



# TopHat: a stylish journey through modular interactive workflows

*Tim Steenvoorden*

Markus Klinik

Nico Naus

Dutch FP Day, January 11, 2019





# Task Oriented Programming (TOP)

## Workflows

coordinate collaboration

## Interactive

driven by user input

## Modular

higher order

elementary building blocks and concepts

labeled transition system

embedding in simply typed  $\lambda$ -calculus  
(with clearly separated semantics)

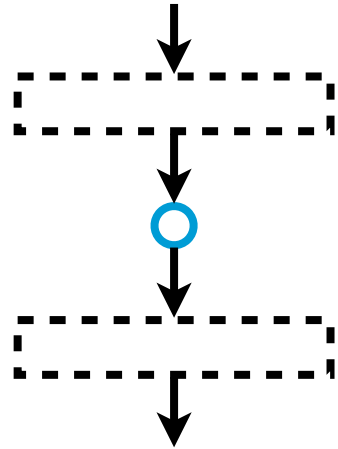
Foundation for formal reasoning  
and comparison to other frameworks



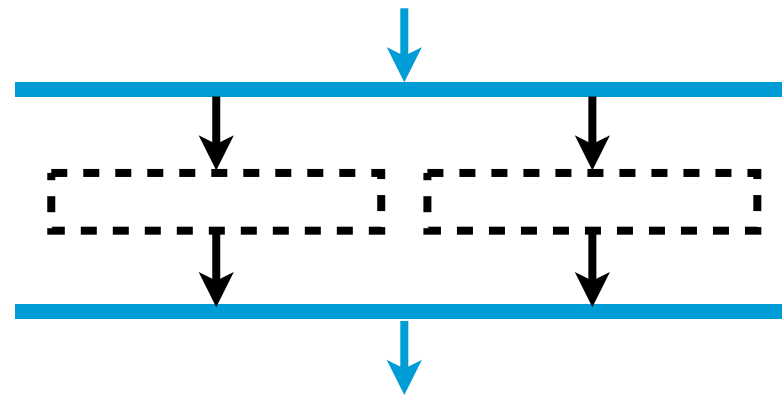


# Concepts

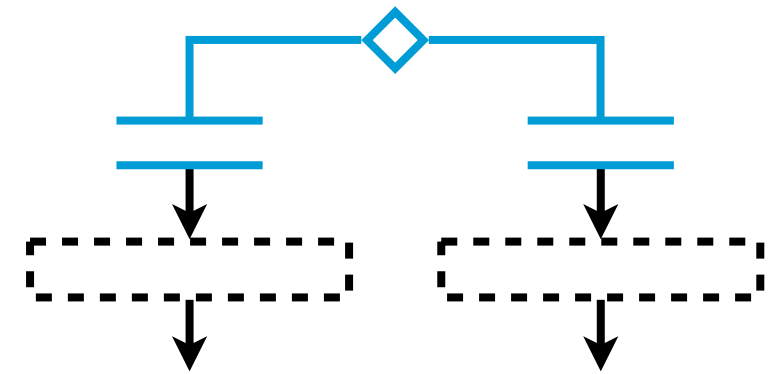
# Collaboration



AFTER EACH OTHER



AT THE SAME TIME



CONDITIONALLY

Communication is taken care of!

“We do  $A$  and  $B$  at the same time, then we continue with  $C$ .”



