Linear Temporal Logic

Presented by

Kevin Browne
Fei Zhao

November 19, 2007

Introduction to Temporal Logic

- Why Temporal Logic?
  - Formulae statically true or false for a given model in classical propositional, predicate logic
  - Most systems are dynamic
  - Property verification for concurrent, reactive systems

- Temporal Logic
  - Formulae are not statically true or false in a model
  - Models are transitions systems
  - Dynamic notion of truth

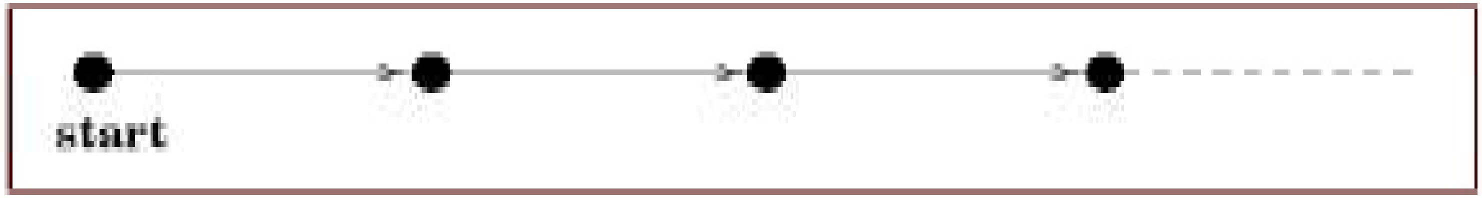
Linear Temporal Logic

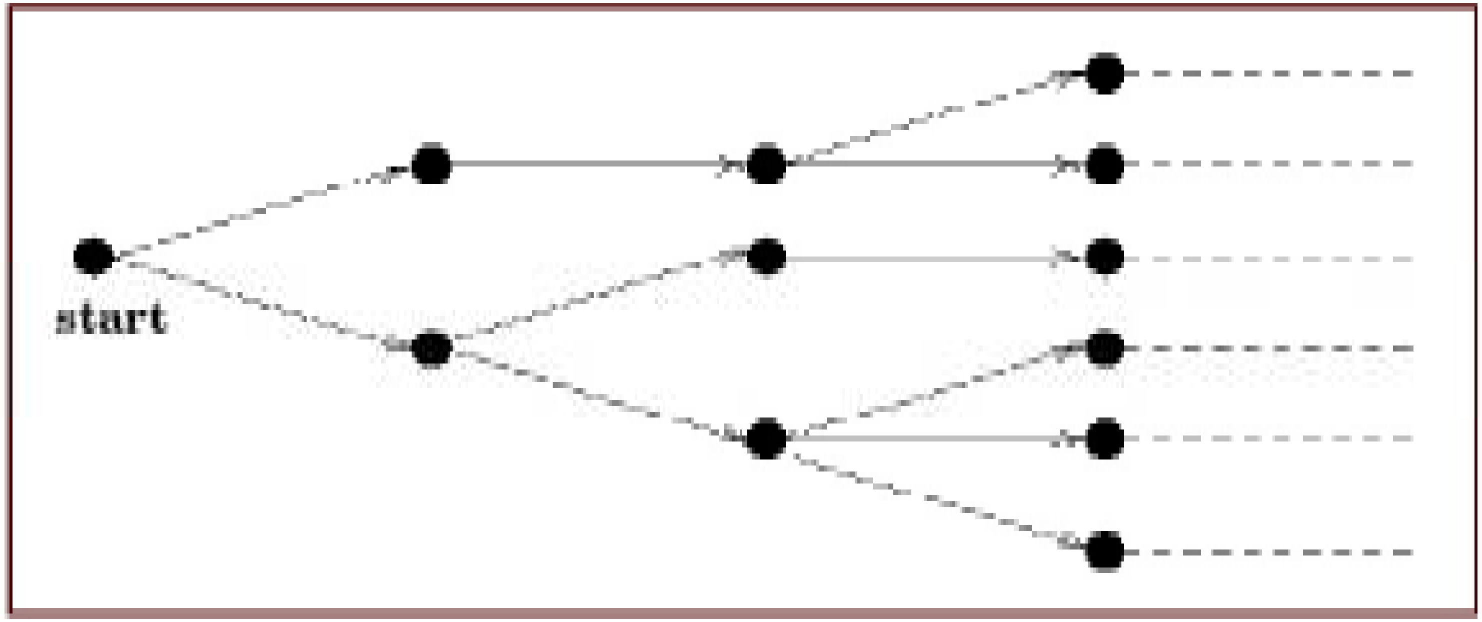
- Timeline is the underlying structure of time in Linear Temporal Logic
- We assume time in LTL is isomorphic to the natural numbers
- Under this assumption, time in LTL:
  - Is discrete
  - Has an initial moment with no predecessors
  - Is infinite into the future
- Timeline is a set of paths: $t_0\ t_1\ t_2 \ldots$

