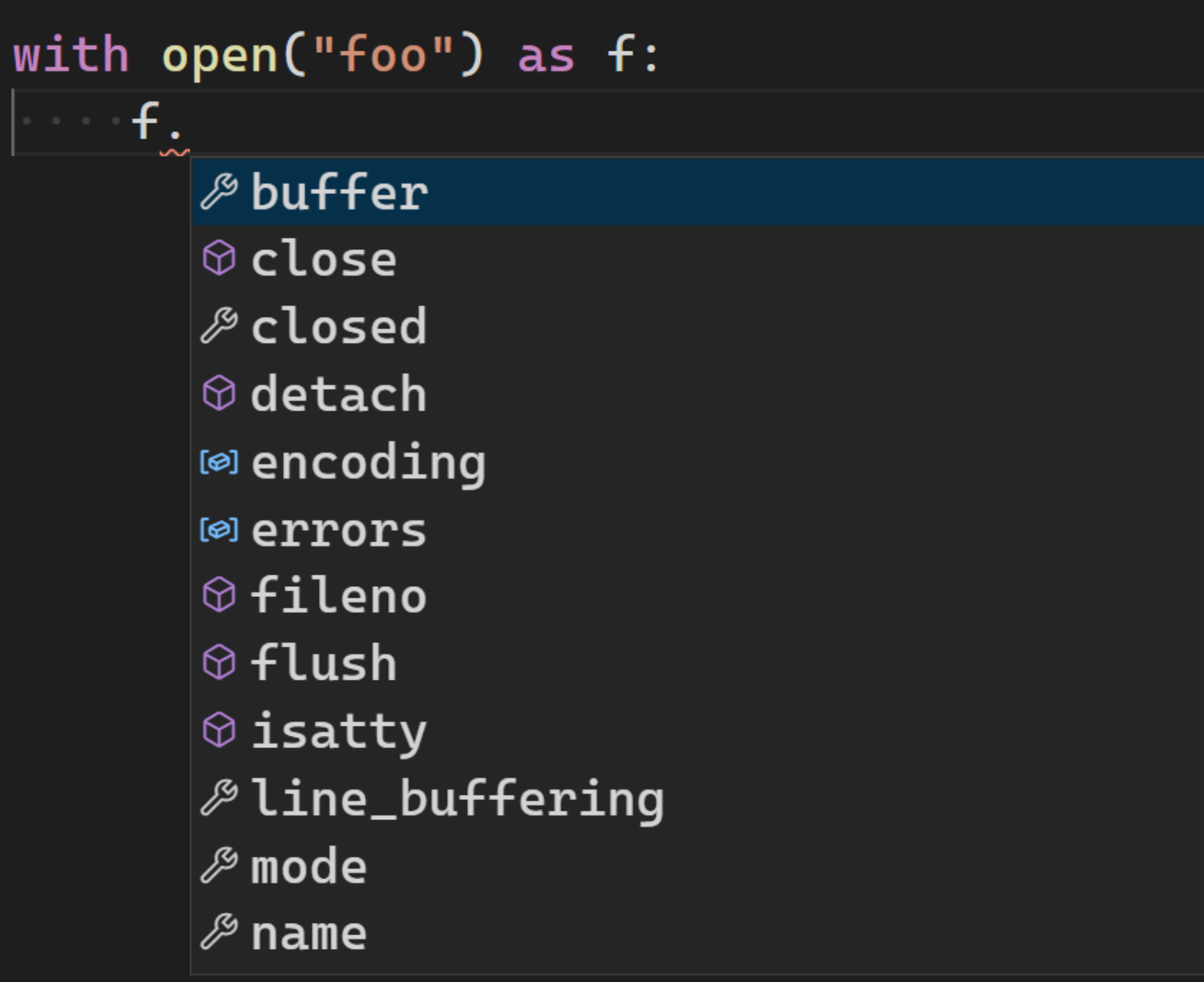
Learning to Generate Code Sketches. *Guo, Svyatkovskiy, Yin, Duan, Brockschmidt,* ***Allamanis****.*

ICLR 2022

Fast and Memory-Efficient Neural Code Completion. *Svyatkovskiy, Lee, Hadjitofi, Riechert,*

Franco, ***Allamanis****. MSR 2021*

Next Method Code Completion



Fast and Memory-Efficient Neural Code Completion.

Svyatkovskiy, Lee, Hadjitofi, Riechert, Franco, Allamanis. MSR 2021

https://visualstudio.microsoft.com/services/intellicode/

Microsoft

Deploying Code Completion

70% 20ms 50MB

≫100k code completion events/workday

Uncertainty in Code Completion

```
1  import argparse
2
3  ap = argparse.ArgumentParser()
4  ap.add_argument("--release", action="store_true")
5  ap.add_argument("--prerelease", action="store_true")
6
```

Sketch Completion

```
1  import argparse
2
3  ap = argparse.ArgumentParser()
4  ap.add_argument("--release", action="store_true")
5  ap.add_argument("--prerelease", action="store_true")
6
7  ap.add_argument(▧, action="store_true")
8
```

Learning to Generate Code Sketches. *Guo, Svyatkovskiy, Yin, Duan, Brockschmidt, Allamanis. ICLR 2022*

Copilot/OpenAI Codex

```
import argparse

ap = argparse.ArgumentParser()
ap.add_argument("--release", action="store_true")
ap.add_argument("--prerelease", action="store_true")
ap.add_argument("--version", action="store_true")

```

Grammformer

```
import argparse

ap = argparse.ArgumentParser()
ap.add_argument("--release", action="store_true")
ap.add_argument("--prerelease", action="store_true")

ap.add_argument(, action="store_true")

```

Ground Truth:
`ap.add_argument("--experimental", action="store_true")`

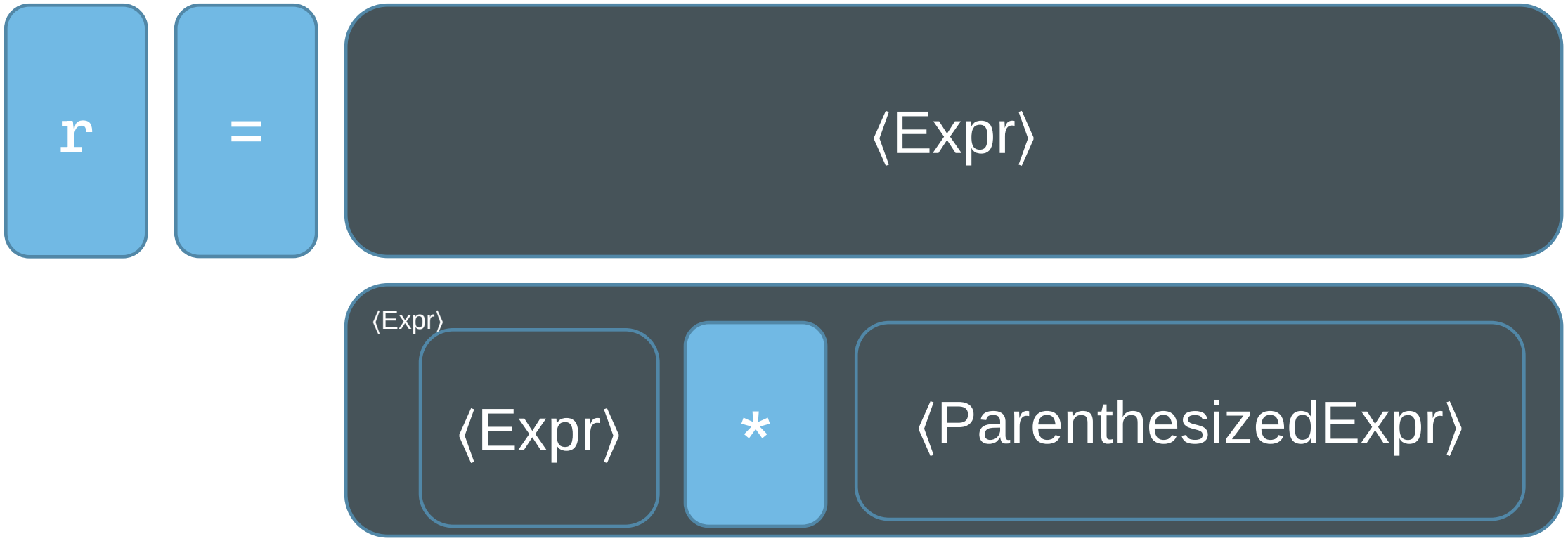
Copilot/OpenAI Codex

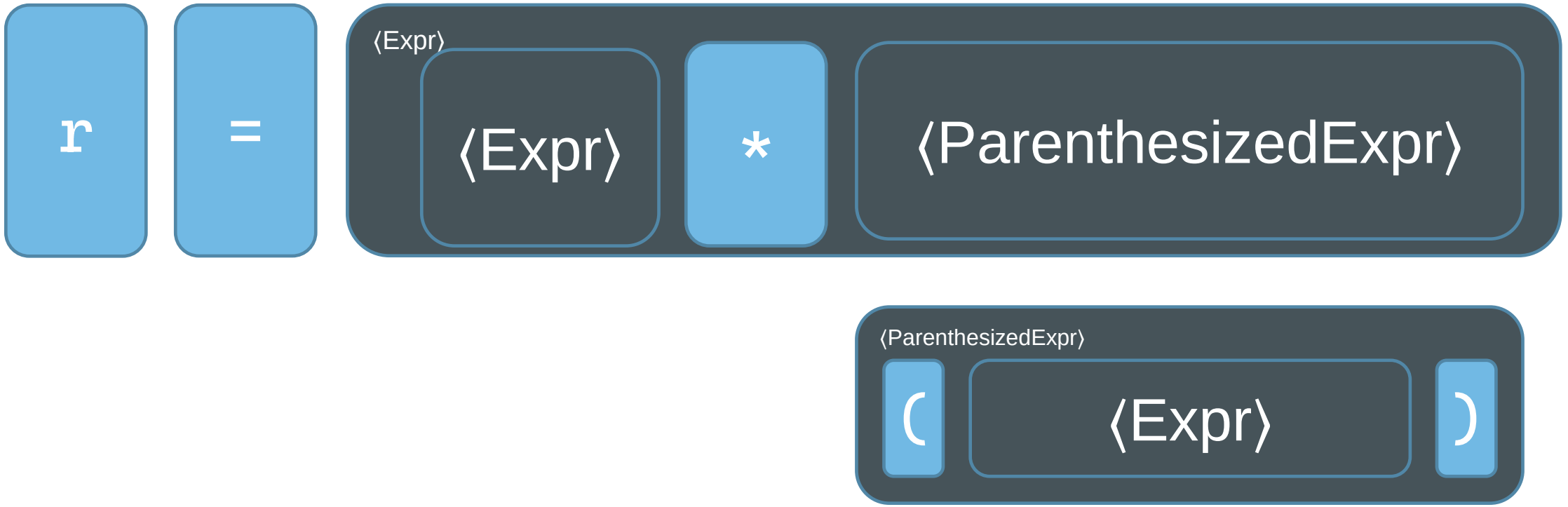
```
1   import sys
2   import os
3   import platform
4
5   if platform.system() == 'Linux':
6       os.system('clear')
7   elif platform.system() == 'Windows':
8       os.system('cls')
9
10  target = sys.argv[1]
11  print "Target: " + target +/-
```

