Bounded Model Checking

(Based on [Biere et al. 1999, Benedetti and Cimatti 2003])

Yih-Kuen Tsay
(original created by Ming-Hsien Tsai)

Dept. of Information Management
National Taiwan University

Outline

- Introduction
- An Illustrative Example
- Part I: Bounded Model Checking for LTL (with future only)
- Part II: Bounded Model Checking for LTL with Past (or Full PTL)
- References:
  [Biere *et al.*] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu, "Symbolic Model Checking without BDD," *TACAS 1999*, *LNCS1579*.
  [Benedetti and Cimatti] M. Benedetti and A. Cimatti, "Bounded Model Checking for Past LTL," *TACAS 2003*, *LNCS 2619*.

Introduction

- In *symbolic model checking*, BDDs had traditionally been used for boolean encodings.
- Drawbacks of BDDs:
  - For large systems (with over a few hundred boolean variables), they can be prohibitively large.
  - Selecting the right variable ordering is often time-consuming or needs manual intervention.
- Propositional decision procedures, or SAT solvers, also operate on boolean expressions, but do not use canonical forms.
- SAT solvers can handle thousands of variables or even more.

Introduction (cont.)

- Basic ideas of bounded model checking (BMC):
  - Consider counterexamples of a particular length k.
  - Generate a propositional formula that is satisfiable iff such a counterexample exists.
  - The propositional formula can be tested for satisfiability by a SAT solver.
- Advantages of BMC:
  - It finds counterexamples very fast.
  - It finds counterexamples of minimal length.
  - It uses much less space than BDD-based approaches.
  - It does not need a manually selected variable ordering or time-consuming dynamic reordering.

An Example

- Consider a three-bit shift register.
- Let $M = \langle X, I, T \rangle$ be its state machine:
  - $X \triangleq \{x[0], x[1], x[2]\}$ contains the three bits.
  - $I(X) \triangleq \mathit{true}$, posing no restriction on the initial states.
  - $T(X, X') \triangleq (x'[0] \Leftrightarrow x[1]) \wedge (x'[1] \Leftrightarrow x[2]) \wedge x'[2]$.
- Suppose we want to check if eventually all three bits are set to 0, i.e., if LTL formula $p \triangleq \Diamond(\neg x[0] \wedge \neg x[1] \wedge \neg x[2])$ holds on all paths in M.
- To do so, we search for a path in M such that $\neg p \triangleq \Box(x[0] \vee x[1] \vee x[2])$ on the path.
- If we succeed, then p does not hold on all paths; otherwise, it does.

An Example (cont.)

- We look for (looping) paths with at most $k+1$ states, for instance $k=2$.
- Let X_i denote the set $\{x_i[0], x_i[1], x_i[2]\}$.
- The first 3 states of such a path can be characterized by the following boolean formula:

$$f_M \triangleq I(X_0) \wedge T(X_0, X_1) \wedge T(X_1, X_2)$$

- A witness for $\neg p$ must contain a loop from X_2 back to X_0, X_1, or X_2:

$$L_i \triangleq T(X_2, X_i)$$

- The path must fulfill the constraints imposed by $\neg p$:

$$S_i \triangleq x_i[0] \vee x_i[1] \vee x_i[2]$$

An Example (cont.)

- The following formula is satisfiable iff there is a counterexample of length 2 for p.

$$f_M \wedge \bigvee_{i=0}^{2} L_i \wedge \bigwedge_{i=0}^{2} S_i$$

- Here is a satisfying assignment:

$$\begin{array}{rl} & x_0[0] = x_0[1] = x_0[2] \\ = & x_1[0] = x_1[1] = x_1[2] \\ = & x_2[0] = x_2[1] = x_2[2] \\ = & 1. \end{array}$$

IM NTU

Part I:

Bounded Model Checking for LTL

Kripke Structures

- A Kripke structure is a tuple $M = (S, I, T, L)$ with
  - a finite set of states S,
  - the set of initial states $I \subseteq S$,
  - a transition relation between states $T \subseteq S \times S$, and
  - the labeling of the states $L : S \to \mathscr{P}(A)$ with atomic propositions A.
- Every state of M is required to have a successor.
- We write $s \to t$ for $(s, t) \in T$.
- For an infinite sequence π of states $s_0, s_1, \ldots$, we define
  - $\pi(i) = s_i$
  - $\pi^i = s_i, s_{i+1}, \ldots$.
- An infinite sequence π is a path if $\pi(i) \to \pi(i+1)$ for all $i \in \mathbb{N}$.

