Examples out of Thin Air

AI-generated Dynamic Context to Assist Program Comprehension by Example

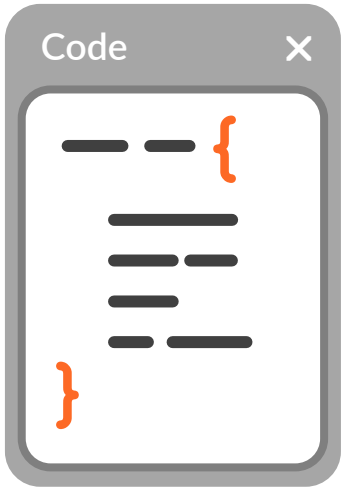
Toni Mattis
Eva Krebs
Martin Rinard
Robert Hirschfeld

Software Architecture Group
HPI, University of Potsdam, Germany

CSAIL
MIT, Cambridge, MA, USA

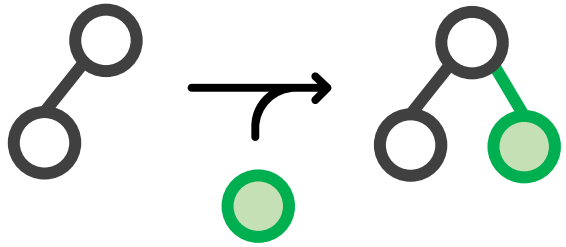
PX/24 | Lund, Sweden | March 11

Seeing Concrete State & Behavior

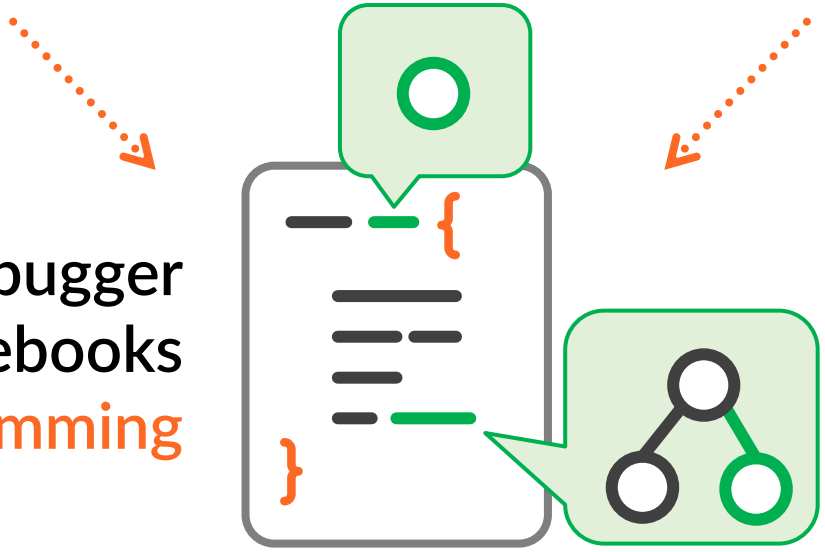


Code
Abstract

Data
Concrete



Behavior
Concrete



Debugger
REPL / Notebooks
Example-based Live Programming

Example-Centric Programming