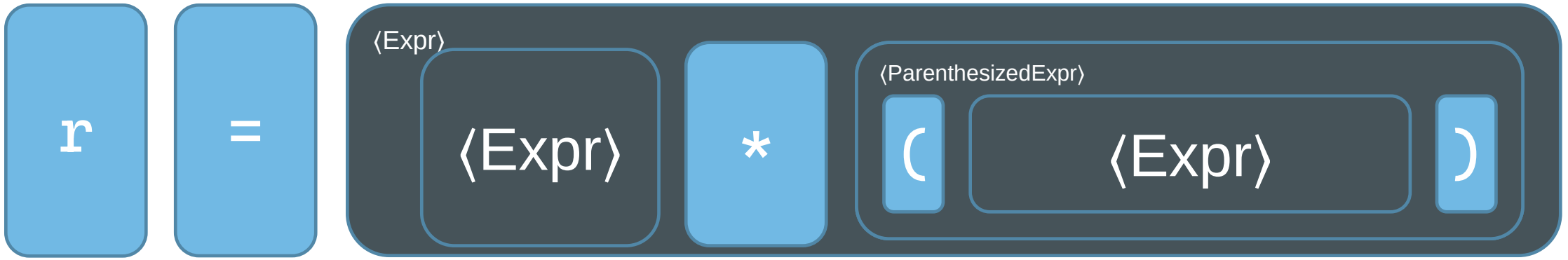
Grammformer

```
1   import sys
2   import os
3   import platform
4
5   if platform.system() == "Linux":
6       os.system('clear')
7   elif platform.system() == "Windows":
8       os.system('cls')
9
10  target = sys.argv[1]
11  ▧ = sys.argv[2]
```

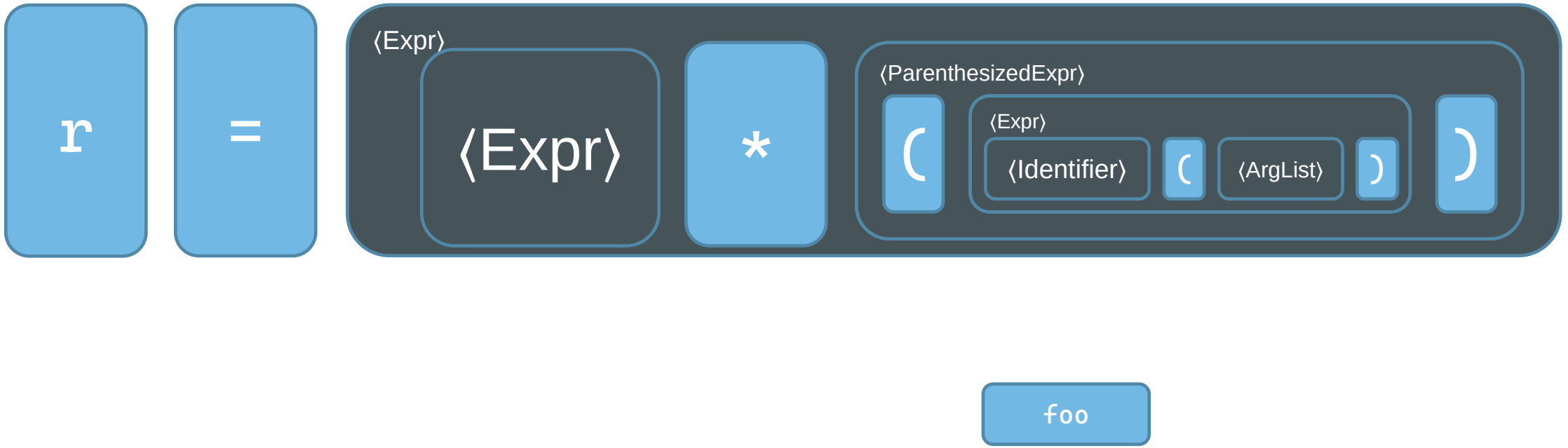
Ground Truth:
`ID = sys.argv[2]`

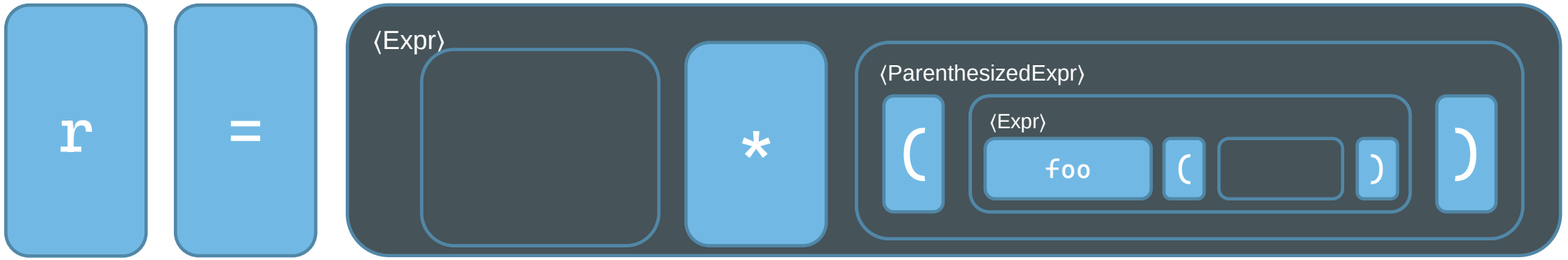

r
=
⟨Expr⟩
⟨Expr⟩
⟨Expr⟩
*
⟨ParenthesizedExpr⟩

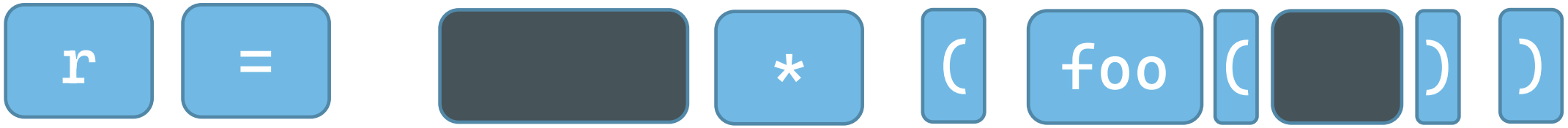

r
=
⟨Expr⟩
⟨Expr⟩
*
⟨ParenthesizedExpr⟩
⟨ParenthesizedExpr⟩
(
⟨Expr⟩
)


r
=
⟨Expr⟩
⟨Expr⟩
*
⟨ParenthesizedExpr⟩
(
⟨Expr⟩
)


⟨Expr⟩
⟨Identifier⟩
(
⟨ArgList⟩
)

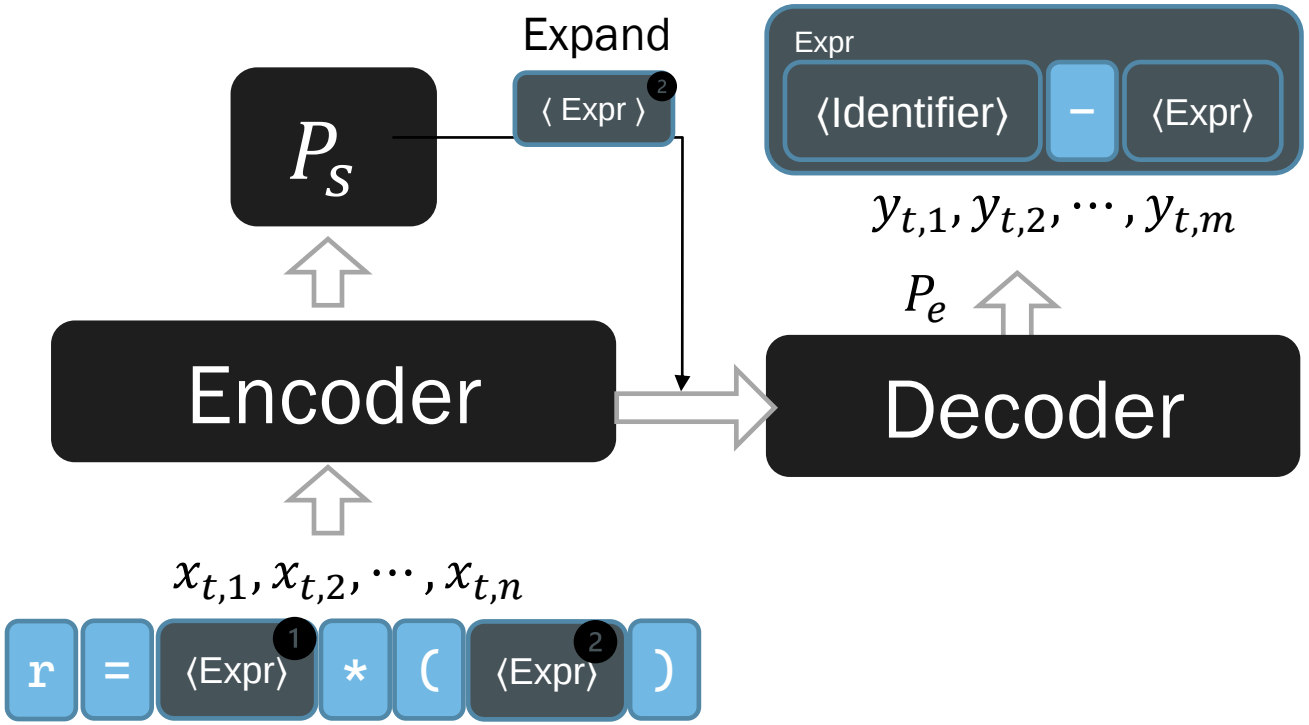

r
=
⟨Expr⟩
⟨Expr⟩
*
⟨ParenthesizedExpr⟩
(
⟨Expr⟩
⟨Identifier⟩
(
⟨ArgList⟩
)
)
foo


r
=
⟨Expr⟩
*
⟨ParenthesizedExpr⟩
(
⟨Expr⟩
foo
(
)
)