Linear Temporal Logic (LTL)

- Let M be a Kripke structure, π be a path in M, and f be an LTL formula (in negation normal form).
- $\pi \models f$ (f is valid along π) is defined as follows:

$$
\begin{array}{lll}
\pi \models p & \text{iff} & p \in L(\pi(0)) \\
\pi \models \neg p & \text{iff} & p \notin L(\pi(0)) \\
\pi \models f \wedge g & \text{iff} & \pi \models f \text{ and } \pi \models g \\
\pi \models f \vee g & \text{iff} & \pi \models f \text{ or } \pi \models g \\
\pi \models \Box f & \text{iff} & \forall j \in [0, \infty). \pi^j \models f \\
\pi \models \Diamond f & \text{iff} & \exists j \in [0, \infty). \pi^j \models f \\
\pi \models \bigcirc f & \text{iff} & \pi^1 \models f \\
\pi \models f \ \mathcal{U} \ g & \text{iff} & \exists j \in [0, \infty). (\pi^j \models g \text{ and } \forall k \in [0, j). \pi^k \models f) \\
\pi \models f \ \mathcal{R} \ g & \text{iff} & \forall j \in [0, \infty). (\pi^j \models g \text{ or } \exists k \in [0, j). \pi^k \models f)
\end{array}
$$

Model Checking

- An LTL formula f is valid in a Kripke structure M, denoted as $M \models \mathbf{A}\ f$, iff $\pi \models f$ for all paths π in M with $\pi(0) \in I$.
- An LTL formula f is satisfiable in a Kripke structure M, denoted as $M \models \mathbf{E}\ f$, iff there is a path π in M such that $\pi \models f$ and $\pi(0) \in I$.
- Given a Kripke structure M and an LTL formula f, the model checking problem is to determine whether $M \models \mathbf{A}\ f$, which is equivalent to determine whether $M \not\models \mathbf{E}\ \neg f$.
- In the following, the problem is restricted to find a witness for formulae of the form $\mathbf{E}\ f$.

Bounded Model Checking

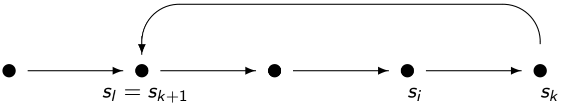
- Consider only a finite prefix of a path that may be a witness of **E** f.
- We restrict the length of the prefix to a certain bound k.
- Generate a propositional formula that is satisfiable iff there is a witness within the bound k.
- The propositional formula can be solved by a SAT solver.
- If there is no witness within bound k, we increase the bound and look for longer and longer possible witnesses.

Infinite Paths from the Prefix

- Though the prefix of a path is finite, it still might represent an infinite path if there is a back loop from the last state of the prefix to any of the previous states.
- If there is no such back loop, then the prefix does not say anything about the infinite behavior of the path.
- Only a prefix with a back loop can represent a witness for $\Box f$.

Loops

- A path π is a (k, l)-loop for $l \leq k$ if
  - $\pi(k) \rightarrow \pi(l)$ and
  - $\pi = u \cdot v^{\omega}$ with
    - $u = \pi(0), \ldots, \pi(l-1)$ and
    - $v = \pi(l), \ldots, \pi(k)$



- A path π is a k-loop if there is an $l \in \mathbb{N}$ with $l \leq k$ for which π is a (k, l)-loop.

Bounded Semantics

- In bounded semantics, we only consider a finite prefix of a path which may or may not be a loop.
- In particular, we only use the first $k + 1$ states of a path to determine the validity of a formula along that path.
- The bounded semantics $\pi \models_k f$ states that the LTL formula f is valid along the path π with bound k.

Bounded Semantics for a Loop

- Let $k \in \mathbb{N}$ and π be a k-loop.
- $\pi \models_k f$ iff $\pi \models f$.
- This is so, because all information about π is contained in the prefix of length k.

