

Inferring Hierarchical Motifs from Execution Traces

Saba Alimadadi, Ali Mesbah, Karthik Pattabiraman

saba@northeastern.edu

a place of mind



All truths are easy to **understand** once they are discovered;
the point is to **discover** them.

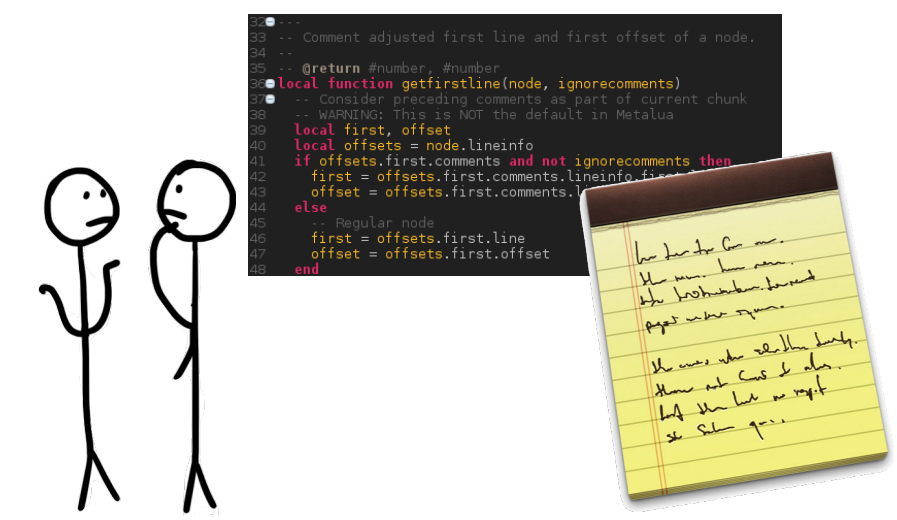
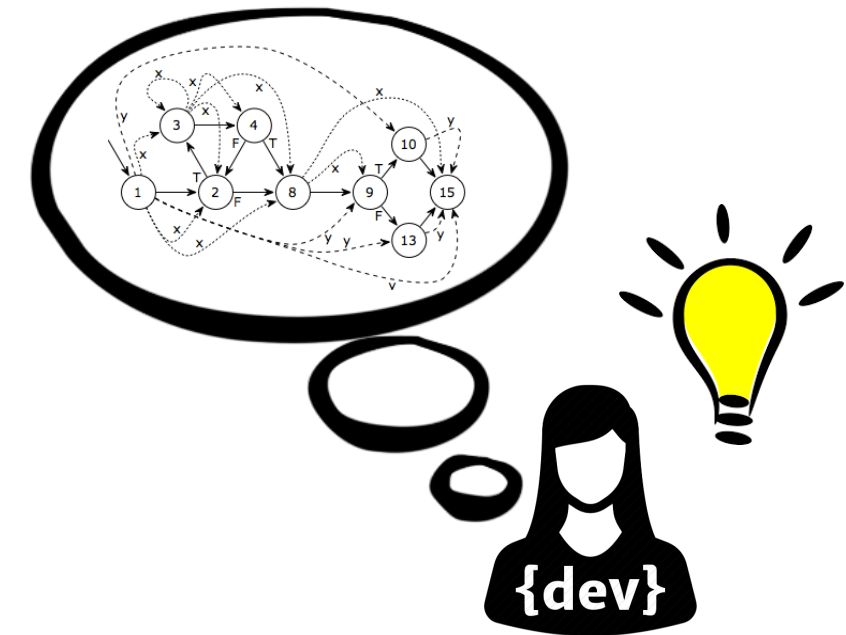
– Galileo Galilei

Search

```
public class MainActivity extends ActionBarActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        // Toggle listener for metroLineSelection  
        ((RadioGroup) findViewById(R.id.metroLineSelection)).setCheckedChangeListener(  
            newCheckedChangeListener() {  
                @Override  
                public void onCheckedChanged(RadioGroup group, int checkedId) {  
                    // Inflate the menu that allows the user to select an item from the action bar if it is present.  
                    getLayoutInflater().inflate(R.menu.main, menu);  
                    return true;  
                }  
            }  
        );  
    }  
}
```

Relate

```
public class MainActivity extends ActionBarActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        // Toggle listener for metroLineSelection  
        ((RadioGroup) findViewById(R.id.metroLineSelection)).setCheckedChangeListener(  
            newCheckedChangeListener() {  
                @Override  
                public void onCheckedChanged(RadioGroup group, int checkedId) {  
                    // Inflate the menu that allows the user to select an item from the action bar if it is present.  
                    getLayoutInflater().inflate(R.menu.main, menu);  
                    return true;  
                }  
            }  
        );  
    }  
}
```

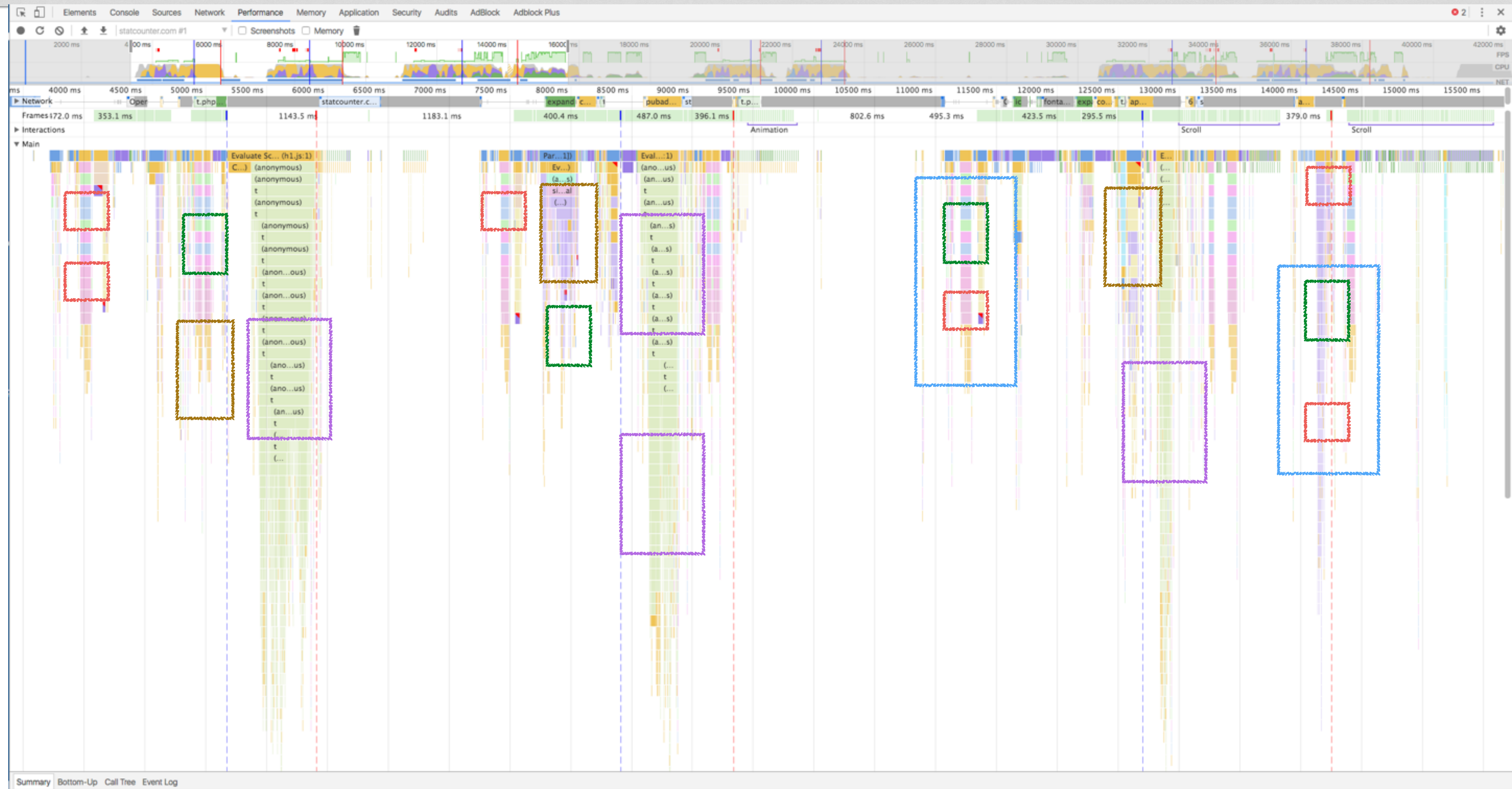


Model

Collect

[Ko et al. '06], [Murphy et al. '95]