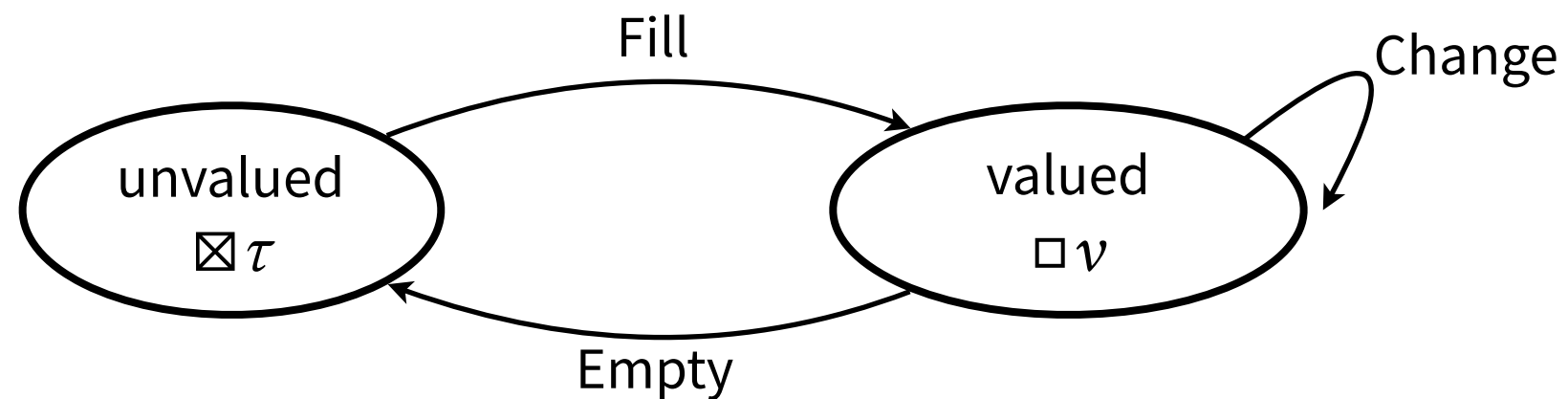
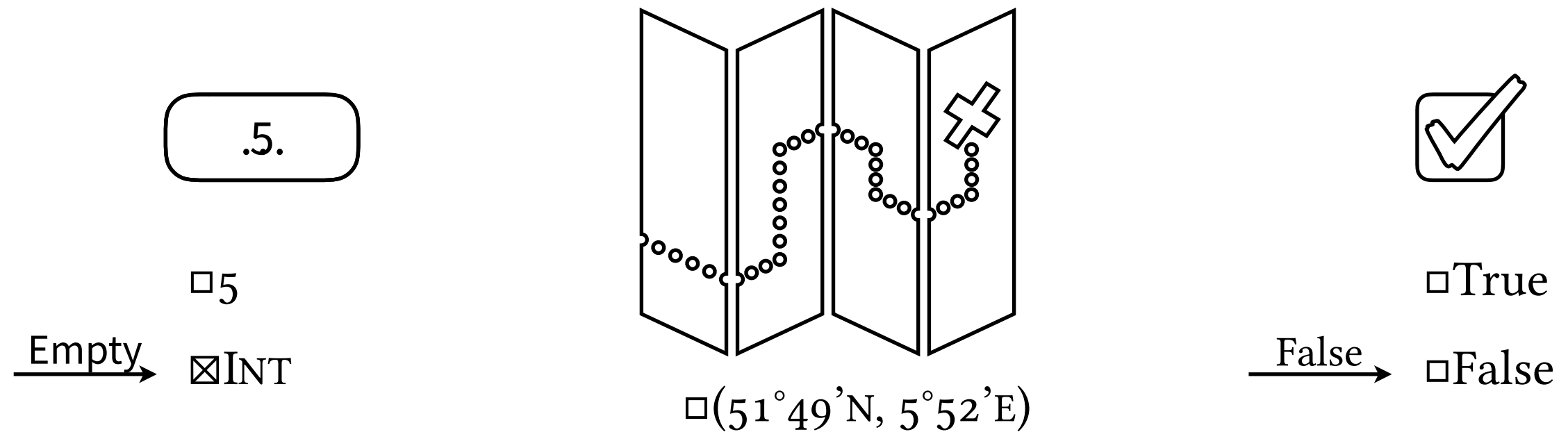
“When I send you  $M$ , you can do  $B$  and I’ll wait for you to send  $N$ .”



