

CS156: Calculus of Computation

- ▶ 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.
 - ▶ Induction.

CS256: Formal Methods for Reactive Systems

- ▶ 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.
- ▶ 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)

PRIME

local y : integer where $y = 1$

l_0 : **loop forever do**

$$\left[\begin{array}{l} \vdots \\ l_5 : \mathbf{print} \ y \\ l_6 : \\ \vdots \\ l_{10} : y \leftarrow y + 1 \\ \vdots \end{array} \right]$$

Output: 2,3,5,7,11,13, ...

- ▶ only primes:

$$\Box[at_l_5 \rightarrow \text{prime}(y)]$$

- ▶ all primes:

$$\forall u. [\text{prime}(u) \rightarrow \Diamond(at_l_5 \wedge y = u)]$$

- ▶ monotonicity (correct order):

$$\forall u. \Box[(at_l_6 \wedge y = u) \rightarrow \Box(at_l_5 \rightarrow y > u)]$$

BAKERY

local y_1, y_2 : **integer where** $y_1 = 0, y_2 = 0$

P_1 ::

loop forever do

l_0 : **noncritical**

l_1 : $y_1 := y_2 + 1$

l_2 : **await** $y_2 = 0 \vee y_1 \leq y_2$

l_3 : **critical**

l_4 : $y_1 := 0$

||

P_2 ::

loop forever do

m_0 : **noncritical**

m_1 : $y_2 := y_1 + 1$

m_2 : **await** $y_1 = 0 \vee y_2 \leq y_1$

m_3 : **critical**

m_4 : $y_2 := 0$

Requirements for BAKERY

- ▶ Mutual exclusion

$$\square \neg (l_3 \wedge m_3)$$

The two processes are not in the critical section simultaneously.

- ▶ One-bounded overtaking

$$l_2 \Rightarrow \neg m_3 \mathcal{W} m_3 \mathcal{W} \neg m_3 \mathcal{W} l_3$$

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

- ▶ Progress

$$l_1 \Rightarrow \diamond l_3$$

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

CS256: Administration

- ▶ TTh 12:50-2:05, Gates B12
- ▶ Instructor: Zohar Manna
TA: Eric Smith
- ▶ Text:

The Temporal Verification of Reactive Systems: Safety
Zohar Manna and Amir Pnueli
Springer-Verlag 1995

- ▶ Prerequisites: CS103, CS156, or equivalent background