An Overview of Linear Temporal Logic

Sruti Goswami

Introduction

Formal verification methods are used to verify the correctness of a computer program. Here the program is modelled. The required properties are specified. Different verification procedures are given to check whether the model satisfies the property. Model checking is one such verification procedure which is based on temporal logic. In temporal logic, a formula need not be true throughout the model. In the model-checking approach, the models are known as transition systems and the formula is a formula in temporal logic. The temporal logic where time is linear is called Linear Temporal Logic.

A transition system is a model used to represent the behaviour of a system that changes over time. There are four components to a transition system. States, actions. transitions and initial state.

Syntax

AP be a finite set of atomic propositions. $\neg$, $\vee$ are logical operators and X, U be temporal operators, then the Backus Naur form of syntax is given by,

$$\phi :\equiv true|p|\phi_1 \vee \phi_2|\neg\phi|X\phi|\phi_1 U \phi_2$$

Here X stands for next and U stands for until operators. From the above operators, we can derive some classical logic operators as well as some new transition operators. They logical operators are $\wedge$, $\rightarrow$, $\leftrightarrow$. The derived temporal operators are F (future), G (global), R(release), W(weak until), M (mighty release).

Semantics

AP be a finite set of atoms. A property is a set of infinite words over the alphabet, the power set of AP. Every letter in that word will be some set of atomic propositions.

Definition 0.1 (Transition System) *A transition system is denoted by $\mu = (S, \leftarrow, L, S_0)$ where S is the set of states, $\rightarrow$ is a binary relation on S such that for all $s \in S$, $\exists s' \in S$ such that $s \rightarrow s'$. $L : S \rightarrow 2^{AP}$ is the labelling function. S_0 is the set of initial states.*

The transition system is said to satisfy a property P if the traces of that system are contained in P.

Let $w = a_0 a_1 ...$ be an word over AP. Let $w_i = a_i$ and $w^i = a_i a_{i+1} ....$ A formula ϕ is said to satisfy the word w if

1. $w \models p$ if $p \in w(0)$
2. $w \models \neg\phi$ if $w \not\models \phi$

3. $w \models \phi \vee \psi$ if $w \models \phi$ or $w \models \psi$
4. $w \models X\phi$ if $w^1 \models \phi$
5. $w \models \phi U \psi$ if there exist $i \geq 0$ such that $w^i \models \psi$ and for all $0 \leq k < i, w^k \models \psi$

A formula ϕ is satisfiable if there is a word w such that $w \models \phi$. A formula ϕ is valid if for all word $w \in 2^{AP}$, $w \models \phi$. The additional logical operators are defined as follows:

1. $\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$
2. $\phi \rightarrow \psi \equiv \neg\phi \vee \psi$
3. $\phi \leftrightarrow \psi$ if $\phi \rightarrow \psi \wedge \psi \rightarrow \phi$
4. $false \equiv \neg true$

The additional transitional operators are defined as follows:

1. $F\phi \equiv trueU\phi$
2. $G\phi \equiv \neg F\neg\phi$
3. $\phi R\psi \equiv \neg(\neg\phi U\neg\psi)$

Equivalences of Formulas

Two LTL formulas are said to be equivalent if the words satisfying both formulas are the same. For example

Distributivity:

1. $X(\phi \vee \psi) \equiv X\phi \vee X\psi$
2. $X(\phi \wedge \psi) \equiv X\phi \wedge X\psi$
3. $X(\phi U\psi) \equiv X\phi UX\psi$
4. $F(\phi \vee \psi) \equiv F\phi \vee F\psi$
5. $G(\phi \wedge \psi) \equiv G\phi \wedge G\psi$
6. $\alpha U(\beta \vee \gamma) \equiv (\alpha U\beta) \vee (\alpha U\gamma)$
7. $(\beta \wedge \gamma)U\alpha \equiv (\beta U\alpha) \wedge (\gamma U\alpha)$

Negations and dualities:

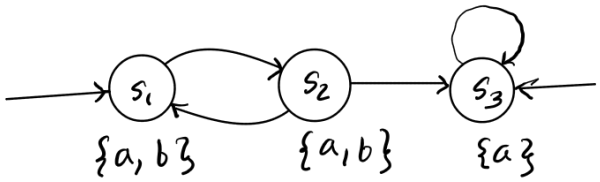
1. $\neg X\phi \equiv X\neg\phi$ (i.e. X is dual to itself).
2. $\neg F\phi \equiv G\neg\phi$ or $\neg G\phi \equiv F\neg\phi$ (i.e. F is dual to G).
3. $\neg(\phi U\psi) \equiv (\neg\phi R\neg\psi)$ or $\neg(\phi R\psi) \equiv (\neg\phi U\neg\psi)$ (i.e. U is dual to R).