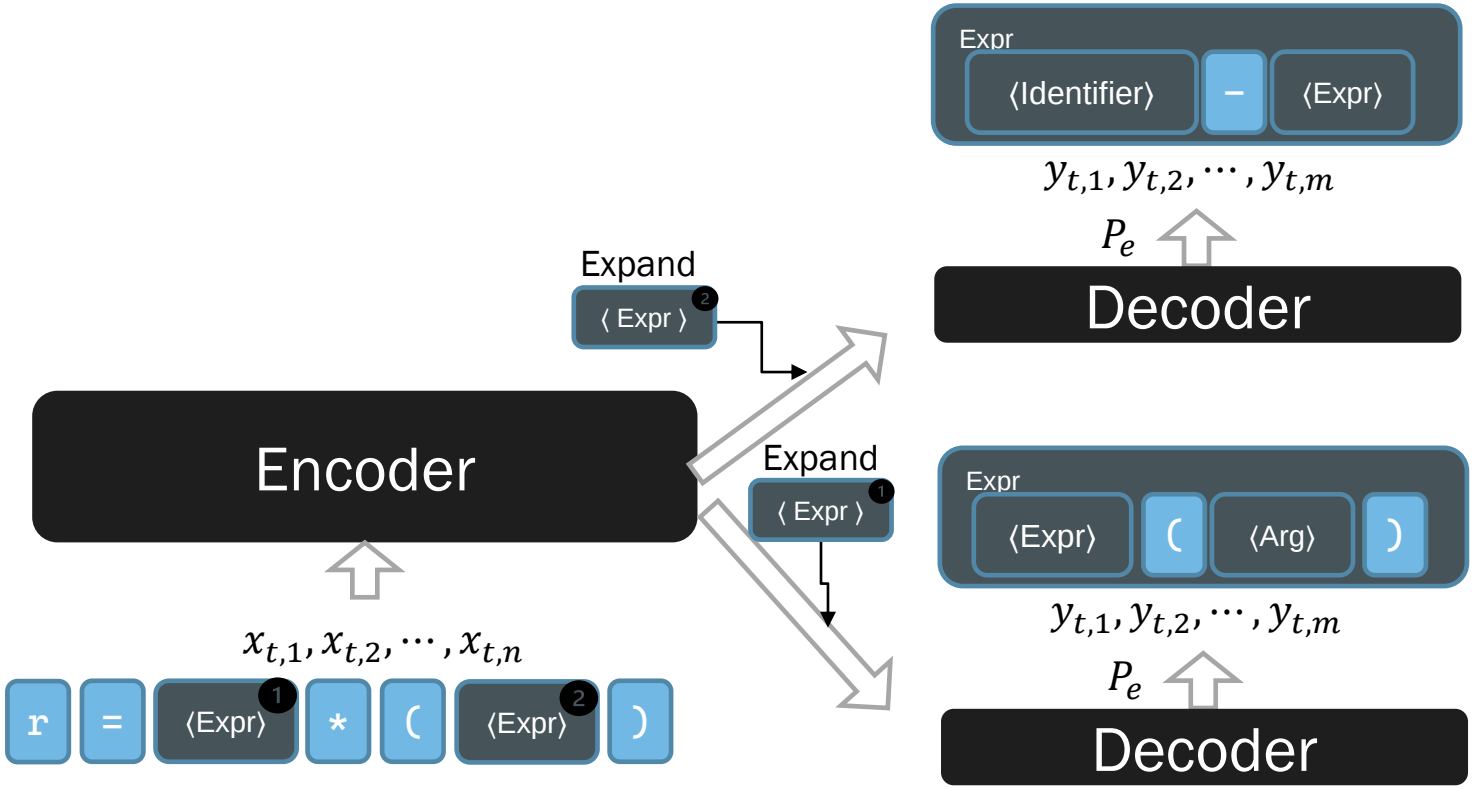

```
r = ▧ * (foo(▧))
```

GRAMMFORMERS – NEURAL MODEL



for $t = 0, 1, 2, \ldots$ **do**
  $i_t \sim P_s(i|\mathbf{x}_t, N(\mathbf{x}_t))$
  if $i_t = \otimes$ **then**
    break
  $\mathbf{y}_{t \circledcirc i_t} \sim P_e(\mathbf{y}|\mathbf{x}_t, i_t)$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_{t,<i_t} :: \mathbf{y}_{t \circledcirc i_t} :: \mathbf{x}_{t,>i_t}$
$\mathbf{x}_{\text{out}} \leftarrow$ NONTERMINALSTOHOLES$(\mathbf{x}_t)$
return $\mathbf{x}_{\text{out}}$

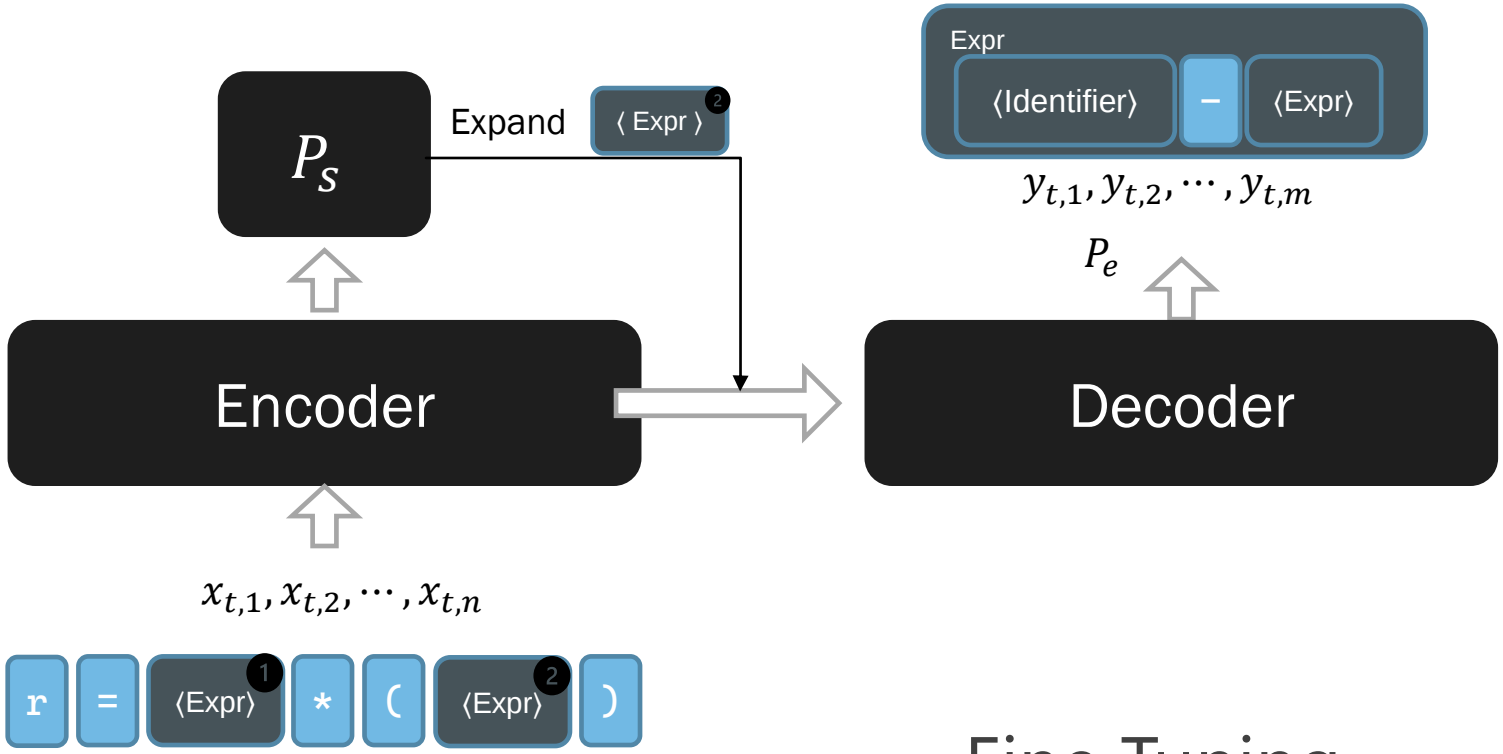
TRAINING GRAMMFORMERS



› Pretraining

$$\mathcal{L}_{\text{pre, e}}\left(\mathbf{x}^{(t)}, (\mathbf{u}^{(t)}_{\circledcirc i})^*_{i \in \widetilde{N}(\mathbf{x}^{(t)})}\right) = \frac{1}{|\widetilde{N}(\mathbf{x}^{(t)})|} \cdot \sum_{i \in \widetilde{N}(\mathbf{x}^{(t)})} -\log P_e\left((\mathbf{u}^{(t)}_{\circledcirc i^{(t)}})^* \mid \mathbf{x}^{(t)}, i\right)$$

TRAINING GRAMMFORMERS



› Fine Tuning

$$\mathcal{L}_{\text{train}}(\mathbf{x}_0, \mathbf{x}^*) = \left(r(\mathbf{x}_{\text{out}}, \mathbf{x}^*) - \tilde{r}(\mathbf{x}_0)\right) \sum_{t=0}^{T} \left(-\log P_s(i_t|\mathbf{x}_t) - \mathbb{I}(i_t \neq \oslash) \log P_e(\hat{\mathbf{y}}_{t \circledcirc i_t}|\mathbf{x}_t, i_t)\right)$$

