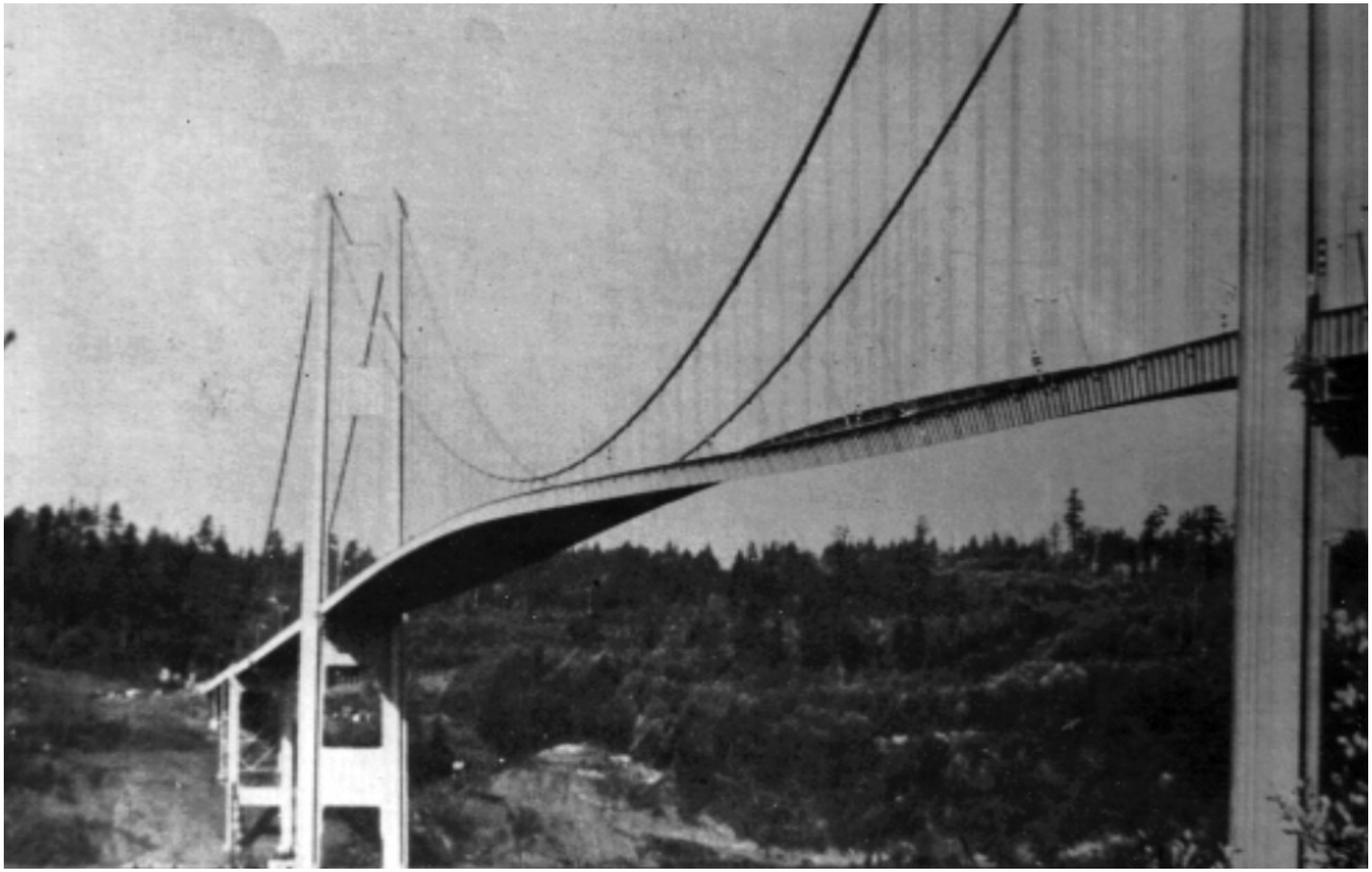
CSE595 Topics in Convergence Research

Model Checking

YoungMin Kwon

Challenger Disaster

Tacoma (WA) Bridge Collapse



https://www.youtube.com/watch?v=j-zczJXSxnw

Chernobyl Nuclear Power Plant Disaster

Ariane 5 Rocket Failure

https://www.youtube.com/watch?v=PK_yguLapgA

Model Checking

- Traditional validations methods
  - Simulation with models
  - Testing on real systems
  - Reasoning (manual or computer aided proof)

