# Problem Comprehension & Metaprogramming

How do we program, and how can computers help?

by Kenny Harvey

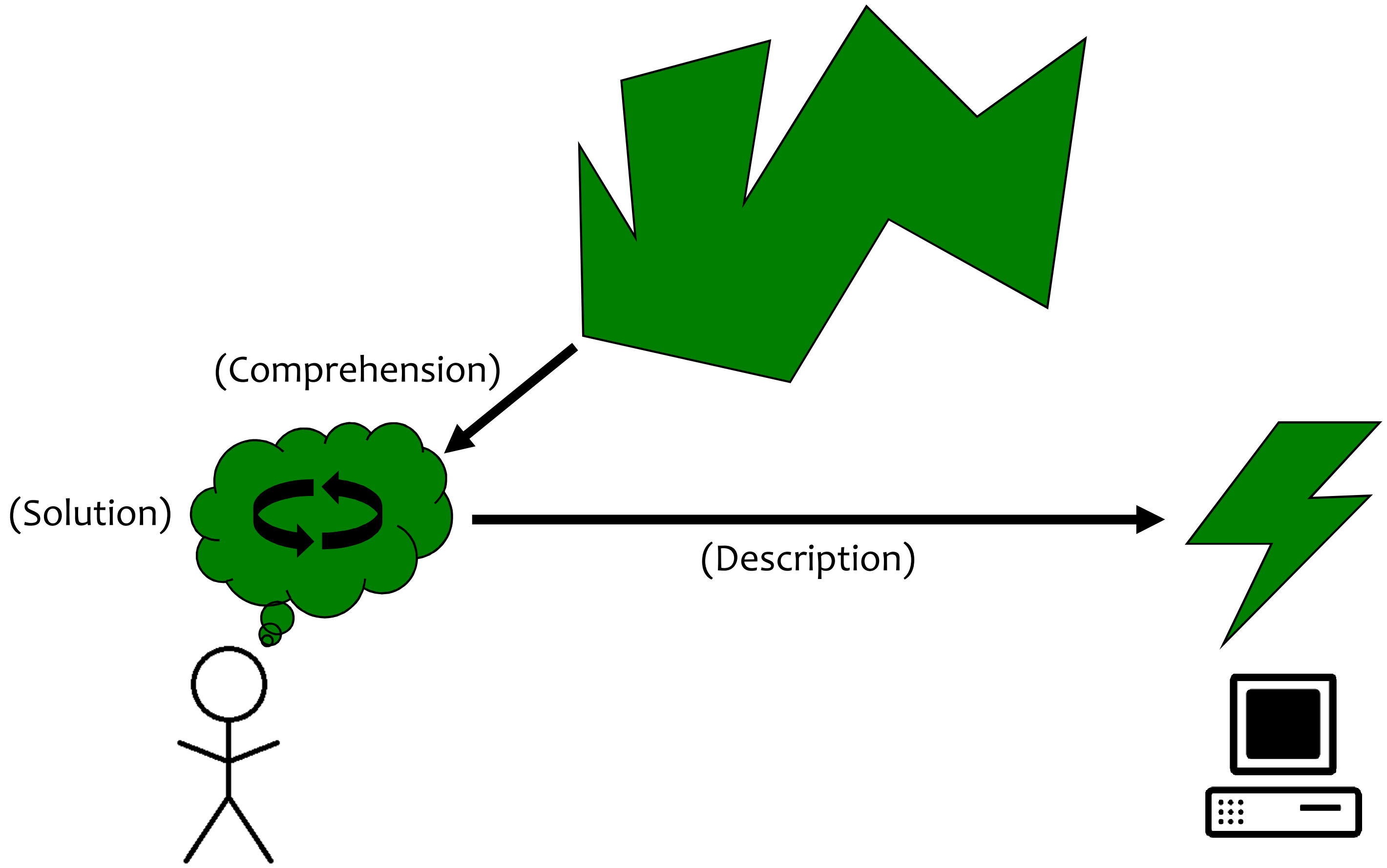
## Agenda

- What is programming (in a general sense)?
- How do we comprehend problems?
- Metaprogramming
- Topics for future work