Edwards 2004

Kanon

Oka et al. 2018

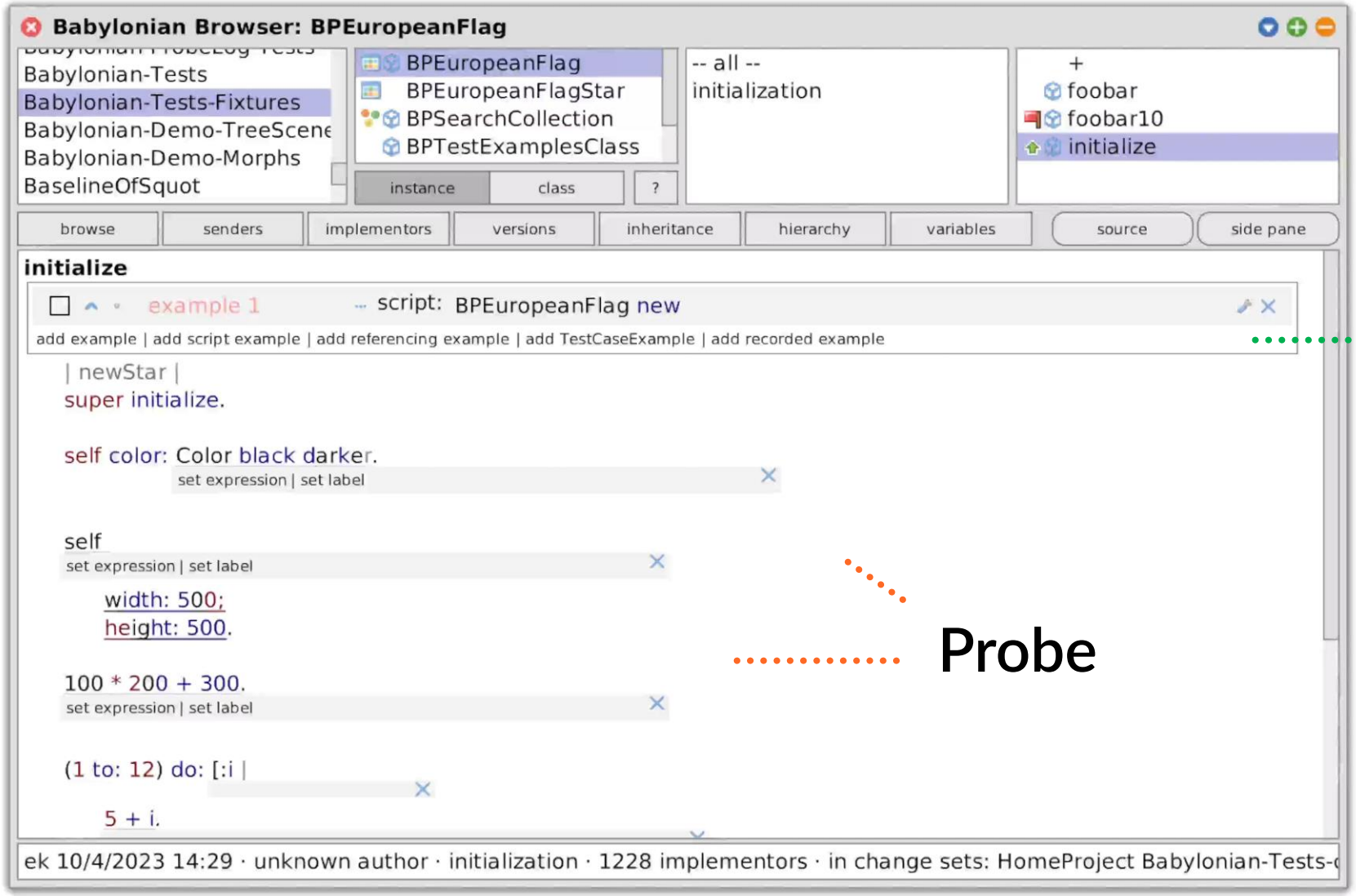
Maniposynth

Hempel & Chugh 2022

Babylonian/G

Krebs et al. 2024

Babylonian Programming | **Babylonian/S**



Classes

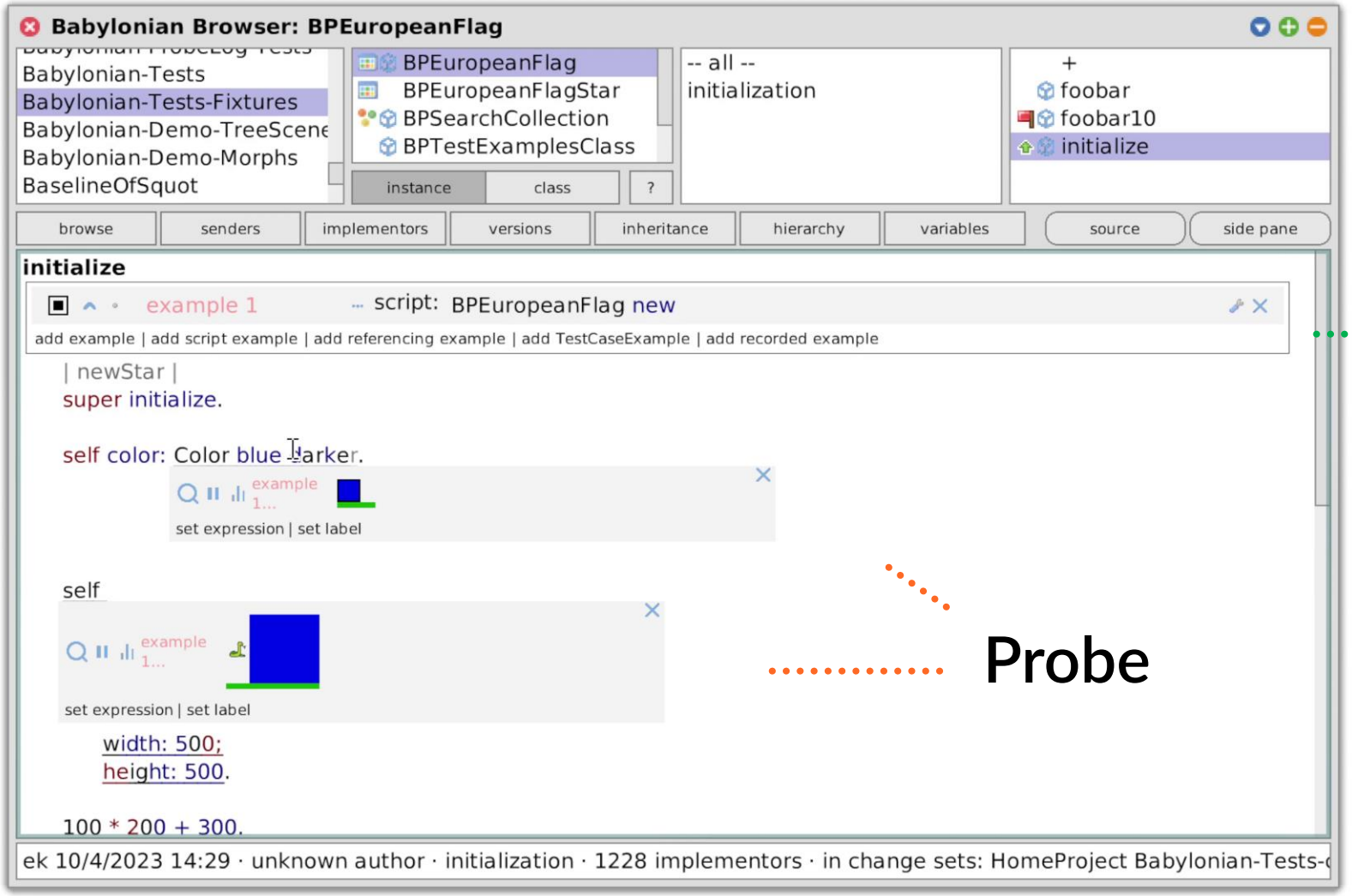
Methods

Example

Probe

Method Source

Babylonian Programming | **Babylonian/S**



Classes

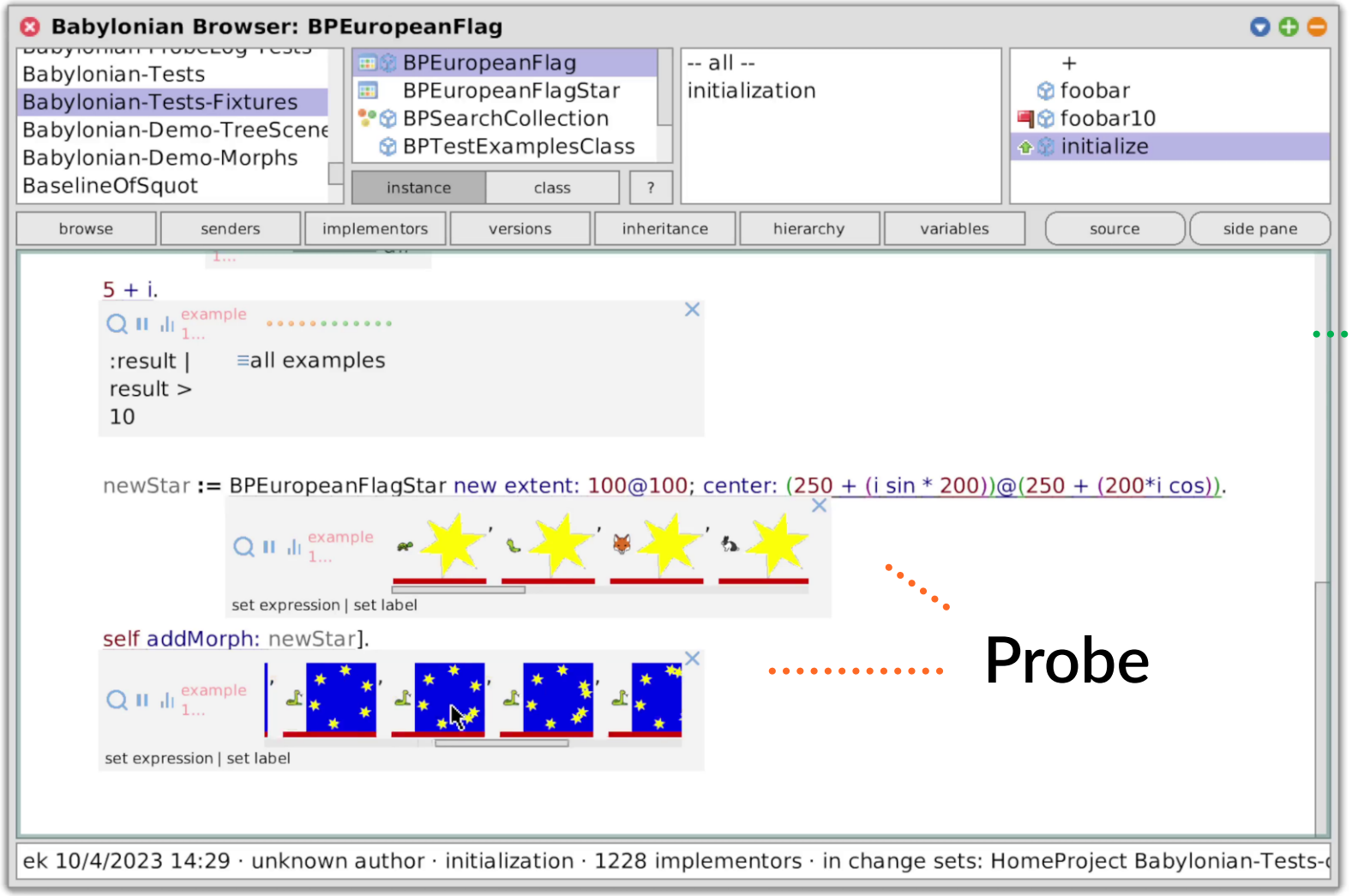
Methods

Example

Method Source

Probe

Babylonian Programming | **Babylonian/S**



Classes

Methods

Example

Method Source

Probe

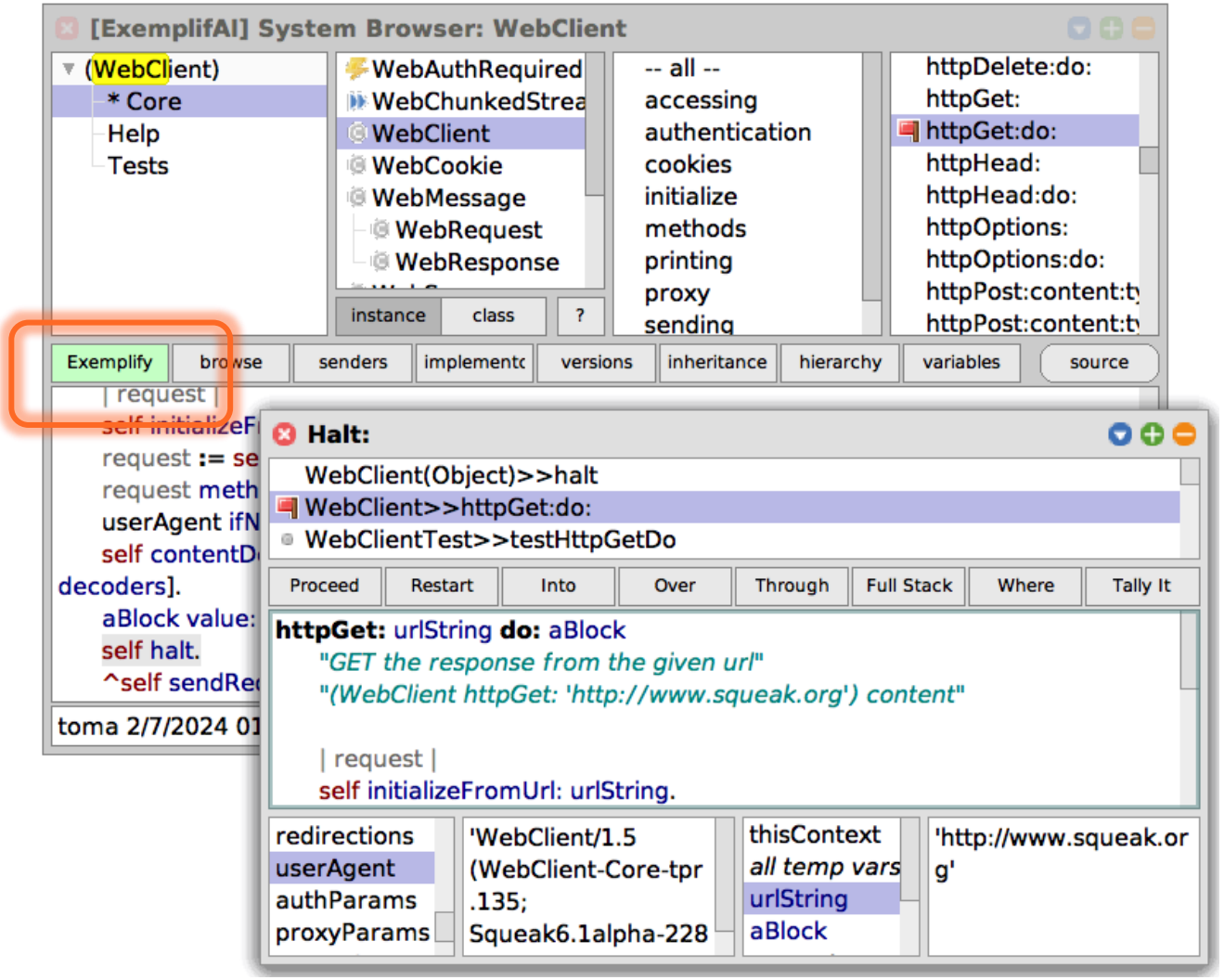
Examples are created manually!