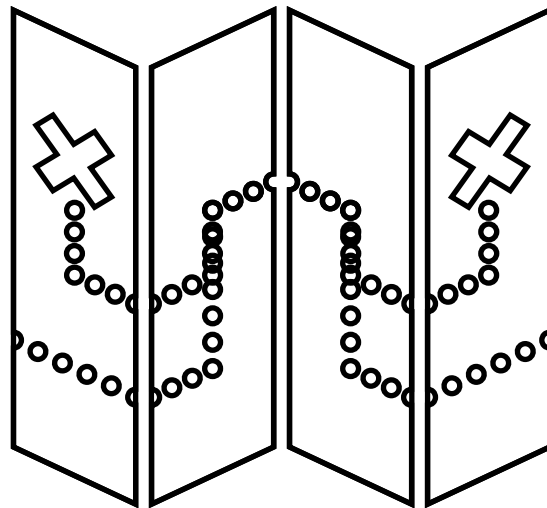


# Building blocks

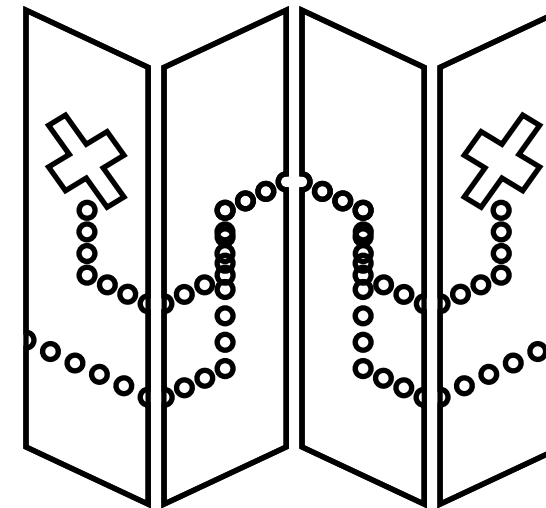
# Editors



# Shared editors



■(51°49'N, 5°52'E)



■(51°49'N, 5°52'E)

(51°49'N, 5°52'W)

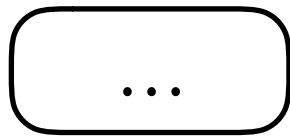
■(51°49'N, 5°52'W)



CAN NOT BE EMPTY!

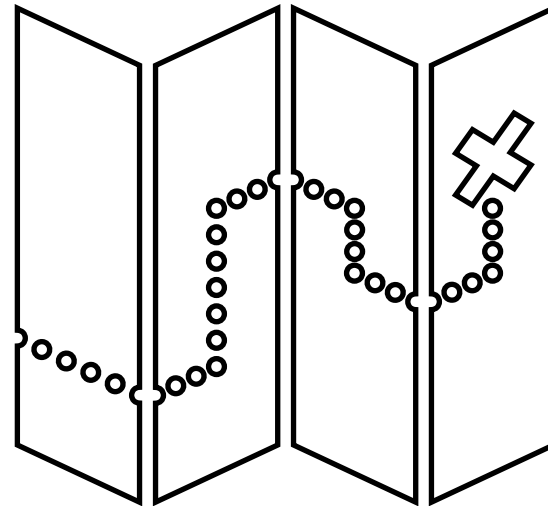
WATCHES A REFERENCE

# Observations



$$\mathcal{V}(\boxtimes \text{INT}) = \perp$$

SOME TASKS DO NOT HAVE A VALUE



$$\begin{aligned} \mathcal{V}(\Box(51^{\circ}49'N, 5^{\circ}52'E)) \\ = (51^{\circ}49'N, 5^{\circ}52'E) \end{aligned}$$



$$\mathcal{V}(\Box \text{True}) = \text{True}$$

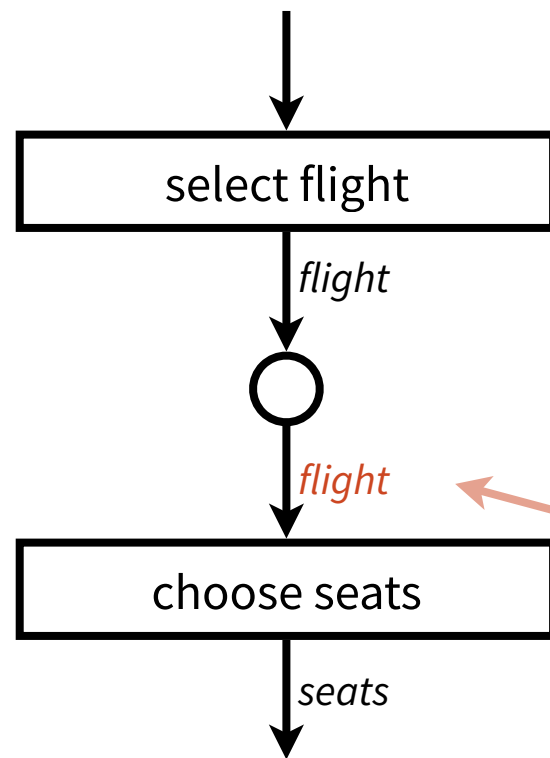
value  $\mathcal{V} : \text{TASK } \tau \rightarrow \text{MAYBE } \tau$