Bounded Semantics without a Loop

- Let $k \in \mathbb{N}$ and π be a path that is not a k-loop.
- $\pi \models_k f$ iff $(\pi, 0) \models_k f$ where

$$
\begin{array}{lll}
(\pi, i) \models_k p & \text{iff} & p \in L(\pi(i)) \\
(\pi, i) \models_k \neg p & \text{iff} & p \notin L(\pi(i)) \\
(\pi, i) \models_k f \wedge g & \text{iff} & (\pi, i) \models_k f \text{ and } (\pi, i) \models_k g \\
(\pi, i) \models_k f \vee g & \text{iff} & (\pi, i) \models_k f \text{ or } (\pi, i) \models_k g \\
(\pi, i) \models_k \Box f & \text{iff} & \mathit{false} \\
(\pi, i) \models_k \Diamond f & \text{iff} & \exists j \in [i, k].(\pi, j) \models_k f \\
(\pi, i) \models_k \bigcirc f & \text{iff} & i < k \text{ and } (\pi, i+1) \models_k f \\
(\pi, i) \models_k f \,\mathcal{U}\, g & \text{iff} & \exists j \in [i, k].((\pi, j) \models_k g \text{ and } \forall n \in [i, j).(\pi, n) \models_k f) \\
(\pi, i) \models_k f \,\mathcal{R}\, g & \text{iff} & \exists j \in [i, k].((\pi, j) \models_k f \text{ and } \forall n \in [i, j].(\pi, n) \models_k g)
\end{array}
$$

- Note: $(\pi, i) \models_k f$ is written as $\pi \models_k^i f$ in the paper.

Bounded Semantics without a Loop (cont.)

- Note that the bounded semantics without a loop imply that the following two dualities no longer hold:
  - the duality of $\square$ and $\diamond$ ($\neg\square f = \diamond\neg f$), and
  - the duality of $\mathcal{U}$ and $\mathcal{R}$ ($\neg(f\ \mathcal{U}\ g) = (\neg f)\ \mathcal{R}\ (\neg g)$).

Reduction to Bounded Model Checking

IM NTU

Lemma

Let h be an LTL formula and π a path, then $\pi \models_k h \Rightarrow \pi \models h$.

Lemma

Let f be an LTL formula and M a Kripke structure. If $M \models \mathbf{E}\ f$ then there exists $k \in \mathbb{N}$ with $M \models_k \mathbf{E}\ f$.

Theorem

Let f be an LTL formula and M a Kripke structure. Then $M \models \mathbf{E}\ f$ iff there exists $k \in \mathbb{N}$ with $M \models_k \mathbf{E}\ f$.

Proof of Lemma 1

Let h be an LTL formula and π a path, then $\pi \models_k h \Rightarrow \pi \models h$.

- Case 1: π is a k-loop.
  - The conclusion follows by the definition.
- Case 2: π is not a loop.
  - Prove by induction over the structure of f and $i \leq k$ the stronger property $\pi \models_k^i h \Rightarrow \pi^i \models h$.

Proof of Lemma 1 (cont.)

$$
\begin{array}{ll}
 & \pi \models_k^i f \; \mathcal{R} \; g \\
\Leftrightarrow & \exists j \in [i,k].(\pi \models_k^j f \text{ and } \forall n \in [i,j].\pi \models_k^n g) \\
\Rightarrow & \exists j \in [i,k].(\pi^j \models f \text{ and } \forall n \in [i,j].\pi^n \models g) \\
\Rightarrow & \exists j \in [i,\infty].(\pi^j \models f \text{ and } \forall n \in [i,j].\pi^n \models g) \\
\Rightarrow & \exists j' \in [0,\infty).(\pi^{i+j'} \models f \text{ and } \forall n' \in [0,j'].\pi^{i+n'} \models g) \\
 & (\text{with } j' = j - i \text{ and } n' = n - i) \\
\Rightarrow & \exists j \in [0,\infty).[(\pi^i)^j \models f \text{ and } \forall n \in [0,j].(\pi^i)^n \models g] \\
\Rightarrow & \forall n \in [0,\infty).[(\pi^i)^n \models g \text{ or } \exists j \in [0,n).(\pi^i)^j \models f] \\
 & (\text{see next slide}) \\
\Rightarrow & \pi^i \models f \; \mathcal{R} \; g
\end{array}
$$

Proof of Lemma 1 (cont.)

$$\exists m[\pi^m \models f \text{ and } \forall l, l \le m.\pi^l \models g] \Rightarrow \forall n[\pi^n \models g \text{ or } \exists j, j < n.\pi^j \models f]$$

- Assume that m is the smallest number such that $\pi^m \models f$ and $\pi^l \models g$ for all l with $l \le m$.
- Case 1: $n > m$.
  - Based on the assumption, there exists $j < n$ such that $\pi^j \models f$ (choose $j = m$).
- Case 2: $n \le m$.
  - Because $\pi^l \models g$ for all $l \le m$ we have $\pi^n \models g$ for all $n \le m$.