Goal:

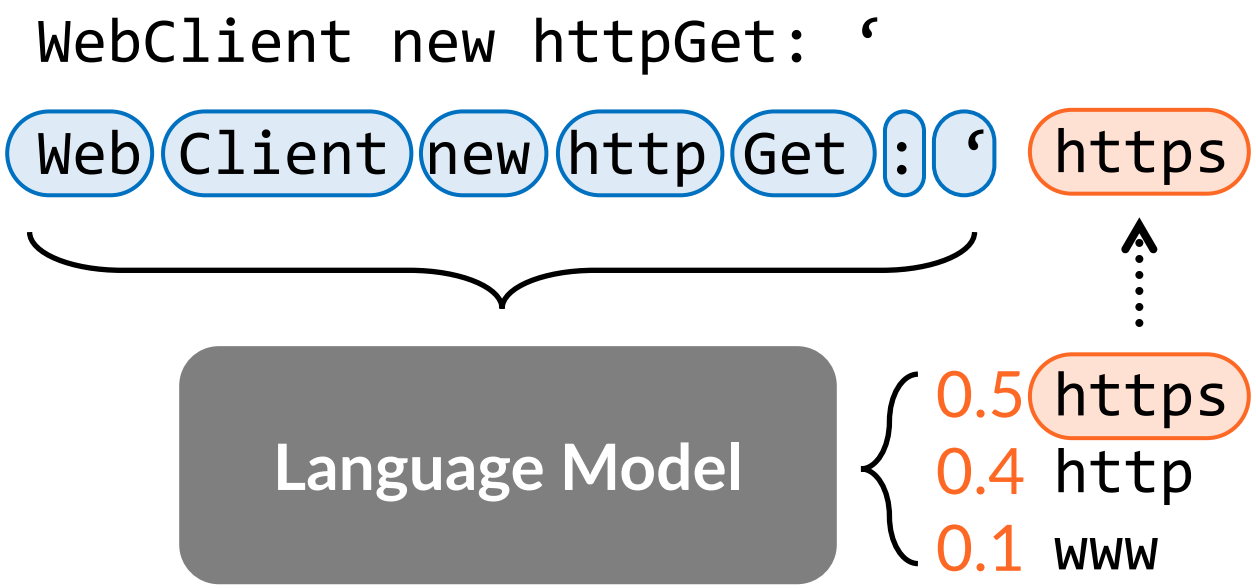
Automatically get [part of] a running program into a state that illustrates its behavior

Opportunity:

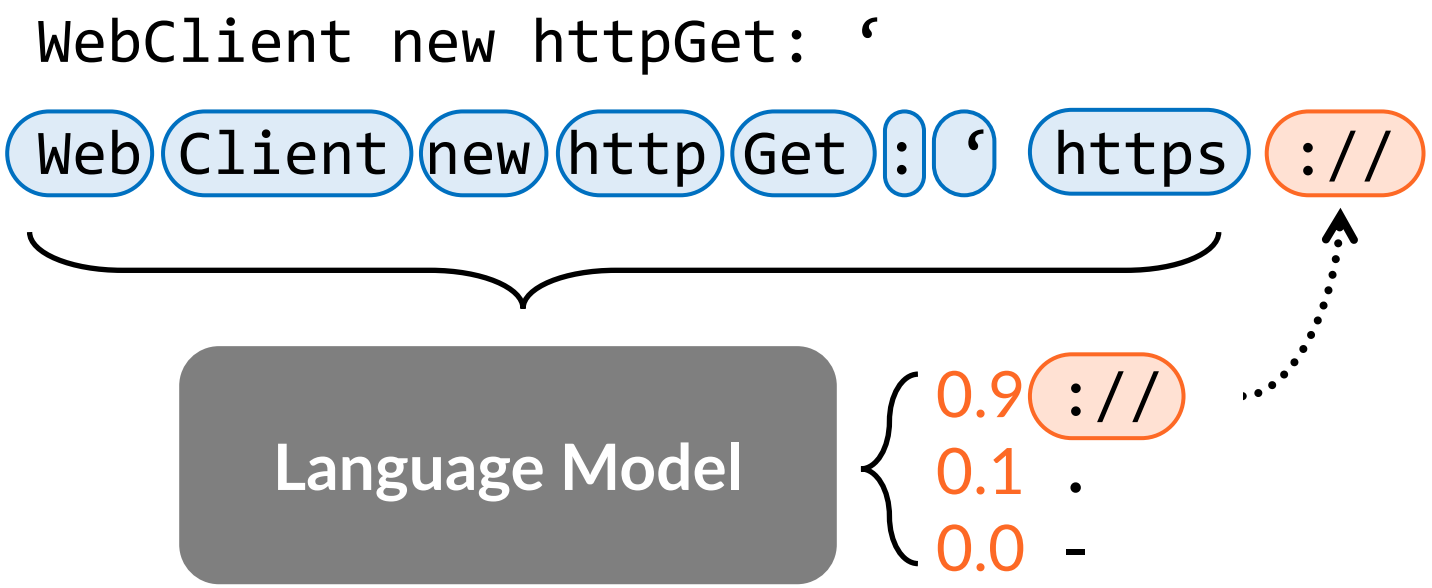
Use “AI” (large language models) to generate such a state


[ExemplifAI] System Browser: WebClient
(WebClient)
* Core
Help
Tests
WebAuthRequired
WebChunkedStrea
WebClient
WebCookie
WebMessage
WebRequest
WebResponse
instance
class
?
-- all --
accessing
authentication
cookies
initialize
methods
printing
proxy
sending
httpDelete:do:
httpGet:
httpGet:do:
httpHead:
httpHead:do:
httpOptions:
httpOptions:do:
httpPost:content:t
httpPost:content:t
Exemplify
browse
senders
implementc
versions
inheritance
hierarchy
variables
source
| request |
request := se
request meth
userAgent ifN
self contentD
decoders].
aBlock value:
self halt.
^self sendRe
toma 2/7/2024 01
Halt:
WebClient(Object)>>halt
WebClient>>httpGet:do:
WebClientTest>>testHttpGetDo
Proceed
Restart
Into
Over
Through
Full Stack
Where
Tally It
httpGet: urlString do: aBlock
"GET the response from the given url"
"(WebClient httpGet: 'http://www.squeak.org') content"
| request |
self initializeFromUrl: urlString.
redirections
userAgent
authParams
proxyParams
'WebClient/1.5 (WebClient-Core-tpr .135; Squeak6.1alpha-228
thisContext
all temp vars
urlString
aBlock
'http://www.squeak.org'

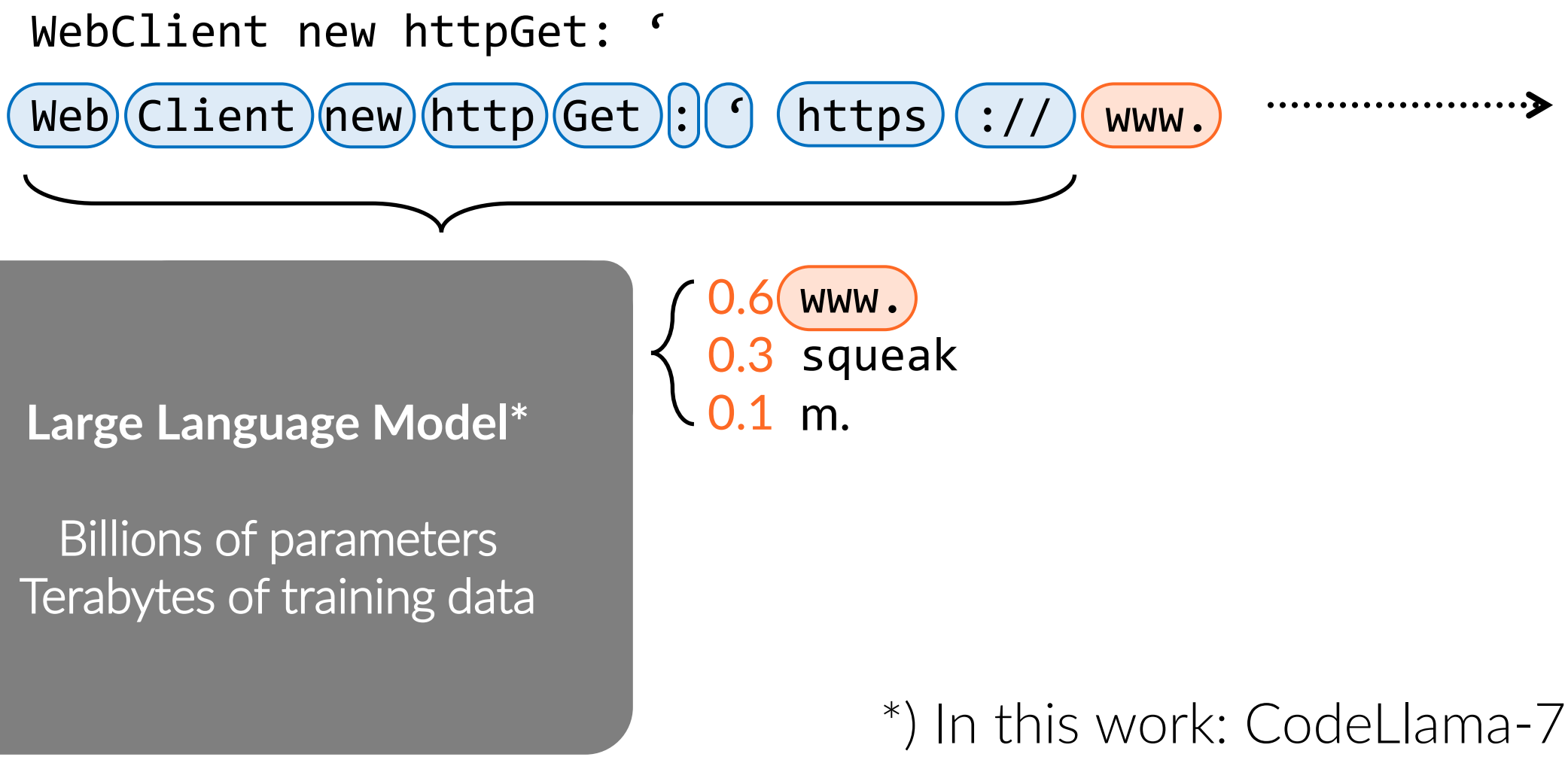
What's an LLM?


