We will wait 10 minutes until 10:40 AM for all students to join into the meeting.

We will start the tutorial at **10:40 AM**.

CS 3SD3 - Concurrent Systems
Tutorial 9

Mahdee Jodayree

McMaster University

Outline

- Announcements
- Review of lecture 15, 16 (Ambiguous slides and examples)
  - Meaning of Constructs and Quantifiers
  - (LTL)Linear Temporal Logic
  - The difference between **Propositional logic** and **temporal logic**
    - Example of temporal logic
  - CTL (Computational Tree Logic)
    - CTL Example

Announcements

- Mid-terms are marked.
- Double-check your mark in the grade section of the avenue.

Constructs and Quantifiers (Lecture 16)

∀ For all

∃ There exists

E There exists

∧ and

∨ or

¬ Not (negation)

Linear Temporal Logic (LTL)

- In logic, linear temporal logic or linear-time temporal logic (LTL) is a modal temporal logic with **modalities referring to time**.
- In LTL, one can encode and formulate about the future of paths,
  - e.g., a condition will **eventually be true**,
  - or condition will be **true until another fact becomes true**, etc.
- It is a fragment of the more complex CTL*, which additionally allows branching time and quantifiers.
- Subsequently LTL is sometimes called propositional temporal logic, abbreviated PTL.[3]
- Linear temporal logic (LTL) is a **fragment** of first-order logic.

What are the differences between propositional logic and temporal logic?

- Propositional logic consists of assigning a **truth value vs** LTL consists of assigning a truth value to every variable at **each point in time**
- Temporal logic is more precise because it encapsulates more precise meaning symbols, such as ***until***, ***eventually***, ***next.***
- Temporal logic extends classical propositional logic with a set of temporal operators.
- Propositional logic is about variables having a **definite truth value**, vs
  - whereas LTL is about variables having a truth value **depending on time.**

Examples Of Temporal Logic

Temporal logic is any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time.

"I am always hungry",

"I will eventually be hungry",

"I will be hungry until I eat something".

It is sometimes also used to refer to **tense logic**

LTL Syntax

$\Phi ::= \bot \mid \top \mid p \mid (\neg\Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \Rightarrow \Phi) \mid$
$(G\Phi) \mid (F\Phi) \mid (X\Phi) \mid (\Phi\ U\ \Phi) \mid (\Phi\ W\ \Phi) \mid (\Phi\ R\ \Phi)$

where p ranges over atomic formulas/descriptions.

- $\bot$ - false, $\top$ - true
- $G\Phi, F\Phi, X\Phi, \Phi\ U\ \Phi, \Phi\ W\ \Phi, \Phi\ R\ \Phi$ are **temporal connections**.
- X means "neXt moment in time"
- F means "some Future moments"
- G means "all future moments (Globally)"
- U means "Until"
- W means "Weak-until"
- R means "Release"
- An LTL formula is evaluated on a path, or a set of paths.
- A set of paths satisfies Φ if every path in the set satisfies Φ.
- Consider the path $\pi \overset{\mathrm{df}}{=} s_1 \rightarrow s_2 \rightarrow \ldots$.
  We write π^i for the suffix starting at s_i, i.e. π^i is $s_i \rightarrow s_{i+1} \rightarrow s_{i+2} \rightarrow \ldots$.

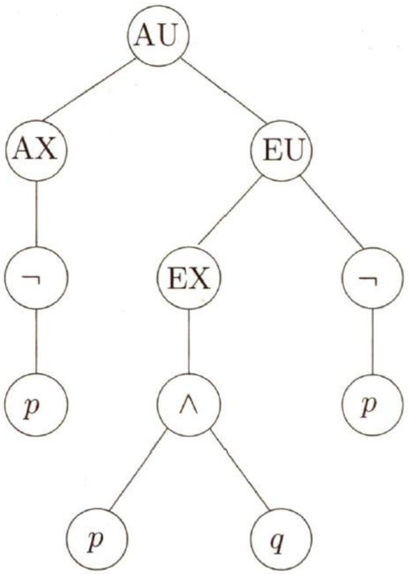
Computational Tree Logic (CTL)

- CTL is a branching-time logic, (meaning that)
- Its model of time is a tree-like structure in which the **future** is not **determined**;
- There are **different paths** in the future, anyone of which might be the `actual' path that is realized.
- We work with a fixed set of atomic formulas/description

$$(p, q, r, \ldots, \text{ or } p_1, p_2, \ldots)$$

- These atoms stand for atomic descriptions of a system, like:
  - *the printer is busy*
  - *there are currently no requested jobs for the printer*
  - *the current content of register R1 is the integer 6*

CTL Example: Parsing Trees



$A[AX\neg p \; U \; E[EX(p \wedge q) \; U \; \neg p]]$

A subformula of a CTL formula Φ is any formula Ψ whose parse tree is a subtree of Φ's parse tree.

CTL Syntax

$$\Phi ::= \bot \mid \top \mid p \mid (\neg\Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \Rightarrow \Phi) \mid$$
$$AX\Phi \mid EX\Phi \mid A[\Phi U \Phi] \mid E[\Phi U \Phi] \mid$$
$$AG\Phi \mid EG\Phi \mid AF\Phi \mid EF\Phi$$

where p ranges over atomic formulas/descriptions.