user interface  $\mathcal{U} : \text{TASK } \tau \rightarrow \text{HTML}$  (or  $\mathcal{U} : \text{TASK } \tau \rightarrow \text{STRING}$  or  $\mathcal{U} : \text{TASK } \tau \rightarrow \dots$ )

possible inputs  $\mathcal{J} : \text{TASK } \tau \rightarrow \text{LIST INPUT}$



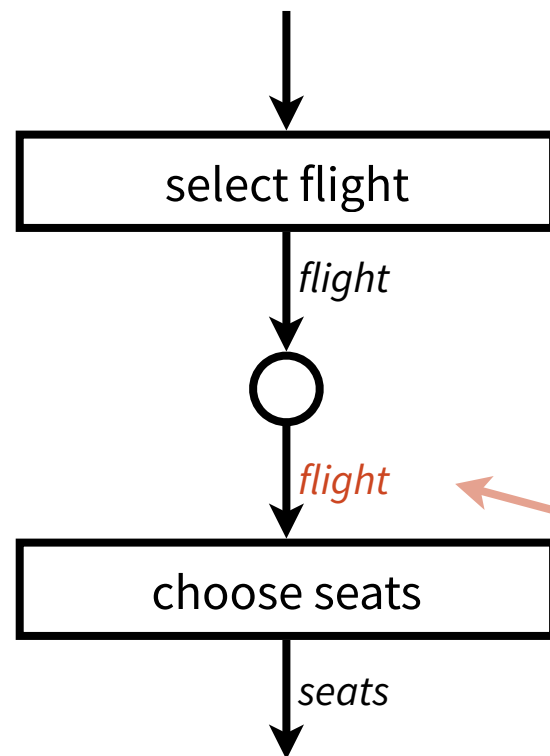
## Steps



select\_flight ► choose\_seats

USE VALUE IN NEXT TASK?

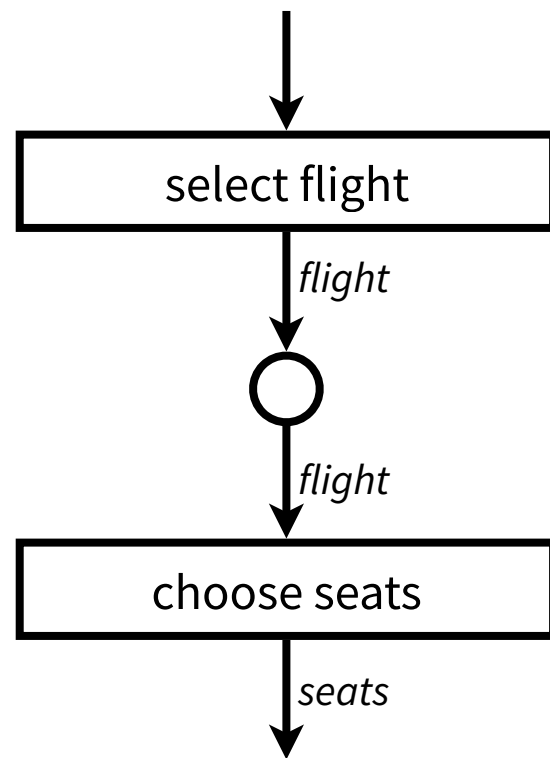
## Steps



select\_flight ►  $\lambda flight.$  choose\_seats *flight*

USE VALUE IN NEXT TASK?

