EECS3342 System Specification and Refinement (Winter 2022)

Q&A - Week 1 Lecture

Thursday, January 20

→ Jan. 31st

↳ online

Lectures/Q&A → available

Hypothetically in-person & exam possible

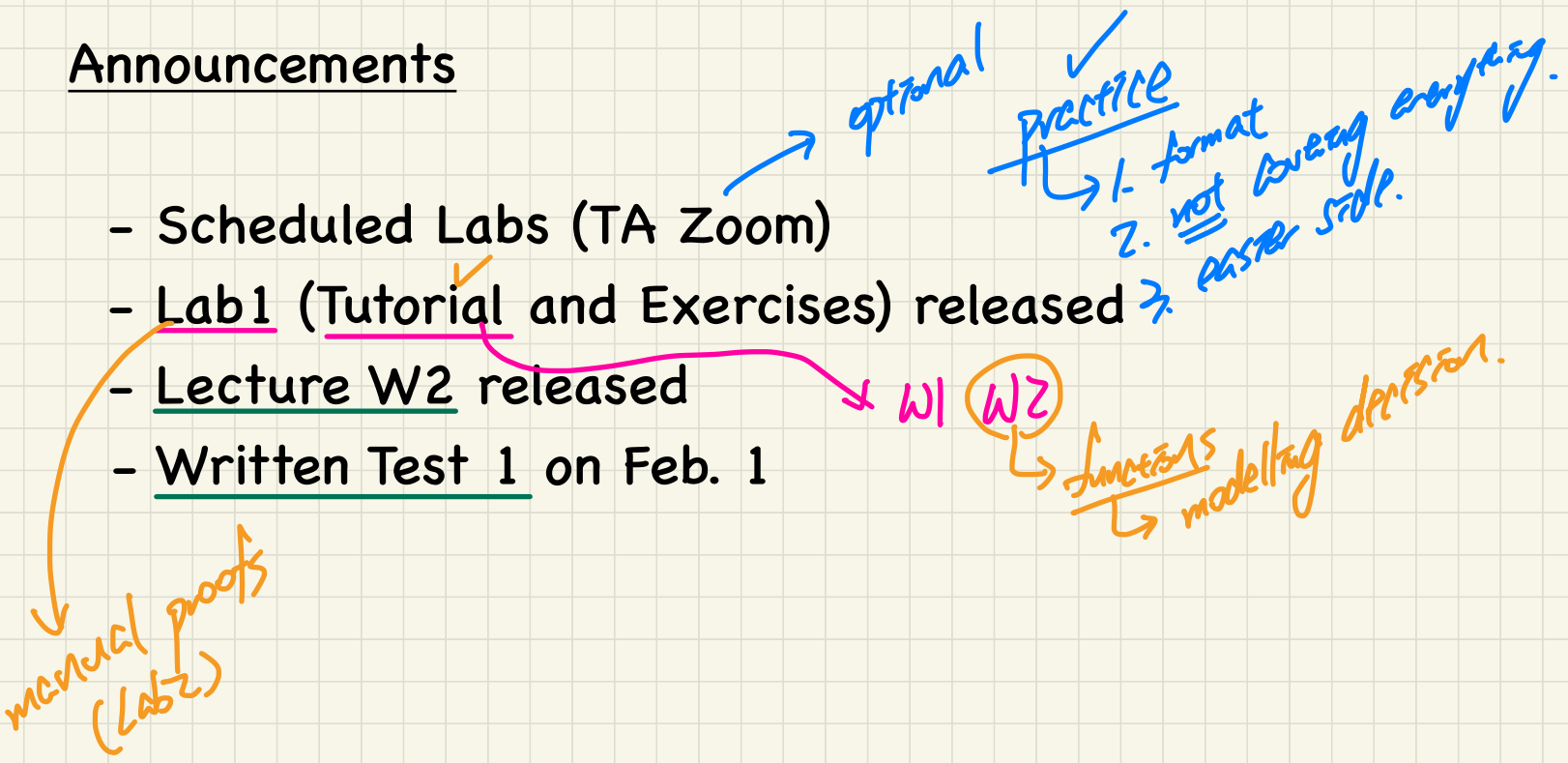
↳ 1. scheduled labs : ≥1 in-person

2. office hours : ≥1 in-person

3. tests : WT2, 3, 4 Prog.

Announcements

- Scheduled Labs (TA Zoom)
- Lab1 (Tutorial and Exercises) released
- Lecture W2 released
- Written Test 1 on Feb. 1



Will we ever need to create our own "informal requirements" for this course or will they be always given for us to conform to?

↗ E- or R-des.

1. No, you'll always be given req. doc.

EECS 4312.