Proof of Lemma 2

Let f be an LTL formula and M a Kripke structure. If $M \models \mathbf{E}\ f$ then there exists $k \in \mathbb{N}$ with $M \models_k \mathbf{E}\ f$.

- If f is satisfiable in M, then there exists a path in the product structure of M and the tableau of f that starts with an initial state and ends with a cycle in the strongly connected component of fair states.
- This path can be chosen to be a k-loop with k bounded by $|S| \cdot 2^{|f|}$ which is the size of the product structure.
- If we project this path onto its first component, the original Kripke structure, then we get a path π that is a k-loop and in addition fulfills $\pi \models f$.
- By definition of the bounded semantics this also implies $\pi \models_k f$.

From BMC to SAT

- Given a Kripke structure M, an LTL formula f, and a bound k, we will construct a propositional formula $[\![M, f]\!]_k$.
- The bounded model checking problem can be reduced in polynomial time to propositional satisfiability.
  - The size of $[\![M, f]\!]_k$ is polynomial in the size of f if common sub-formulae are shared.
  - It is quadratic in k and linear in the size of the propositional formulae for T, I, and the $p \in A$.

Unfolding the Transition Relation

- For a Kripke structure M and $k \in \mathbb{N}$,

$$[\![M]\!]_k \triangleq I(s_0) \wedge \bigwedge_{i=0}^{k-1} T(s_i, s_{i+1})$$

Trans. of an LTL formula without a Loop

- For an LTL formula f and $k, i \in \mathbb{N}$, with $i \leq k$,

$$
\begin{array}{lcl}
[\![p]\!]_k^i & \triangleq & p(s_i) \\
[\![\neg p]\!]_k^i & \triangleq & \neg p(s_i) \\
[\![f \wedge g]\!]_k^i & \triangleq & [\![f]\!]_k^i \wedge [\![g]\!]_k^i \\
[\![f \vee g]\!]_k^i & \triangleq & [\![f]\!]_k^i \vee [\![g]\!]_k^i \\
[\![\Box f]\!]_k^i & \triangleq & \mathit{false} \\
[\![\Diamond f]\!]_k^i & \triangleq & \bigvee_{j=i}^{k} [\![f]\!]_k^j \\
[\![\bigcirc f]\!]_k^i & \triangleq & \text{if } i < k \text{ then } [\![f]\!]_k^{i+1} \text{ else } \mathit{false} \\
[\![f \;\mathcal{U}\; g]\!]_k^i & \triangleq & \bigvee_{j=i}^{k} ([\![g]\!]_k^j \wedge \bigwedge_{n=i}^{j-1} [\![f]\!]_k^n) \\
[\![f \;\mathcal{R}\; g]\!]_k^i & \triangleq & \bigvee_{j=i}^{k} ([\![f]\!]_k^j \wedge \bigwedge_{n=i}^{j} [\![g]\!]_k^n)
\end{array}
$$

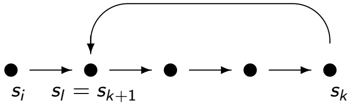
Trans. of an LTL formula for a Loop

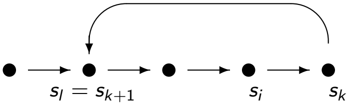
- For an LTL formula f and $k, l, i \in \mathbb{N}$, with $l, i \leq k$,

$$
\begin{array}{lcl}
{}_{l}[\![p]\!]_{k}^{i} & \triangleq & p(s_i) \\
{}_{l}[\![\neg p]\!]_{k}^{i} & \triangleq & \neg p(s_i) \\
{}_{l}[\![f \wedge g]\!]_{k}^{i} & \triangleq & {}_{l}[\![f]\!]_{k}^{i} \wedge {}_{l}[\![g]\!]_{k}^{i} \\
{}_{l}[\![f \vee g]\!]_{k}^{i} & \triangleq & {}_{l}[\![f]\!]_{k}^{i} \vee {}_{l}[\![g]\!]_{k}^{i}
\end{array}
$$

