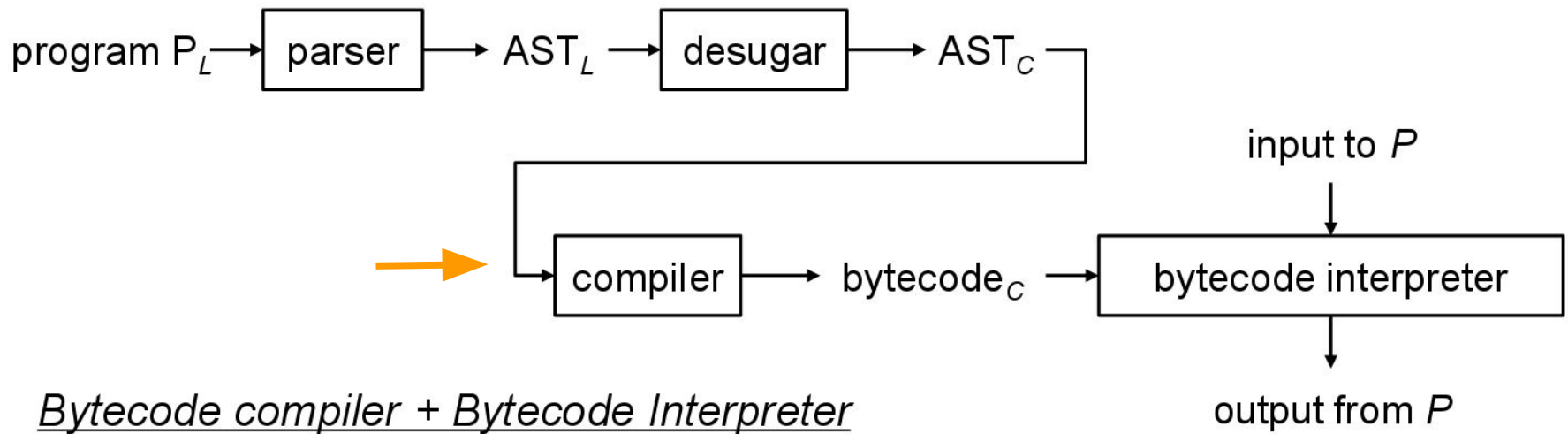
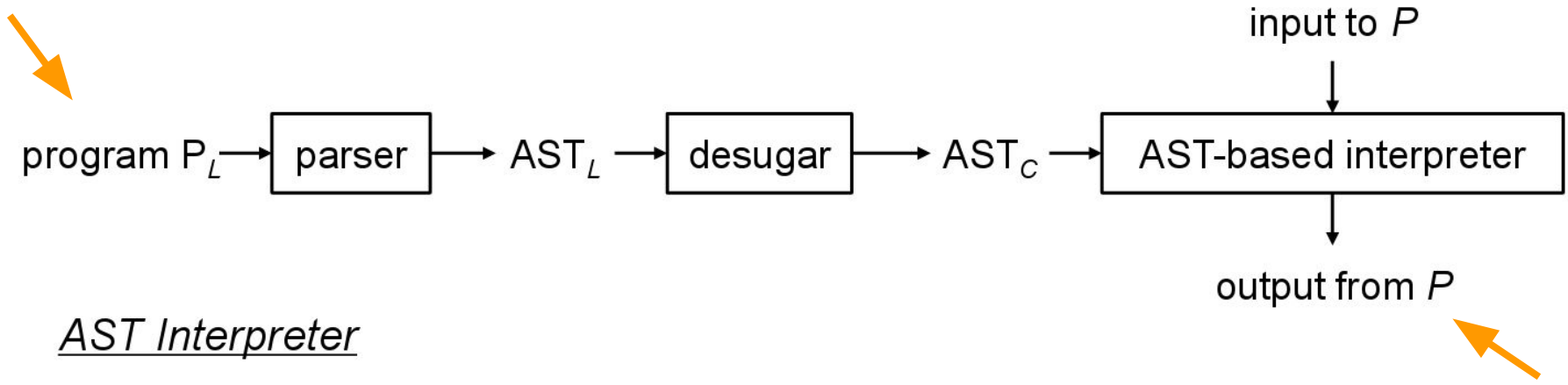


Byte Code Interpretation

Section 3

Bigger picture.



Representation 1: The original code

```
def foo (y) {  
    def x = (1 + 2) - 3  
  
    x + y  
  
}  
  
print fun(0)
```

Representation 2: AST

```
[
  {
    "type": "def",
    "name": {
      "type": "id",
      "name": "foo"
    },
    "value": {
      "body": [ ... ],
      "type": "lambda",
      "arguments": [ ... ]
    }
  },
  {
    "type": "print",
    "value": {
      "function": { ... },
      "type": "call",
      "arguments": [ ... ]
    }
  }
]
```

HW2: Execute the AST!!

But...

- I want co-routines :(
- I want speed :'(

Representation: Bytecode

```
[
  {
    "type": "lambda",
    "arguments": [
      {
        "type": "id",
        "name": "y"
      }
    ],
    "target": "#btc-reg-2",
    "body": [
      {
        "type": "int-lit",
        "value": 1,
        "target": "#btc-reg-6"
      },
      {
        "type": "int-lit",
        "value": 2,
        "target": "#btc-reg-7"
      },
      {
        "type": "+",
        "operand1": "#btc-reg-6",
        "operand2": "#btc-reg-7",
        "target": "#btc-reg-5"
      },
      ...
    ]
  }
]
```

PA2: Interpreting bytecode

```
def eval(bytecode_array) { // program to be executed
    pc = 0
    while (true) {
        curr_instr = bytecode_array[pc]
        switch (curr_instr.op) {
            case '+':
                ... // do + on args (as in HW2)
                pc++ // jump to next bc instr ...
            }
        }
    }
}
```


What do we do for function calls?

Stacks of bytecode instruction frames.

What should a frame include?

case 'call':

```
// make stack frame
```

```
// push new frame to stack
```

```
break;
```

Instruction array
Instruction pointer
Environment
Target register

Exercise

Dry run the bytecode in the following example:

<http://tinyurl.com/bytecode-exercise>

Track the values in registers, the frame being executed and program counters as you execute the code. Figure out the proper behavior for each instruction.

If unsure or stuck, ask for help!

Arrowlets

Section 3

Exercise:

<http://tinyurl.com/arrowlets>

Instructions within document.

If stuck, ask for help! =)