Execution Traces: Complex and Overwhelming



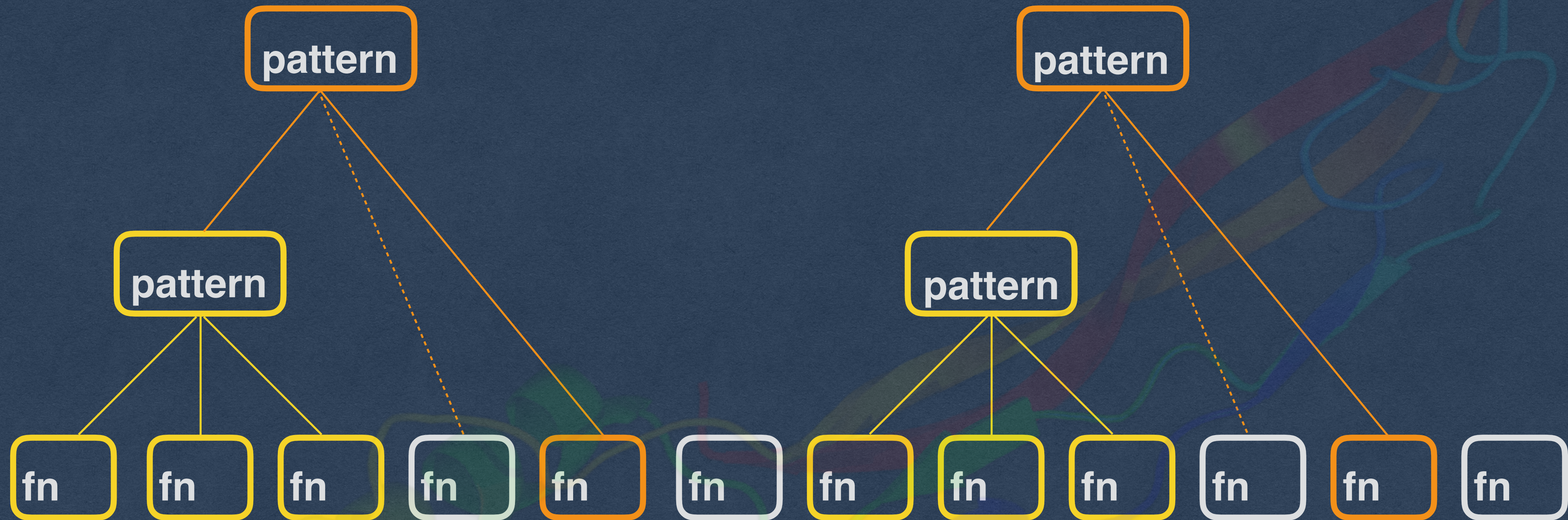
recurring patterns?