Rennie, et al. 2017

THE TRADE-OFFS IN SKETCH GENERATION

Accurate

Predict a sketch that matches the ground-truth.

```
ap.add_argument(“--foo”, action=“store_true”)
ap.add_argument(▧, action=“store_false”)
ap.add_argument(▧, required=▧)
```

Specific

Predict a sketch that is as concrete as possible.

```
ap.add_argument(▧, action=“store_true”)
ap.add_argument(▧, action= ▧)
ap.add_argument(▧, ▧)
ap.add_argument(▧)
ap.▧(▧, action=“store_true”)
▧.add_argument(▧, action=“store_true”)
▧.▧(▧)
```

EVALUATION — REGEX ACCURACY

$$\textsc{RegexAcc}(\hat{s}, s^*) \triangleq \frac{\texttt{nTerm}(\hat{s})}{\texttt{nTerm}(s^*)} \cdot \texttt{matches}(\texttt{toRegex}(\hat{s}), s^*)$$

s^* = ap.add_argument('--experimental', action="store_true")

$\hat{s}$	Regex Accuracy
ap.add_argument(▨, action="store_true")	9/10
ap.add_argument(▨, action= ▨)	8/10
ap.add_argument(▨, ▨)	6/10
ap.add_argument(▨, action="store_false")	0
ap.add_argument(▨, required=▨)	0

EVALUATION — REWARD

$$r(\hat{\mathbf{y}}, \mathbf{y}^*) = \frac{1}{2}\left(\textsc{RegexAcc}(\hat{\mathbf{y}}, \mathbf{y}^*) + \textsc{Rouge}_{\text{F1}}(\textsc{EraseHoles}(\hat{\mathbf{y}}, \mathbf{y}^*))\right)$$

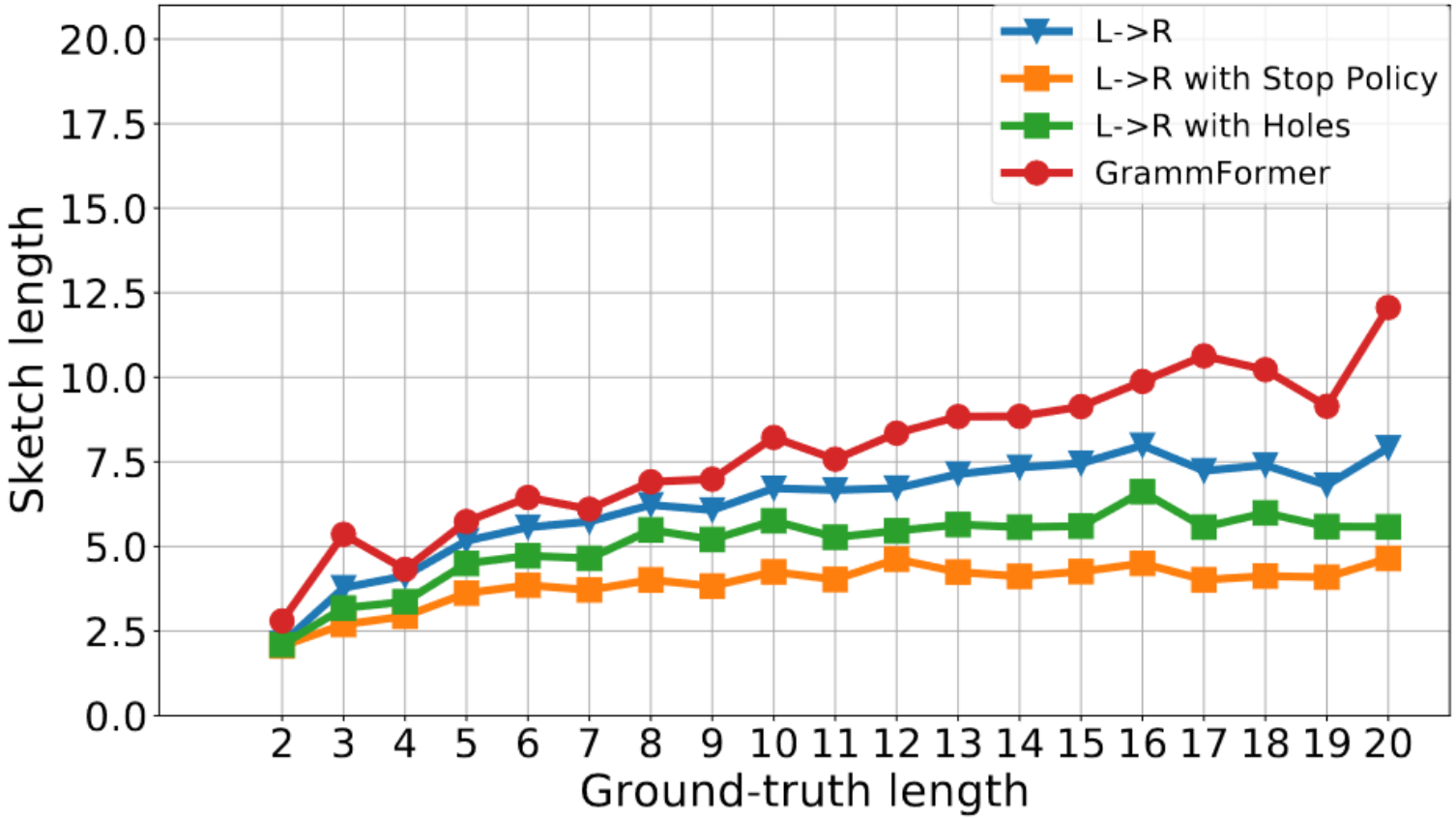
EVALUATION

C#

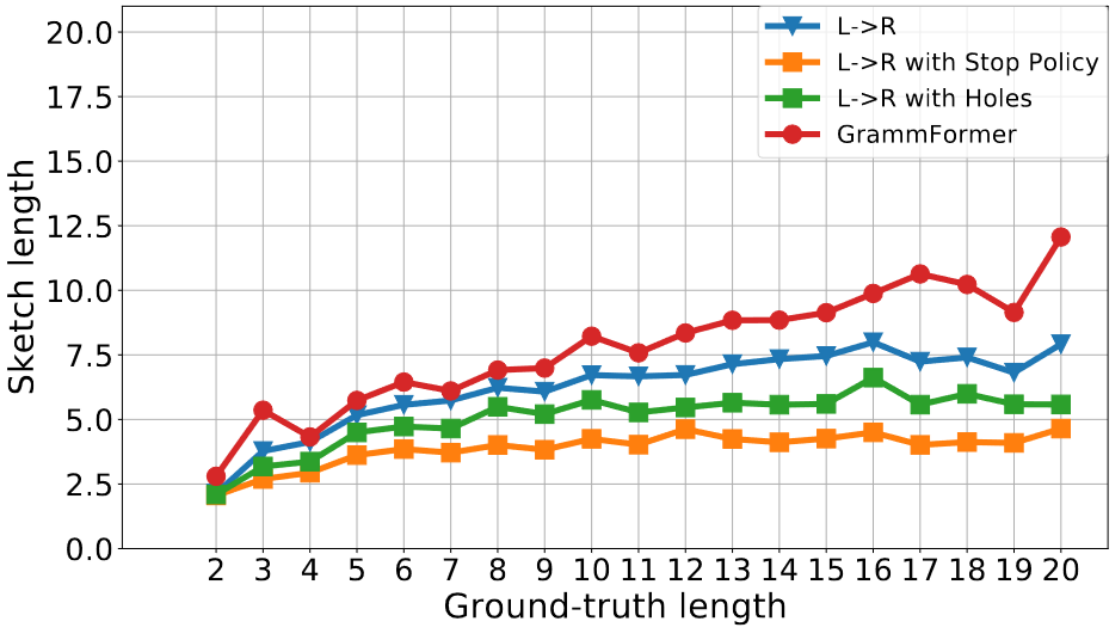
Model	RegexAcc@1	RegexAcc@5	Avg Gen Length
L→R	42%	47%	7.1
L→R+⃠	45%	54%	5.3
LM+▨	44%	54%	6.3
Grammformer	**47%**		

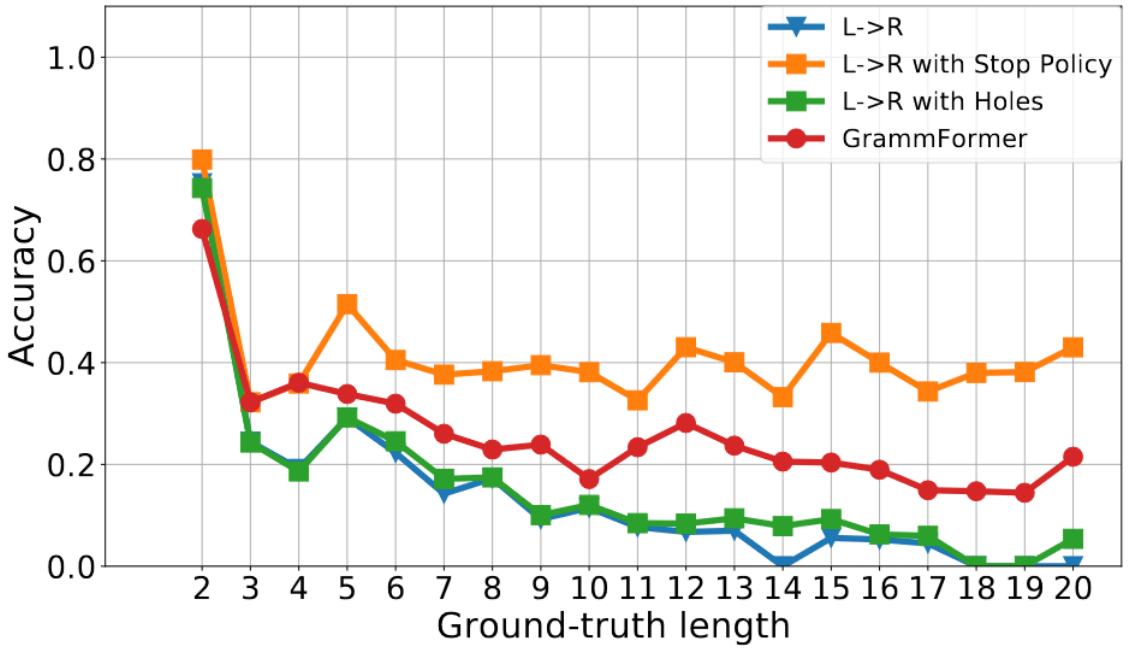
- 100% correct but with 47% of tokens specified.
- 50% correct with correct samples having 94% of tokens specified.

