Feb 11
Safety and Liveness

Vijay Chidambaram

Safety and Liveness

- Safety: "bad things don't happen (ever)"
- Liveness: "good things happen eventually"
- We will formalize these properties to help reason about them
- We will build on the theory we have learnt so far

Property

- Property: A set of infinite sequences of program states
- A program satisfies property P if each of its histories H is a subset of P
- We will use Buchi automata to specify properties
- A buchi automaton M accepts the sequence of program states in the property it specifies

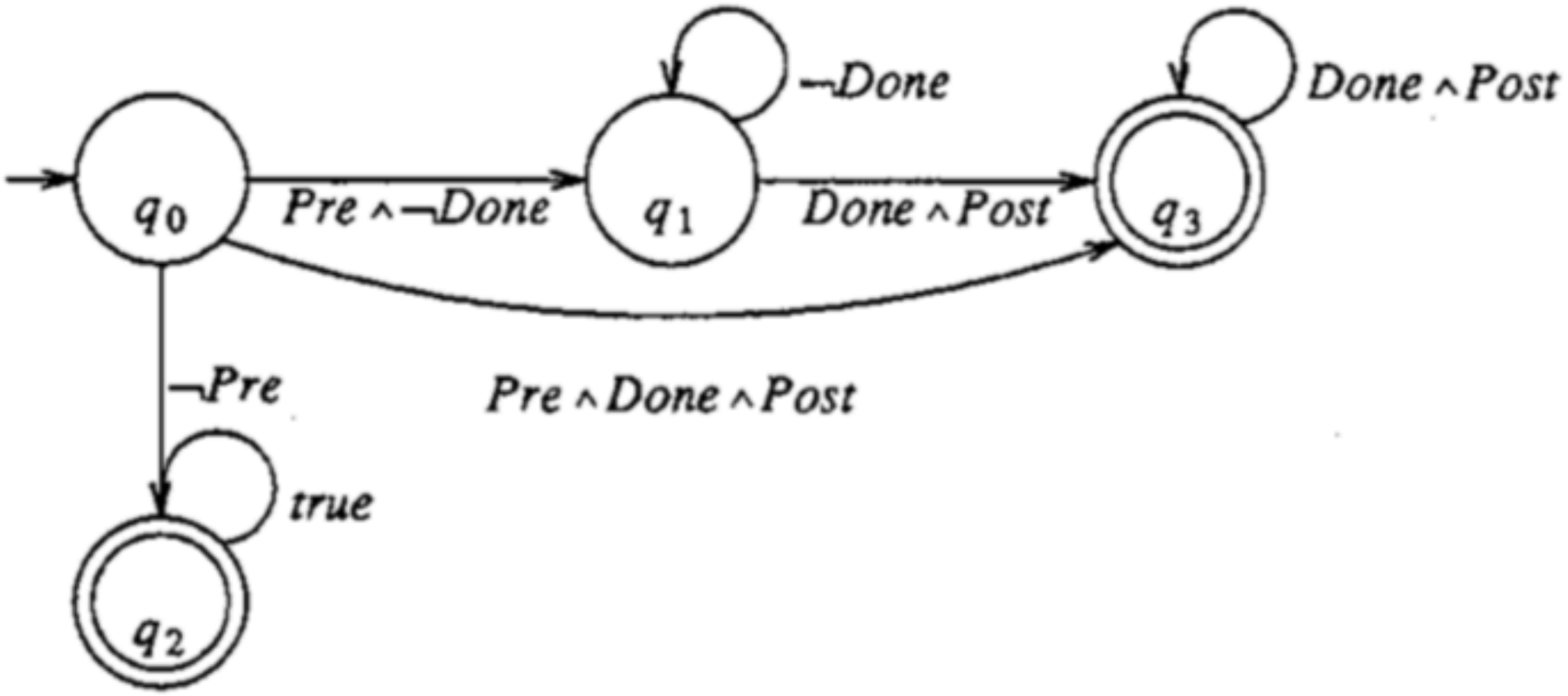
¬Done
Done ∧ Post
q0
Pre ∧ ¬Done
q1
Done ∧ Post
q3
¬Pre
Pre ∧ Done ∧ Post
true
q2

Fig. 1. m_{tc}

Buchi Automatons

- A buchi automaton contains a start state and a set of accepting states
- Arcs between states are labelled with predicates called transition predicates
- If a state does not have an arc for a given program state, we say an undefined transition occurs
- A buchi automaton is reduced if there is a path from every state to an accepting state
- Non-deterministic automatons have multiple start states or multiple transition arcs for the same input

Buchi Automatons

- Formally, a Buchi automaton m for a property of a program rc is a five-tuple (S, Q, Q-start, Q-accept, D)

- S is the set of program states of m,
  Q is the set of automaton states of m,
  D (Q, S) is the transition function of m.

-

Specifying Safety

Safety:

$$(\forall \sigma : \sigma \in S^{\omega} : \sigma \models P$$
$$\Leftrightarrow (\forall i : 0 \leq i : (\exists \beta : \beta \in S^{\omega} : \sigma[..i]\ \beta \models P))), \qquad (3.1)$$

- All runs are a subset of property P
- For a reduced Buchi automaton m, define its closure cl(m) to be the corresponding Buchi automaton in which every state has been made into an accepting state.
- The closure of m can be used to determine whether the property specified by m is a safety property.
- It rejects only by attempting an undefined transition (a "bad thing").
- If m and cl(m) accept the same language then m recognizes a safety property.