Trans. of an LTL formula for a Loop

$$
\begin{array}{lcl}
{}_l[\![\Box f]\!]_k^i & \triangleq & \bigwedge_{j=\min(i,l)}^{k} {}_l[\![f]\!]_k^j \\
{}_l[\![\Diamond f]\!]_k^i & \triangleq & \bigvee_{j=\min(i,l)}^{k} {}_l[\![f]\!]_k^j \\
{}_l[\![\bigcirc f]\!]_k^i & \triangleq & {}_l[\![f]\!]_k^{succ(i)}
\end{array}
$$





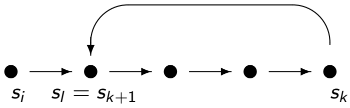
- Let $k, l, i \in \mathbb{N}$, with $l, i \leq k$.

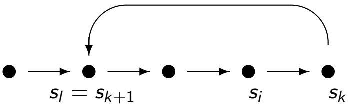
$$
succ(i) \triangleq \begin{cases} i+1 & \text{for } i < k \\ l & \text{for } i = k \end{cases}
$$

Trans. of an LTL formula for a Loop (cont.)

$$
\begin{aligned}
{}_l[\![f\ \mathcal{U}\ g]\!]_k^i &\triangleq \bigvee_{j=i}^{k}\left({}_l[\![g]\!]_k^j \wedge \bigwedge_{n=i}^{j-1} {}_l[\![f]\!]_k^n\right) \vee \\
&\ \bigvee_{j=l}^{i-1}\left({}_l[\![g]\!]_k^j \wedge \bigwedge_{n=i}^{k} {}_l[\![f]\!]_k^n \wedge \bigwedge_{n=l}^{j-1} {}_l[\![f]\!]_k^n\right)
\end{aligned}
$$

$$
\begin{aligned}
{}_l[\![f\ \mathcal{R}\ g]\!]_k^i &\triangleq \bigwedge_{j=\min(i,l)}^{k} {}_l[\![g]\!]_k^j \vee \\
&\ \bigvee_{j=i}^{k}\left({}_l[\![f]\!]_k^j \wedge \bigwedge_{n=i}^{j} {}_l[\![g]\!]_k^n\right) \vee \\
&\ \bigvee_{j=l}^{i-1}\left({}_l[\![f]\!]_k^j \wedge \bigwedge_{n=i}^{k} {}_l[\![g]\!]_k^n \wedge \bigwedge_{n=l}^{j} {}_l[\![g]\!]_k^n\right)
\end{aligned}
$$





Loop Condition

- The loop condition L_k is used to distinguish paths with bound k which are loops or not loops.
- For $k, l \in \mathbb{N}$, let
  - ${}_lL_k \triangleq T(s_k, s_l)$
  - $L_k \triangleq \bigvee_{l=0}^{k} {}_lL_k$.

General Translation

- Let f be an LTL formula, M a Kripke structure, and $k \in \mathbb{N}$.

$$[\![M, f]\!]_k \triangleq [\![M]\!]_k \wedge ((\neg L_k \wedge [\![f]\!]_k^0) \vee (\bigvee_{l=0}^{k} ({}_l L_k \wedge {}_l[\![f]\!]_k^0)))$$

Note: is the term $\neg L_k$ redundant?

Theorem

$[\![M, f]\!]_k$ is satisfiable iff $M \models_k \mathbf{E}\ f$.

Corollary

$M \models \mathbf{A}\ \neg f$ iff $[\![M, f]\!]_k$ is unsatisfiable for all $k \in \mathbb{N}$.

Bounds for LTL

- LTL model checking is known to be PSPACE-complete.
- A polynomial bound on k with respect to the size of M and f for which $M \models_k \mathbf{E}\, f \Leftrightarrow M \models \mathbf{E}\, f$ is unlikely to be found.

Theorem

Given an LTL formula f and a Kripke structure M, let $|M|$ be the number of states in M, then $M \models \mathbf{E}\, f$ iff there exists $k \leq |M| \times 2^{|f|}$ with $M \models_k \mathbf{E}\, f$.