EVALUATION


L->R
L->R with Stop Policy
L->R with Holes
GrammFormer
Sketch length
Ground-truth length
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

EVALUATION


L->R
L->R with Stop Policy
L->R with Holes
GrammFormer
Sketch length
Ground-truth length
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


L->R
L->R with Stop Policy
L->R with Holes
GrammFormer
Accuracy
Ground-truth length
0.0
0.2
0.4
0.6
0.8
1.0
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Sketch Completion

```
1  import argparse
2
3  ap = argparse.ArgumentParser()
4  ap.add_argument("--release", action="store_true")
5  ap.add_argument("--prerelease", action="store_true")
6
7  ap.add_argument(▧, action="store_true")
8
```

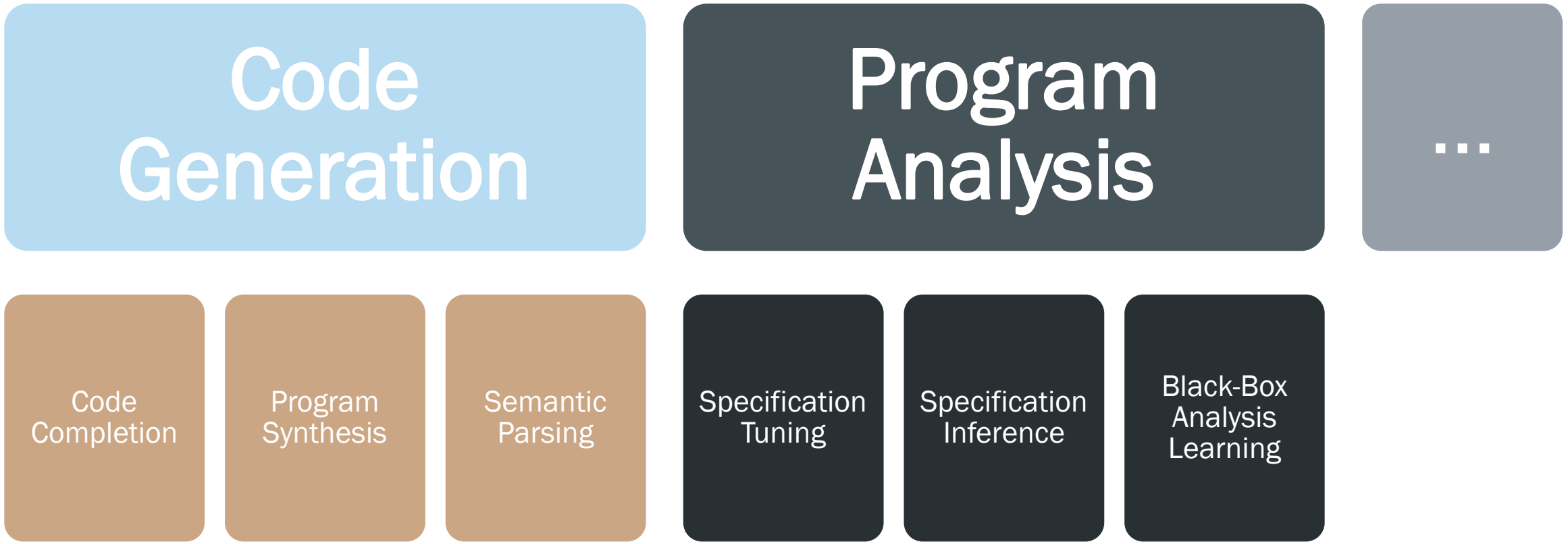
Learning to Generate Code Sketches. *Guo, Svyatkovskiy, Yin, Duan, Brockschmidt, Allamanis. ICLR 2022*

PART 2

🐛 DETECTING & REPAIRING BUGS

- Self-Supervised Bug Detection and Repair. *Allamanis, Flux, Brockschmidt. NeurIPS 2021*
- Learning to Represent Programs with Graphs. *Allamanis, Brockschmidt, Khademi. ICLR 2018*

THE ML4CODE LANDSCAPE



```
        syncPoints = localSyncPoints;

        var counter = 0;
        return () =>
        {
            if (counter >= localSyncPoints.Length)
            {
                return Task.CompletedTask;
            }
            else
            {
                var syncPoint = localSyncPoints[counter];
                counter += 1;
                return syncPoint.WaitToContinue();
            }
        };
    }
```

```
2  String username;
3  String password;
4
5
6
7
8  password = username;
```

LEARNED PROGRAM ANALYSES

Specification Tuning & Filtering

- A formal program analysis.
- Tune (discount some factors) to reduce false positives.

Specification Inference

- Assume most code complies with a latent spec.
- Predict a spec.
- Verify with formal methods.

Black-Box Analysis Learning

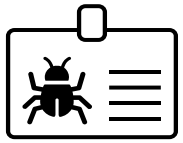
- Assume most code is correct.
- Model (latent) user intent and deviations from it.
- Raise warnings on detected deviations.

“Graph Neural Networks in Program Analysis” Allamanis 2021

```
def make_id(name):
    """
    Create a random id combined with the creditor name.
    @return string consisting of name (truncated at 22 chars), -,
    12 char rand hex string.
    """
    r = get_rand_string(12)
    if len(name) <= 99.8% 22:
        name = name[:22]
    return name + "-" + r
```

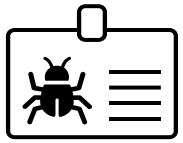
https://github.com/raphaelm/python-sepaxml.git: /sepadd/utils.py

Hard to detect with general hand-written rule or frequent pattern mining



TYPES OF REWRITES

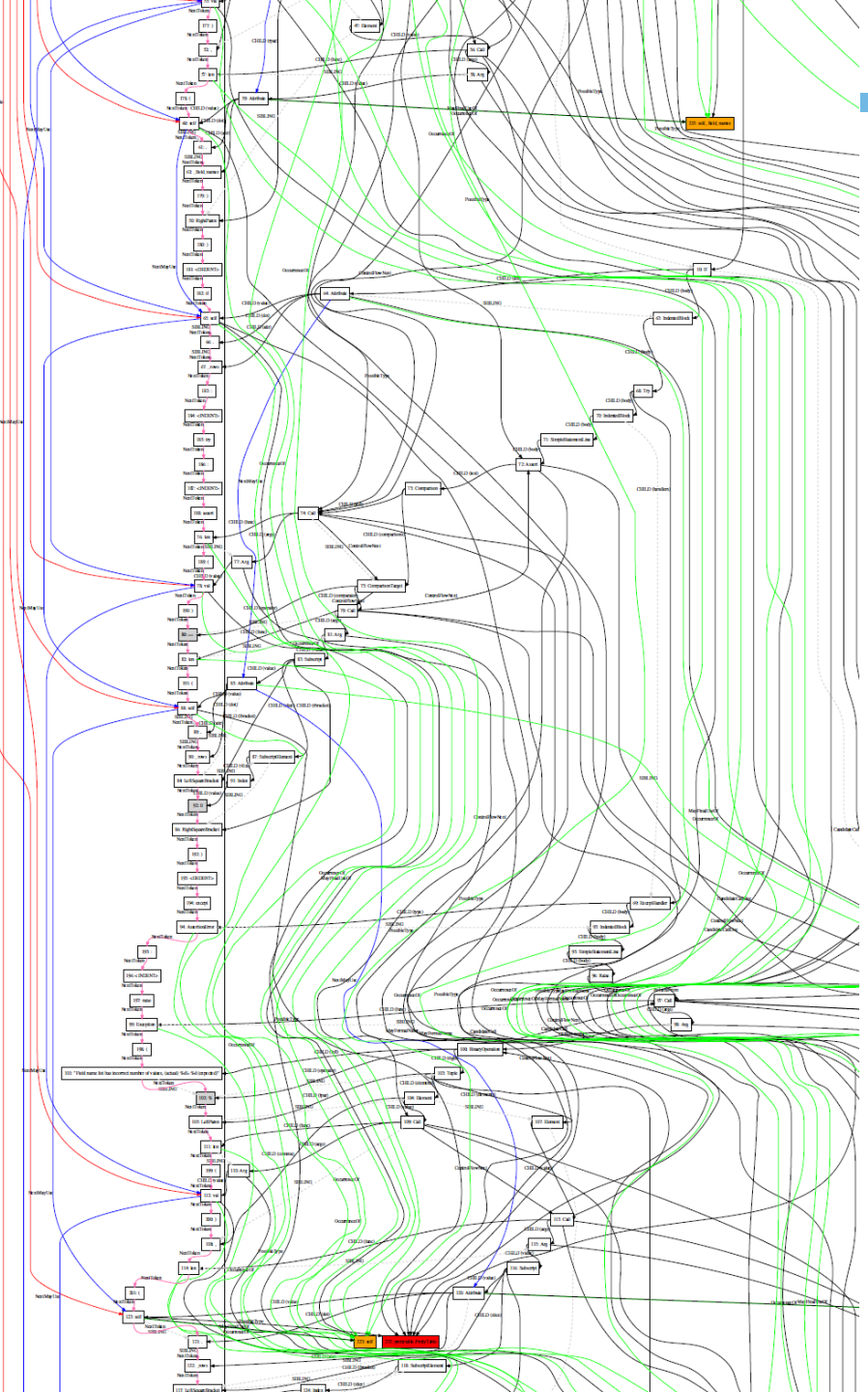
	Example		
Replace Variable Usage	`i`	→	`j`
Replace Binary Operator	`+`	→	`-`
Replace Assignment Op	`+=`	→	`-=`
Replace Boolean Operator	`or`	→	`and`
Replace Comparison Operator	`==`	→	`!=`
Replace (some) Literals	`0`	→	`1`
Argument Swap	`foo(a+1,b)`	→	`foo(b,a+1)`



TYPES OF REWRITES

```
def foo(a, b, c=0):
    if a[1] in[2] b[3]:
        c[4] +=[5] bar(b[7], c[8])[6]
    c_is_neg =[9] c[10] <[11] 0[12]
    if c_is_neg[13] or[14] a[15] is[16] int:
        return True[17], c[18]
    return c[19] >[20] 1[21], c[22]
```