- Model Checking
  - Automatic techniques for verifying concurrent systems
  - Always terminate with yes/no answer

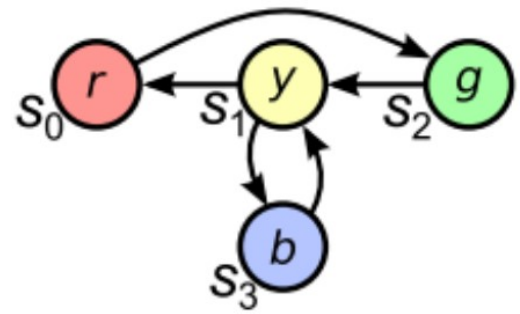
Model Checking Process

- Modeling
  - Convert a design into a formalism accepted by a model checking tool
- Specification
  - State the properties that the design must satisfy
  - Temporal logics are commonly used
- Verification
  - Check whether the model satisfies the specification
  - Need to analyze the traces for the negative results

Modeling Systems

- State
  - Snapshot or instantaneous description of the system
- Transition
  - Change of states
- Computation
  - Infinite sequence of states where the change of states is defined by the transition

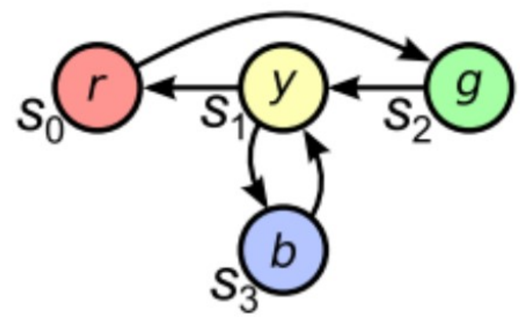
Kripke Structure



- A state transition graph
  - A set of states
  - A set of transitions between states
  - A function that labels each state with a set of properties that are true in this state
  - Paths in a Kripke structure model computations of the system

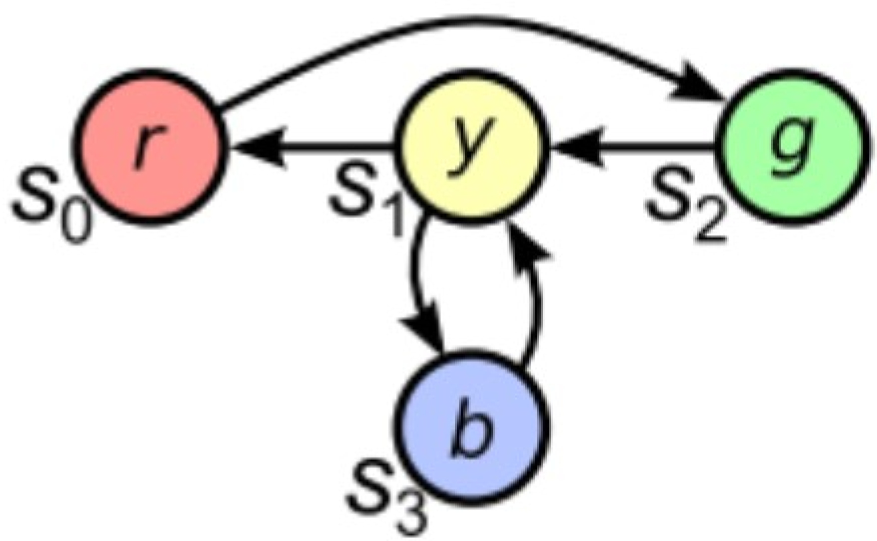
Kripke Structure

- Formally, a Kripke structure M over a se of atomic propositions AP is a four tuple $M=(S,S_0,R,L)$, where
  - S is a finite set of states
  - $S_0 \subseteq S$ is the set of initial states
  - $R \subseteq SxS$ is a transition relation
  - $L: S \rightarrow 2^{AP}$ is a function that labels each with the set of atomic propositions that are true in that state