- For the subset of LTL formulae that involves only temporal operators $\Diamond$ and $\Box$, LTL model checking is NP-complete.
- For this subset of LTL formulae, there exists a bound on k linear in the number of states and the size of the formula.

Bounds for LTL (cont.)

Definition (Loop Diameter)

A Kripke structure is lasso shaped if every path p starting from an initial state is of the form $u_p v_p^\omega$, where u_p and v_p are finite sequences of length less or equal to u and v, respectively. The loop diameter of M is defined as (u, v).

Theorem

Given an LTL formula f and a lasso-shaped Kripke structure M, let the loop diameter of M be (u, v), then $M \models \mathbf{E}\, f$ iff there exists $k \leq u + v$ with $M \models_k \mathbf{E}\, f$.

Part II:

Bounded Model Checking for LTL with Past

Note: (k, l)-loop here corresponds to $(k-1, l)$-loop in Part I. For easy cross-referencing with the original paper, we have not attempted to unify the notion.

Propositional Temporal Logic

- The full propositional temporal logic (PTL) is LTL with past operators.

$$
\begin{array}{lll}
(\pi, i) \models \ominus f & \text{iff} & i > 0 \text{ and } (\pi, i-1) \models f \\
(\pi, i) \models \circleddash f & \text{iff} & i = 0 \text{ or } (\pi, i-1) \models f \\
(\pi, i) \models \diamondminus f & \text{iff} & \exists j, j \leq i.(\pi, j) \models f \\
(\pi, i) \models \boxminus f & \text{iff} & \forall j, j \leq i.(\pi, j) \models f \\
(\pi, i) \models f \; \mathcal{S} \; g & \text{iff} & \exists j, j \leq i.((\pi, j) \models g \text{ and } \forall k, j < k \leq i.(\pi, k) \models f) \\
(\pi, i) \models f \; \mathcal{T} \; g & \text{iff} & \forall j, j \leq i.((\pi, j) \models g \text{ or } \exists k, j < k \leq i.(\pi, k) \models f)
\end{array}
$$

- Every PTL formula can be converted into the negation normal form.

Extend the Translation without Loops

- Let $k, i \in \mathbb{N}$ with $i \leq k$.

$$
\begin{array}{lcl}
[\![\ominus f]\!]_k^i & \triangleq & \begin{cases} \mathit{false} & i = 0 \\ [\![f]\!]_k^{i-1} & i > 0 \end{cases} \\
[\![\circleddash f]\!]_k^i & \triangleq & \begin{cases} \mathit{true} & i = 0 \\ [\![f]\!]_k^{i-1} & i > 0 \end{cases} \\
[\![\diamondminus f]\!]_k^i & \triangleq & \bigvee_{j=0}^{i} [\![f]\!]_k^j \\
[\![\boxminus f]\!]_k^i & \triangleq & \bigwedge_{j=0}^{i} [\![f]\!]_k^j \\
[\![f \ \mathcal{S} \ g]\!]_k^i & \triangleq & \bigvee_{j=0}^{i}([\![g]\!]_k^j \wedge \bigwedge_{n=j+1}^{i} [\![f]\!]_k^n) \\
[\![f \ \mathcal{T} \ g]\!]_k^i & \triangleq & \bigwedge_{j=0}^{i}([\![g]\!]_k^j \vee \bigvee_{n=j+1}^{i} [\![f]\!]_k^n)
\end{array}
$$

Extend the Translation with Loops

- The extension is not straightforward.
- For example, consider the path $01(2345)^\omega$ which can be seen as a $(6, 2)$-loop.
  - In the future case, the encoding of a specification is based on the idea that, for every time in the encoding, exactly one successor time exists.
  - Past formulae do not enjoy the above property.
    - The predecessor of 2 may be 1 or 5.

$x =$	0	1	2	3	4	5	2	3	4	5	2	3	4	5	2
time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

The Solution: Intuition

- The formula $\Diamond(x = 2 \wedge \diamondminus(x = 3 \wedge \diamondminus(x = 4 \wedge (\diamondminus(x = 5)))))$ is true in all the occurrences of $x = 2$ after the fourth.
- The key idea is that every formula has a finite discriminating power for events in the past.
- When evaluated sufficiently far from the origin of time, a formula becomes unable to distinguish its past sequence from infinitely many other past sequences with a "similar" behavior.
- The idea is then to collapse the undistinguishable versions of the past together into the same equivalence class.