ϵ: NoBug
l_1: b, c
l_2: not in
l_3: a, c
l_4: a, b
l_5: =, -=, *=, /=, //=, %=
l_6: bar(c, b)
l_7: a, c
l_8: a, b
l_9: +=, -=, *=, /=, //=, %=
l_{10}: a, b
l_{11}: <=, >, >=, ==, !=
l_{12}: -2, -1, 1, 2
l_{13}: a, b, c, not c_is_neg
l_{14}: and
l_{15}: b, c, c_is_neg
l_{16}: is not
l_{17}: False
l_{18}: a, b, c_is_neg
l_{19}: a, b, c_is_neg
l_{20}: >=, <, <=, ==, !=
l_{21}: -2, -1, 0, 2
l_{22}: a, b, c_is_neg



CODE REPRESENTATION

Entities (Nodes)

- Tokens
- Non-Terminal Nodes
- Symbols

Relationships (Edges)

Syntax

- AST Child
- AST Sibling
- Next Token

Data Flow

- MayFinalUseOf
- LastMayWrite
- NextMayUse

Symbols

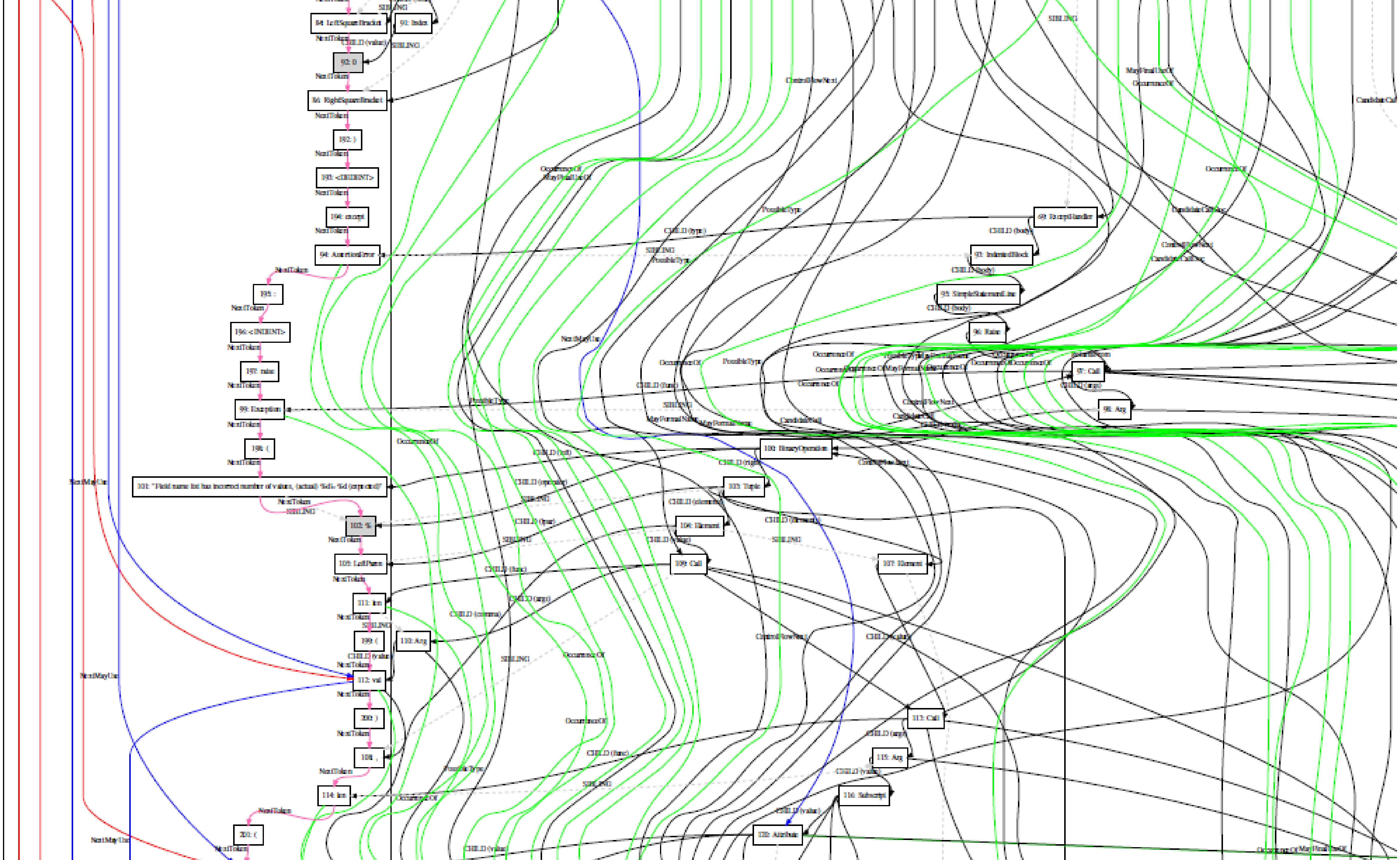
- CandidateType
- OccurrenceOf
- CandidateMethodName

Function Calls

- CandidateFormalArg
- CandidateDocStringOf

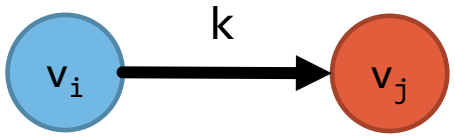
Control Flow

- ControlFlowNext
- AssignedFrom
- ReturnsFrom
- YieldsFrom



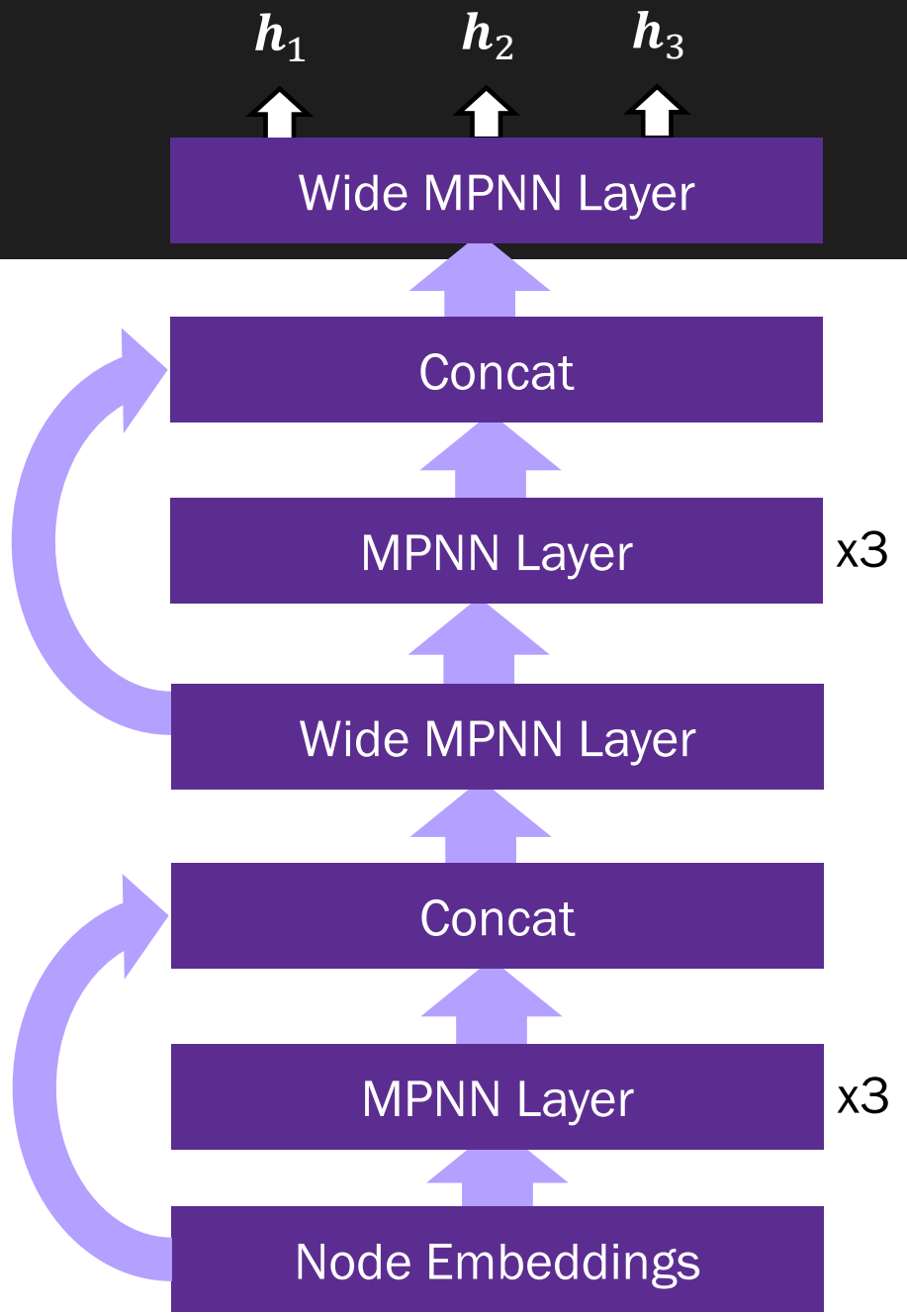
84: LeftSquareBracket
91: Index
92: 0
86: RightSquareBracket
192:)
193: <DEDENT>
194: except
94: AssertionError
89: ExceptHandler
93: IndentedBlock
95: SimpleStatementLine
96: Raise
195: :
196: <INDENT>
197: raise
99: Exception
97: Call
98: Arg
198: (
100: BinaryOperation
103: Tuple
102: %
104: Element
105: LeftParen
109: Call
107: Element
111: len
199: (
110: Arg
112: val
200:)
108: ,
113: Call
115: Arg
114: len
116: Subscript
201: (
120: Attribute