Temporal equivalences:

1. $F\phi \equiv FF\phi$, $G\phi \equiv GG\phi$
2. $\phi U\psi \equiv \phi U(\phi U\psi)$
3. $G\phi \equiv \phi \wedge X(G\phi)$, $F\phi \equiv \phi \vee X(F\phi)$
4. $\phi U\psi \equiv \psi \vee (\phi \wedge X(\phi U\psi))$

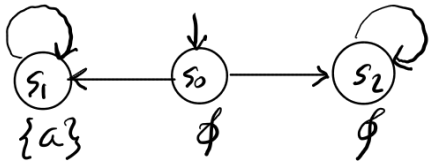
Examples

1. Consider the following transition system over the set of atomic propositions $\{a, b\}$.



   - The above system TS satisfies Ga because all the states s_1, s_2, s_3 satisfoes a.
   - $s_1 \models X(a \wedge b)$. Because $s_2 \models a \wedge b$. $s_2, s_3 \not\models X(a \wedge b)$. So $TS \not\models X(a \wedge b)$. Because s_3 is successor to both s_2, s_3 but $s_3 \not\models a \wedge b$.
   - $TS \models \neg G(\neg b \rightarrow G(a \wedge \neg b))$. $s_3 \models \neg b$ and also $s_3 \models a \wedge \neg b$.
   - $TS \models bU(a \wedge \neg b)$. since there is a path $\{s_1, s_2\}^\omega$ that does not satisfy $a \wedge \neg b$. Note that the path $\{s_1 s_2\} s_3 *^\omega$ satisfies $bU(a \wedge \neg b)$.

2. Again consider the following transition system with $AP = \{a\}$. There is a LTL formula



   $\phi = Fa$ such that $TS \not\models \phi$ and $TS \not\models \neg\phi$. The initial path $s_0(s_2)^\omega \not\models Fa$. But the initial path $s_0(s_1)^\omega \models Fa$ ie.$s_0(s_2)^\omega \not\models \neg Fa$.

Formal Verification Properties

Safety property:

A property P (a set of words) over AP is said to be a safety property if there exists a set of bad prefixes such that P is the set of all words which do not start with a bad prefix. This is used to verify if something bad never happens. Safety properties can be expressed as the LTL formula

$$G\neg\phi$$

Liveness property:

Suppose a property P holds in some future state. If that holds for every state in the transition system then this property is called liveness property. Liveness property as LTL formula is given by

$$GF\phi$$

LTL Based Model Checking

For a transition system to satisfy a formula (set of infinite word) ϕ, all the traces of the system satisfy ϕ. Alternatively, if there exists a trace that satisfies $\neg\phi$, we can say that $TS \models \neg\phi$. We consider the following steps.

- We convert the LTL formula $\neg\phi$ to a non deterministic Buchi(NBA) automaton $A_{\neg\phi}$. There is a set of infinite words that satisfy the formula. We construct an automaton corresponding to the formula $\neg\phi$ whose language is exactly equal to the words of $\neg\phi = \psi$. We look for all the subformulas of the formula ψ, look at the properties of the formula expansions and check the and-not and until compatibilities. Using these compatibility criteria, we remove the incompatible states and also add a single initial state. We now use word compatibility, X compatibility and U compatibility to define transitions among the states. Now using the until eventuality condition we define the accepting states. If the formula is not in the form of $\alpha U \beta$, we make all states to be accepting. This will result in a generalised NBA. A run is accepted if visits accepting states infinitely often. Every GNBA can be converted into an NBA.
- We convert the transition system TS to a non deterministic Buchi automaton A_{TS}.
- $L(A)$ be the language of the automaton A. We check whether $L(A_{\neg\phi}) \cap L(A_{TS})$ is empty using the nested DFS algorithm.

However, this process results in an NBA of exponential size.

Reference

1. Logic in Computer Science: Modelling and Reasoning about Systems by Mark D. Ryan and Michael Huth.
2. Principles of Model Checking by Joost-Pieter Katoen.
3. Model Checking - Course by B. Srivatsan.
4. Linear temporal logic - Wikipedia article