- $\bot$ - false, $\top$ - true
- $\underbrace{AX, EX, AG, EG, AU, EU, AF, EF}_{\text{all pairs, each starts with either } A \text{ or } E}$ are **temporal connections**.
- A means "along All paths" (inevitably)
- E means "along at least (there Exists) one path" (possibly)
- X means "neXt state"
- F means "some Future state"
- G means "all future states (Globally)"
- U means "Until"
- X, F, G, U cannot occur without being preceded by A or E.
- every A or E must have one of X, F, G, U to accompany it.

Examples of CTL Formulas

- An upwards traveling elevator at the second floor does not change its direction when it has passengers wishing to go to the fifth floor:

$$AG(floor = 2 \wedge direction = up \wedge ButtonPressed5 \Rightarrow A[direction = up \ U \ floor = 5])$$

- The elevator can remain idle on the third floor with its doors closed:

$$AG((floor = 3 \wedge idle \wedge door = closed) \Rightarrow EG(floor = 3 \wedge idle \wedge door = closed))$$

CTL Syntax

$$\Phi ::= \bot \mid \top \mid p \mid (\neg\Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \Rightarrow \Phi) \mid AX\Phi \mid EX\Phi \mid A[\Phi U \Phi] \mid E[\Phi U \Phi] \mid AG\Phi \mid EG\Phi \mid AF\Phi \mid EF\Phi$$

where p ranges over atomic formulas/descriptions.

- $\bot$ - false, $\top$ - true
- $\underbrace{AX, EX, AG, EG, AU, EU, AF, EF}_{\text{all pairs, each starts with either } A \text{ or } E}$ are **temporal connections**.
- A means "along All paths" (inevitably)
- E means "along at least (there Exists) one path" (possibly)
- X means "neXt state"
- F means "some Future state"
- G means "all future states (Globally)"
- U means "Until"
- X, F, G, U cannot occur without being preceded by A or E.
- every A or E must have one of X, F, G, U to accompany it.

Practical Patterns of Specifications (1)

What kind of practically relevant properties can we check with formulas of CTL?

Suppose atomic descriptions include some words as busy, requested, ready, etc.

- It is possible to get a state where started holds but ready does not hold:

$$EF(started \wedge \neg ready)$$

- For any state, if a request (of some resource) occurs, then it will eventually be acknowledged:

$$AG(requested \Rightarrow AF\ acknowledged)$$

- A certain process is enabled infinitely often on every computation path:

$$AG(AF\ enabled)$$

- Whatever happens, a certain process will eventually be permanently deadlocked:

$$AF(AG\ deadlock)$$

CTL examples

Examples: Let

P mean "I like chocolate"

Q mean "It's warm outside."

AG= along **all** paths in **all future** states (globally)

EF= along **at least** one path, in some future state

AF= along **all** paths , in some future state

EG= along **at least** one path, in **all future** states(globally

- **AG**.P

  "I will like chocolate from now on, no matter what happens."
- **EF**.P

  "It's possible I may like chocolate some day, at least for one day."
- **AF**.**EG**.P

  "It's always possible (AF) that I will suddenly start liking chocolate for the rest of time." (Note: not just the rest of my life, since my life is finite, while **G** is infinite).

CTL examples

EG= along **at least** one path, in **all** future states (globally)
AF= along **all** paths , in **some** future state

•**EG**.**AF**.P

"Depending on what happens in the future (E), it's possible that for the rest of time (G), I'll be guaranteed at least one (AF) chocolate-liking still ahead of me.

However, if something ever goes wrong, then all bets are off and there's no guarantee about whether I'll ever like chocolate."

The two following examples show the difference between CTL and CTL*, as they allow for the until operator to not be qualified with any path operator (**A** or **E**):

AG= along **all** paths in **all future** states(globally)
EF= along **at least** one path, in some future state
AF= along **all** paths , in some future state
EG= along **at least** one path, in **all future** states(globally)
U = until
X = next state
EX = along **at least** (there exist) one path, next state
Q mean "It's warm outside.“

- **AG**(P**U**Q)

  "From now until it's warm outside, I will like chocolate every single day. Once it's warm outside, all bets are off as to whether I'll like chocolate anymore. Oh, and it's guaranteed to be warm outside eventually, even if only for a single day."
- **EF**((**EX**.P)**U**(**AG**.Q))

  "It's possible that: there will eventually come a time when it will be warm forever (AG.Q) and that before that time there will always be *some* way to get me to like chocolate the next day (EX.P)."

Equivalences

- Two CTL formulas Φ and Ψ are said to be **semantically equivalent** if any state in any model which satisfies one of them also satisfies the other, write then: $\Phi \equiv \Psi$.
- $$\left.\begin{array}{l} \neg AF\Phi \equiv EG\neg\Phi \\ \neg EF\Phi \equiv AG\neg\Phi \end{array}\right\} \text{de Morgan rules}$$
- $$\neg AX\Phi \equiv EX\neg\Phi$$
- $$AF\Phi \equiv A[\top \; U \; \Phi]$$
  $$EF\Phi \equiv E[\top \; U \; \Phi]$$

AG= along **all** paths in **all future** states(globally)
EF= along **at least** one path, in some future state
AF= along **all** paths , in some future state
EG= along **at least** one path, in **all future** states(globally)
U = until
X = next state
EX = along **at least** (there exist) one path, next state

Any Questions?