

## Homework 3

— How to handle multiple files?

Observation 1: Everything you write in  
a file called `file.ml` is  
implicitly placed in a module  
called `File`

`file.ml`

→

`struct File : File`

`file.mli` ← interface

May or may not  
exist

signature `File`

Observation 2: In class, I frequently

wrote `#use "file.ml"`

wrote `#use file.ml`

Directive to the top-level

Literally like `#include "file.ml"` in C.

Observation 3: If I want to access some member `v` of a module `M`, I have two options:

Option 1  
Fully qualified name  
`M.v`

Option 2  
Open the module  
`open M`  
`v`

Q2 · AST for `Imp` defined in `imp.ml`

<code>aexp</code>	<code>bexp</code>	<code>cmd</code>
<code>Int</code>	<code>Bool</code>	<code>Output</code>
<code>Var</code>	<code>Lt</code>	<code>Asgn</code>
..		

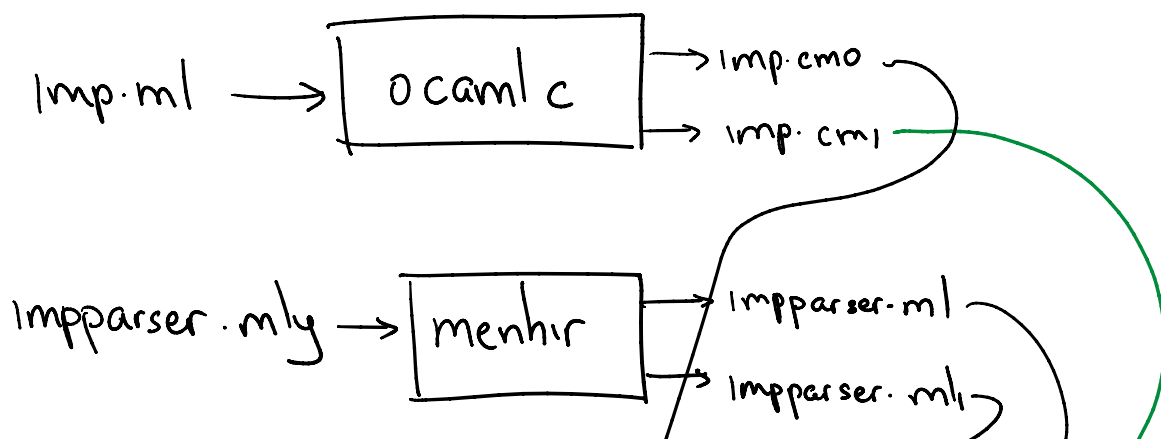
Var	Lt	Asgn
Plus	:	Seq
Minus	:	:

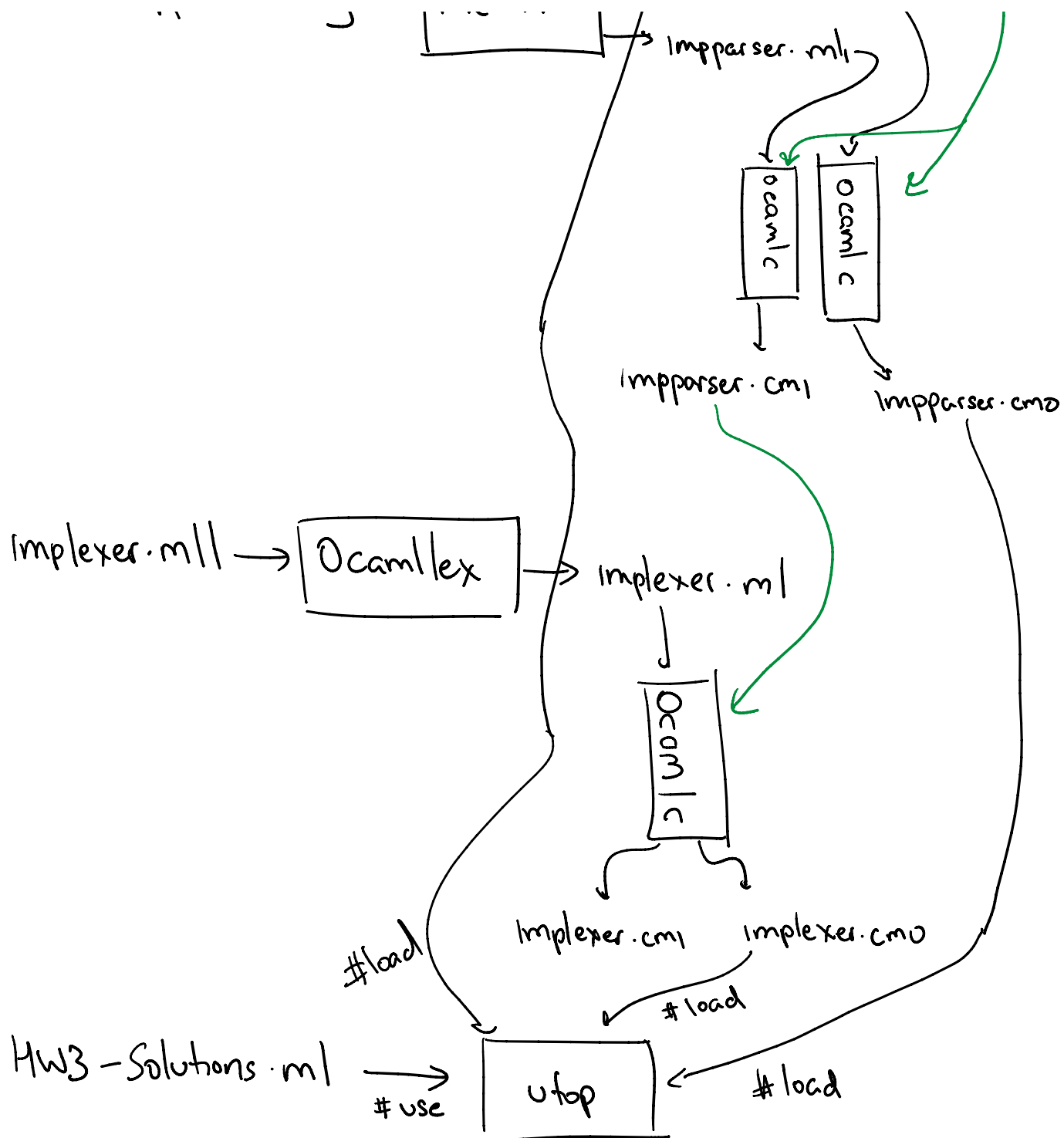
In `impparser.mly`, to access these constructs, either

- ① open `Imp`, or
- ② Write `Imp.v` for each thing `v`.

② Menhir / Ocamllex / Ocamlc / Utop

↓  
Just a super sophisticated preprocessor





Ambiguities in if statements without fi

if — then — else if — then —  
   '            '                    else —

if — then — else if — then — else —  
⇒ if — then —  
else (if — then —  
else —)

---

Q1: How do we parse

true and true and true

↙ ↘  
true and (true and true) (true and true) and true

true and false or true and true

↙ ↘  
(true and (false or true)) and true

(true and false) or (true and true)

3 - 4 + 5

↙ ↘

$$(3-4)+5 \qquad 3-(4+5)$$

Output 3; skip; Output 4

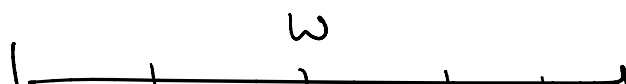
$$(Output\ 3); (skip; Output\ 4) \qquad (Output\ 3; skip); Output\ 4$$

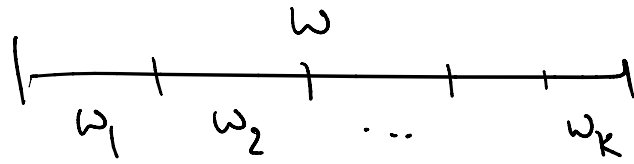
### Q3: Matching regular expressions

Does the string "abcdefg" match the pattern  $a^*$ ?

$a^*$  = all strings which can be broken into  $k$  parts ( $k \geq 0$ )

$$w = w_1 w_2 \dots w_k$$





where each part  $w_i$  matches  $a$ .

Does  $abcdefg$  match  $\frac{a^*}{\downarrow}$ ?

$\{ \_, a, aa, aaa, \dots \}$

$(a+b)^*$

$\{ \_, a, b, aa, ab, ba, bb, \dots \}$

$a^* \cdot b^*$

$\{ \_, a, aa, aaa, \dots$

$ab, aab, a^2ab, \dots$

$abb, aabb, aaabb, \dots$

$\vdots$

$\}$