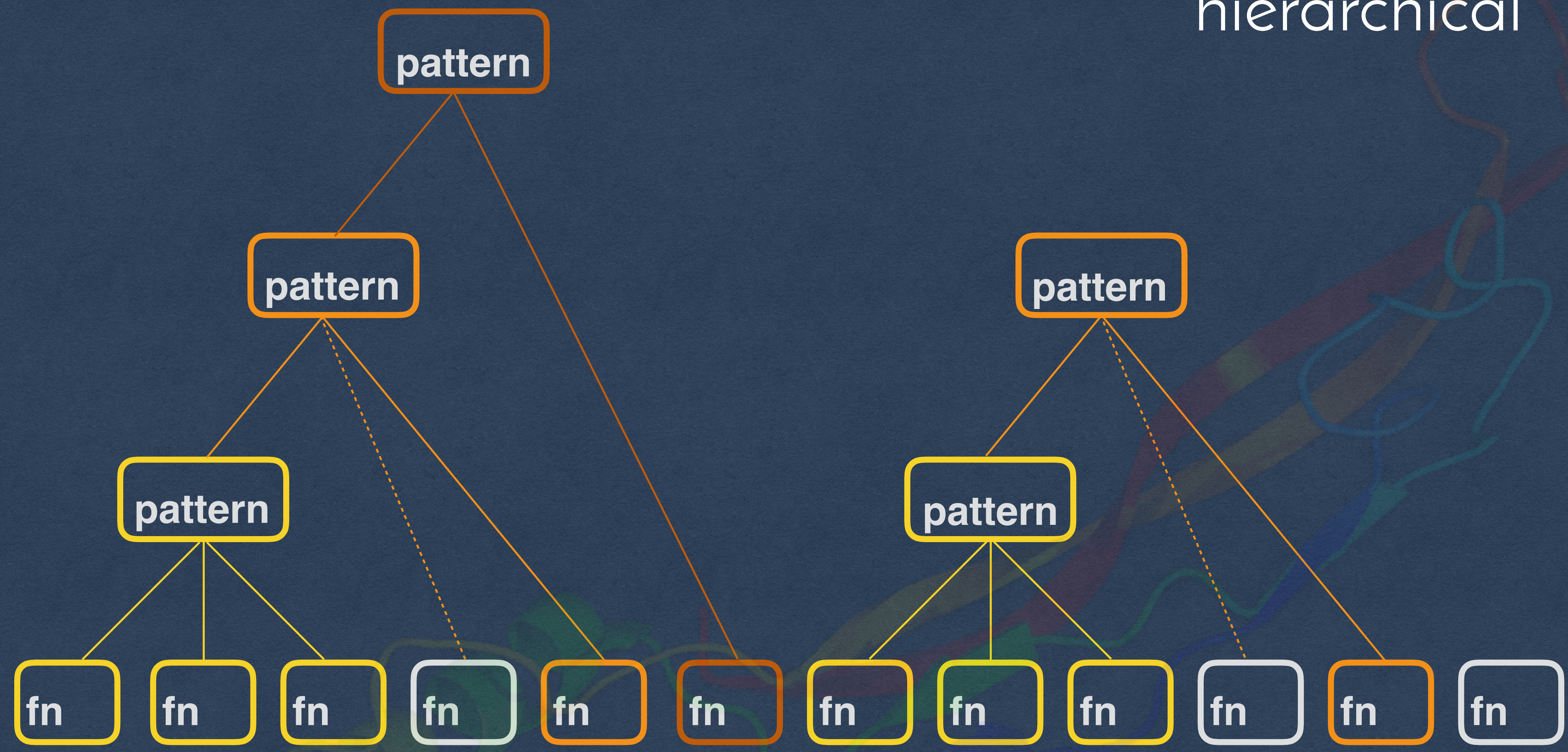
previously unknown



tolerant of small changes



hierarchical

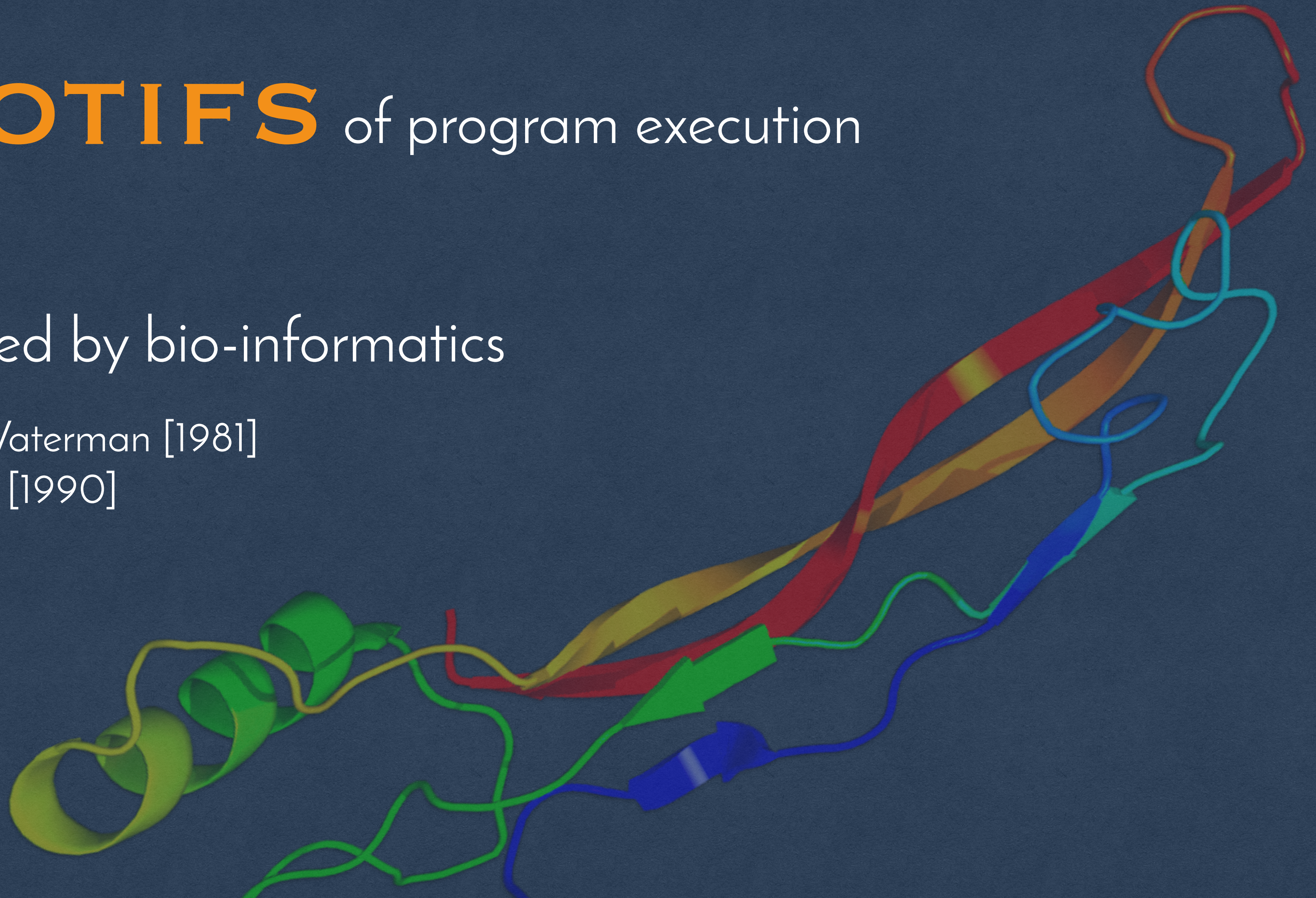


MOTIFS of program execution

Inspired by bio-informatics

Smith-Waterman [1981]

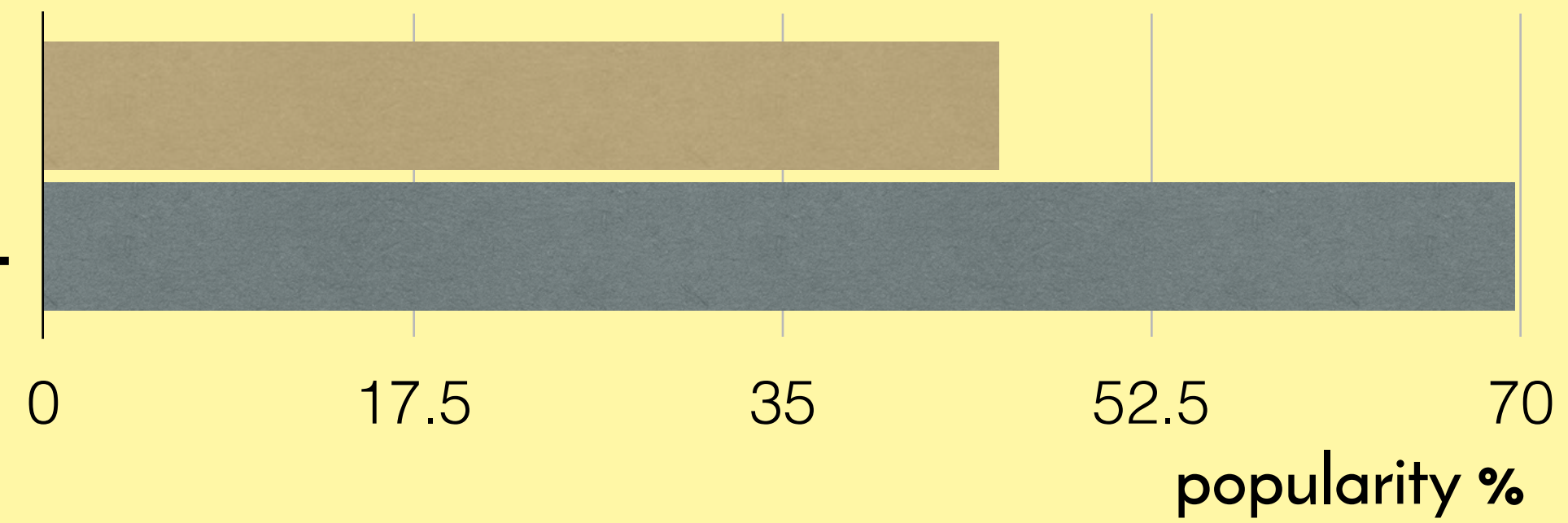
BLAST [1990]