2. Formalize the req. into

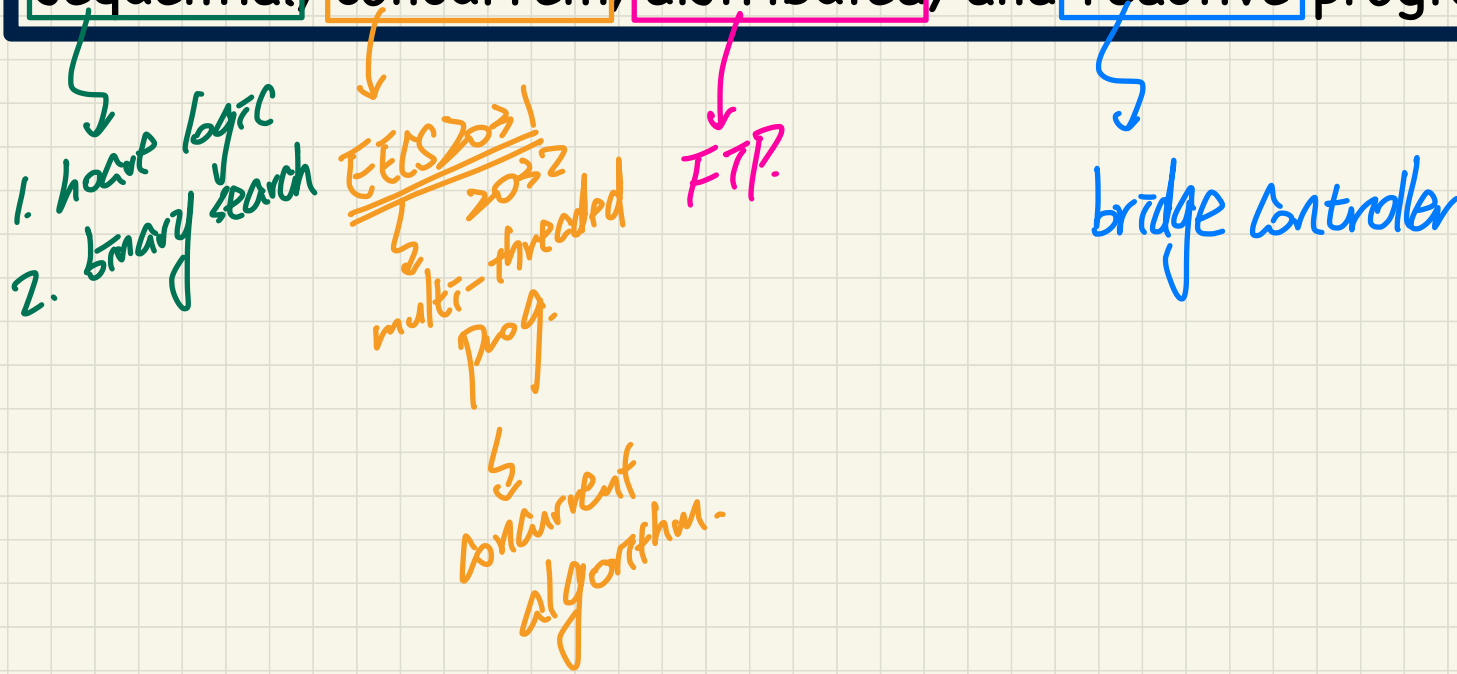
modelling decisions

Context + machine

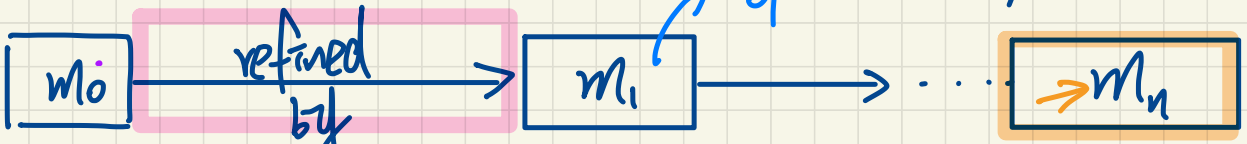
Context: constants, axioms, theorems

machine: variables, invariant, event (param., guards, action)

What would be some real world examples of the sequential, concurrent, distributed, and reactive programs?



Model - Based Development



- What defines the most abstract model? → a small # of vars & const, constraints.
- Does it just satisfy one requirement? → a small # of req.
- What defines the most concrete model → sufficiently close the code
- Does it satisfy all requirements? → ideally.
- When do we stop adding refinements? → Ch. 2
- How do we know what details to add in a subsequent refinement?
- What does it mean to prove a refinement? Will it be formal or informal?
- How complex will the models be in this class? → each ref. triggers extra POs.
- Will you teach us the refinement process in detail?
- Because you mentioned we will have to do this ourselves? YES
- When you are marking our work do you value correctness over creativity?

case by case study

Prove/Disprove Logical Quantifications

- Prove or disprove: $\forall x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \Rightarrow x > 0.$

show true / show false

True: For #s 1, 2, ..., 10, all > 0

- Prove or disprove: $\forall x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \Rightarrow x > 1.$

Disprove: 1. (T) $1 \in \mathbb{Z} \land 1 \le 1 \le 10 \Rightarrow 1 > 1$ (F)

- Prove or disprove: $\exists x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \land x > 1.$

Prove. Witnesses (2, 3, ..., 10) I cannot be the witness

$3 \in \mathbb{Z} \land 1 \le 3 \le 10 \land 3 > 1$ (T)

- Prove or disprove that $\exists x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \land x > 10$?