## What is programming (in a general sense)?

- ~~“Writing computer programs”~~ – circular/loose
- “Describing a solution to a computer, so it may solve a problem”
- Requires translation from the logical definitions of problems, through the abstract concepts in which humans think, to the electrical signals computers understand

## What is programming (in a general sense)?



Describing a solution to a computer, so it may solve a problem

## What is programming (in a general sense)?

"Theory of Programming Behavior" – Brooks (1977) [1]

| | | |
|---|---|---|
| • Understanding | (Comprehension) | ??? |
| • Method-finding | (Solution) | Metaprog. |
| • Coding | (Description) | Most PL |

## Agenda

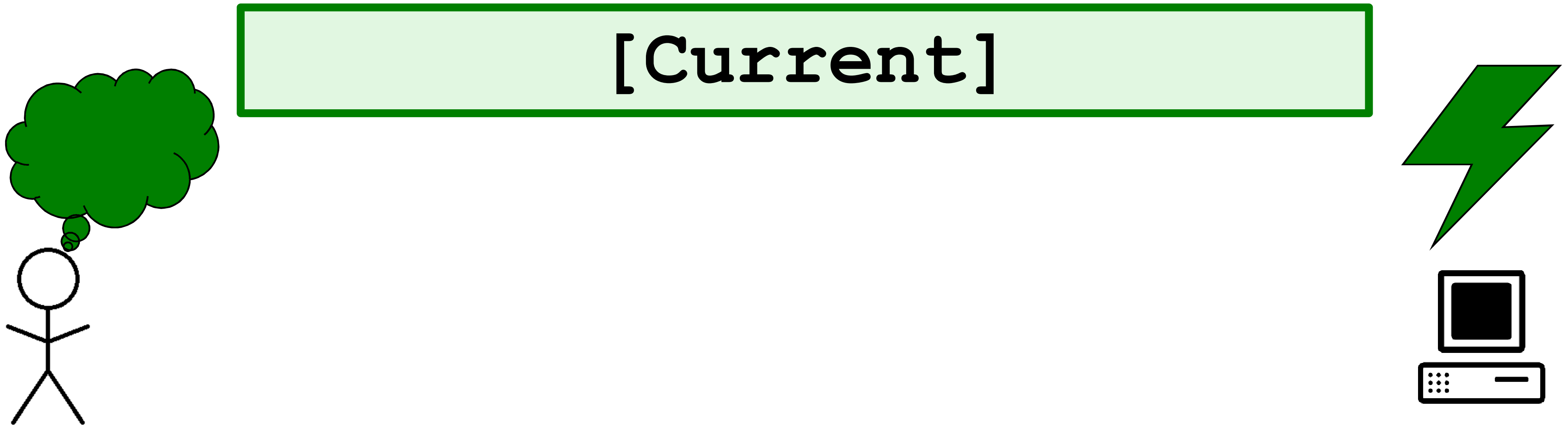
- What is programming (in a general sense)?
- How do we comprehend problems?
- Metaprogramming
- Topics for future work

## How do we comprehend problems?

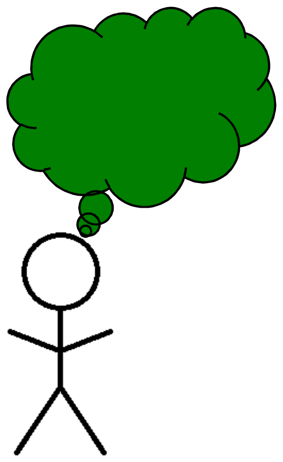
No formal definitions, but empirically:

- Pattern-recognition – Brooks (1977) [1]
  - “Genius is having seen it before” – Prof. Aho
- Verbal/Numerical WM – Siegmund (2014) [2]
- Levels of abstraction

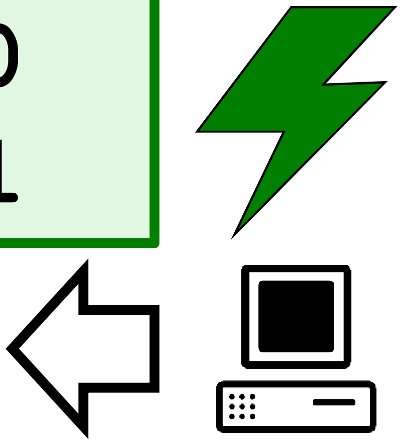
## How do we comprehend problems?