## Steps

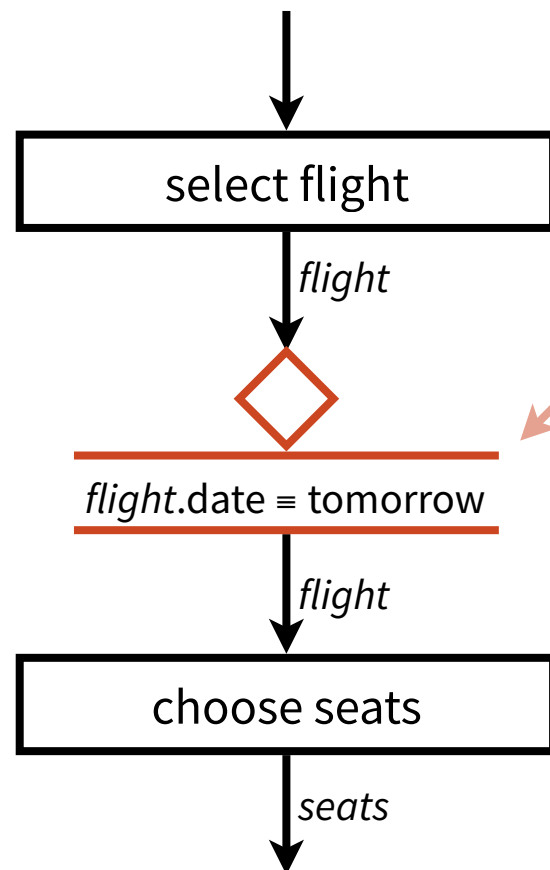


$\text{select\_flight} \triangleright \lambda \text{flight}. \text{choose\_seats } \text{flight}$

WHEN TO PROCEED TO THE NEXT TASK?

$$\Rightarrow \mathcal{V}(\text{select\_flight}) = v$$

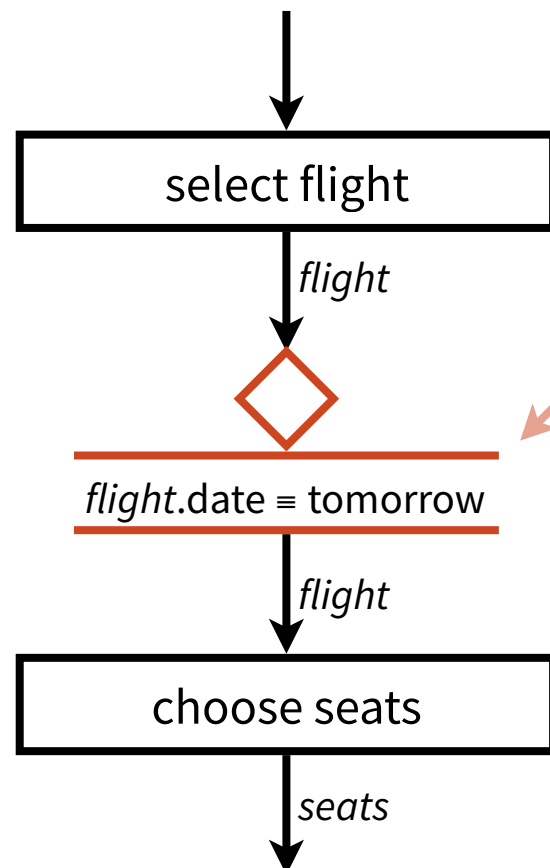
# Guarded steps



ONLY WHEN GUARD IS TRUE?

select\_flight  $\triangleright \lambda flight. \text{choose\_seats } flight$

# Guarded steps



ONLY WHEN GUARD IS TRUE?

$select\_flight \triangleright \lambda flight.$

**if**  $flight.date \equiv tomorrow$

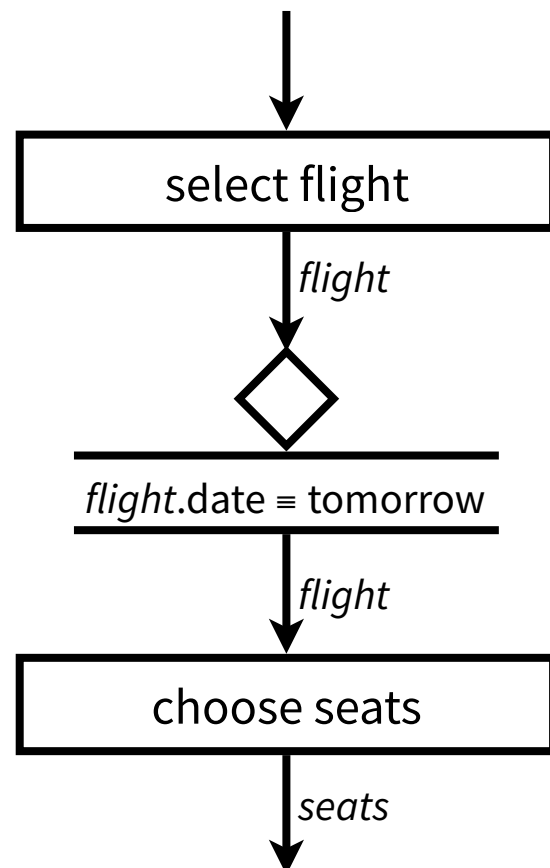
**then**  $choose\_seats\ flight$

**else**  $\downarrow$

FAILURE:

A TASK THAT NEVER ENDS  
AND CAN NOT HANDLE USER INPUT

# Guarded steps



$\text{select\_flight} \triangleright \lambda \text{flight.}$   
**if**  $\text{flight.date} \equiv \text{tomorrow}$   
**then**  $\text{choose\_seats } \text{flight}$   
**else**  $\downarrow$

WHEN TO PROCEED TO THE NEXT TASK?

$\Rightarrow \mathcal{V}(\text{select\_flight}) = v$   
**and**  $\dots (v) \Downarrow t$  **where**  $t \neq \downarrow$



USING HOST LANGUAGE SEMANTICS!



# Calculus

# Grammar

Take a  $\lambda$ -calculus...

$e ::=$		Expressions		
$\lambda x : \tau. e$	$e_1 e_2$	– abstraction, application		
$x$	$c$	$e_1 \star e_2$	– variable, constant, operation	
<b>if</b> $e_1$ <b>then</b> $e_2$ <b>else</b> $e_3$	$\langle \rangle$	– branch, unit		
$\langle e_1, e_2 \rangle$	<b>fst</b> $e$	<b>snd</b> $e$	– pair, projections	
<b>ref</b> $e$	<b>!</b> $e$	$e_1 := e_2$	$l$	– references, location
$p$				– pretask
$c ::=$		Constants		
$B$	$I$	$S$	– boolean, integer, string	



# Grammar

Take a  $\lambda$ -calculus...

$e ::=$	Expressions
$\lambda x : \tau. e$   $e_1 e_2$	– abstraction, application
$x$   $c$   $e_1 \star e_2$	– variable, constant, operation
<b>if</b> $e_1$ <b>then</b> $e_2$ <b>else</b> $e_3$   $\langle \rangle$	– branch, unit
$\langle e_1, e_2 \rangle$   <b>fst</b> $e$   <b>snd</b> $e$	– pair, projections
<b>ref</b> $e$   $!e$   $e_1 := e_2$   $l$	– references, location
$p$	– pretask
$c ::=$	Constants
$B$   $I$   $S$	– boolean, integer, string

...embed a workflow language

$t ::=$	Tasks
$\square v$   $\boxtimes \tau$   $\blacksquare l$	– editors
$t_1 \blacktriangleright e_2$   $t_1 \triangleright e_2$	– steps
$\not\downarrow$   $t_1 \bowtie t_2$	– fail, combination
$t_1 \blacklozenge t_2$   $e_1 \diamond e_2$	– choices

COMBINATION OF TWO TASKS

CHOICE BETWEEN TWO TASKS

# Semantics

Two layers  $\Rightarrow$  two semantics



$$\frac{\text{S-THENSTAY} \quad t_1, s \rightsquigarrow t'_1, s'}{t_1 \blacktriangleright e_2, s \rightsquigarrow t'_1 \blacktriangleright e_2, s'} \quad \mathcal{V}(t'_1, s') = \perp$$

$$\frac{\text{S-THENFAIL} \quad t_1, s \rightsquigarrow t'_1, s' \quad e_2 \ v_1, s' \downarrow t_2, s''}{t_1 \blacktriangleright e_2, s \rightsquigarrow t'_1 \blacktriangleright e_2, s'} \quad \mathcal{V}(t'_1, s') = v_1 \wedge \mathcal{F}(t_2, s'')$$

$$\frac{\text{S-THENCONT} \quad t_1, s \rightsquigarrow t'_1, s' \quad e_2 \ v_1, s' \downarrow t_2, s''}{t_1 \blacktriangleright e_2, s \rightsquigarrow t_2, s'''} \quad \mathcal{V}(t'_1, s') = v_1 \wedge \neg \mathcal{F}(t_2, s'')$$