WebClient new httpGet: ‘
Web
Client
new
http
Get
:
‘
https
Language Model
0.5 https
0.4 http
0.1 www

What's an LLM?


WebClient new httpGet: ‘
Web
Client
new
http
Get
:
‘
https
://
Language Model
0.9
://
0.1
.
0.0
-

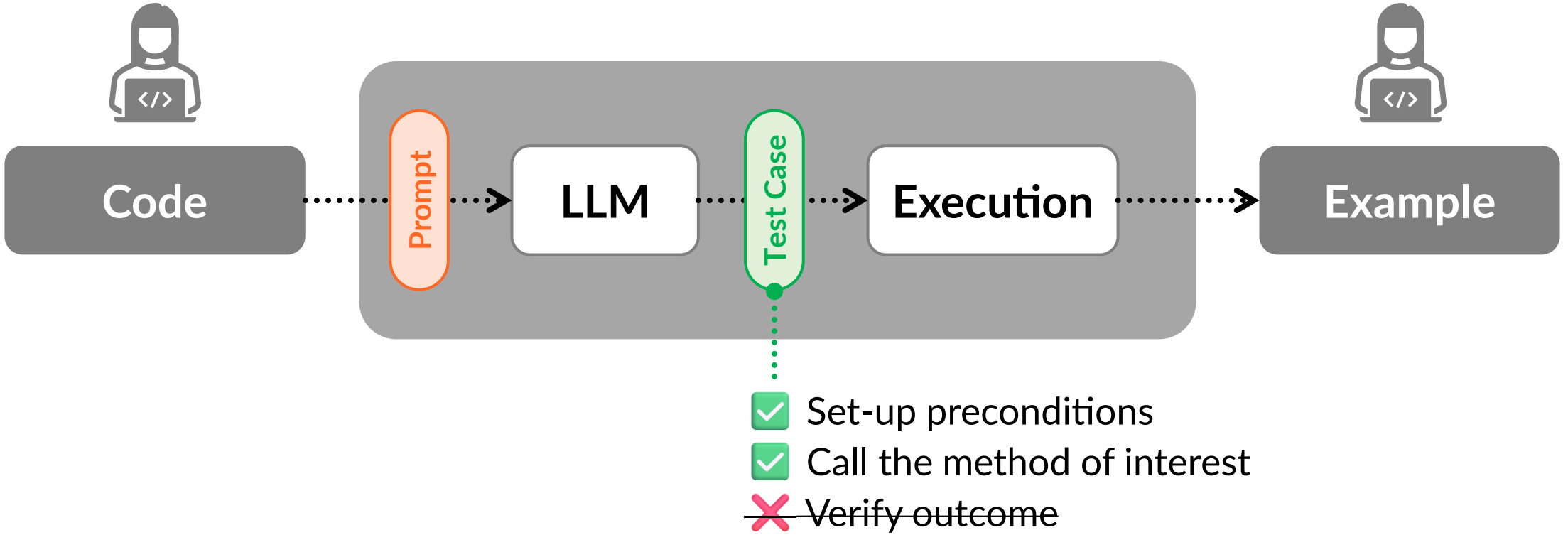
What's an LLM?



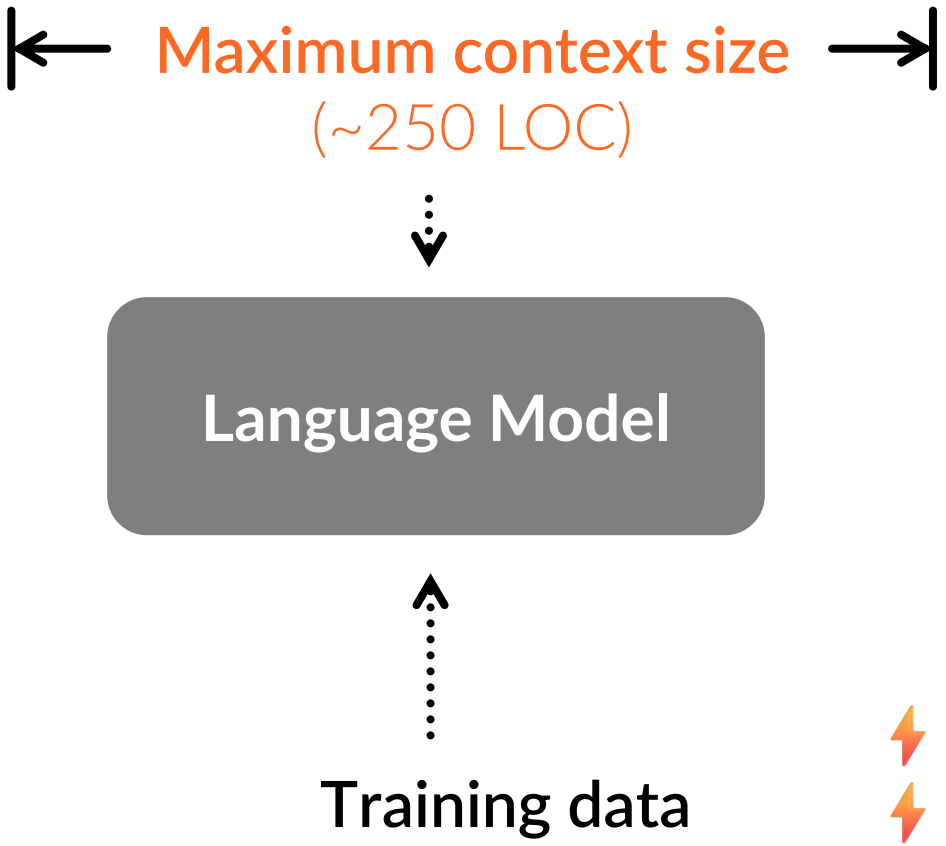
*) In this work: CodeLlama-7B

Approach

LLMs can generate **code** but not **run-time state** (yet)

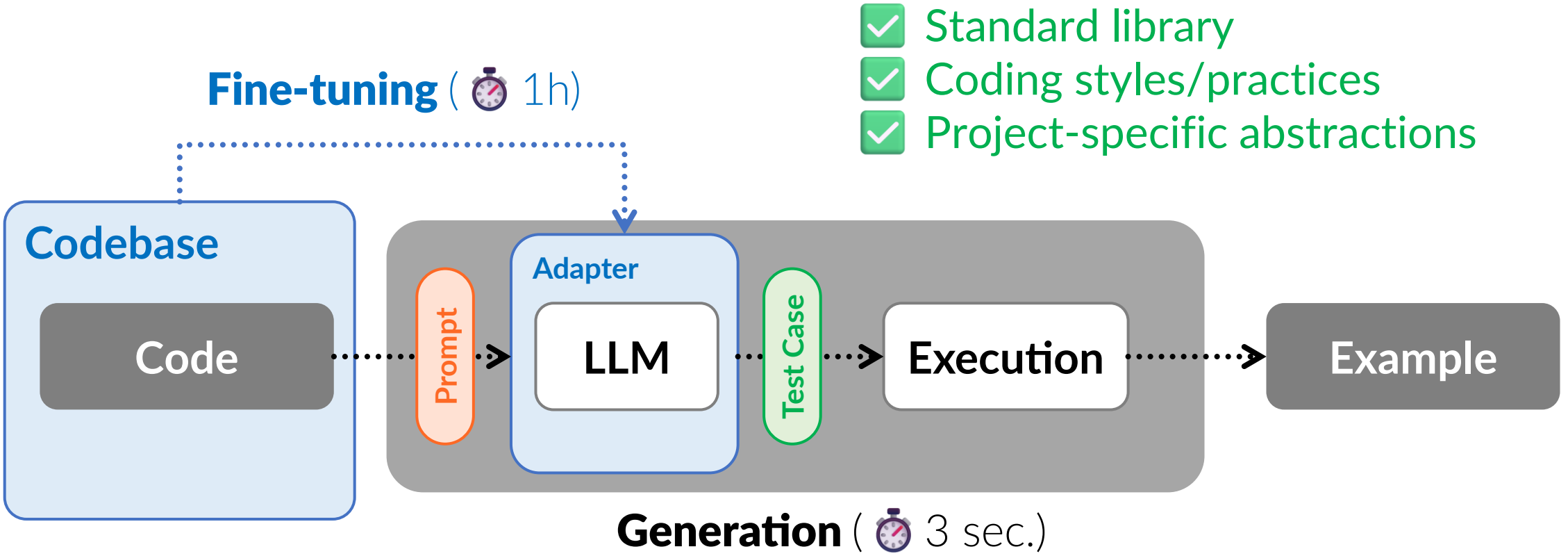


Approach | **Limitations**

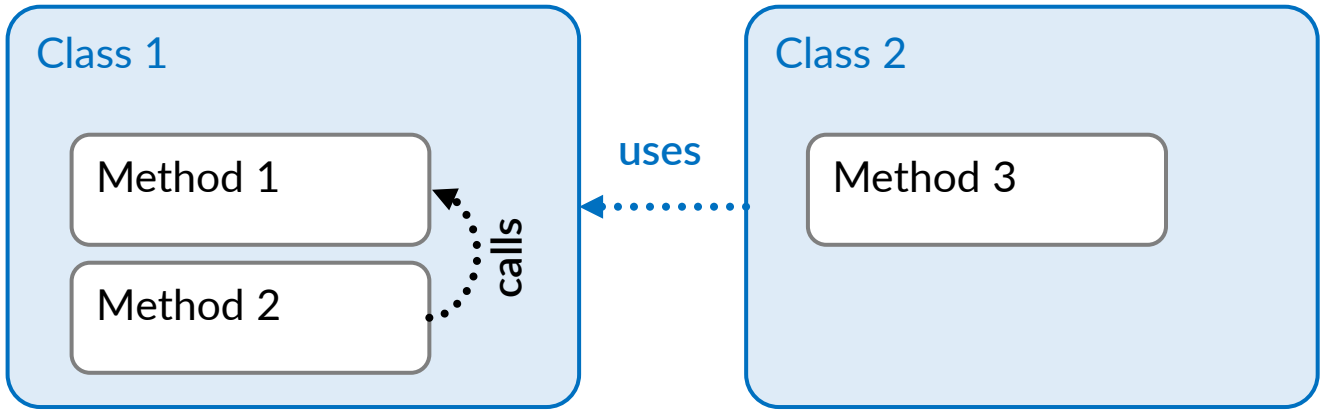


- No project-specific knowledge
- Mixed versions and distributions
- Out of date

Approach | **Fine-tuning**



Approach | **Fine-tuning**



Problem structure:

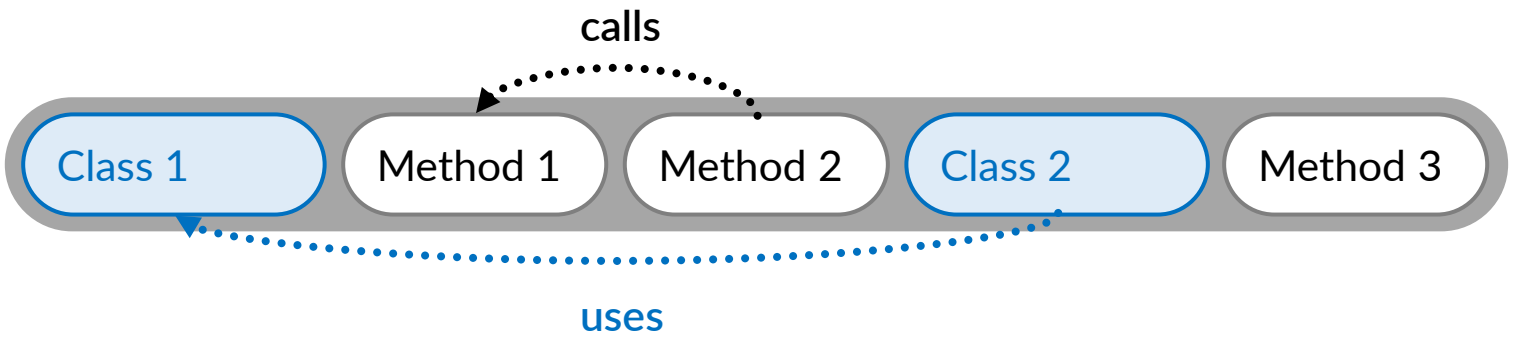
calls/uses

Code

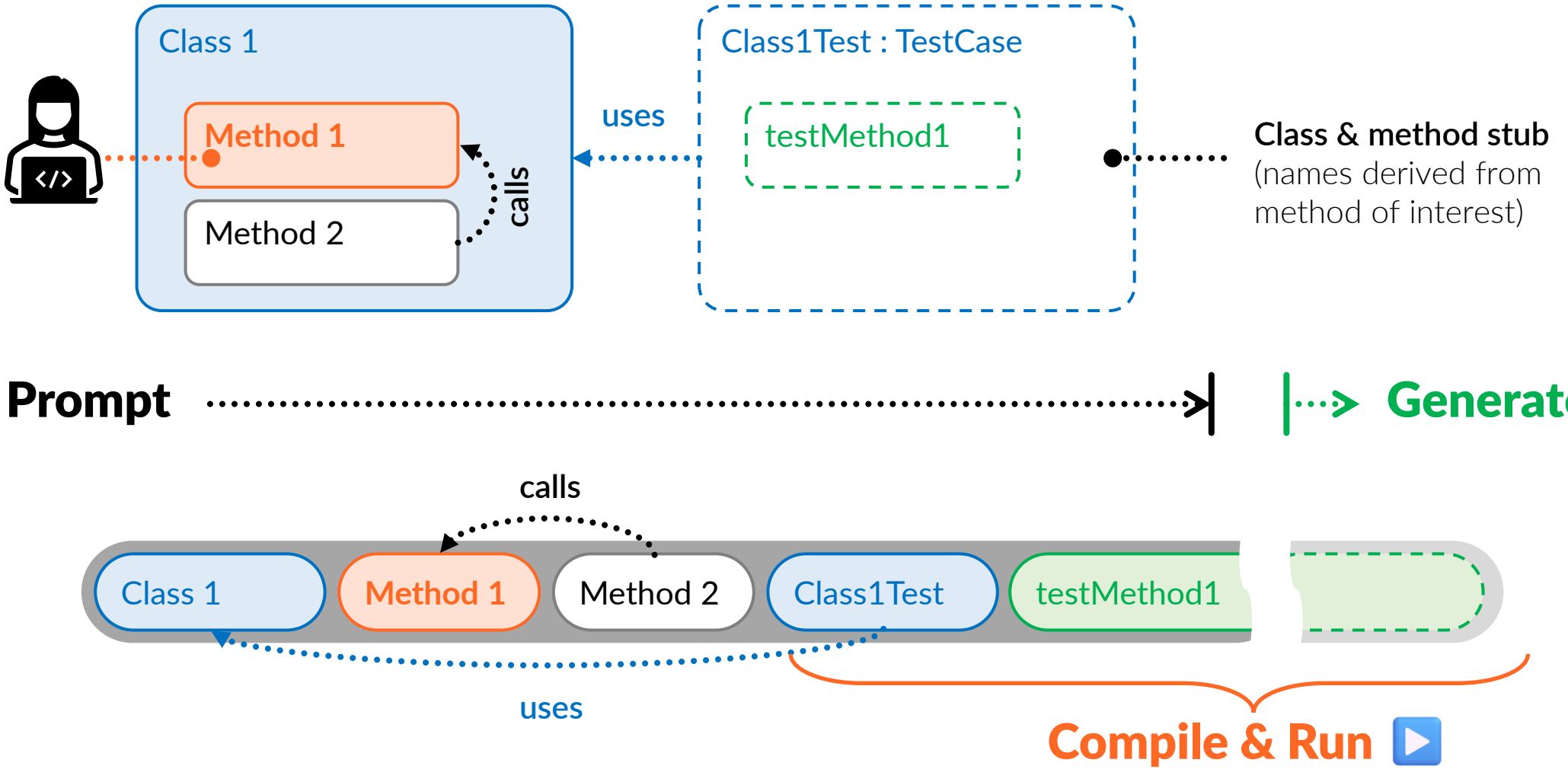
Example

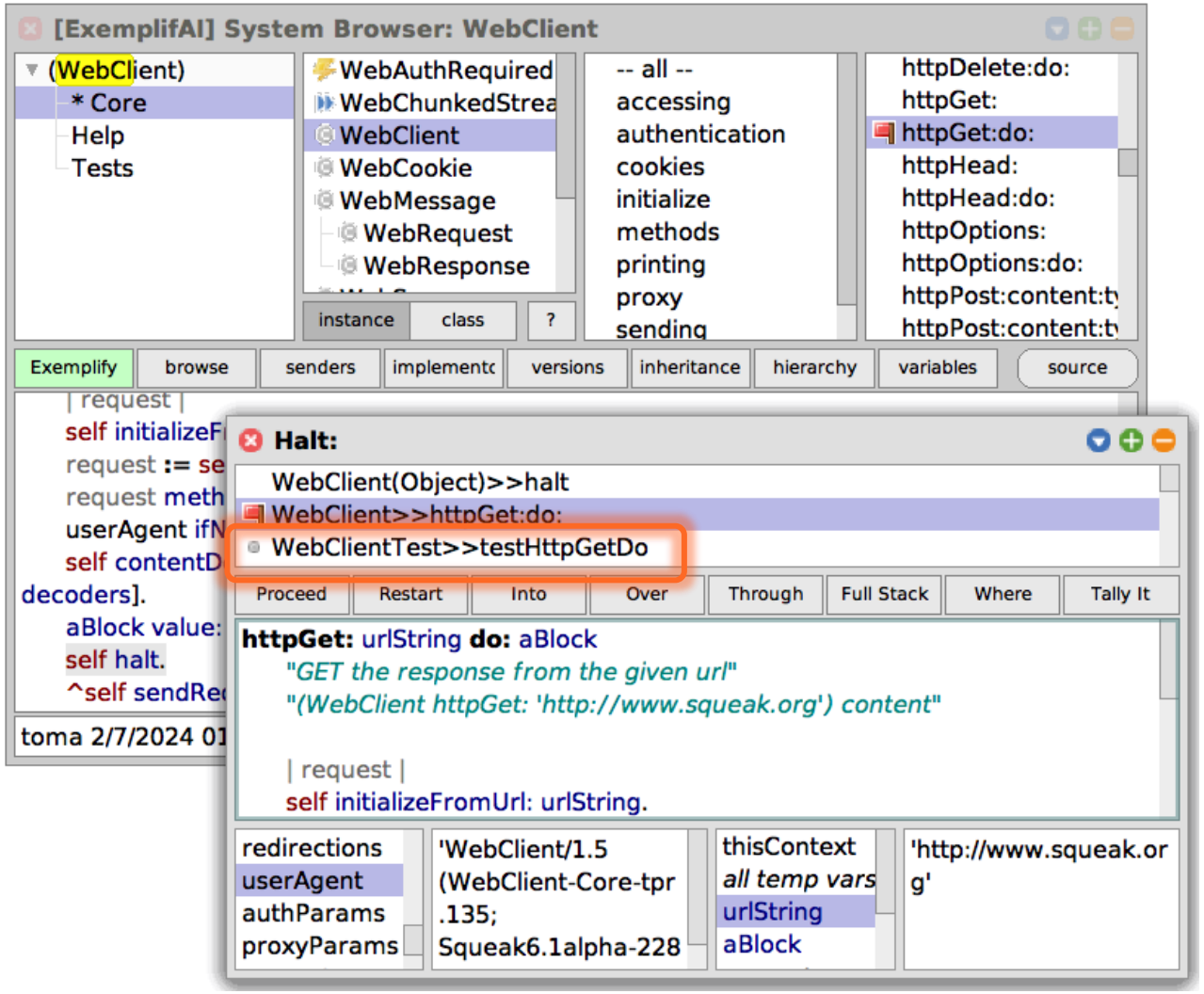
LLM completes left-to-right

Fine-tuning Data

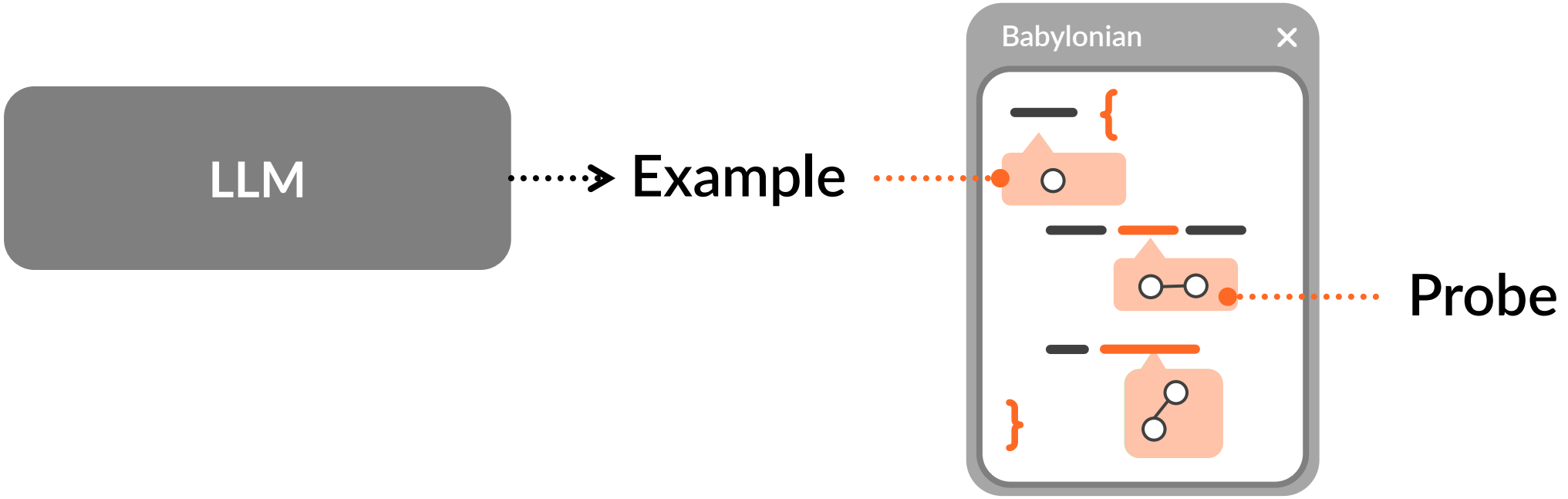


Approach | **Generation**




[ExemplifAI] System Browser: WebClient
(WebClient)
* Core
Help
Tests
WebAuthRequired
WebChunkedStrea
WebClient
WebCookie
WebMessage
WebRequest
WebResponse
instance
class
?
-- all --
accessing
authentication
cookies
initialize
methods
printing
proxy
sending
httpDelete:do:
httpGet:
httpGet:do:
httpHead:
httpHead:do:
httpOptions:
httpOptions:do:
httpPost:content:t
httpPost:content:t
Exemplify
browse
senders
implementc
versions
inheritance
hierarchy
variables
source
| request |
self initializeF
request := se
request meth
userAgent ifN
self contentD
decoders].
aBlock value:
self halt.
^self sendRe
toma 2/7/2024 01
Halt:
WebClient(Object)>>halt
WebClient>>httpGet:do:
WebClientTest>>testHttpGetDo
Proceed
Restart
Into
Over
Through
Full Stack
Where
Tally It
httpGet: urlString do: aBlock