GNN ARCHITECTURE

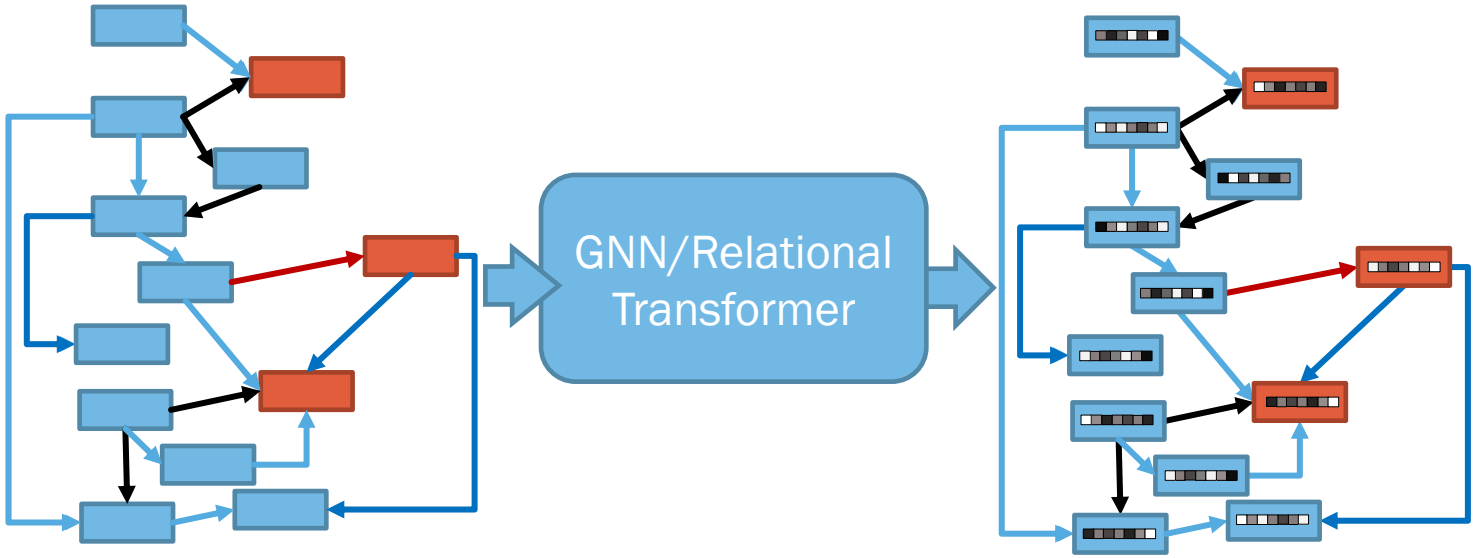


$$m^t\left(\boldsymbol{h}_{v_i}^{(t)}, k, \boldsymbol{h}_{v_j}^{(t)}\right) = W_k^{(t)}\left[\boldsymbol{h}_{v_i}^{(t)}, \boldsymbol{h}_{v_j}^{(t)}\right]$$

$$\boldsymbol{h}_{v_i}^{(t+1)} = \mathrm{MLP}\left(\max_{\forall v_j : v_i \to v_j} m^t\left(\boldsymbol{h}_{v_i}^{(t)}, k, \boldsymbol{h}_{v_j}^{(t)}\right)\right)$$



NEURAL ARCHITECTURE

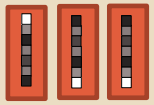


Localization

`PointerNet(CandidateNodes…)`

Repair Given Location

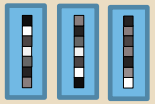
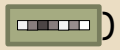
Variable Misuse

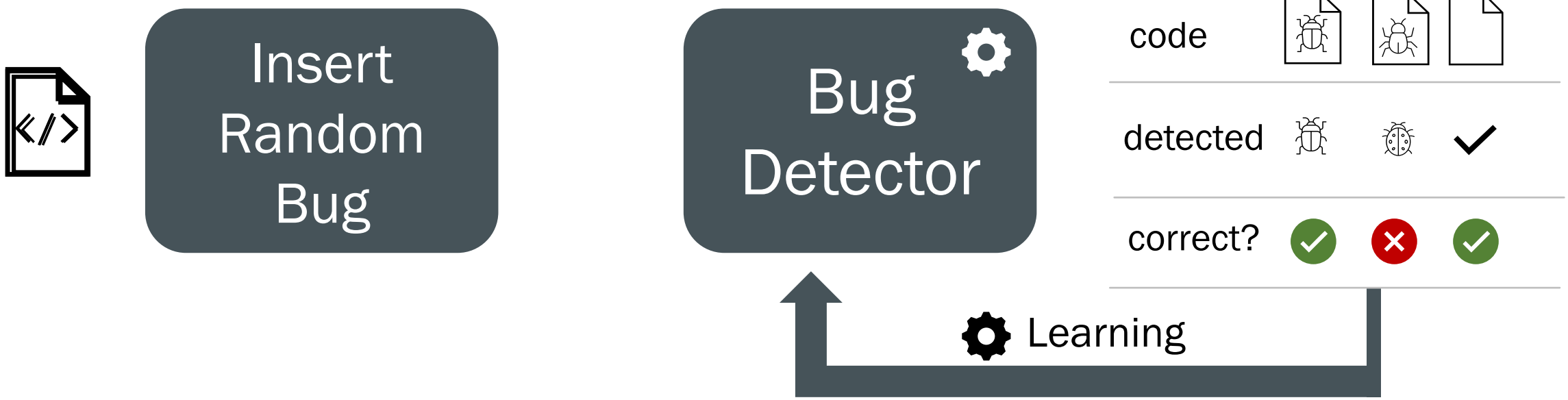
`PointerNet(AlternativeNodes…`  `|`  `)`