## How do we comprehend problems?

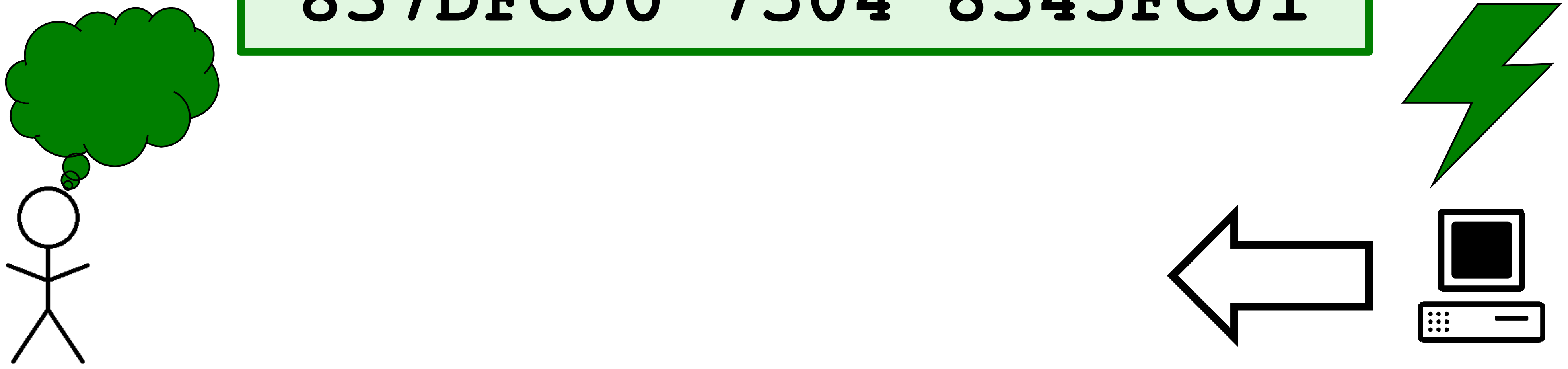


```
10000011011111011111
11000000000001110101
00000100100000110100
01011111110000000001
```