"GET the response from the given url"
"(WebClient httpGet: 'http://www.squeak.org') content"
| request |
self initializeFromUrl: urlString.
redirections
userAgent
authParams
proxyParams
'WebClient/1.5 (WebClient-Core-tpr .135; Squeak6.1alpha-228
thisContext
all temp vars
urlString
aBlock
'http://www.squeak.org'

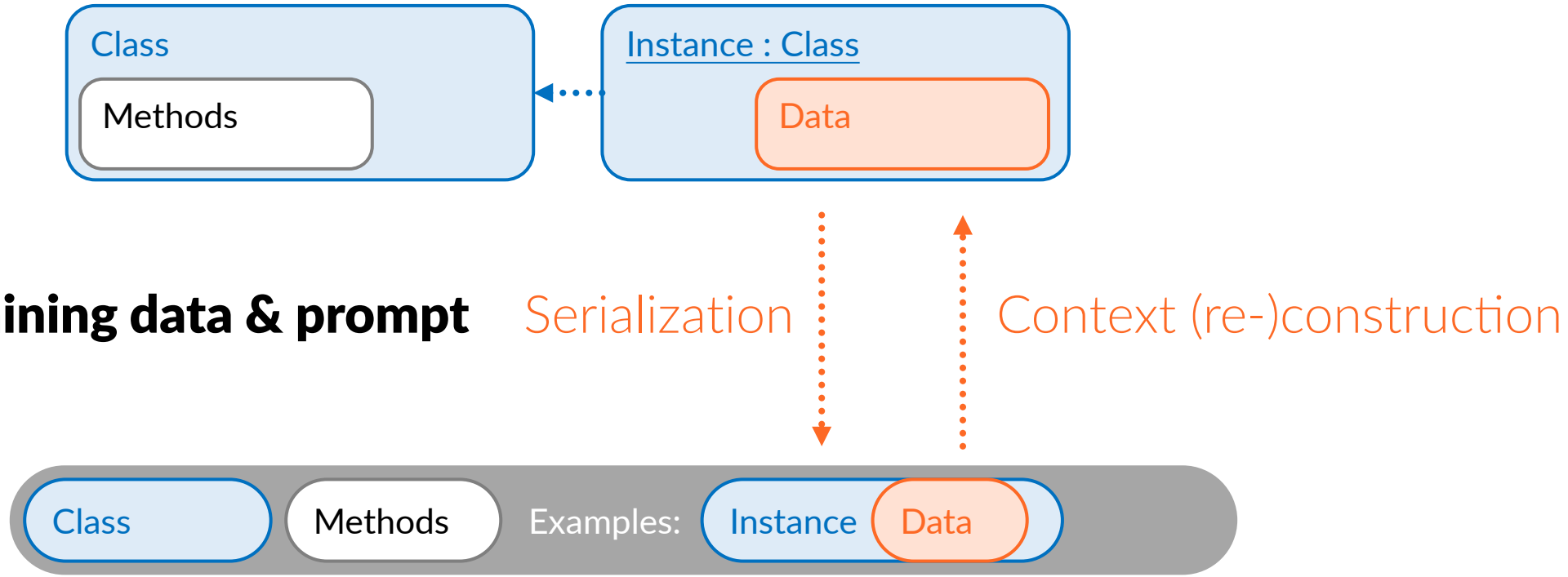
Next Steps | **Babylonian Integration**



- Study usefulness
- Interactive refinement

Next Steps | **Directly to Objects**

Goal: Remove indirection over test code



Next Steps | **Data Sources**

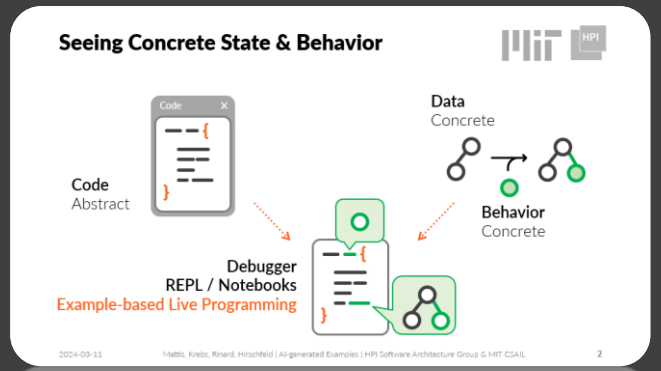
Repository and Community Resources

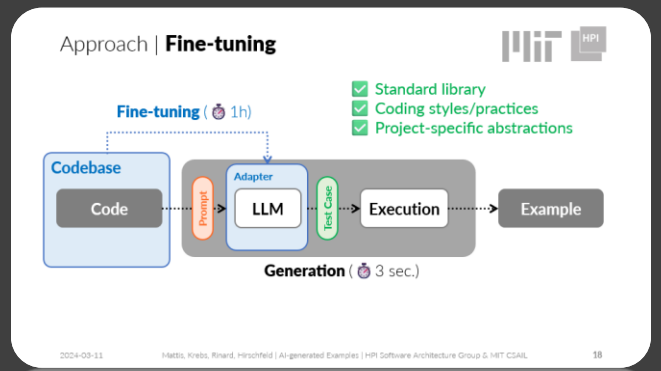
- Examples from documentation, mailing lists, …
- Ethical/legal considerations when using community content for training (plagiarism, attribution, consent, …)

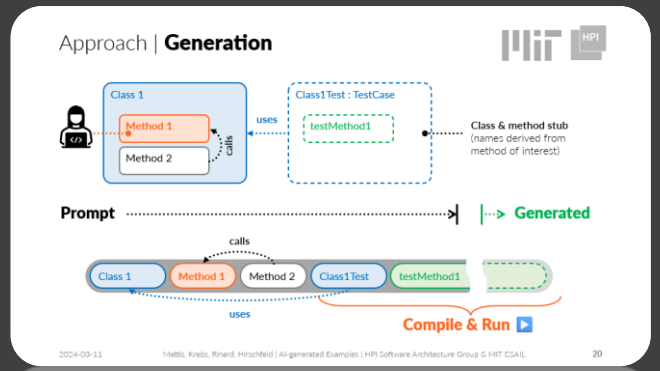
Summary

- (Fine-tuned) LLM that generates examples in Squeak/Smalltalk
- LLMs generate abstract code, live programming is about concrete data and behavior and feedback

How can we better co-design LP + LLMs?