Linear vs. Branching

- Linear-time Temporal Logic
  - Time as a set of paths
  - Each path is a sequence of moments
  - At each moment, only one possible next future moment
- Computational Tree Logic (Branching)
  - Time as a tree
  - Root as present moment
  - Branches out into the future

Linear vs. Branching (Intuition)





Linear vs. Branching (Alessandro Artable, "Formal Methods")

Syntax

The rules for generating Linear Temporal Logic (LTL):

- Each φ is a formula
- If φ and ψ are formulae then $\neg \varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ are formulae
- If φ and ψ are formulae then $\mathbf{X}\, \varphi$, $\mathbf{F}\, \varphi$, $\mathbf{G}\, \varphi$, $\varphi\, \mathbf{U}\, \psi$, $\varphi\, \mathbf{W}\, \psi$, $\varphi\, \mathbf{R}\, \psi$

Syntax (cont'd)

Defined in Backus Naur Form:

$$\varphi ::= p \mid \text{True} \mid \text{False}$$
$$\mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi$$
$$\mid \mathbf{X}\,\varphi \mid \mathbf{F}\,\varphi \mid \mathbf{G}\,\varphi \mid \varphi\,\mathbf{U}\,\psi \mid \varphi\,\mathbf{W}\,\psi$$
$$\mid \varphi\,\mathbf{R}\,\psi$$

Atomic propositions: **p** $\in$ Atoms

Boolean operators: $\wedge$ $\vee$ $\neg$ $\rightarrow$

Temporal operators: **X G F U W R**

Well Formed Formulae

The well formed formulae in LTL follow the same rules as well formed formulae in classical propositional logic, plus the rules for temporal operators.

The following are **NOT** well formed formulae:

- **U** r – since **U** is a binary operator, not unary
- p **G** q – since **G** is a unary operator, not binary

Syntax Examples

- $\mathbf{F}\, p \wedge \mathbf{G}\, q \rightarrow p\, \mathbf{W}\, r$
- $\mathbf{F}\, (p \rightarrow \mathbf{G}\, r) \vee \neg\, q\, \mathbf{U}\, p$
- $p\, \mathbf{W}\, (q\, \mathbf{W}\, r)$
- $\mathbf{G}\, \mathbf{F}\, p \rightarrow \mathbf{F}(q \vee s)$

More Practical Syntax Examples

- $((x = 0) \wedge (x + 3)) \rightarrow \mathbf{X}\ (x = 3)$
- lottery-win → **G** rich
- send → **F** receive
- start-lecture → talk **U** end-lecture
- born → alive **U** dead