Kripke Structure

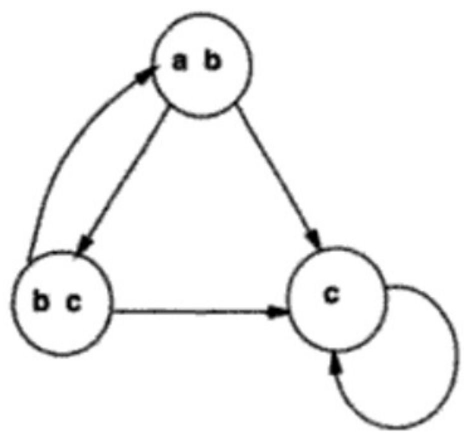
- Example
  - States: $\{S_0, S_1, S_2, S_3\}$
  - Transitions: $\{(S_0,S_2), (S_1,S_0), (S_1,S_3), (S_2,S_1), (S_3,S_1)\}$
  - Labeling function: $L(S_0)=r$, $L(S_1)=y$, $L(S_2)=g$, $L(S_3)=b$



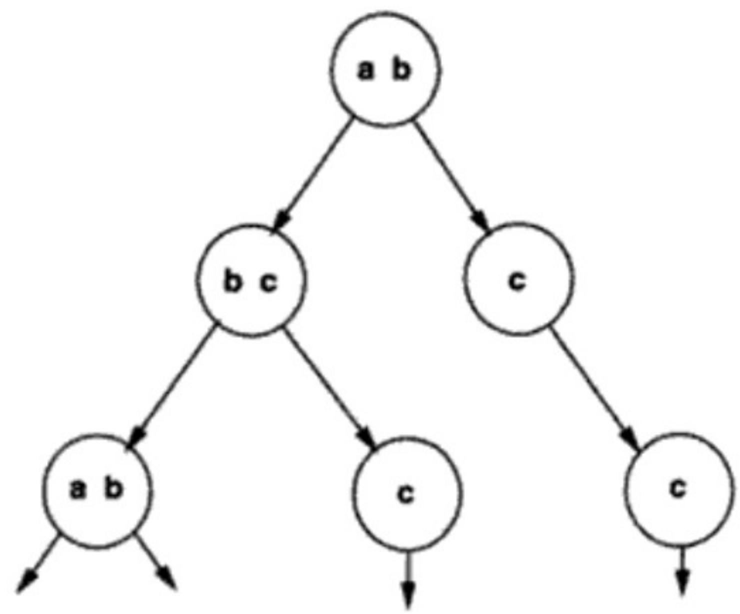
Specification

- Properties of a system can be described by temporal logics
- Temporal logics are a logic with temporal operators as well as logical operators
- Sequences of state transitions of a system can be described by temporal logics

Temporal Logic



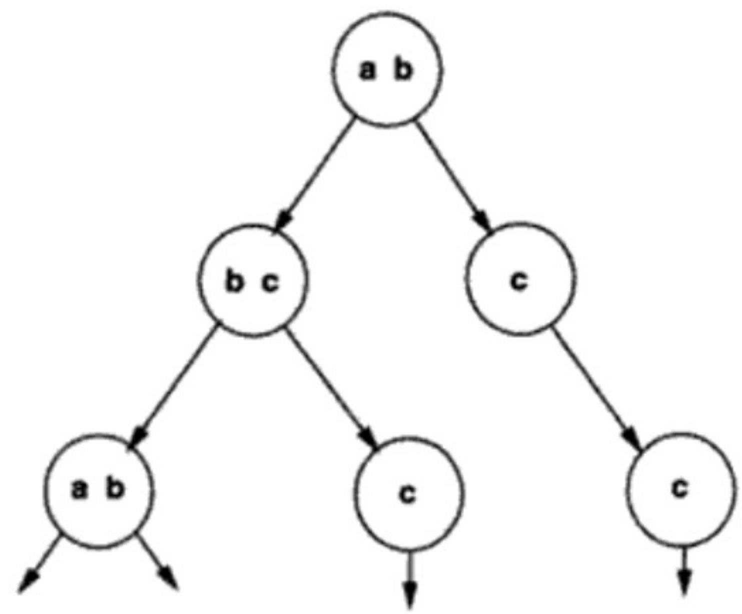
A Kripke Structure



Unwind State Graph

Temporal Logic

- Logical operators
  - ~: not
  - /\: and
  - \/: or
- Temporal operators
  - X: next
  - <>: eventually
  - [ ]: always
  - U: until
  - R: release
- Path quantifiers
  - A: for all computation paths
  - E: for some computation paths