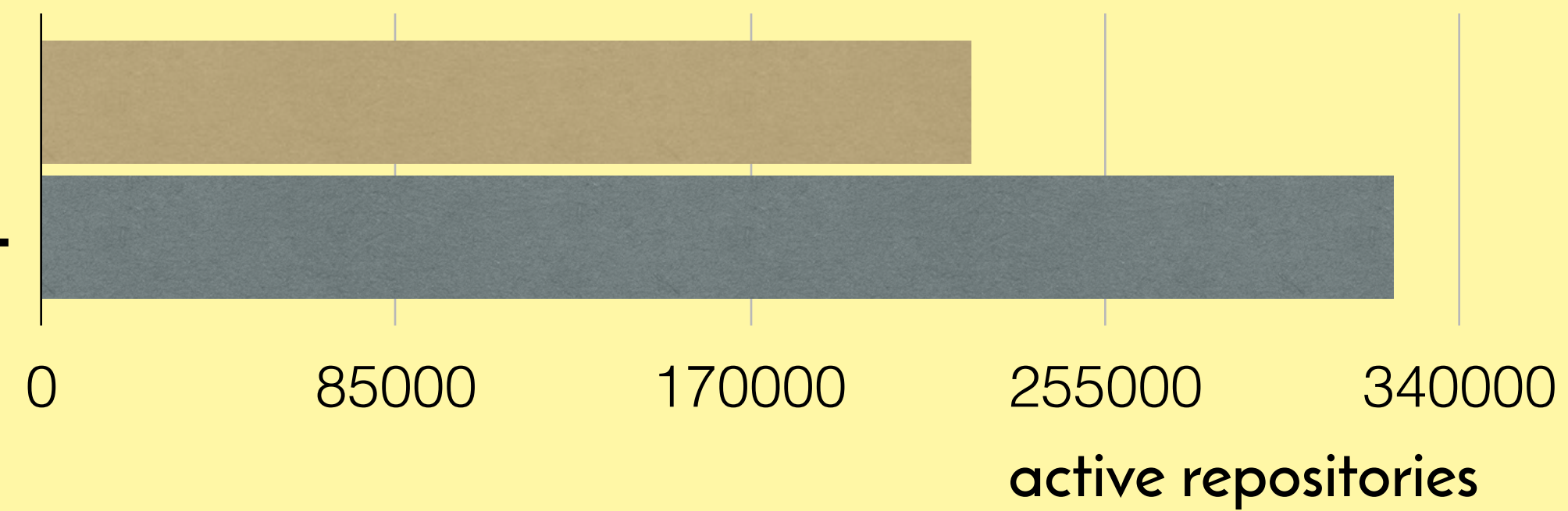
Most Popular Programming Language



Java
JavaScript



Java
JavaScript



JS

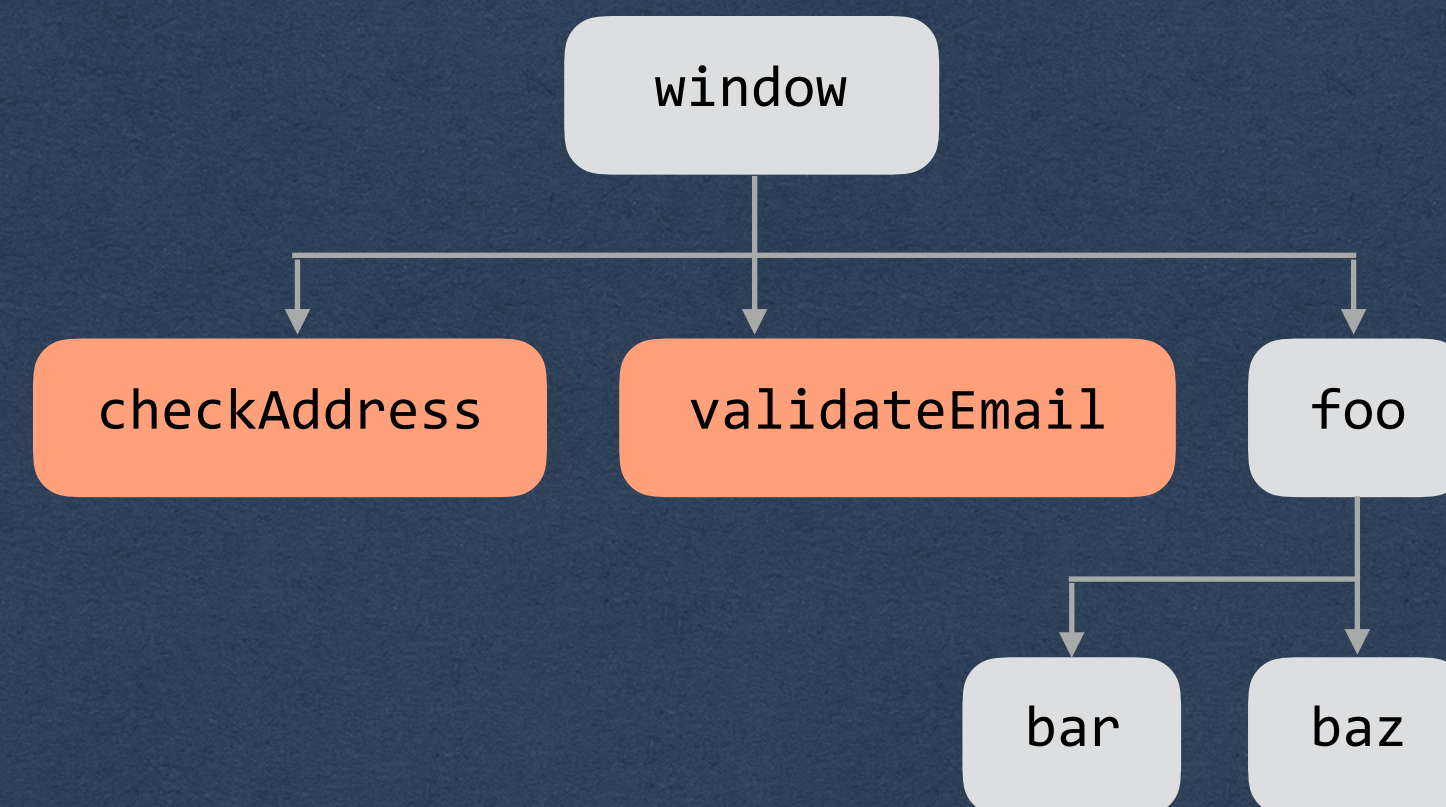
Email:

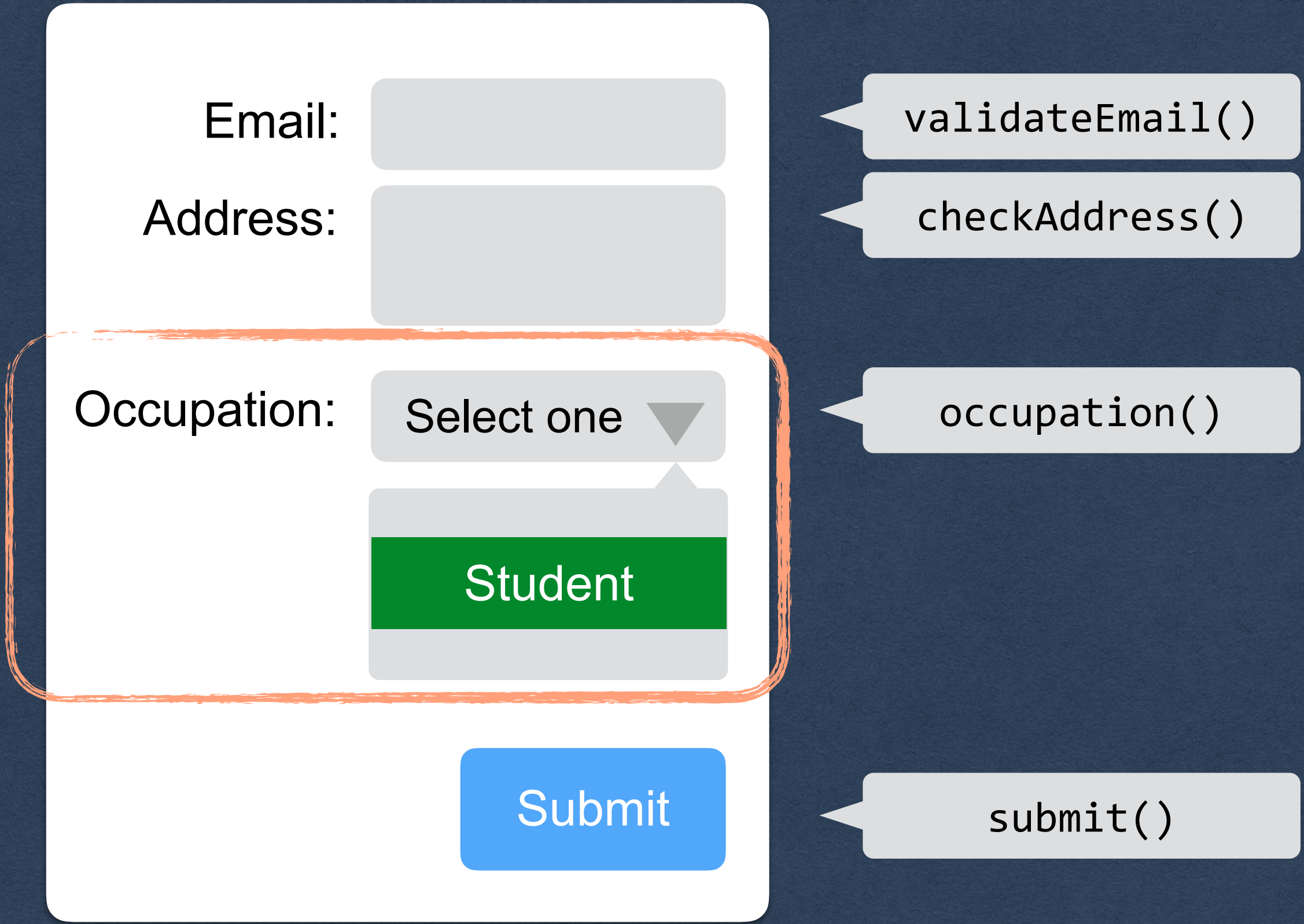
Address:

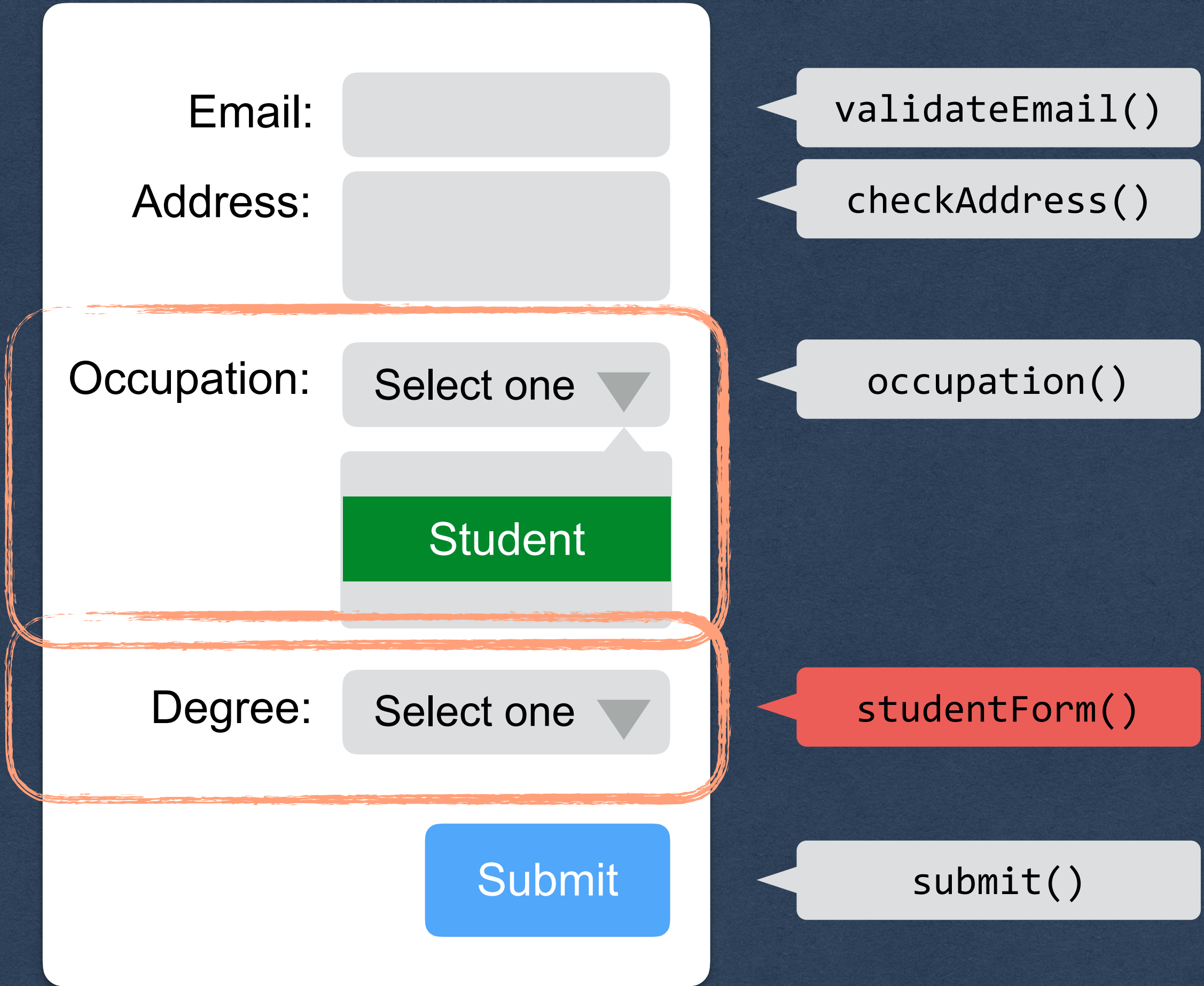
Occupation:

```
function validateEmail () {  
  // do stuff  
}
```

```
function checkAddress () {  
  // do more stuff  
}
```







Email:

Address:

Occupation:

Degree:

`validateEmail()`

`checkAddress()`

`occupation()`

`studentForm()`

`submit()`

Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Email:	<input type="text"/>	validateEmail()
Address:	<input type="text"/>	checkAddress()
Occupation:	<input type="text"/>	occupation()
	<input type="text" value="Student"/>	studentForm()
Degree:	<input type="text"/>	submit()
	<input type="button" value="Submit"/>	

Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Query

```
baz()  
foo()  
validateEmail()  
checkAddress()  
occupation()  
studentForm()  
submit()  
change()  
transition()
```

Email:

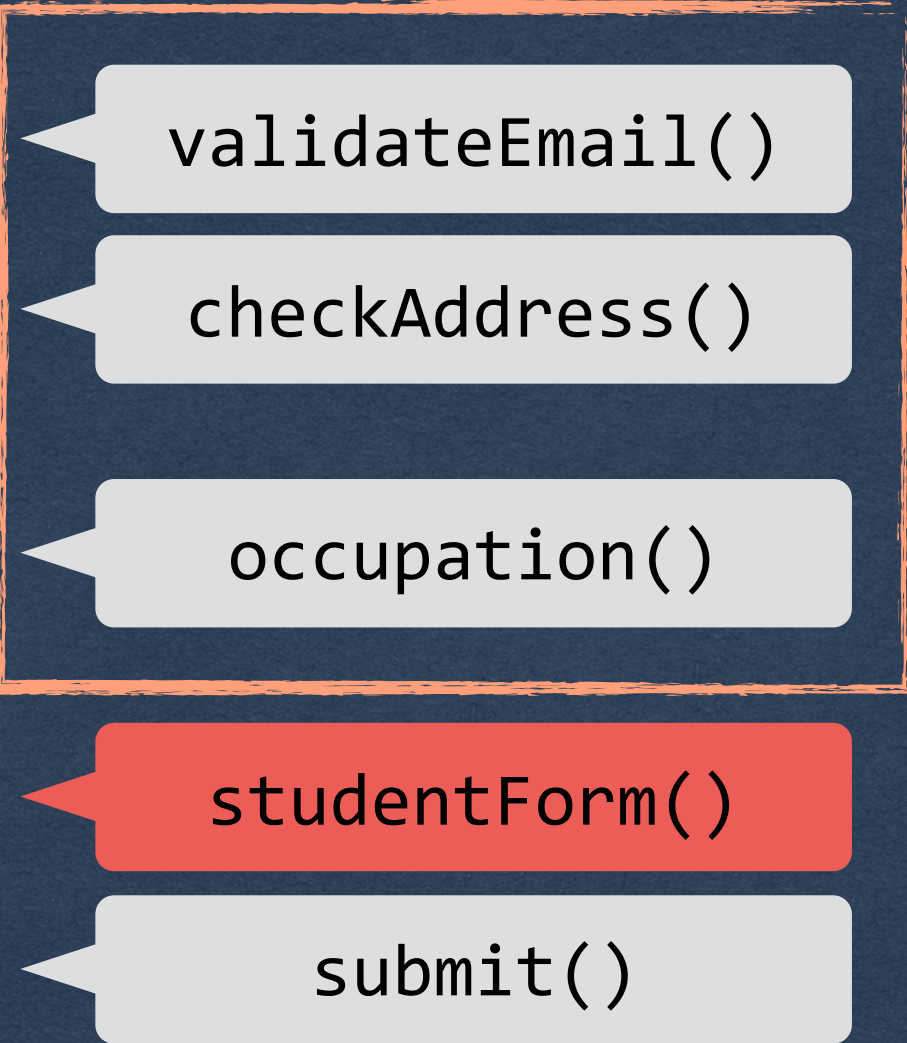
Address:

Occupation:

Student

Degree:

Submit



Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Query

```
baz()  
foo()  
validateEmail()  
checkAddress()  
occupation()  
studentForm()  
submit()  
change()  
transition()
```

Email:	<input type="text"/>	validateEmail()
Address:	<input type="text"/>	checkAddress()
Occupation:	<input type="text"/>	occupation()
	<input type="text" value="Student"/>	studentForm()
Degree:	<input type="text"/>	submit()
	<input type="button" value="Submit"/>	

Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Query

```
baz()  
foo()  
validateEmail()  
checkAddress()  
occupation()  
studentForm()  
submit()  
change()  
transition()
```

	validateEmail	checkAddress	occupation	studentForm	submit
validateEmail	0	0	0	0	0
checkAddress	0	2	0	0	0
occupation	0	0	4	2	0
submit	0	0	2	6	4

Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Query

```
baz()  
foo()  
validateEmail()  
checkAddress()  
occupation()  
studentForm()  
submit()  
change()  
transition()
```

	validateEmail	checkAddress	occupation	studentForm	submit
validateEmail	0	0	0	0	0
checkAddress	0	0	4	2	0
occupation	0	0	2	6	4
submit	0	0	0	4	4

Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Query

```
baz()  
foo()  
validateEmail()  
checkAddress()  
occupation()  
studentForm()  
submit()  
change()  
transition()
```

	validateEmail	checkAddress	occupation	studentForm	submit
validateEmail	0	2	0	0	0
checkAddress	0	0	4	2	0
occupation	0	0	2	6	4
submit	0	0	0	4	6

Database

```

bar()
change()
transition()
grouped()
stacked()
index.html:11()
showPopup()
foo()
baz()