Transition System

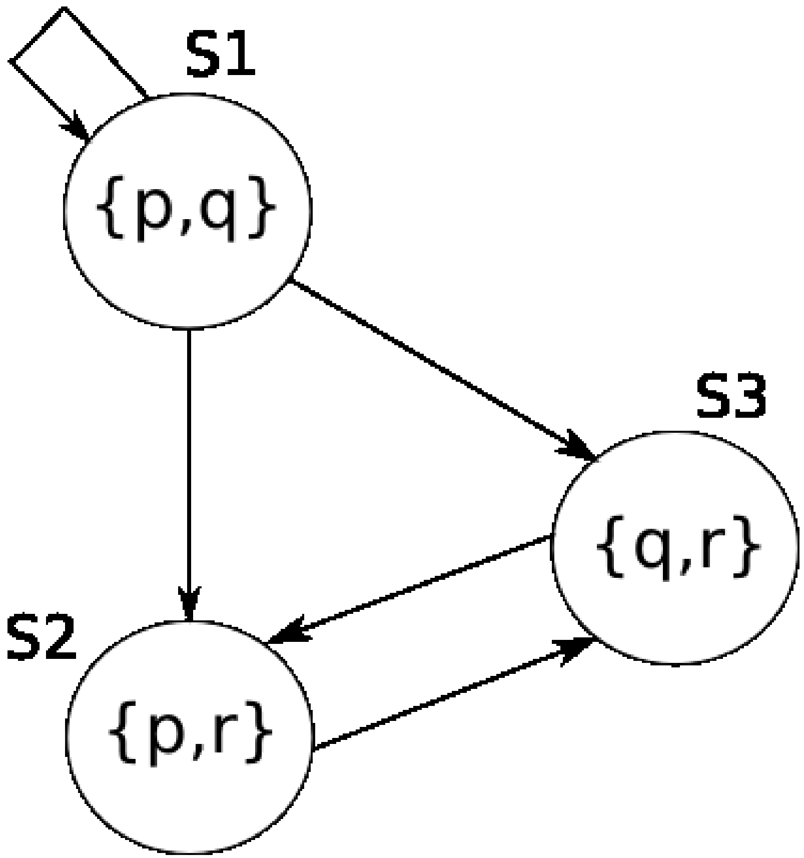
Timeline as a linear time structure **M=(S, →,L)** where

- S is a set of states,
- → is a transition relation such that every s∈S has some r∈S with s→r
- L : S → PowerSet (Atoms) is a labeling of each state with a set of atomic propositions in Atoms.

Semantics are given with respect to a path $\boldsymbol{\pi = s_1\ s_2\ s_3 \ldots}$

Suffix of trace starting at s_i is defined as $\boldsymbol{\pi_i = s_i\ s_{i+1}\ s_{i+2} \ldots}$

Transition System Example



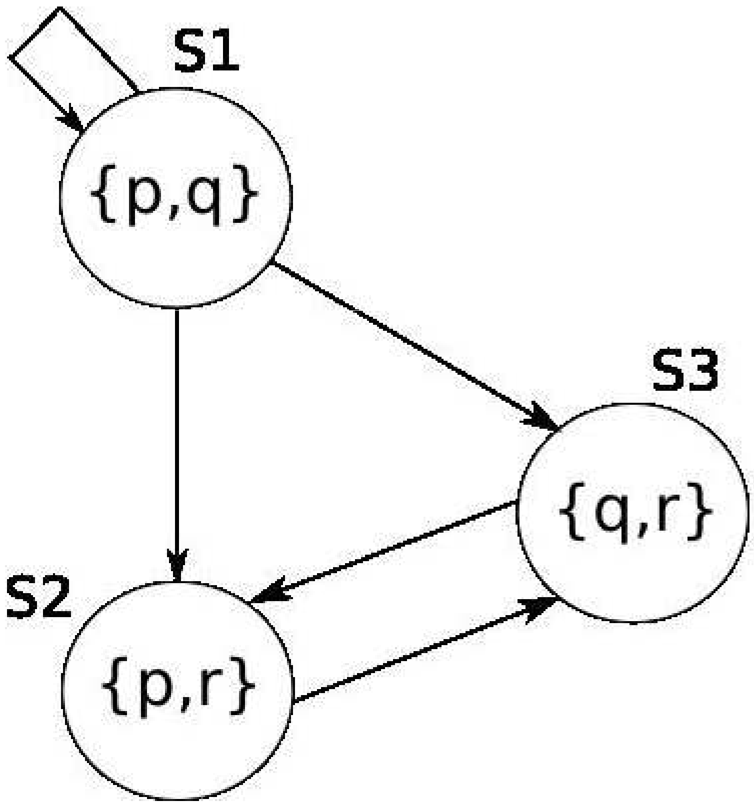
M = (S,→,L) is as follows:

S = {S1,S2,S3}

→ = {(S1,S1),(S1,S3),
(S1,S2),(S2,S3),
(S3,S2)}

L(S1) = {p,q}
L(S2) = {p,r}
L(S3) = {q,r}

Transition System Example



The set of paths is limited by what we can construct from the given states and transitions. So

π = S1,S1,S1,S1,S1,S1,S1,….

and

π = S1,S1,S3,S2,S3,S2,….

are in M (read: can be defined by it). But

π = S1,S1,S3,S2,**S1,S1**….

is not!

Semantics

- $\pi \models \text{True}$
- not $\pi \models \text{False}$
- $\pi \models p$ iff $p \in L(s_1)$ $(\pi = s_1\, s_2\, s_2 \ldots)$
- $\pi \models \neg\varphi$ iff not $\pi \models \varphi$
- $\pi \models \varphi \wedge \psi$ iff $\pi \models \varphi$ and $\pi \models \psi$
- $\pi \models \varphi \vee \psi$ iff $\pi \models \varphi$ or $\pi \models \psi$
- $\pi \models \varphi \rightarrow \psi$ iff $\pi \models \varphi$ if $\pi \models \psi$
- $\pi \models X\, \varphi$ iff $\pi_2 \models \varphi$

  (holds iff φ holds at the next state)
- $\pi \models F\, \varphi$ iff $\exists i \geq 1\ \pi_i \models \varphi$

  (at some future state φ is true)

Semantics (cont'd)

- $\pi \models G\,\varphi$ iff $\forall i \geq 1\ \pi_i \models \varphi$

  (at all the future states, φ is true)
- $\pi \models \varphi\,U\,\psi$ iff $\exists i \geq 1\ \pi_i \models \psi$ and $\forall\, j < i\ \pi_j \models \varphi$

  (φ is true until ψ is true)
- $\pi \models \varphi\,W\,\psi$ iff $(\exists i \geq 1\ \pi_i \models \psi$ and $\forall\, j < i\ \pi_j \models \varphi)$

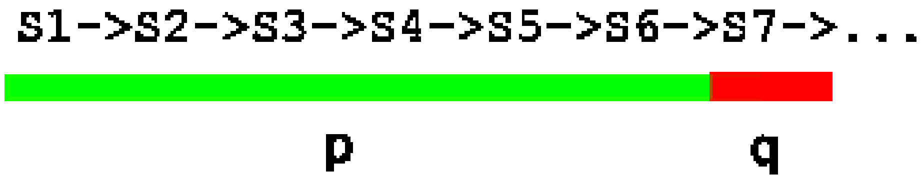
  or $\forall\, k \geq 1\ \pi_k \models \varphi$

  (φ is true until ψ is true, or φ is always true)
- $\pi \models \varphi\,R\,\psi$ iff $(\exists i \geq 1\ \pi_i \models \varphi$ and $\forall j<i\ \ \pi_j \models \psi)$

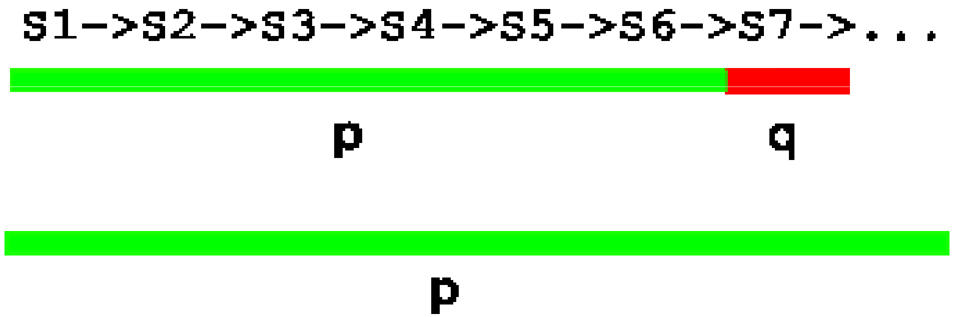
  or $\forall\, k \geq 1\ \pi_k \models \psi)$