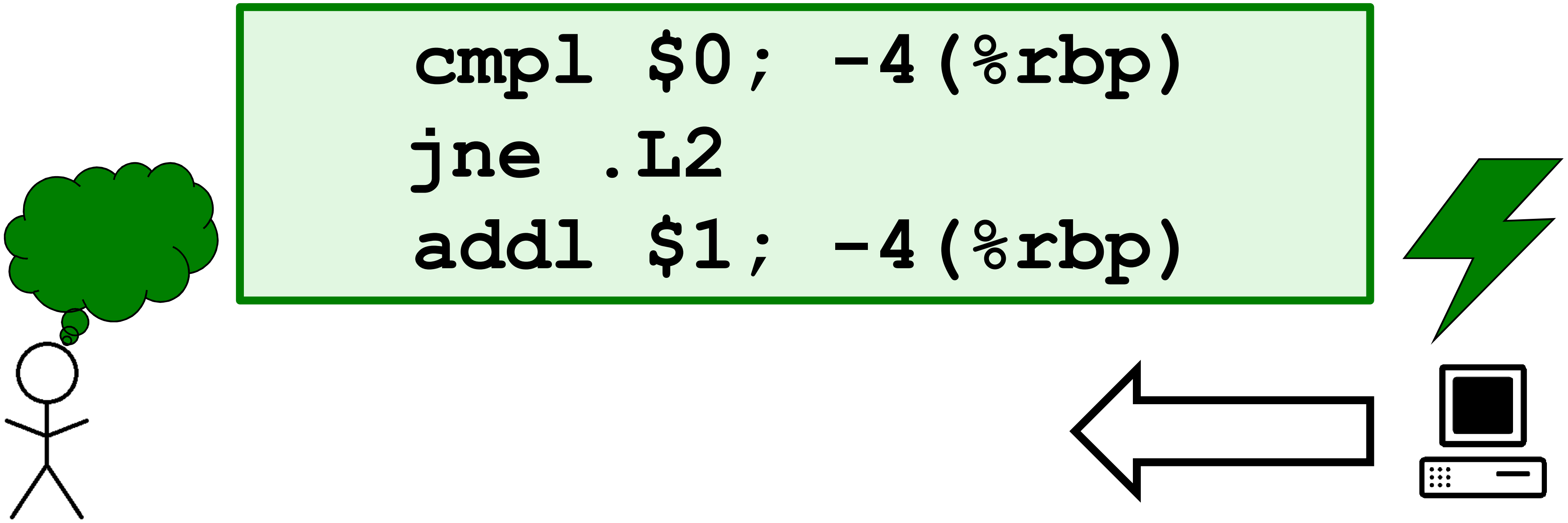


## How do we comprehend problems?

```
837DFC00 7504 8345FC01
```

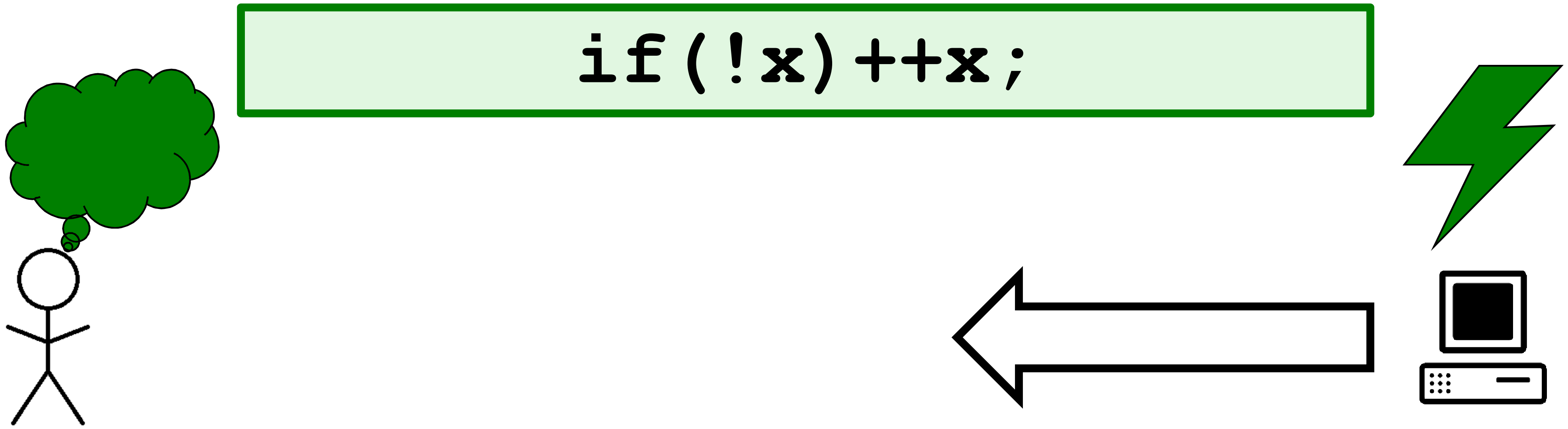


## How do we comprehend problems?



```
cmpl $0; -4(%rbp)
jne .L2
addl $1; -4(%rbp)
```

## How do we comprehend problems?

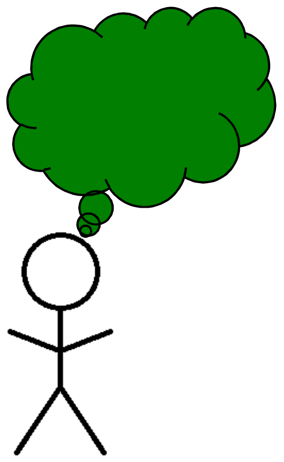


```
if(!x)++x;
```

## How do we comprehend problems?

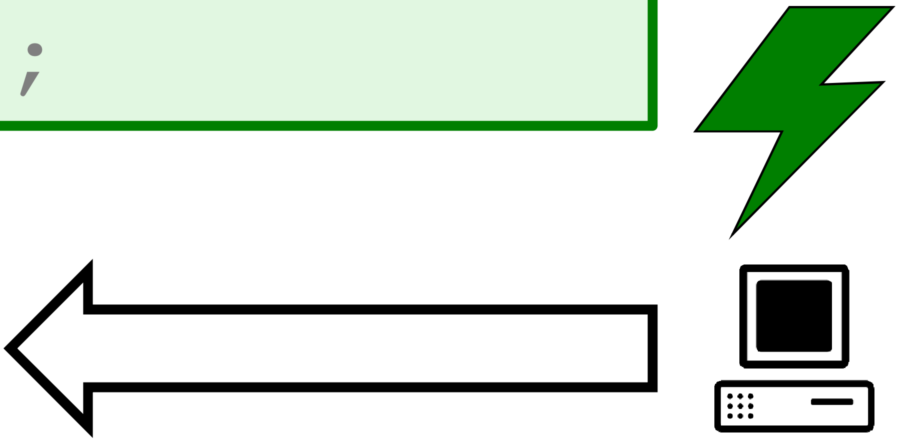
Secondary Notation – Schrepfer (2009) [3]

```
if( !x )
++x;
```



How can we bridge this gap?

Hint: We're programmers!



## Agenda

- What is programming (in a general sense)?
- How do we comprehend problems?
- Metaprogramming