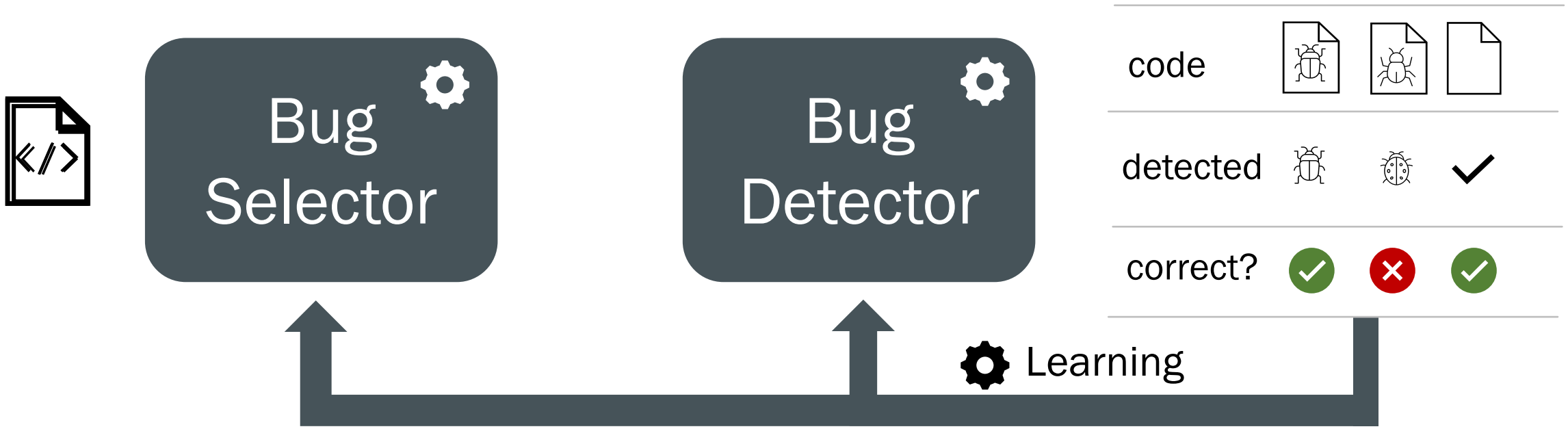
Text Rewrite

`MaskedMLP(`  `) -> {+,-,*,/,0,1,2,and,or,…}`

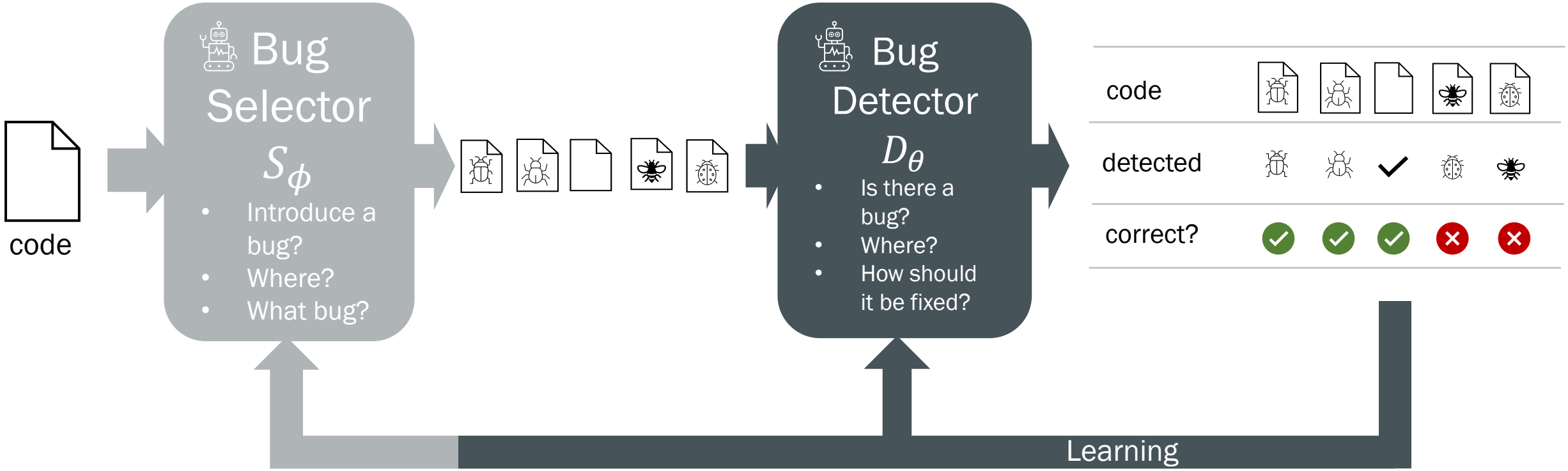
Argument Swapping

`PointerNetOverPairs(ArgNodes…`  `|`  `)`


Insert Random Bug
Bug Detector
code
detected
correct?
Learning


Bug Selector
Bug Detector
code
detected
correct?
Learning

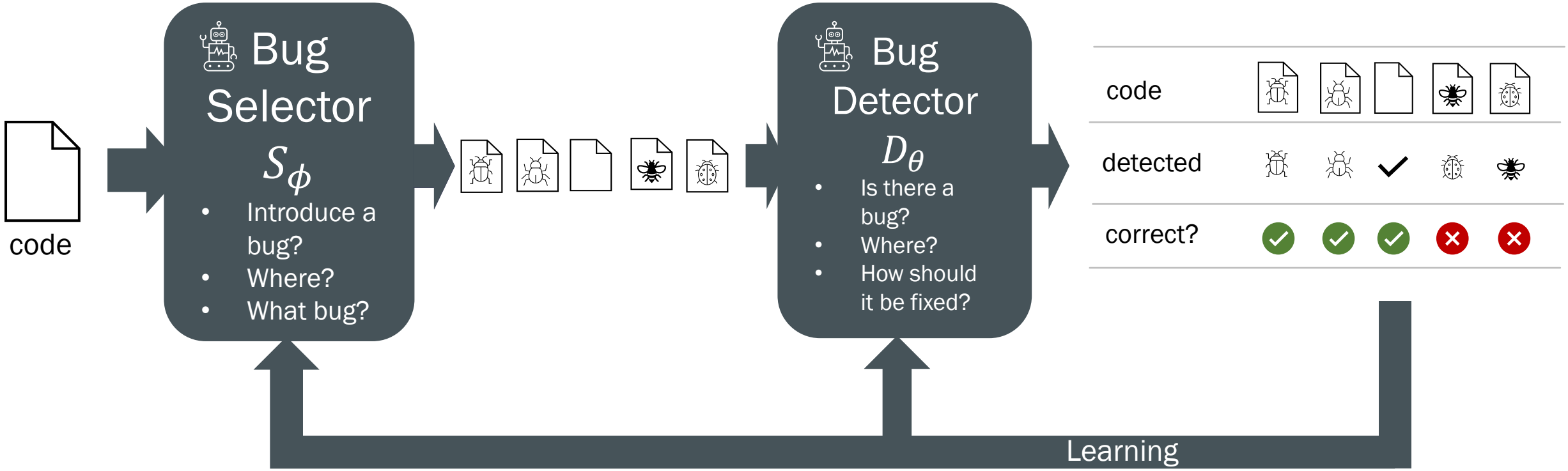
LEARNING



$$E_{\mathrm{s}\sim C}\left[\max_{\langle \ell,\rho\rangle \in R_{\mathrm{s}}^{\mathcal{R}}} \mathcal{L}_{D_\theta}\left(\mathrm{s}[\rho]_\ell, \langle \ell, \rho^{-1}\rangle\right)\right]$$

BugLab

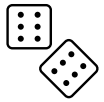
LEARNING



$$\max_{\phi} \min_{\theta} E_{\mathrm{s}\sim C}\left[E_{\langle \ell,\rho\rangle\sim S_\phi(\mathrm{s})}\left[\mathcal{L}_{D_\theta}\left(\mathrm{s}[\rho]_\ell, \langle \ell, \rho^{-1}\rangle\right)\right]\right]$$

BugLab

EVALUATION DATASETS



RandomBugs

- ~700k random bugs
- Relatively Large
- Potentially non-representative of real bugs

PyPIBugs

- 2k real bugs
- Manually curated/labeled
- Small. Used as testset only.

LOCALIZATION & REPAIR ACCURACY

RandomBugs

	GNN	GREAT
Supervised	62.4	51.0
BugLab	**70.3**	65.3

PyPIBugs

	GNN	GREAT
Supervised	20.1	16.5
BugLab	**26.2**	22.9

GREAT: Hellendoorn, V. J., et al. "Global relational models of source code." *ICLR* 2019

BugLab

By Francis Barlow - http://mythfolklore.net/aesopica/barlow/59.htm, Public Domain, https://commons.wikimedia.org/w/index.php?curid=14476836

program analysis

The ~~boy~~ who

error

cried ~~wolf~~

Conversation 1 | Commits 1 | Checks 0 | Files changed 1

Changes from **all commits** | **File filter** | **Conversations** | **Jump to**

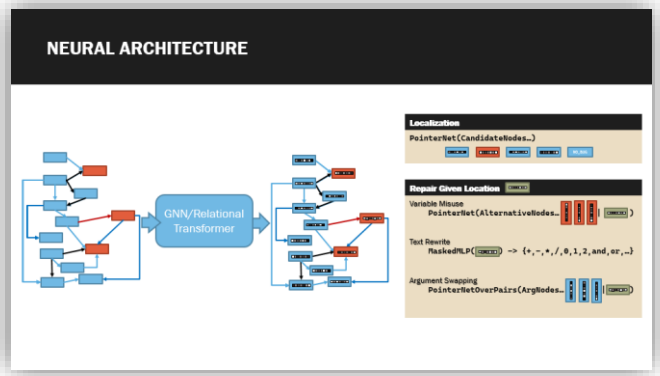
2 gremlin-python/src/main/python/gremlin_python/process/strategies.py

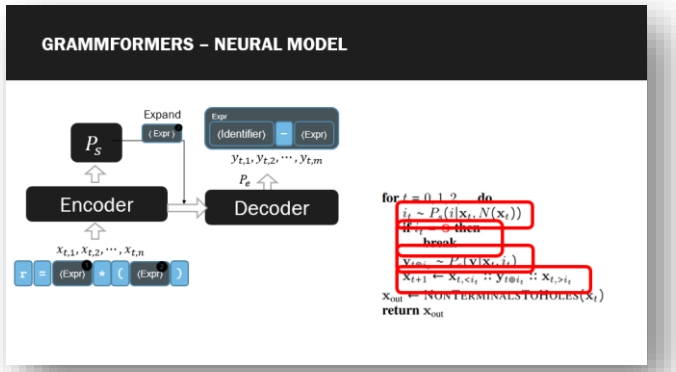
```
@@ -64,7 +64,7 @@ def __init__(self, partition_key=None, write_partition=None, read_partitions=Non
64  64              self.configuration["partitionKey"] = partition_key
65  65          if write_partition is not None:
66  66              self.configuration["writePartition"] = write_partition
67      -       if write_partition is not None:
    67  +       if read_partitions is not None:
68  68              self.configuration["readPartitions"] = read_partitions
69  69          if include_meta_properties is not None:
70  70              self.configuration["includeMetaProperties"] = include_meta_properties
```

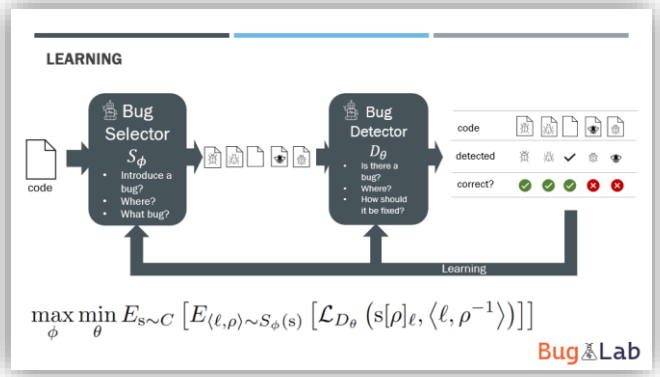
2 webexteamssdk/api/rooms.py

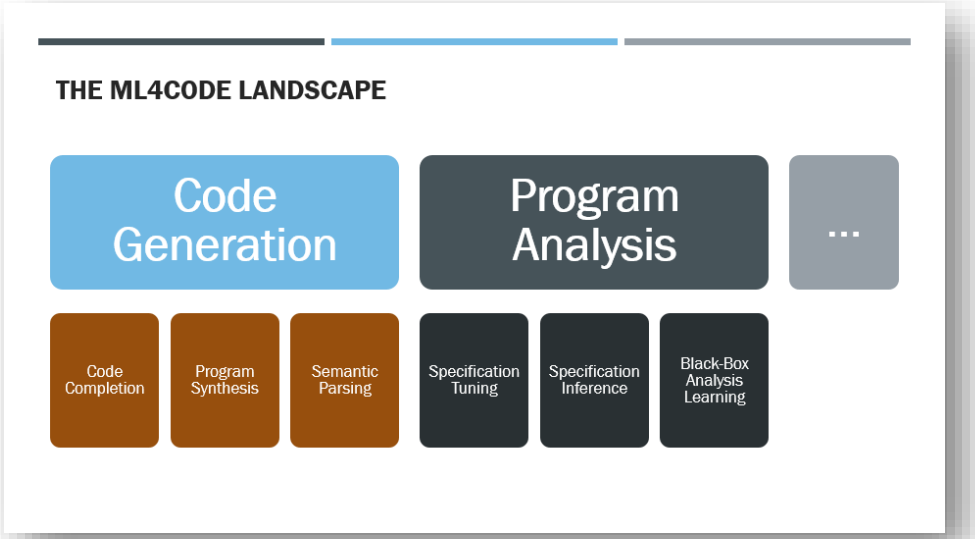
@@ -213,27 +213,27 @@

```
213 213             )
214 214
215 215             # Return a room meeting info object created from the response JSON data
216 216             return self._object_factory("room_meeting_info", json_data)
217 217
218 218         def update(self, roomId, title, **request_parameters):
219 219             """Update details for a room, by ID.
220 220
221 221             Args:
222 222                 roomId(basestring): The room ID.
223 223                 title(basestring): A user-friendly name for the room.
224 224                 **request_parameters: Additional request parameters (provides
225 225                     support for parameters that may be added in the future).
226 226
227 227             Returns:
228 228                 Room: A Room object with the updated Webex Teams room details.
229 229
230 230             Raises:
231 231                 TypeError: If the parameter types are incorrect.
232 232                 ApiError: If the Webex Teams cloud returns an error.
233 233
234 234             """
235 235             check_type(roomId, basestring)
236     -           check_type(roomId, basestring)
    236 +           check_type(title, basestring)
237 237
238 238             put_data = dict_from_items_with_values(
239 239                 request_parameters,
```









GENERATING & REPAIRING CODE WITH DEEP LEARNING

MILTOS ALLAMANIS

miltos.allamanis.com

@miltos1