  (ψ is true until φ is true, or ψ is always true)

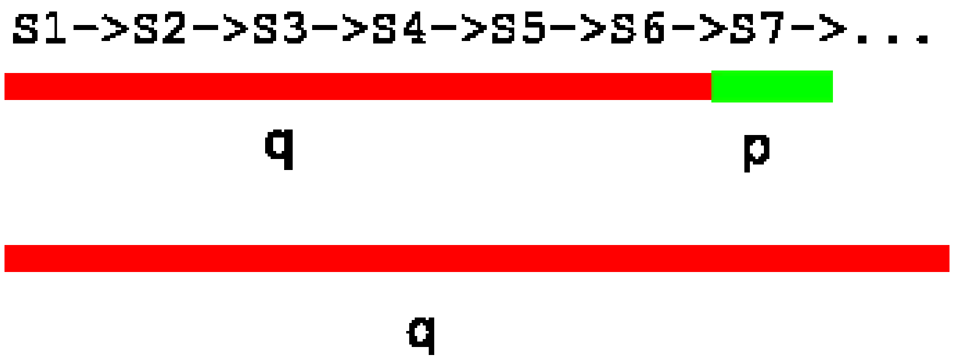
Semantics (Intuition)



p **U** q (p until q)

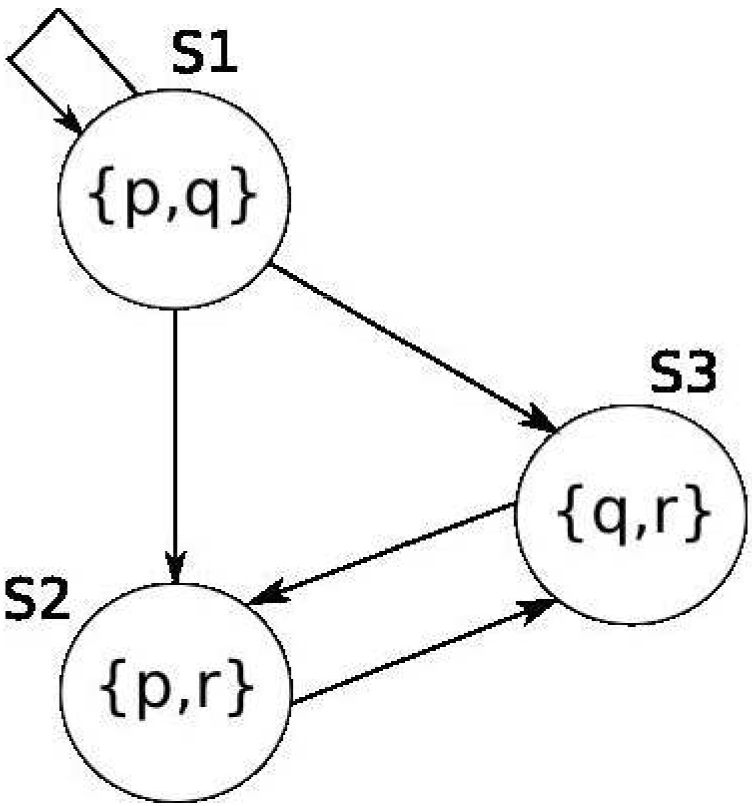


p **W** q (p weak until q)



p **R** q (p release q)