Disprove: For 1, 2, ..., 10 each value ≤ 10 meaning that false > 10

Axiom vs. Theorem

$\mathbb{N}_1 = \{1, 2, 3, \cdots +\infty\} = \{x \mid x > 0\}$

```
C0
CONTEXT
    C0
SETS
  ∘ ACCOUNT    >carrier set: abstract without the need to enumerate content of the set
CONSTANTS
  ∘ c    >credit limit (ENV3)
AXIOMS
  ∘ axm1: c ∈ ℕ1 not theorem >// not theorem means an axiom; theorem means a proof is needed.
  ∘ thm1: c > 0 theorem >
END
```

axiom: no proofs needed; may be used to prove theorems

theorem: proof needed

$c \in \mathbb{N}_1 \Rightarrow c > 0$

positive number

$\{x \mid x \in \mathbb{Z} \wedge x > 0\}$

hypotheses

$axm1 \wedge axm2 \cdots \wedge axm_n \Rightarrow thm1$

goal

$c \in \mathbb{N}_1$ (typing constraint) declare

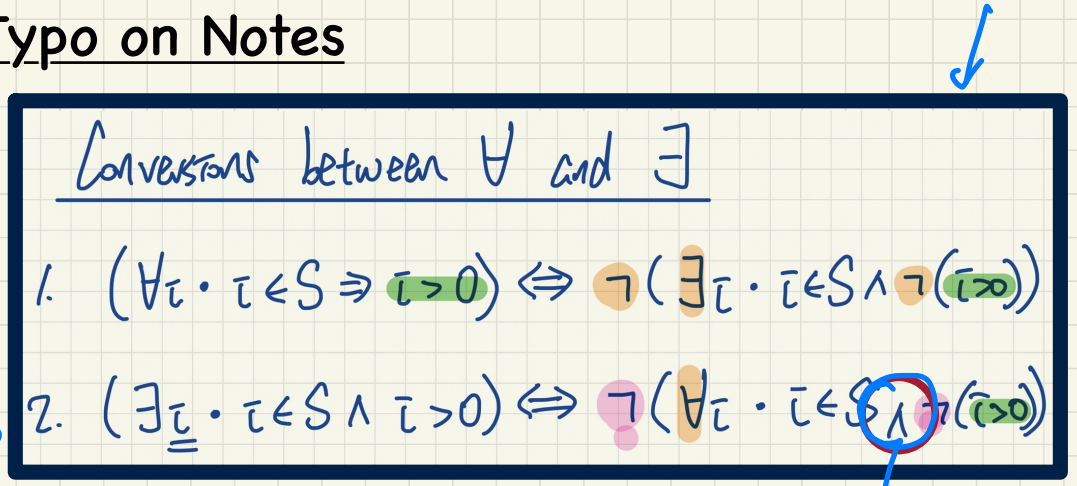
$c > 0$ provable from the typing constraint.

Java: `int c` ≡ $c \in int$

Context

- names of constants (C_1, C_2)
- typing constraint of declared names
  (e.g. $C_1 \in \mathbb{N}\setminus\{0\}$, $C_2 \in \mathbb{N}$)
- theorems
  - $C_1 > 0$ ✓ — > 0
  - $C_2 \geq 0$ ✓ — ≥ 0 — $> 0 \vee = 0$
    - $C_2 > 0$ ✗
    - $C_2 = 0$ ✗
  - $C_1 + C_2 > 0$ ✓
  - $C_1 \cdot C_2 > 0$ ✗ counter-example
    - ① $C_1 > 0 \wedge C_2 > 0 \Rightarrow C_1 \cdot C_2 > 0$
    - ② $C_1 > 0 \wedge C_2 = 0 \Rightarrow C_1 \cdot C_2 = 0$

Typo on Notes



Conjunction to implication?

$$f \in \{a, b\} \leftrightarrow \{1, 2\}$$

$$f = \{(a, 1), (b, 1)\}$$

$S \leftrightarrow T$

$$g \in \{(a, 1), (b, 2)\} \leftrightarrow \{x, y\}$$

$$g = \{((a, 1), y), ((b, 2), x)\}$$

$$dom(g) = \{(a, 1), (b, 2)\}$$