Specifying Liveness

- The thing to observe about a liveness property is that no partical execution is irremediable since if some partial execution were irremediable, then it would be a "bad thing" (and thus a safety property)

- A buchi automaton m specifies a liveness property if and only if its closure accepts every input

- For all finite inputs, there exists an infinite sequence of states that result in the property P being maintained

Liveness: $(\forall \alpha : \alpha \in S^* : (\exists \beta : \beta \in S^\omega : \alpha\beta \models P))$. (3.4)

Partioning into safety and liveness

- Given a Buchi automaton m, it is not difficult to construct Buchi automata Safe(m) and Live(m) such that Safe(m) specifies a safety property, Live(m) specifies a liveness property, and the prop- erty specified by m is the intersection of those speci- fied by Safe(m) and Live(m).

- Safe(m) = closure of m

- Live(m) = m augmented with a trap accepting state and all other state transitioning on undefined input to the trap state

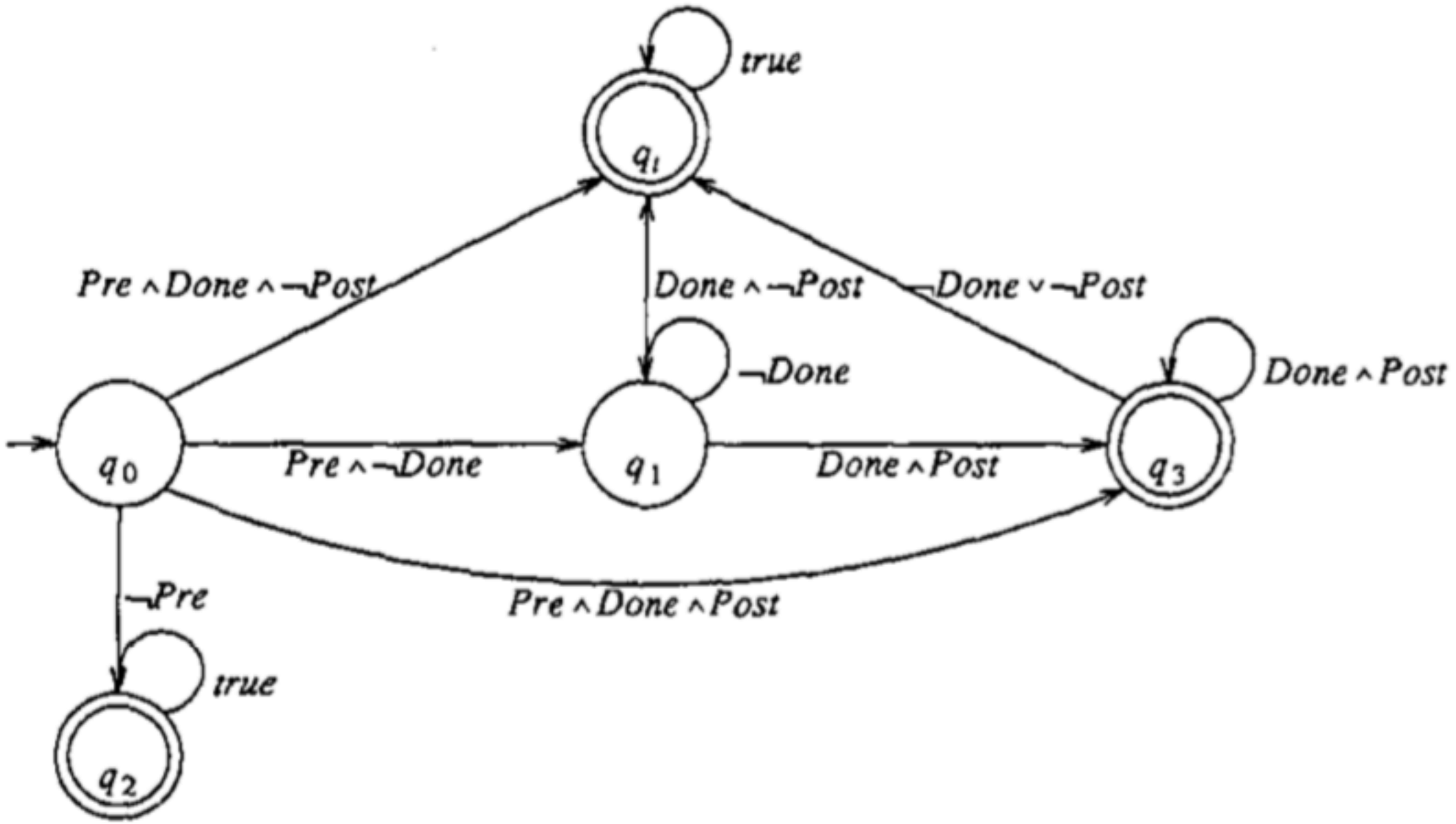

Fig. 8. *Live*(m_{tc})