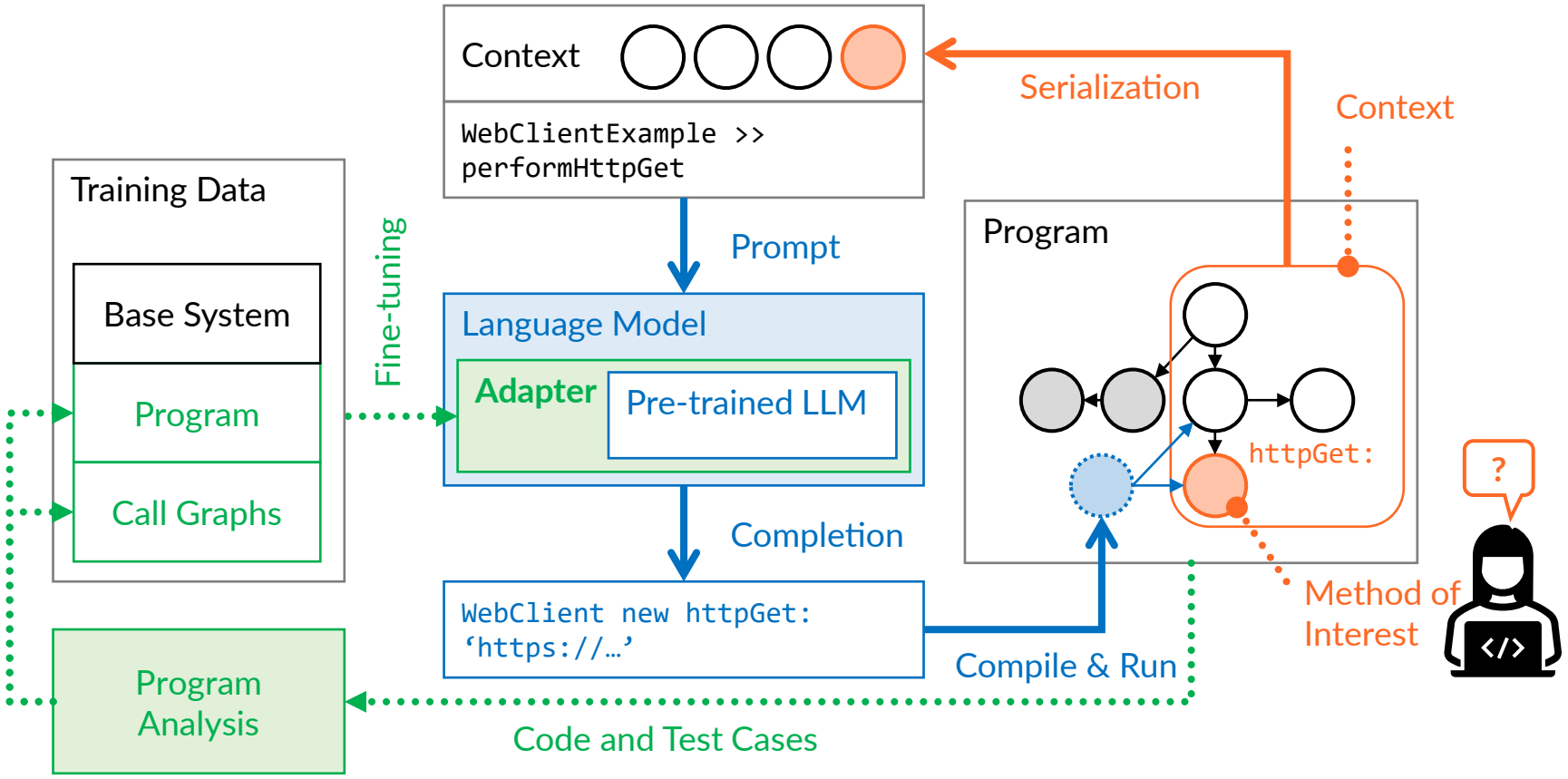




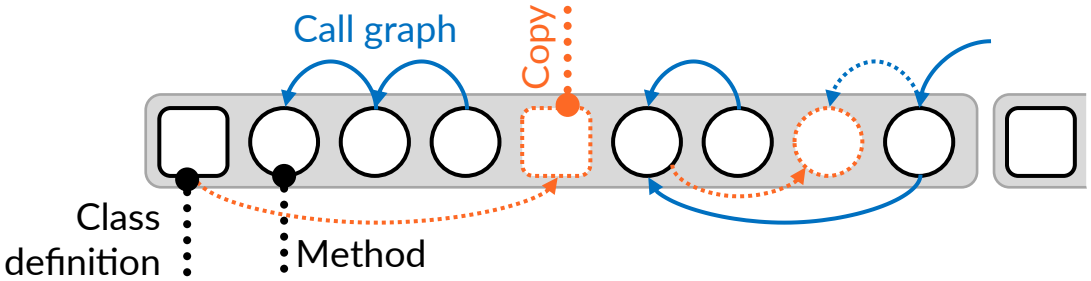


Backup Slides

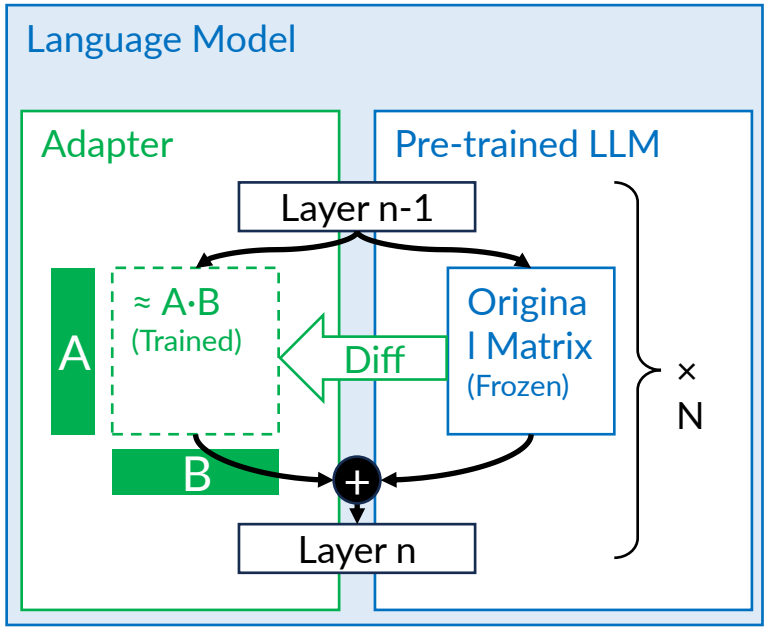
Architecture Overview



Fine-tuning



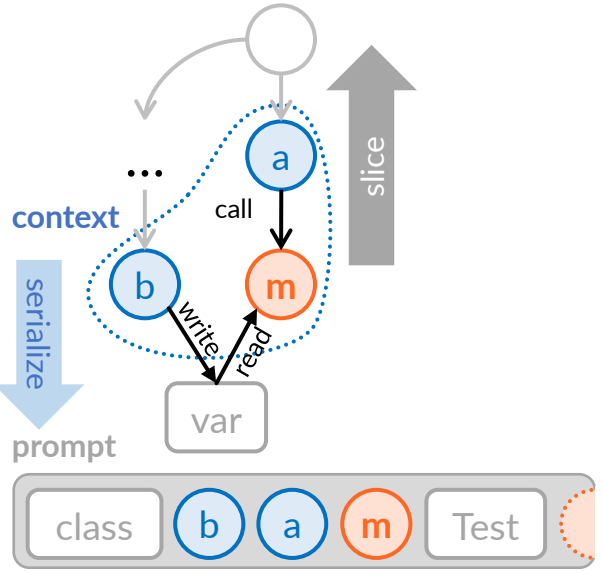
- Filtering (80% of Methods)
- Insert redundant definitions



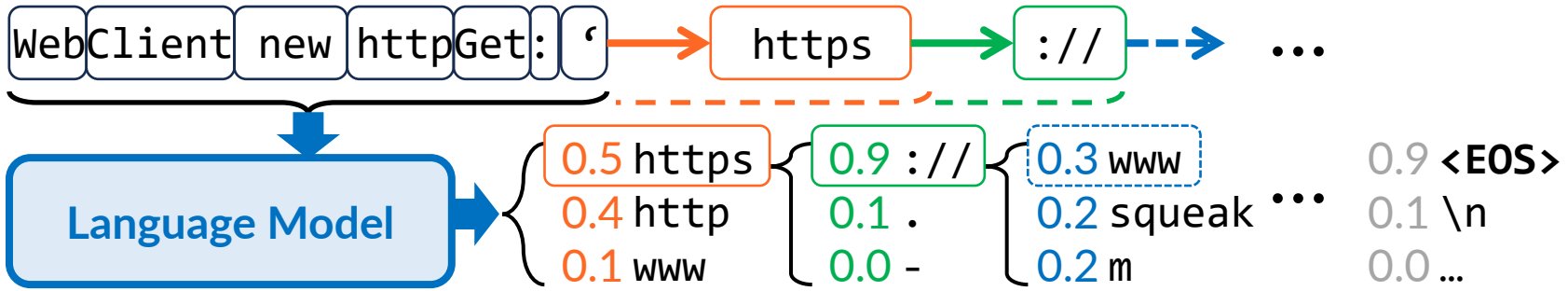
- Low-rank Adaptation (LoRA)
- Train <1% of model weights