- Topics for future work

## Metaprogramming

- Even more circular definitions
- "Describing a solution to a computer, so it may solve describing solutions to computers, so they may solve problems"
- C macros, JITs, Lex/YACC
- Not just Comprehend → Solve → Describe…

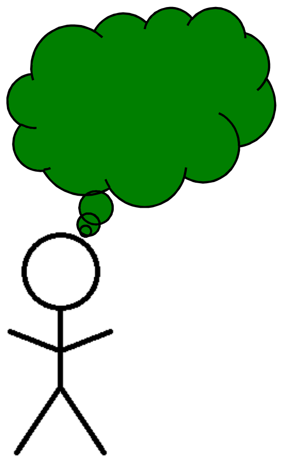
## Metaprogramming

- Task is to comprehend, solve, and describe:
  - Comprehension
  - Solution
  - Description

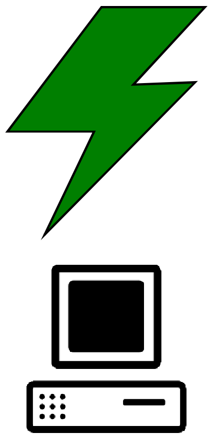
| | Comprehension | Solution | Description |
|---|---|---|---|
| Comprehend | ? | Abstraction | Linguistic WM |
| Solve | ? | Cog. Psych. | Lang. Theory |
| Describe | ML? | Meta Lang. | Grammars |