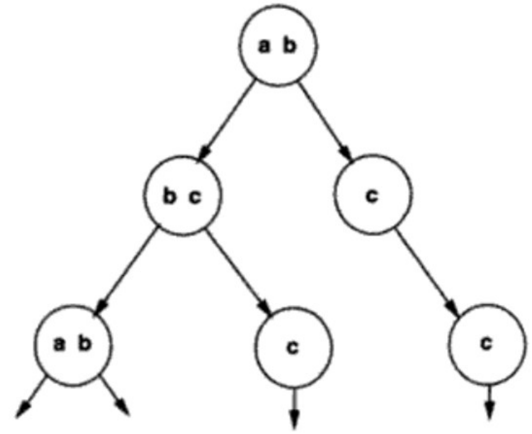
Temporal Logic

- State Formula and Path Formula

  - If $p \in AP$, then p is a state formula.
  - If f and g are state formulas, then $\neg f$, $f \vee g$ and $f \wedge g$ are state formulas.
  - If f is a path formula, then $\mathbf{E}\ f$ and $\mathbf{A}\ f$ are state formulas.

  - If f is a state formula, then f is also a path formula.
  - If f and g are path formulas, then $\neg f$, $f \vee g$, $f \wedge g$, $\mathbf{X}\ f$, $\mathbf{F}\ f$, $\mathbf{G}\ f$, $f\ \mathbf{U}\ g$, and $f\ \mathbf{R}\ g$ are path formulas.

Formal Semantics



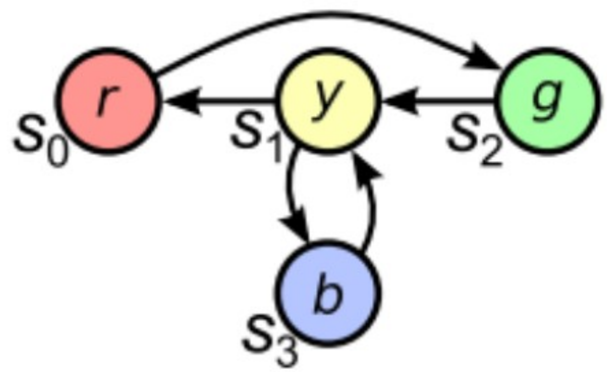
$M, s \models p$	$\Leftrightarrow$	$p \in L(s)$.
$M, s \models \neg f_1$	$\Leftrightarrow$	$M, s \not\models f_1$.
$M, s \models f_1 \vee f_2$	$\Leftrightarrow$	$M, s \models f_1$ or $M, s \models f_2$.
$M, s \models f_1 \wedge f_2$	$\Leftrightarrow$	$M, s \models f_1$ and $M, s \models f_2$.
$M, s \models \mathbf{E}\ g_1$	$\Leftrightarrow$	there is a path π from s such that $M, \pi \models g_1$.
$M, s \models \mathbf{A}\ g_1$	$\Leftrightarrow$	for every path π starting from s, $M, \pi \models g_1$.
$M, \pi \models f_1$	$\Leftrightarrow$	s is the first state of π and $M, s \models f_1$.
$M, \pi \models \neg g_1$	$\Leftrightarrow$	$M, \pi \not\models g_1$.
$M, \pi \models g_1 \vee g_2$	$\Leftrightarrow$	$M, \pi \models g_1$ or $M, \pi \models g_2$.
$M, \pi \models g_1 \wedge g_2$	$\Leftrightarrow$	$M, \pi \models g_1$ and $M, \pi \models g_2$.
$M, \pi \models \mathbf{X}\, g_1$	$\Leftrightarrow$	$M, \pi^1 \models g_1$.
$M, \pi \models \mathbf{F}\, g_1$	$\Leftrightarrow$	there exists a $k \geq 0$ such that $M, \pi^k \models g_1$.
$M, \pi \models \mathbf{G}\, g_1$	$\Leftrightarrow$	for all $i \geq 0$, $M, \pi^i \models g_1$.
$M, \pi \models g_1\, \mathbf{U}\, g_2$	$\Leftrightarrow$	there exists a $k \geq 0$ such that $M, \pi^k \models g_2$ and for all $0 \leq j < k$, $M, \pi^j \models g_1$.
$M, \pi \models g_1\, \mathbf{R}\, g_2$	$\Leftrightarrow$	for all $j \geq 0$, if for every $i < j$ $M, \pi^i \not\models g_1$ then $M, \pi^j \models g_2$.