Semantic Notion Applied



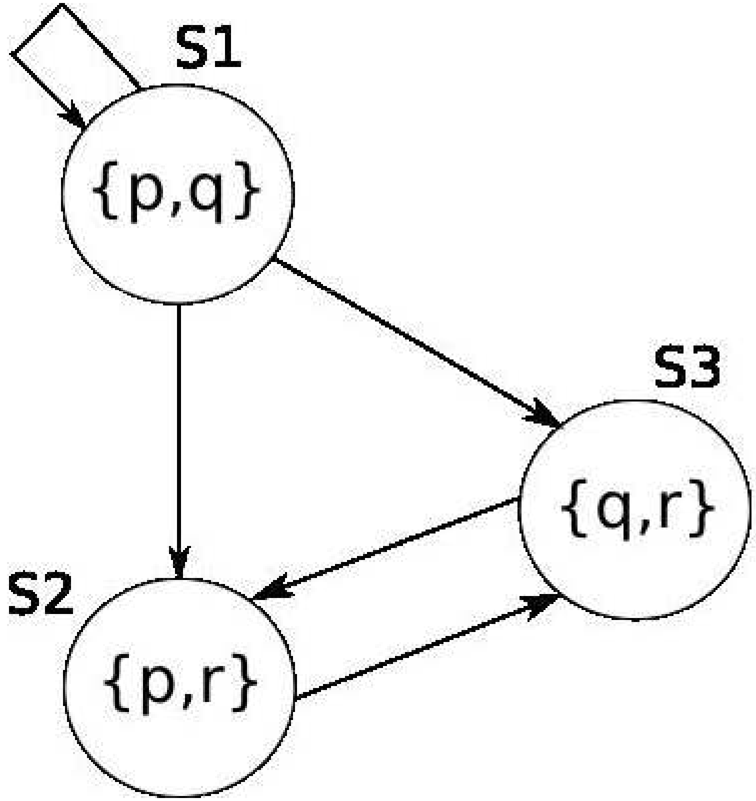
From this model, we can derive different (infinite) paths:

π = S1,S1,S1,S1,S1,S1,S1,….

π = S1,S1,S2,S3,S2,S3,,…

π = S1,S1,S3,S2,S3,S2,…

Semantic Notion Applied (Cont'd)



Given a path π, and a formula φ, we can now evaluate the truth of that formula.

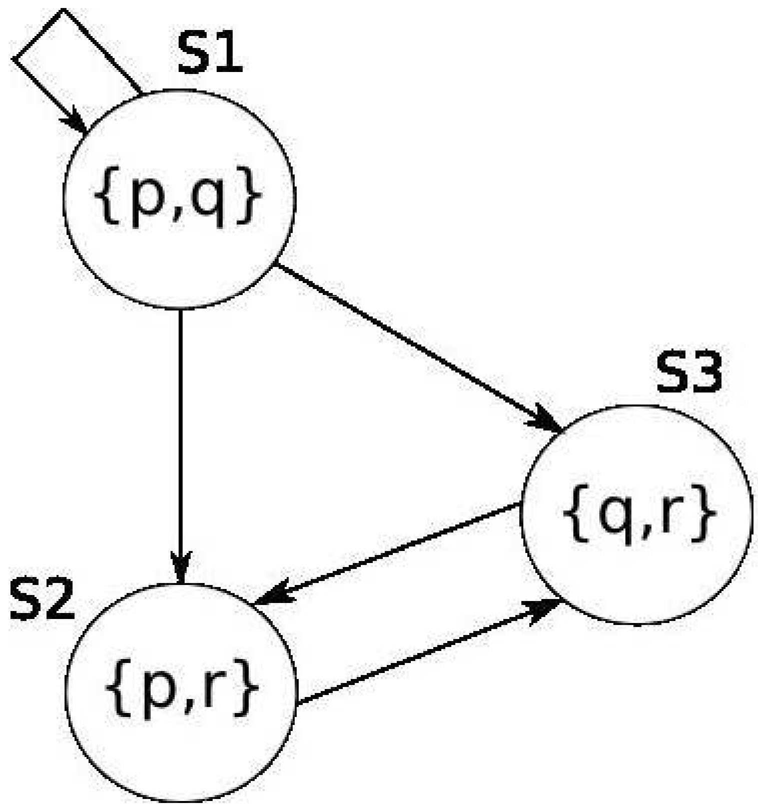
So for:

π = S1,S1,S1,S1,S1,S1,S1,….