## Metaprogramming

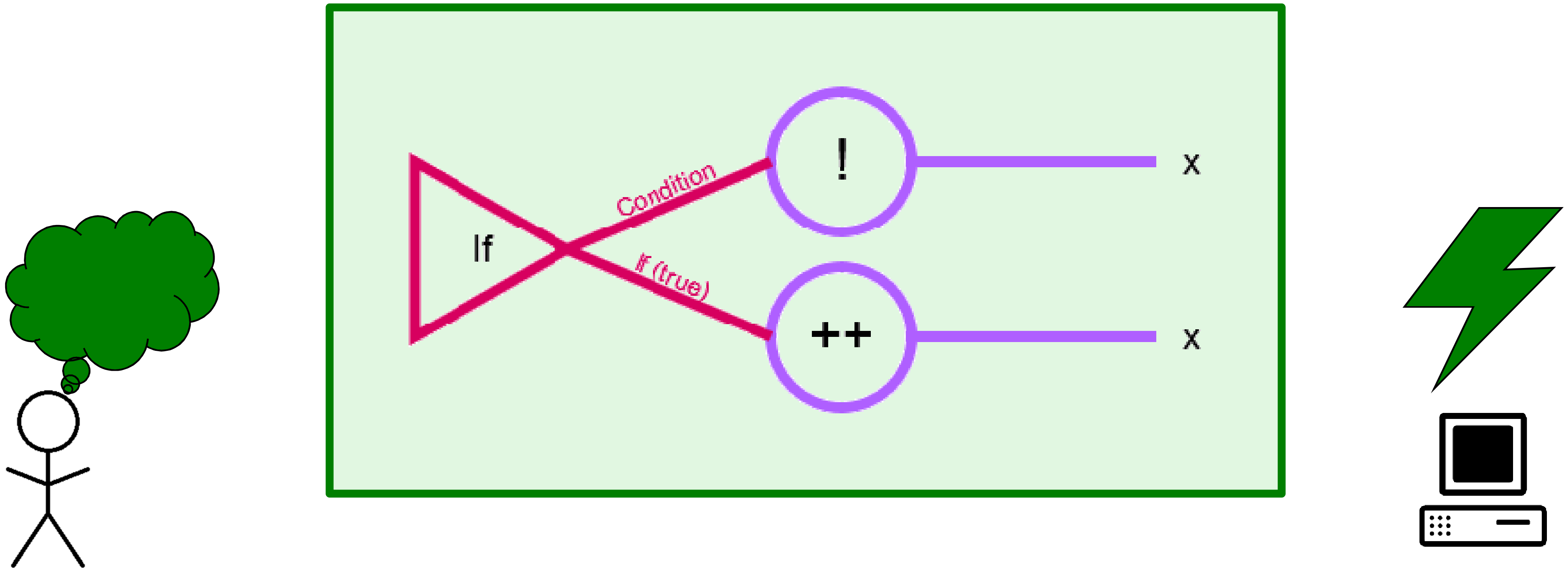
- Let’s revisit our abstraction
  - Does it suggest comprehension?
  - Can we improve?