# Semantics

Two layers  $\Rightarrow$  <sup>three</sup> ~~two~~ semantics

$e \downarrow v$  **STANDARD BIG STEP SEMANTICS**

$p \rightsquigarrow t$  **SPECIAL TASK SEMANTICS**

But interaction...  $\Rightarrow$  additional layer

$t \xrightarrow{i} t'$  **HANDLING OF USER INPUT**

$$\frac{\text{S-THENSTAY} \quad t_1, s \rightsquigarrow t'_1, s'}{t_1 \blacktriangleright e_2, s \rightsquigarrow t'_1 \blacktriangleright e_2, s'} \quad \mathcal{V}(t'_1, s') = \perp$$

$$\frac{\text{S-THENFAIL} \quad t_1, s \rightsquigarrow t'_1, s' \quad e_2 \ v_1, s' \downarrow t_2, s''}{t_1 \blacktriangleright e_2, s \rightsquigarrow t'_1 \blacktriangleright e_2, s'} \quad \mathcal{V}(t'_1, s') = v_1 \wedge \mathcal{F}(t_2, s'')$$

$$\frac{\text{S-THENCONT} \quad t_1, s \rightsquigarrow t'_1, s' \quad e_2 \ v_1, s' \downarrow t_2, s''}{t_1 \blacktriangleright e_2, s \rightsquigarrow t_2, s'''} \quad \mathcal{V}(t'_1, s') = v_1 \wedge \neg \mathcal{F}(t_2, s'')$$

□5

$\xrightarrow{\text{Empty}} \boxtimes \text{INT}$

■(51°49'N, 5°52'E)

$\xrightarrow{(51°49'N, 5°52'W)} \blacksquare(51°49'N, 5°52'W)$



# Summary



= the essence of task oriented programming

## Language + Formal semantics

$t ::=$

|  $\square v$  |  $\boxtimes \tau$  |  $\blacksquare l$   
|  $t_1 \blacktriangleright e_2$  |  $t_1 \triangleright e_2$   
|  $\not\downarrow$  |  $t_1 \bowtie t_2$   
|  $t_1 \blacklozenge t_2$  |  $e_1 \lozenge e_2$

Tasks

- editors
- steps
- fail, combination
- choices

Proved progress & preservation

Implemented in Idris

## Still to do...

- Task equality
- Pre- and postconditions
- Symbolic execution

IDEAS APPRECIATED!





Thank you

# Summary



Language & formal semantics



Proved progress & preservation



Implemented in Idris



Essence of task oriented programming

# Grammar

## TAKE A $\lambda$ -CALCULUS...

$e ::=$   
 $\mid \lambda x : \tau. e \mid e_1 e_2$   
 $\mid x \mid c \mid e_1 \star e_2$   
 $\mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \mid \langle \rangle$   
 $\mid \langle e_1, e_2 \rangle \mid \text{fst } e \mid \text{snd } e$   
 $\mid \text{ref } e \mid !e \mid e_1 := e_2 \mid l$   
 $\mid p$   
 $c ::=$   
 $\mid B \mid I \mid S$

### Expressions

- abstraction, application
- variable, constant, operation
- branch, unit
- pair, projections
- references, location
- pretask

### Constants

- boolean, integer, string

## ...EMBED A WORKFLOW LANGUAGE

$t ::=$

$\mid \square v \mid \boxtimes \tau \mid \blacksquare l$   
 $\mid t_1 \blacktriangleright e_2 \mid t_1 \triangleright e_2$   
 $\mid \text{⚡} \mid t_1 \bowtie t_2$   
 $\mid t_1 \blacklozenge t_2 \mid e_1 \diamond e_2$

### Tasks

- editors
- steps
- fail, combination
- choices

RACE BETWEEN TWO TASKS

COMBINATION OF TWO TASKS





## Summary

- Language for modular interactive workflows
- Essence of task oriented programming
- Formal semantics
- Proved progress & preservation
- Implemented in Idris

$t ::=$

|  $\square v$  |  $\boxtimes \tau$  |  $\blacksquare l$   
|  $t_1 \blacktriangleright e_2$  |  $t_1 \triangleright e_2$   
|  $\not\downarrow$  |  $t_1 \bowtie t_2$   
|  $t_1 \blacklozenge t_2$  |  $e_1 \lozenge e_2$