```

```

foo()
bar()
foo()
bar()
baz()
validateEmail()
checkAddress()
occupation()
submit()

```

Query

```

baz()
foo()
validateEmail()
checkAddress()
occupation()
studentForm()
submit()
change()
transition()

```

submit
studentForm
occupation
checkAddress
validateEmail

validateEmail
checkAddress
occupation
submit

MOTIF

validateEmail
checkAddress
occupation
ABSTRACT
submit

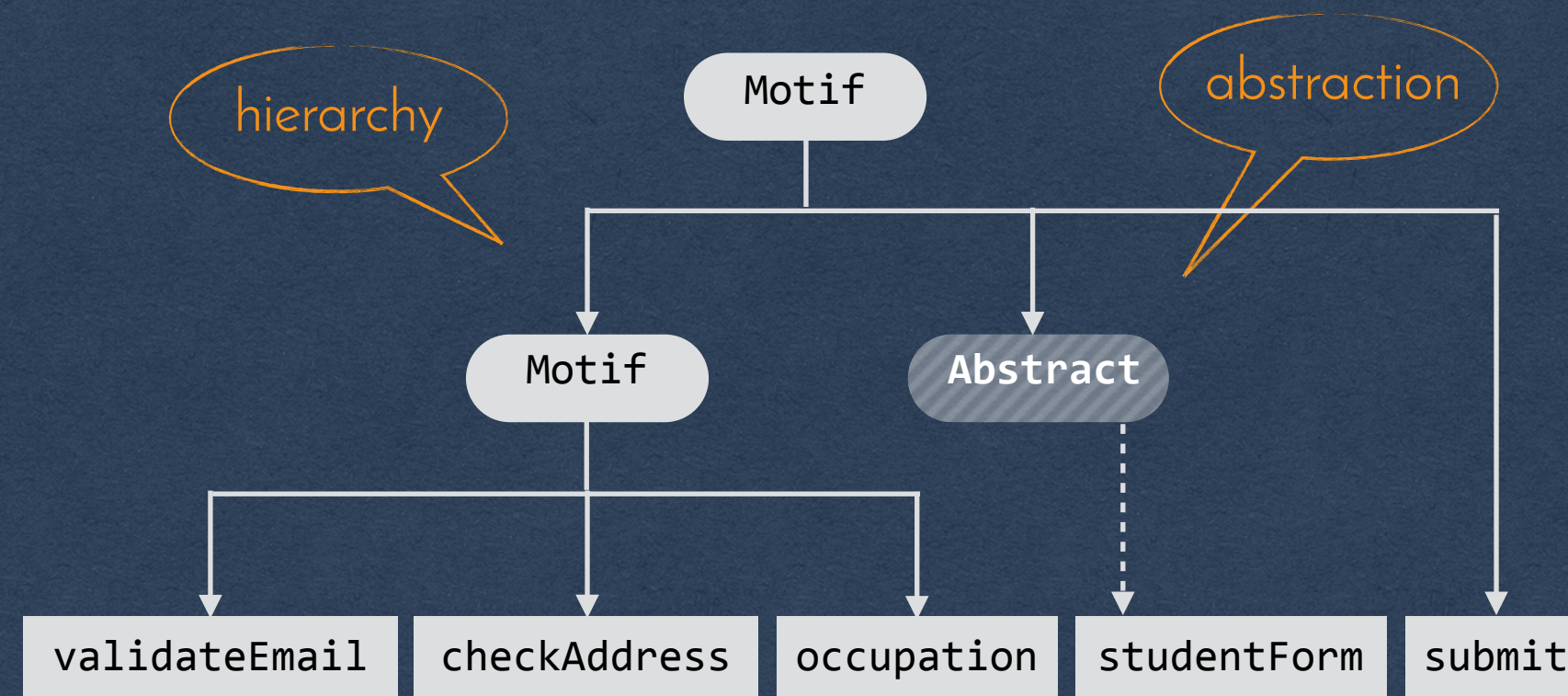
Database

```
bar()  
change()  
transition()  
grouped()  
stacked()  
index.html:11()  
showPopup()  
foo()  
baz()
```

```
foo()  
bar()  
foo()  
bar()  
baz()  
validateEmail()  
checkAddress()  
occupation()  
submit()
```

Query

```
baz()  
foo()  
validateEmail()  
checkAddress()  
occupation()  
studentForm()  
submit()  
change()  
transition()
```



submit
studentForm
occupation
checkAddress
validateEmail

validateEmail
checkAddress
occupation
submit

MOTIF
validateEmail
checkAddress
occupation
ABSTRACT
submit

SABALAN

A

- validateEmail
- checkAddress
- occupation
- studentForm
- submit
- refresh
- updateRegistry
- informServer
- successMessage
- getNewList
- animateChange
- setDefault

B

	motif1	motif2	motif3	motif4	motif5
validateEmail	■	■			
checkAddress	■	■			
occupation	■	■			
studentForm	■				
submit	■				
refresh					
updateRegistry					
informServer					
successMessage					
getNewList					
animateChange					
setDefault					

C

DB Motifs

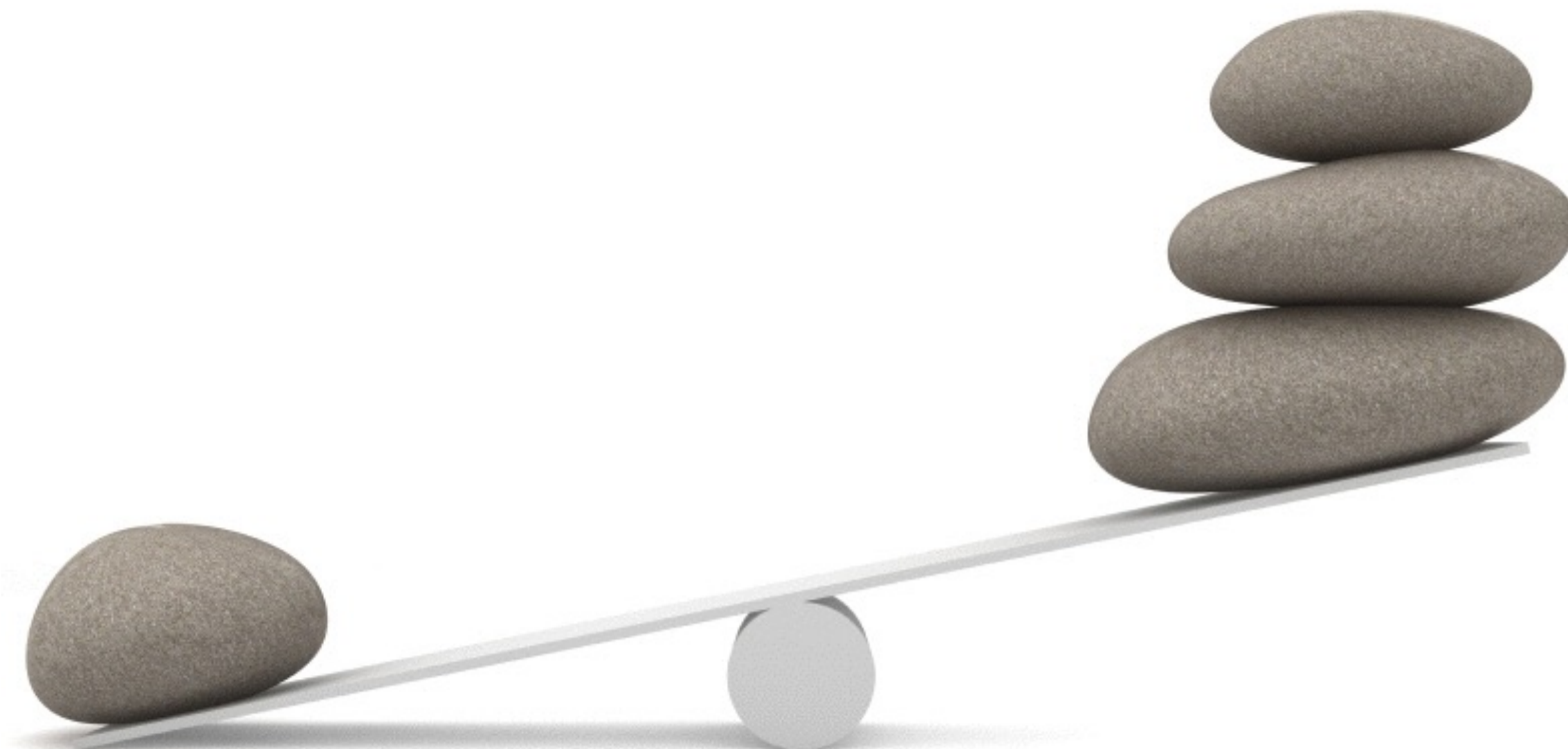
E index.html

```
$("#email").addEventListener("change", validateEmail, fa
$(".addr").click(checkAddress);
$(".dropdown-content").addEventListener("change", occupa
function occupation () {
    eval(this.value + "JobForm()");
}
function avademicJobFo
var label = document
var degreeType = doc
var option1 = docuem
option1.value = "Und
}
function validateEmail
// do stuff
}
function checkAddress
```