Temporal Logic

- LTL (Linear Temporal Logic)
  - A f, where f has unrestricted use of logical and temporal operators but without path quantifiers
- CTL (Computation Tree Logic)
  - Temporal operators must be immediately preceded by path quantifiers
- CTL* (Computation Tree Logic)
  - Logical operators, temporal operators, and path quantifiers can be used without restriction

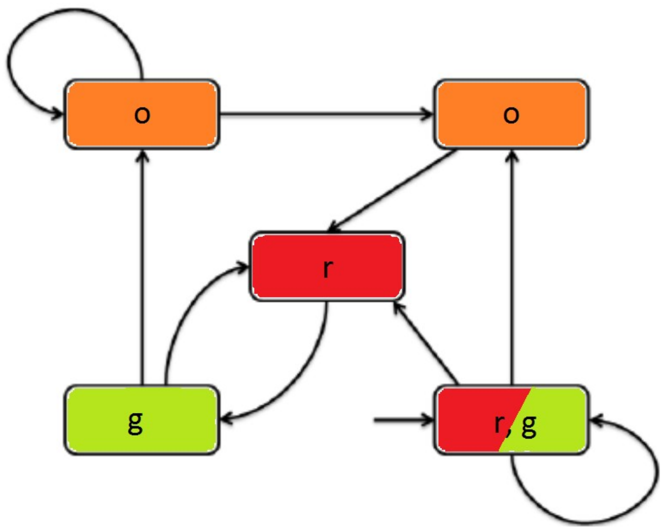
Linear Temporal Logic

- Suppose that S_0 is the initial state
  - r
  - X g, XXX r, XX g
  - <> y, <> b
  - [] <> y, <> [] y
  - (r \/ g) U y
  - [] (y -> X(r \/ b))



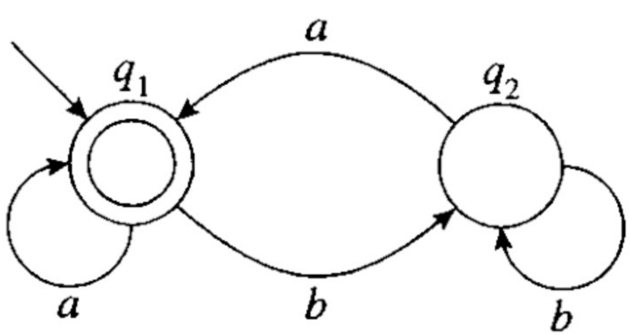
Linear Temporal Logic

- r, r /\ g, r \/ o
- X o, X r, X g
- X (o \/ r \/ g)
- X o -> XX r
- X o -> XXX g
- <> r, <> g, <> o
- [] (g -> X (o \/ r))
- [] ((r /\ ~g) -> X g)
- []<> o, []<> r, []<> g
- []<> (o \/ r)



Model Checking

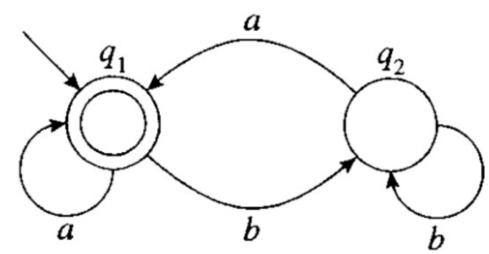
- Finite Automata (FA)
  - Accept or reject a finite string
  - Regular expressions can be converted to an FA



$\epsilon \mid (a|b)^* a$

Model Checking

- A finite automaton A is a quintuple $A = (\Sigma, Q, \Delta, Q^0, F)$
  - Σ is the finite alphabet
  - Q is the finite set of states
  - $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation
  - $Q^0 \subseteq Q$ is the set of initial states
  - $F \subseteq Q$ is the set of final states



- A finite string s is accepted by A iff there is a run for s that ends with a state in F

Model Checking

- Büchi automata
  - A Büchi automaton B has the same representation as a finite automaton B = (Σ,Q,Δ,Q^0,F)
  - An infinite string s is accepted by B iff a state in F appears infinitely often in a run
- An LTL formula can be converted to a Büchi automaton

Model Checking

- LTL Model Checking
  - Build a Büchi automaton $B_{\sim f}$ for the negation of a given specification f
  - Build an intersection automaton $I_{\sim f}$ between a Kripke structure model and $B_{\sim f}$
  - If $I_{\sim f}$ accepts a string, the string is a counterexample witnessing the violation of the specification.

Links to Lecture Slides

- http://www3.cs.stonybrook.edu/~youngkwon/cse595/ModelChecking1.pdf
- http://www3.cs.stonybrook.edu/~youngkwon/cse595/ModelChecking2.pdf