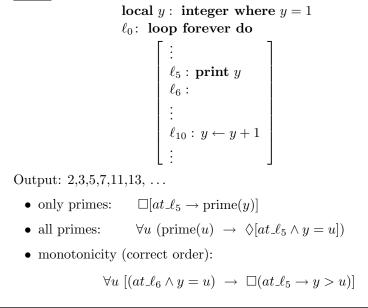
CS156: Topics

- Verification of **sequential** programs.
 - No concurrency.
 - Programs (should) always terminate.
 - Observable at start (input) and end (output) of execution.
- Logical foundations:
 - FOL.
 - Invariants and ranking functions.
 - Verification conditions.
 - Decision procedures.
 - Invariant generation.
 - Induction.

1

PRIME



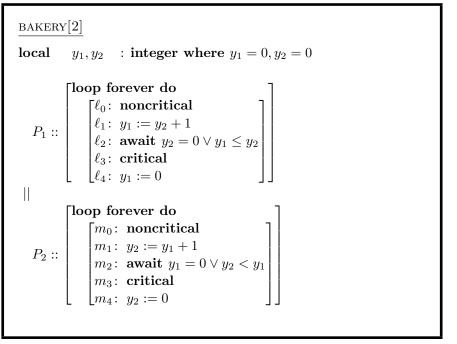
CS256: Topics

- Verification of **reactive systems**.
 - Highly concurrent.
 Concept of fairness.

Properties: mutual exclusion, freedom from deadlock.

- Programs need not terminate (e.g., OS, web server).
 But some components must terminate (e.g., IO handler).
- Observable throughout execution.
 And the environment affects execution.
- \bullet Logical foundations: Everything from CS156 plus
 - temporal logics
 linear (LTL), branching (CTL), alternating (ATL) time
 - automata theory and connection with temporal logics infinite strings (linear) and trees (branching, alternating)

2



Requirements for BAKERY[2]

• Mutual exclusion

 $\Box \neg (\ell_3 \land m_3)$

The two processes are not in the critical section simultaneously.

• One-bounded overtaking

 $\ell_2 \Rightarrow \neg m_3 \mathcal{W} m_3 \mathcal{W} \neg m_3 \mathcal{W} \ell_3$

Once P_1 waits to get access, P_2 can enter its critical section at most once.

• Progress

$$\ell_1 \Rightarrow \Diamond \ell_3$$

Once P_1 shows interest in entering its critical section, it eventually gets access to the critical section.

5

CS256: Administration

- TTh 11:00-12:15, Gates B12
- Instructor: Zohar Manna TA: Matteo Slanina
- \bullet Text:

The Temporal Verification of Reactive Systems: Safety Zohar Manna and Amir Pnueli

• Prerequisites: CS103, CS156, or equivalent background

6