Lecture 6 - January 26

Model Checking

Introduction
Linear-time Temporal Logic (LTL): Syntax

Announcement

- Lab1 Part 2 tutorial videos released
  + Help: Scheduled Office Hours & flexible TA hours
  + ≈ 2 hours
    * debugging using labels, error trace, state graph
    * PlusCal vs. Auto-Translated TLA+ Predicates
- Optional Textbook for Model Checking and Program Verification
  Logic in Computer Science:
  Modelling and reasoning about systems
  by M. Huth and M. Ryan

Use of Model Checking in Industry

Pentium FDIV bug: https://en.wikipedia.org/wiki/Pentium_FDIV_bug

*The Pentium FDIV bug is a hardware bug affecting the **floating-point unit (FPU)** of the early Intel Pentium processors. Because of the bug, the processor would return incorrect binary floating point results when dividing certain pairs of high-precision numbers.*

*In December 1994, Intel **recalled** the defective processors ... In its 1994 annual report, Intel said it incurred "**a $475 million pre-tax charge** ... to recover replacement and write-off of these microprocessors."*

*In the aftermath of the **bug** and subsequent **recall**, there was a marked increase in the use of formal verification of hardware floating point operations across the **semiconductor industry**. Prompted by the discovery of the bug, a technique ... called "word-level **model checking**" was developed in 1996. Intel went on to use **formal verification** extensively in the development of later CPU architectures. In the development of the Pentium 4, symbolic trajectory evaluation and **theorem proving** were used to **find a number of bugs that could have led to a similar recall incident** had they gone undetected.*

checking at the machine-instruction level

Formal Verification: Proof Based vs. Check Based

$\Gamma \vdash \phi$

property formula

derivable from left to right using the relevant inference rules (deduction)

system formula (before-after preds of actions, guards, invariants)

undecidable problem

Even if there exists a seq. of inference rule application between Γ and ϕ, a theorem prover cannot always find it.

$\mathcal{M} \models \phi$

satisfies

invariant, temporal properties (LTL, CTL)

system description (usually formulated as a LTS) a graph!

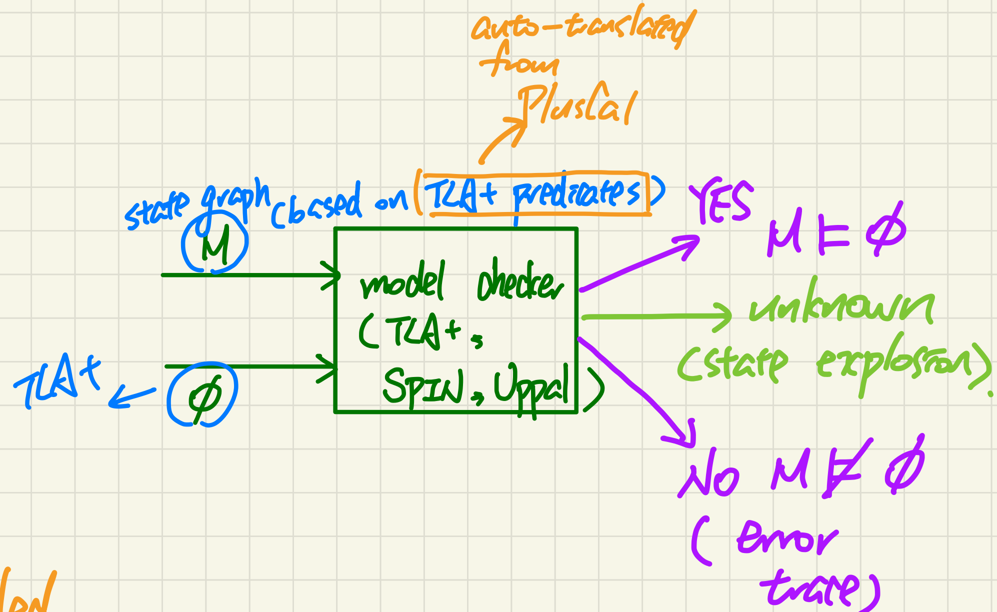
labeled transition system

automated ⇒ - build the graph - traverse the graph to check.

Temporal Logic

- Syntax : structure
- Semantics : meaning

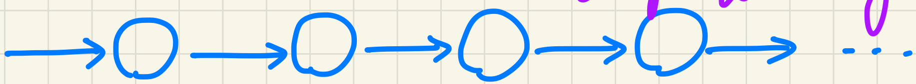
↳ (1) how to express
(2) how to check
(3) when the check failed, how to interpret the error trace

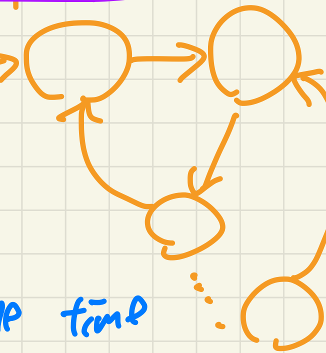


state graph (with a cycle).

(Computation) Path:

must be finite (and small enough to be fit to memory).





an $\boxed{\text{infinite seq.}}$ of states that models the time

trace

(model checking is natural for reactive system).

LTL Syntax: Context-Free Grammar

F $\boxed{G\,\phi}$
G $\boxed{F\,\phi}$

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\perp$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

base cases.

propositional logic

atomic description (which itself does not contain any operators).

unary operators

eventually

binary operators

implicitly use $\forall$ or $\exists$.

$\phi = \underline{\underline{\phi}} \wedge \phi = p \wedge \underline{\underline{\phi}} = p \wedge X\underline{\underline{\phi}} = \boxed{p \wedge X\top}$ from the grammar.

valid LTL formula derivable

Operator Precedence

(1) $F\phi_1 \Rightarrow \phi_2$

$\hookrightarrow$ (a) $F(\phi_1 \Rightarrow \phi_2)$ (b) $(F\phi_1) \Rightarrow \phi_2$ ✓

not what (1) means

Precedence

X, F, G /* unary LTL op */

U, W, R /* binary LTL op */

$\neg$
$\wedge$
$\vee$
$\Rightarrow$

] logical op.

$\underline{X\, p \Rightarrow q}$

Letter	symbol
X	$\bigcirc$
F	$\Diamond$
G	$\Box$

$\underline{X \phi}$

$\equiv$

$\bigcirc \phi$

$F G \phi$

$\equiv$

$\Diamond \Box \phi$