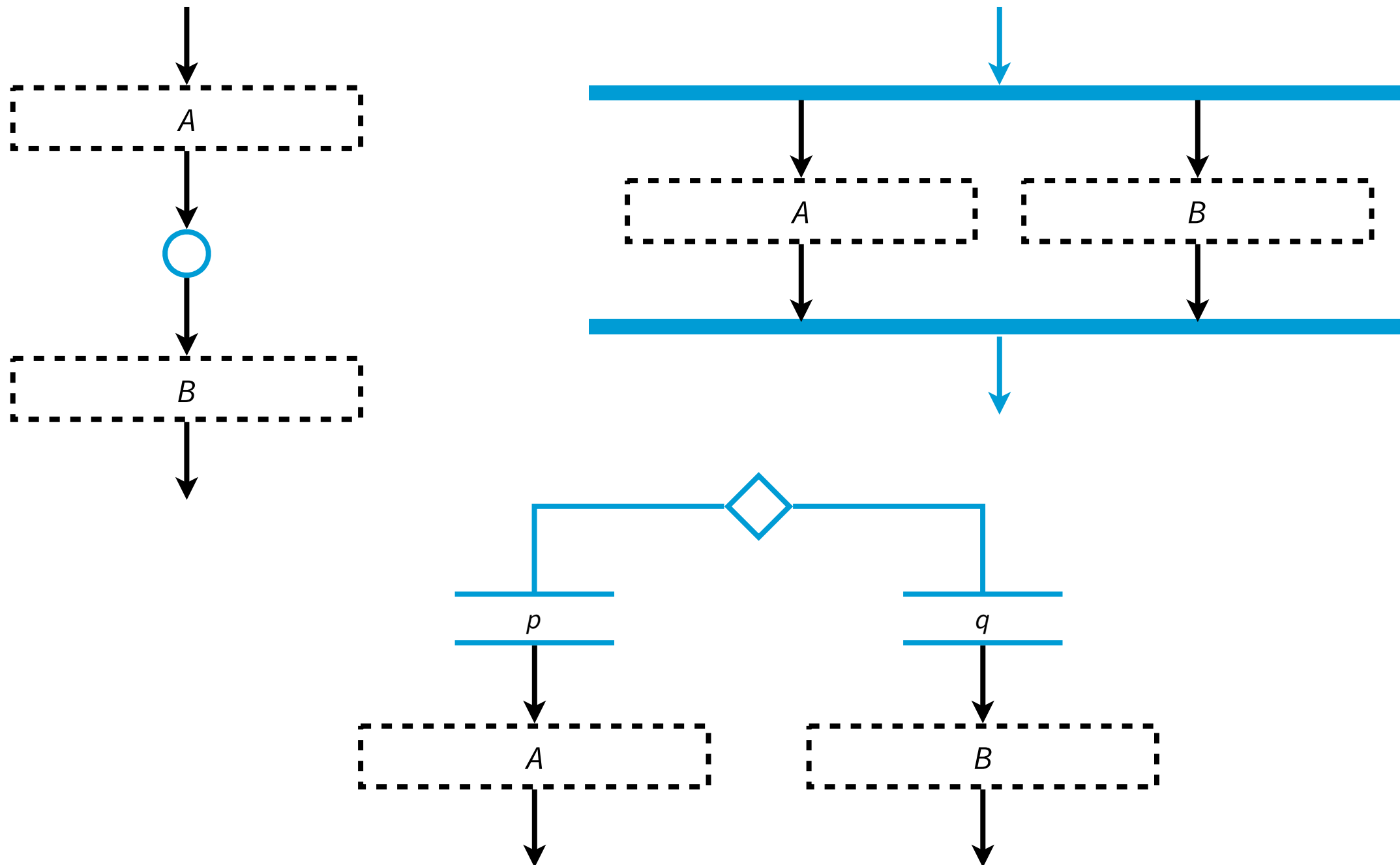
Tasks

- editors
- steps
- fail, combination
- choices

LANGUAGE  
+  
FORMAL SEMANTICS



# Tasks



$l$ ,  $\boxtimes$   $\square$   $\boxminus$   $\blacksquare$   $\square$   $\blacksquare$   $\square$   $\blacksquare$   $\square$   $\blacksquare$   $\square$

$l$ ,  $\boxtimes$   $\square$   $\boxminus$   $\blacksquare$   $\square$   $\blacksquare$   $\square$   $\blacksquare$   $\square$   $\blacksquare$   $\square$

$\blacksquare$   $l$ ,  $\boxtimes$   $\square$   $\boxminus$   $\blacksquare$   $\square$   $\blacksquare$   $\square$   $\blacksquare$   $\square$   $\blacksquare$   $\square$