π |= p is true

by the rule that "π |= p iff p ∈ L(s_1)".

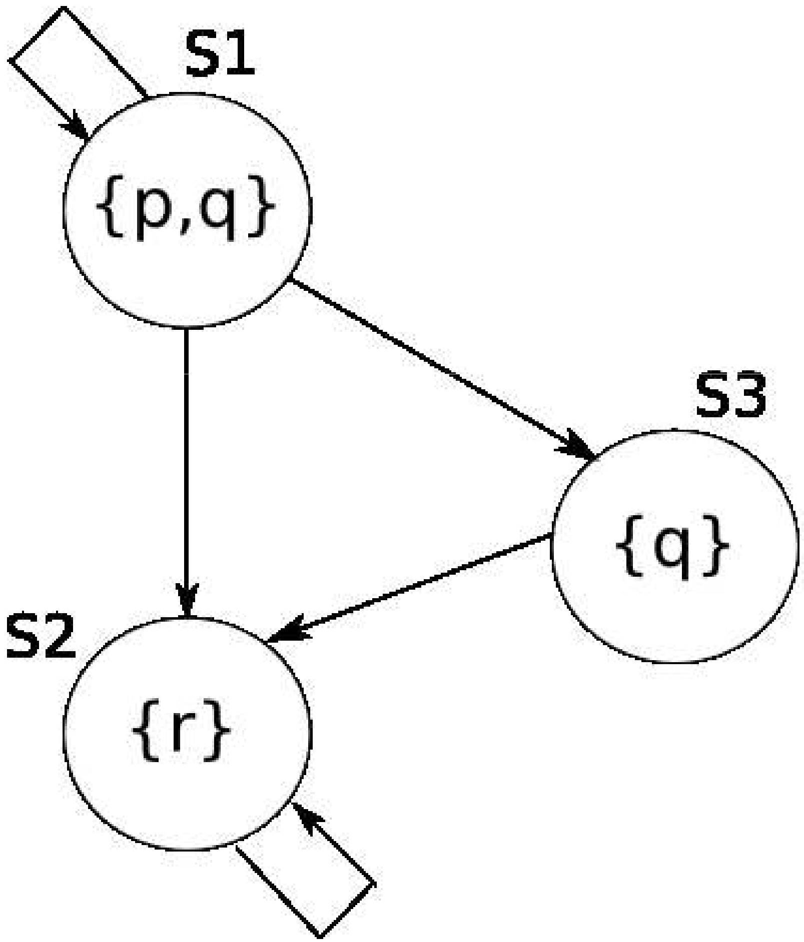
Semantics Example



Given the model to the left, we can then say for
π = S1,S1,S3,S2,S3,S2,S3,…

$\pi \models p$ is true
$\pi \models q$ is true
$\pi \models p \wedge q$ is true
$\pi \models r$ is false
$\pi \models \mathbf{F}\ p$ is true
$\pi \models \mathbf{X}\ p$ is true
$\pi \models \mathbf{X}\ (\mathbf{X}\ p)$ is false
$\pi \models \mathbf{G}\ r$ is false
$\pi \models \mathbf{G}\ (\mathbf{X}\ (\mathbf{X}\ r))$ is true
$\pi \models p\ \mathbf{U}\ r$ is true