D

	motif1		motif2
validateEmail	■	validateEmail	■
checkAddress	■	checkAddress	■
occupation	■	occupation	■
studentForm	■		
submit	■		

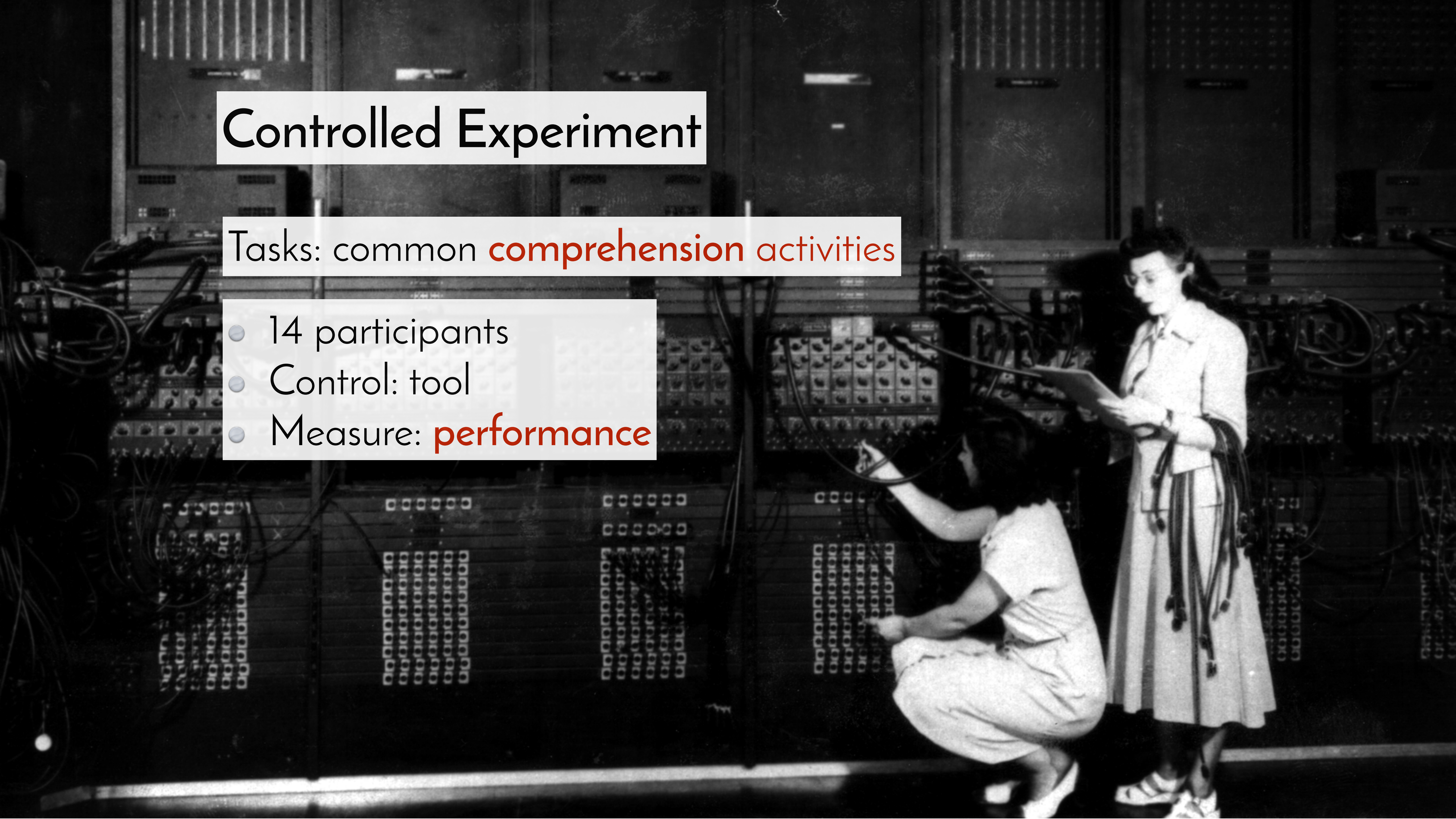
Does using **SABALAN** improve
performance of developers?