Past Temporal Horizon

- The past temporal horizon (PTH) $\tau_\pi(f)$ of a PTL formula f with respect to a (k, l)-loop π (with period $p = k - l$) is the smallest value $n \in \mathbb{N}$ such that

$$\forall i, l \leq i < k.((\pi, i + np) \models f \text{ iff } (\forall n' > n.(\pi, i + n'p) \models f)).$$

PTH of a PTL Formula

- The PTH $\tau(f)$ of a PTL formula f is defined as $\tau(f) \triangleq \max_{\tau\in\Pi} \tau_\pi(f)$ where Π is the set of all the paths which are (k,l)-loops for some $k > l \geq 0$.

Theorem

Let f and g be PTL formulae. Then, it holds that:

- $\tau(p) = 0$, *when* $p \in A$ *and* $\tau(f) = \tau(\neg f)$;
- $\tau(\circ f) \leq \tau(f)$, *when* $\circ \in \{\bigcirc, \Diamond, \Box\}$;
- $\tau(\circ f) \leq \tau(f) + 1$, *when* $\circ \in \{\ominus, \circleddash, \diamondminus, \boxminus\}$;
- $\tau(f \circ g) \leq \max(\tau(f), \tau(g))$, *when* $\circ \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$;
- $\tau(f \circ g) \leq \max(\tau(f), \tau(g)) + 1$, *when* $\circ \in \{\mathcal{S}, \mathcal{T}\}$;

- The PTH of a PTL formula is bounded by its structure regardless of the particular path π.

Borders and Intervals

- We call
  - $\mathbf{LB}(n) \triangleq l + np$ the n-th left border of π,
  - $\mathbf{RB}(n) \triangleq k + np$ the n-th right border of π, and
  - the interval $M(n) \triangleq [0, \mathbf{RB}(n))$ the n-th main domain of a (k, l)-loop.
- We call
  - $\mathbf{LB}(f) \triangleq \mathbf{LB}(\tau(f))$ the left border of f,
  - $\mathbf{RB}(f) \triangleq \mathbf{RB}(\tau(f))$ the right border of f, and
  - $M(f) \triangleq M(\tau(f))$ the main domain of f.

Borders and Intervals (cont.)

$x =$	0	1	2	3	4	5	2	3	4	5	2	3	4	5	2
time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

- $\textbf{LB}(0) = 2$
- $\textbf{RB}(0) = 6$
- $\textbf{LB}(1) = 6$
- $\textbf{RB}(1) = 10$

Projection of Points

- Let $i \in \mathbb{N}$.
- The projection of the point i in the n-th main domain of a (k,l)-loop is $\rho_n(i)$, defined as

$$\rho_n(i) \triangleq \begin{cases} i & i < \mathbf{RB}(n) \\ \rho_n(i-p) & \text{otherwise} \end{cases}$$

- The projection of the point i onto the main domain of f is defined as $\rho_f(i) \triangleq \rho_{\tau(f)}(i)$.

Projection of Intervals

- The projection of the interval $[a, b)$ onto the main domain of f is defined as $\rho_f([a, b)) \triangleq \rho_{\tau(f)}([a, b))$.

Lemma

For an open interval $[a, b)$*,*

$$\rho_n([a,b)) = \begin{cases} \emptyset & \text{if } a = b, \text{ else} \\ [a, b) & \text{if } b < \mathbf{RB}(n), \text{ else} \\ [\min(a, \mathbf{LB}(n)), \mathbf{RB}(n)) & \text{if } b - a \geq p, \text{ else} \\ [\rho_n(a), \rho_n(b)) & \text{if } \rho_n(a) < \rho_n(b), \text{ else} \\ [\rho_n(a), \mathbf{RB}(n)) \cup [\mathbf{LB}(n), \rho_n(b)) & \end{cases}$$

Extended Projection of Intervals

- An extended intervals is of the form $[a, b)$ where b is possibly less than a (or even it is equal to ∞).
- Let $[a, b)$ be an extended interval.
- The extended projection of $[a, b)$ onto the n-th main domain of a (k, l)-loop is defined as follows

$$\rho^*_n([a, b)) \triangleq \begin{cases} \rho^*_n([a, \max(a, \mathbf{RB}(n)) + p)) & b = \infty \\ \rho^*_n([a, b + p)) & b < a \\ \rho^*_n([a, b)) & \text{otherwise} \end{cases}$$

- As before, $\rho^*_f([a, b)) \triangleq \rho^*_{\tau(f)}([a, b))$.