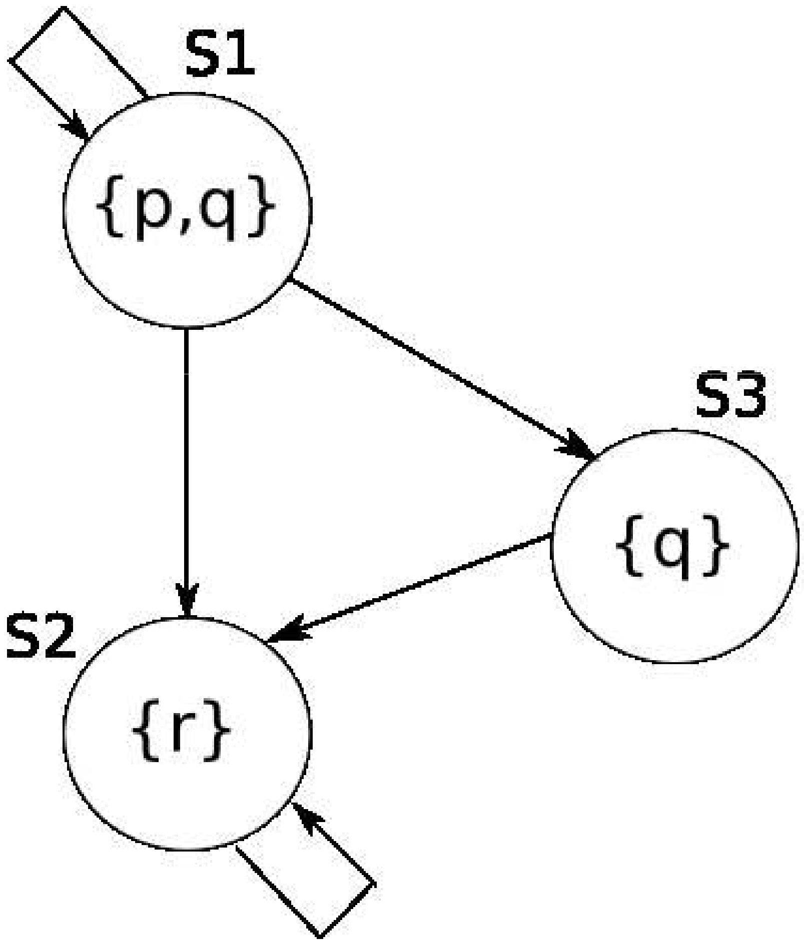
Another Semantics Example



Given the model to the left, we can then say for
π = S1,S1,S3,S2,S2,….

π |= p is true
π |= **X** p is true
π |= **X** (**X** p) is false
π |= p **U** r is false
π |= q **U** r is true
π |= q **W** r is true
π |= r **R** q is true
π |= **F** p is true
π |= **F** (**X** (**X** p)) is false

Same Model, Different Paths



Given the model to the left, we can then say for
π = S1,S1,S3,S2,S2,….

π |= p **U** r is **FALSE**

But for:

π = S1,S2,S2,S2,….

π |= p **U** r is **TRUE**

Truth is no longer static for a given model. Different paths may evaluate differently for the same formula!

Further Definitions

Entailment: f |= y iff $\forall M, \forall i \in N. (M, \pi_i) \models f \Rightarrow (M, \pi_i) \models y$

Equivalence: f ≡ y iff $\forall M, \forall i \in N. (M, \pi_i) \models f \Leftrightarrow (M, \pi_i) \models y$

Satisfiable: An LTL formula φ is satisfiable iff there exists a linear time structure **M = (S,→,L)** such that **M,** π **|= φ.** Any such structure defines a model of φ.

Valid: A formula φ is valid iff for all linear time structures **M = (S, →,L)**, we have **M,** π **|= φ ,** and write **|= φ .**

Examples:

Some significant validities

- $\models \mathbf{G} \neg p \equiv \neg \mathbf{F} p$
- $\models \mathbf{F} \neg p \equiv \neg \mathbf{G} p$
- $\models \mathbf{X} \neg p \equiv \neg \mathbf{X} p$

Satisfiable or valid

- $p \rightarrow \mathbf{F} q$

  satisfiable formula but not valid
- $\mathbf{G} (p \rightarrow \mathbf{F} q) \rightarrow (p \rightarrow \mathbf{F} q)$

  valid formula

Conclusion

- Linear Temporal Logic is a useful and accessible framework for modeling systems which involve changes occurring over time!
- Questions?

Bibliography

Logic in Computer Science, Michael Huth and Mark Ryan

Handbook of Theoretical Computer Science (Temporal and Modal Logic, Ch. 16), E. Allen Emerson

Formal Methods (Lecture 3: Linear Temporal Logic), Allesandro Artale