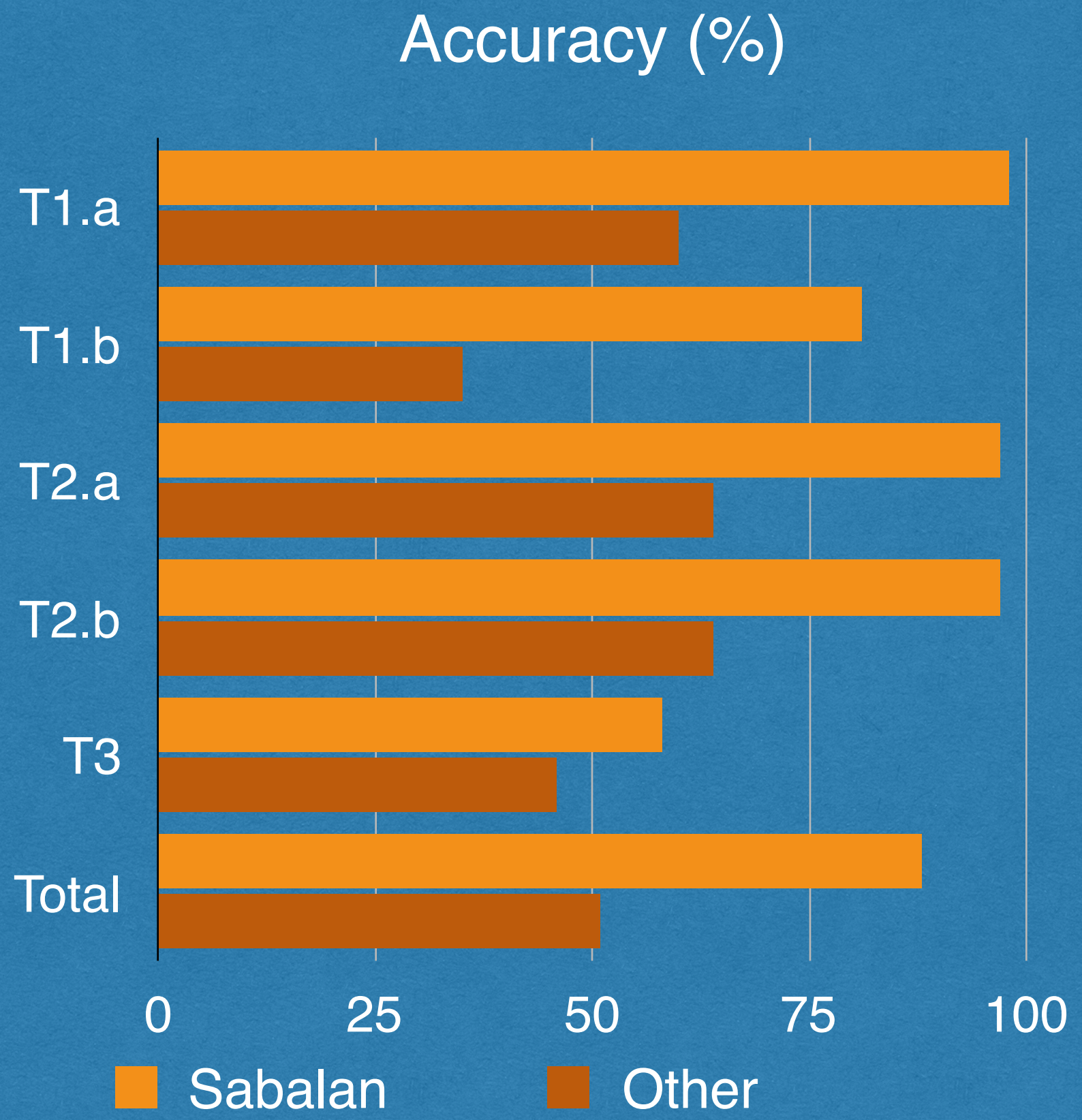
Controlled Experiment

Tasks: common **comprehension** activities

- 14 participants
- Control: tool
- Measure: **performance**

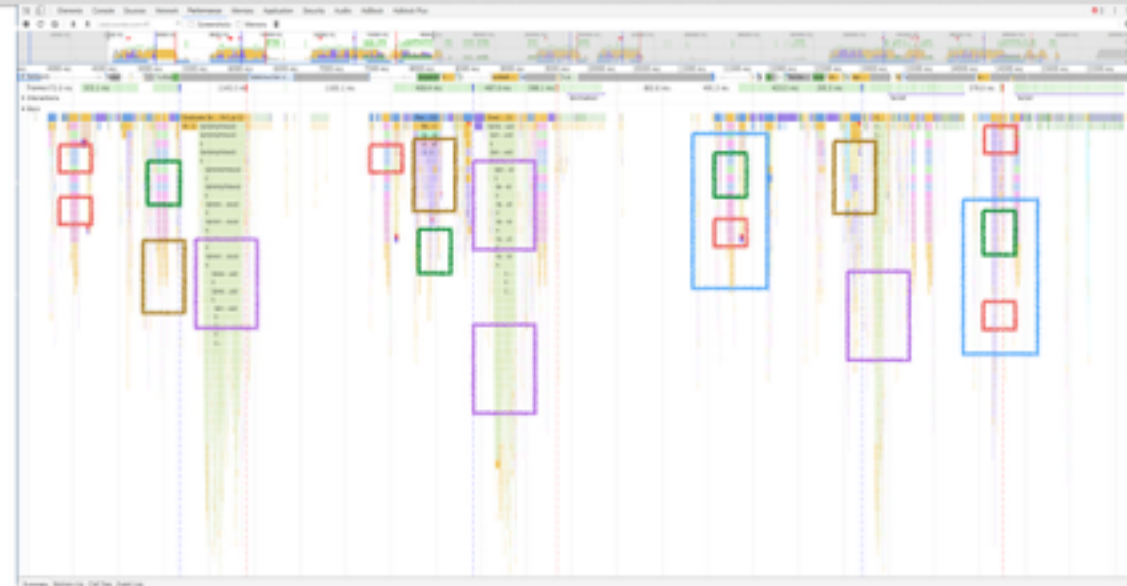


Fast is fine, but **accuracy** is everything.



54 pp more accuracy

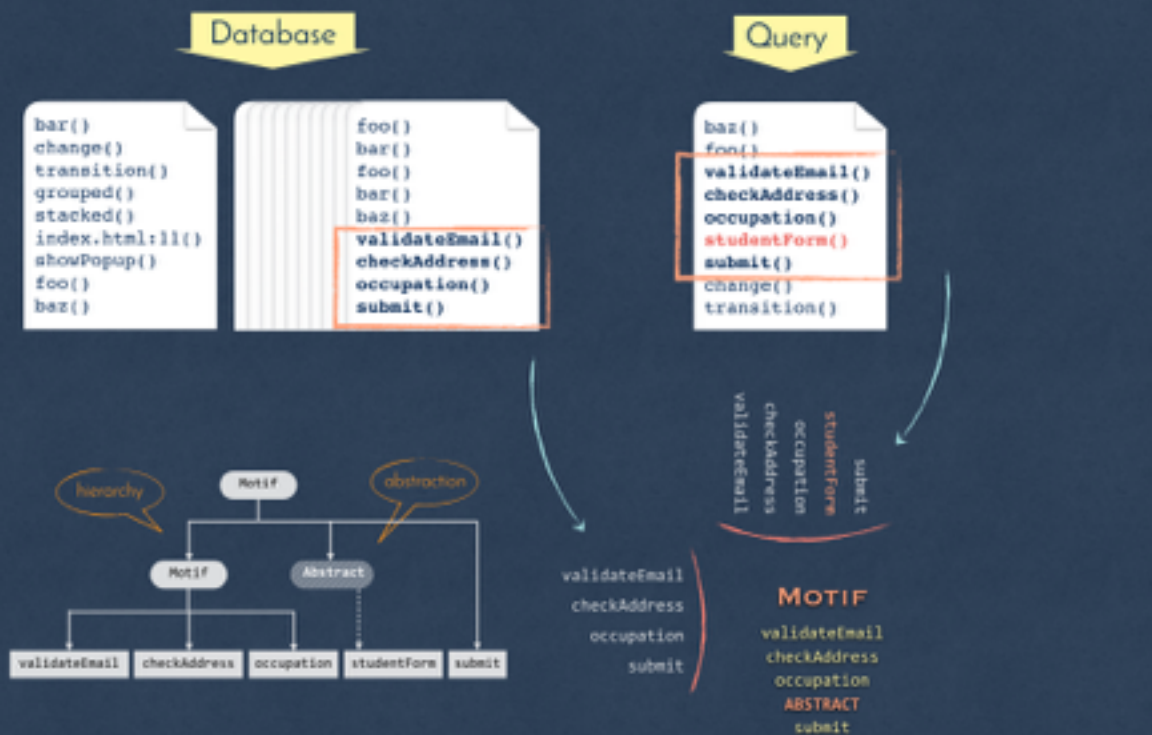
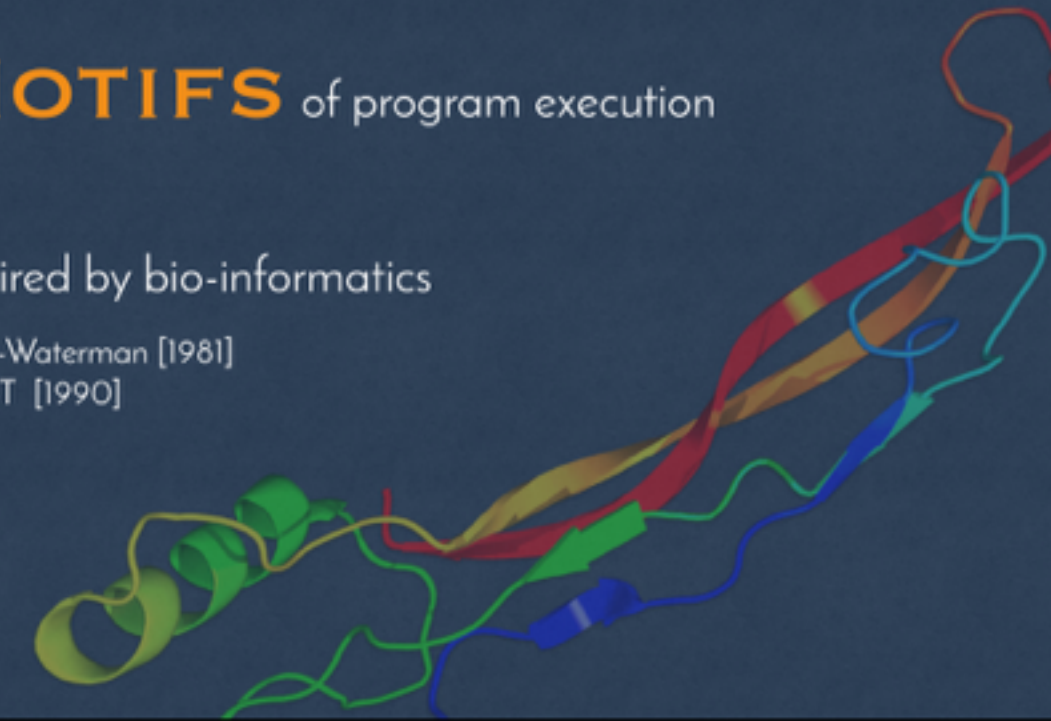
Execution Traces: Complex and Overwhelming



MOTIFS of program execution

Inspired by bio-informatics

Smith-Waterman [1981]
BLAST [1990]



Fast is fine, but **accuracy** is everything.