Equivalent Counterparts

Theorem

For

- *any PTL formula f,*
- *any (k, l)-loop π, and*
- *any extended interval $[a, b)$,*

a point $i \in [a, b)$ such that $(\pi, i) \models f$ exists iff a point $i' \in \rho_f^*([a, b))$ exists such that $(\pi, i') \models f$.

Extend the Translation with Loops

- The translation of a PTL formula on a (k, l)-loop π at time point i (with $k, l, i \in \mathbb{N}$ and $0 \leq l < k$) is a propositional formula inductively defined as follows.

$$
\begin{array}{lcl}
{}_l[\![p]\!]_k^i & \triangleq & p^{\rho_0(i)} \\
{}_l[\![\neq p]\!]_k^i & \triangleq & \neq p^{\rho_0(i)} \\
{}_l[\![f \wedge g]\!]_k^i & \triangleq & {}_l[\![f]\!]_k^{\rho_f(i)} \wedge {}_l[\![g]\!]_k^{\rho_g(i)} \\
{}_l[\![f \vee g]\!]_k^i & \triangleq & {}_l[\![f]\!]_k^{\rho_f(i)} \vee {}_l[\![g]\!]_k^{\rho_g(i)}
\end{array}
$$

Extend the Translation with Loops (cont.)

$$
\begin{array}{lcl}
{}_l[\![\Diamond f]\!]_k^i & \triangleq & \bigvee_{j\in\rho_f^*([i,\infty))} {}_l[\![f]\!]_k^j \\
{}_l[\![\Box f]\!]_k^i & \triangleq & \bigwedge_{j\in\rho_f^*([i,\infty))} {}_l[\![f]\!]_k^j \\
{}_l[\![f\ \mathcal{U}\ g]\!]_k^i & \triangleq & \bigvee_{j\in\rho_g^*([i,\infty))}\left({}_l[\![g]\!]_k^j \wedge \bigwedge_{n\in\rho_f^*([i,j))} {}_l[\![f]\!]_k^n\right) \\
{}_l[\![f\ \mathcal{R}\ g]\!]_k^i & \triangleq & \bigwedge_{j\in\rho_g^*([i,\infty))}\left({}_l[\![g]\!]_k^j \vee \bigvee_{n\in\rho_f^*([i,j))} {}_l[\![f]\!]_k^n\right) \\
{}_l[\![\ominus f]\!]_k^i & \triangleq & i > 0 \wedge {}_l[\![f]\!]_k^{\rho_f(i-1)} \\
{}_l[\![\circledsim f]\!]_k^i & \triangleq & i = 0 \vee {}_l[\![f]\!]_k^{\rho_f(i-1)} \\
{}_l[\![\diamondminus f]\!]_k^i & \triangleq & \bigvee_{j\in\rho_f^*([0,i])} {}_l[\![f]\!]_k^j \\
{}_l[\![\boxminus f]\!]_k^i & \triangleq & \bigwedge_{j\in\rho_f^*([0,i])} {}_l[\![f]\!]_k^j \\
{}_l[\![f\ \mathcal{S}\ g]\!]_k^i & \triangleq & \bigvee_{j\in\rho_g^*([0,i])}\left({}_l[\![g]\!]_k^j \wedge \bigwedge_{n\in\rho_f^*((j,i])} {}_l[\![f]\!]_k^n\right) \\
{}_l[\![f\ \mathcal{T}\ g]\!]_k^i & \triangleq & \bigwedge_{j\in\rho_g^*([0,i])}\left({}_l[\![g]\!]_k^j \vee \bigvee_{n\in\rho_f^*((j,i])} {}_l[\![f]\!]_k^n\right)
\end{array}
$$

Correctness of the Translation

Theorem

For any PTL formula f, a (k, l)-loop path π in M such that $\pi \models f$ exists iff $[\![M]\!]_k \wedge {}_lL_k \wedge {}_l[\![f]\!]_k^0$ is satisfiable.