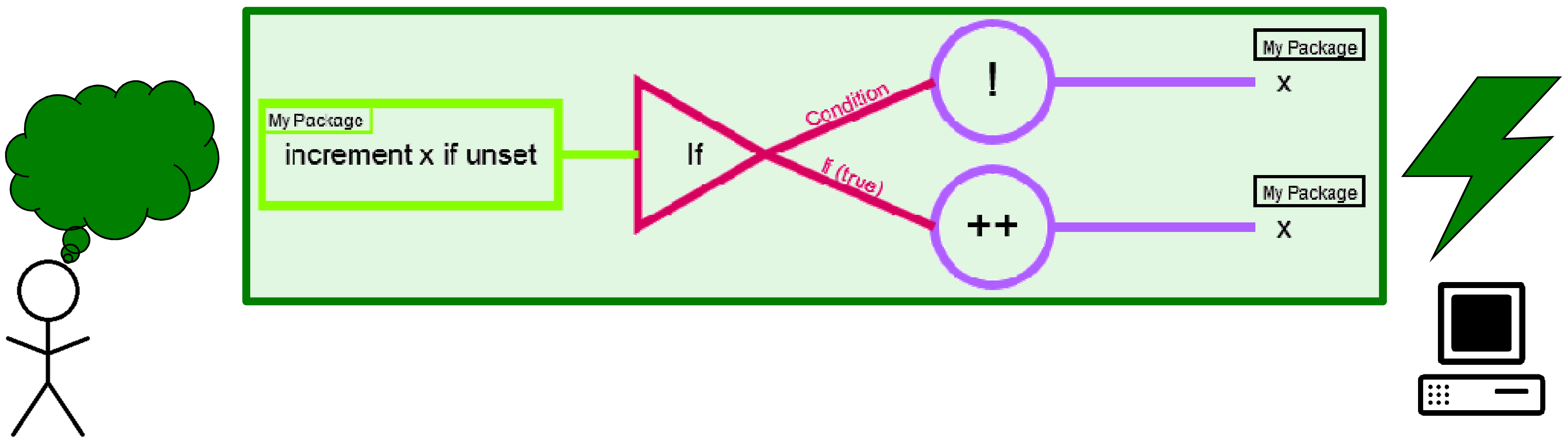
```
if( !x )
++x;
```



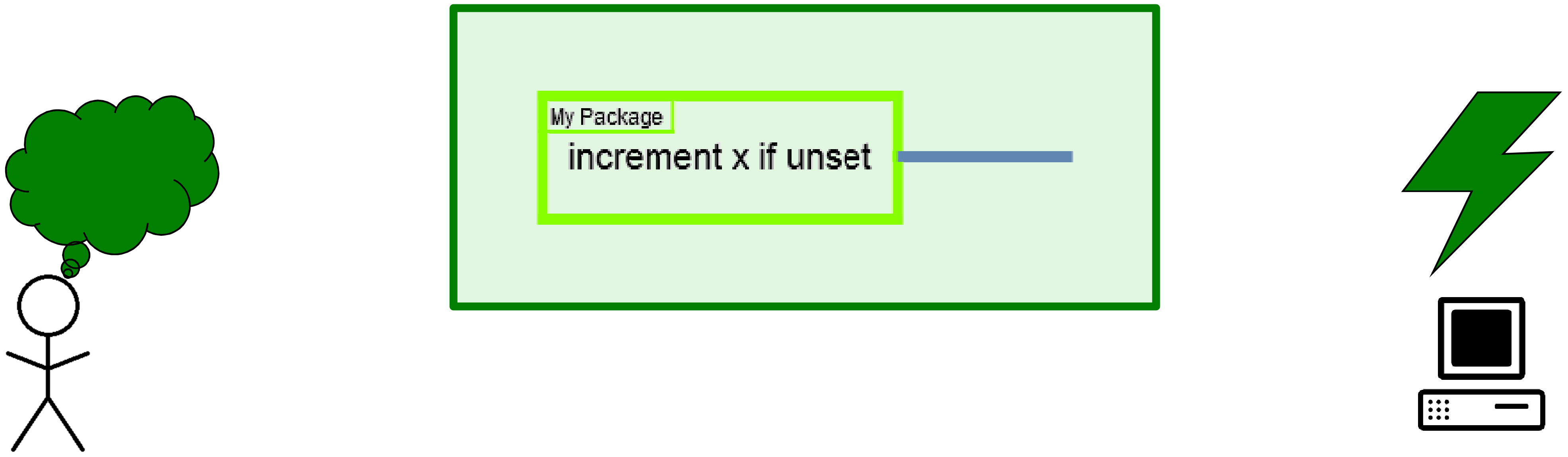
## Metaprogramming



# Metaprogramming



## Metaprogramming



## Demo

http://harveyserv.ath.cx/adhoc_demo/ [5]

## Agenda

- What is programming (in a general sense)?
- How do we comprehend problems?
- Metaprogramming
- Topics for future work

## Topics for Future Work

- Semester project: logic database, autocomplete
- More software engineering tools
  - Tags for “Done”, “Tested”, could produce “Taint” system
  - Auto-generate tests
- Easier searching / browsing
- Background Cog. Psych. literature for UI
- Mobile coding!

## References

1. Brooks R. Towards a theory of the cognitive processes in computer programming. Int J of Man-Machine Studies, 1977. **51**(2): p197-211.
2. Siegmund J K, et al. Understanding Understanding Source Code with Functional Magnetic Resonance Imaging. Comm. of the ACM, 2014.
3. Schrepfer M W, et al. The Impact of Secondary Notation on Process Model Understanding. Lecture Notes in Biz Info Proc., 2009. **39**: p161-175.
4. Myers B A. Taxonomies of Visual Programming and Program Visualization. J of Vis. Lang. and Comp., 1990. **1**(1): p. 97-123.
5. http://harveyserv.ath.cx/adhoc_demo/