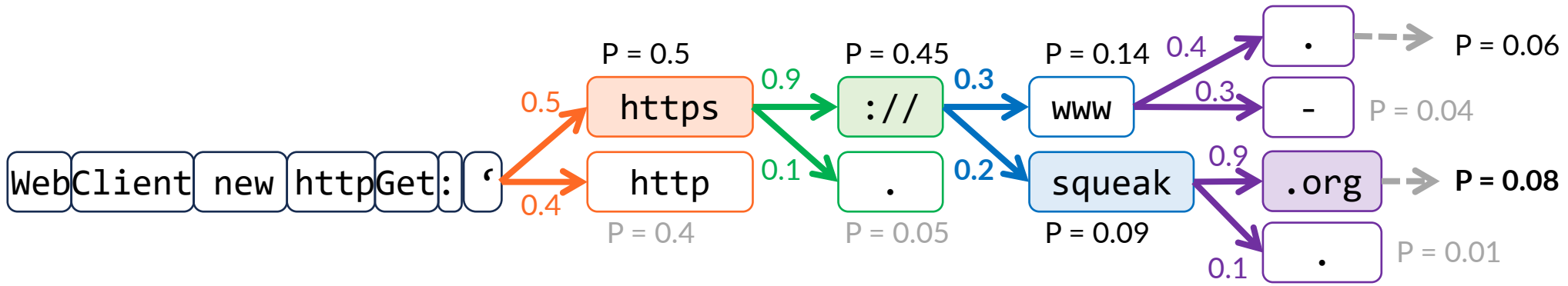
Prompt Construction

- Follow Dependencies
  - Call Graph
    - Dynamically sampled for project
    - Statically approximated for rest of environment
  - Data dependencies
    - Symbolic execution → Data flow graph
    - Link inter-method DFGs to a full DFG with help of call graph
- Include immediate dependents
- **Future work:** Ablation study



Greedy vs. Search





Example Prompt


```
<class>
Object subclass: #WebClient
    instanceVariableNames: 'flags server scheme timeout userAgent …
    classVariableNames: 'DebugLog FlagAcceptCookies FlagAllowRedirect …
    category: 'WebClient-Core'
</class>
<method>
WebClient >> httpGet: urlString do: aBlock
    "GET the response from the given url"
</method>
<method>
WebClientTest >> testHttpGetDo
    "Create a WebClient instance and call httpGet:do: with an example"
```

```
  | client |
  client := WebClient new.
  client httpGet: 'http://www.squeak.org'
  do: [:request |
       self assert: request method = 'GET'].
</method>
```

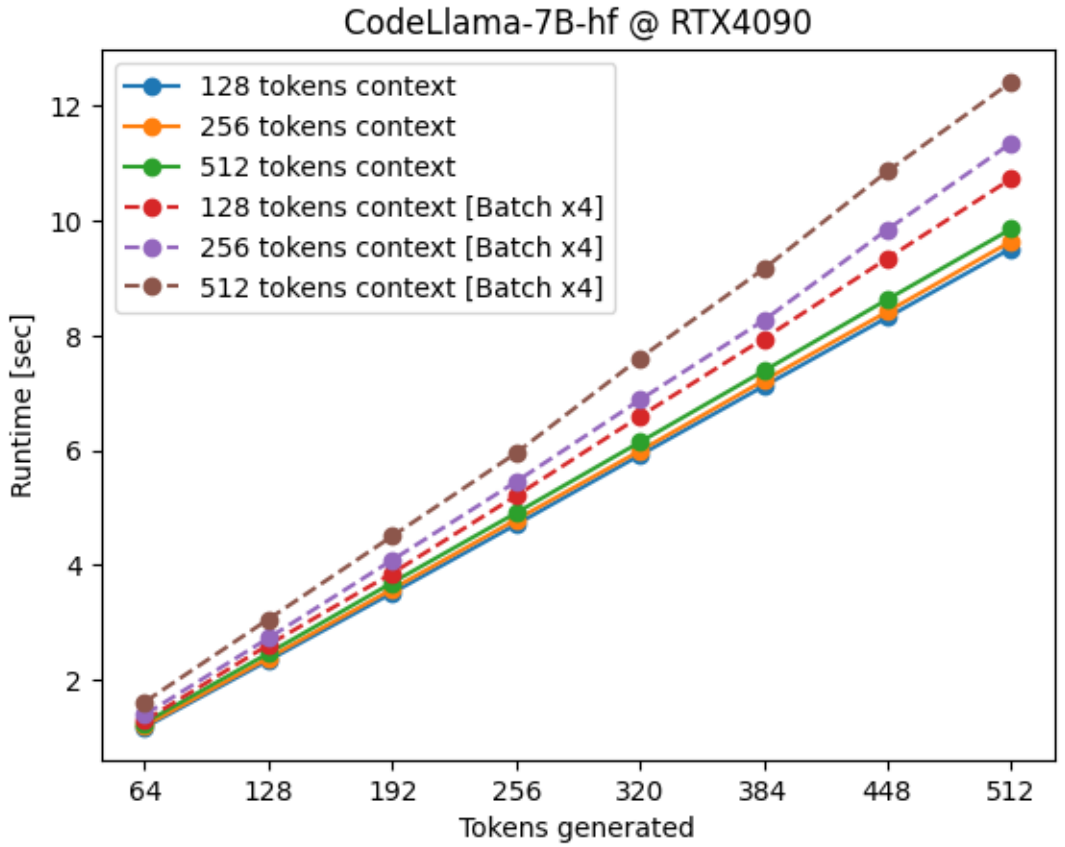
Context
Class, related Methods

Target Method

Test Stub

Generated
Output

Benchmark




```
------------------------------------------------------------
parallel generations: 1
in      out
        64      128     192     256     320     384     448     512
128     1.2     2.3     3.5     4.7     5.9     7.1     8.3     9.5
256     1.2     2.4     3.6     4.8     6.0     7.2     8.4     9.6
384     1.2     2.4     3.6     4.9     6.1     7.3     8.5     9.8
512     1.3     2.5     3.7     4.9     6.1     7.4     8.6     9.9
640     1.3     2.5     3.7     5.0     6.2     7.5     8.7     10.0
768     1.3     2.6     3.8     5.1     6.3     7.5     8.8     10.1
896     1.3     2.6     3.8     5.1     6.3     7.6     8.9     10.1
1024    1.4     2.6     3.9     5.1     6.4     7.7     9.0     10.3
------------------------------------------------------------
parallel generations: 4
in      out
        64      128     192     256     320     384     448     512
128     1.3     2.6     3.9     5.2     6.6     8.0     9.3     10.7
256     1.4     2.7     4.1     5.5     6.9     8.3     9.8     11.4
384     1.5     2.9     4.3     5.7     7.2     8.8     10.3    12.0
512     1.6     3.1     4.5     6.0     7.6     9.2     10.9    12.4
640     1.7     3.2     4.8     6.4     8.0     9.7     --      --
768     1.9     3.4     5.0     6.7     --      --      --      --
896     2.1     3.7     --      --      --      --      --      --
1024    --      --      --      --      --      --      --      --
------------------------------------------------------------
parallel generations: 16
in      out
        64      128     192     256     320     384     448     512
128     1.7     3.5     --      --      --      --      --      --
```

Training Parameters

- 292k LOC → 5.5M tokens → 5.3k × (4 × 256-token)-batches
- 2 epochs á 20 min (~ 40 min. training)

```
100%|██████████| 5318/5318 [19:54<00:00,  4.45it/s]
Warning: Evaluation disabled. Overfitting.
epoch=1: train_epoch_loss=tensor(1.0421) eval_epoch_loss=tensor(1.0421)
```

```
lr = 3e-4

optimizer = torch.optim.AdamW(model.parameters(), lr=lr)

lr_scheduler = get_linear_schedule_with_warmup(
    optimizer=optimizer,
    num_warmup_steps=0,
    num_training_steps=(len(train_dataloader) * num_epochs),
)
```

[15]:
```
peft_config = LoraConfig(
    task_type=TaskType.CAUSAL_LM,
    inference_mode=False,
    target_modules=['q_proj', 'v_proj'],
    r=16,
    lora_alpha=32,
    lora_dropout=0.05,
    bias="all"
)
```

Data Cleaning (Number of Methods)

passed:	**66020 (81.13%)**	
denied:	546 (0.67%)	Manual Blacklist (Fonts)
empty:	223 (0.27%)	
exceeded blocks:	1616 (1.99%)	Nesting/Complexity
exceeded class variables:	6 (0.01%)	
exceeded classes:	416 (0.51%)	Low Coupling
exceeded keywords:	1197 (1.47%)	Arguments <= 3
exceeded lines:	4561 (5.6%)	
exceeded variables:	1075 (1.32%)	Low State
too many assertions:	1129 (1.39%)	Small Tests
too few letters:	4586 (5.64%)	Magic numbers/Base64/…
total LOC:	**292330**	
inserted 4806 repeated class definitions		Redundancy

Hardware

- NVidia RTX 4090, 24GB Video Memory
  - CodeLlama uses 14GB for inference, 23GB for training
- Ryzen 7 7800X3D
- 64GB DDR5 Memory
- PCIe 4.0 SSD