LTL can be more succinct

Kamal Lodaya

The Institute of Mathematical Sciences, Chennai

Work in progress with A.V. Sreejith

Logic LTL

$\alpha ::= p \in Prop \mid \neg\, \alpha \mid \alpha\ \vee\ \beta \mid \mathsf{X}\alpha \mid \mathsf{F}\alpha \mid \phi\, \mathsf{U}\, \beta$

- $\mathsf{F}\alpha$ can be abbreviated as $true\mathsf{U}\alpha$.
- Let $\mathsf{X}^m\alpha$ abbreviate the m-fold iteration $\mathsf{X}\ldots\mathsf{X}\alpha$.
- We use notation like $LTL[\mathsf{X},\mathsf{U}]$, $LTL[\mathsf{X},\mathsf{F}]$, $LTL[\mathsf{X}^m,\mathsf{F}]$, ...to abbreviate various fragments of LTL.
- In the last case we can distinguish between $LTL^{bin}[\mathsf{X}^m,\mathsf{F}]$ and $LTL^{un}[\mathsf{X}^m,\mathsf{F}]$ depending on whether the iteration index m is written in binary or in unary notation.
- Can also consider “past” modalities (e.g. $\mathsf{P}\alpha$ as mirror of $\mathsf{F}\alpha$.

Semantics of LTL

As usual, the semantics for LTL is given by a state sequence $\sigma : \mathcal{N} \to \wp(Prop)$ or word over the alphabet $\wp(Prop)$.

- $\sigma, i \models p$ iff $p \in \sigma(i)$
- $\sigma, i \models \mathsf{X}\alpha$ iff $\sigma, i + 1 \models \alpha$
- $\sigma, i \models \mathsf{X}^m\alpha$ iff $\sigma, i + m \models \alpha$
- $\sigma, i \models \mathsf{F}\alpha$ iff for some $k : i \leq k : \sigma, k \models \alpha$
- $\sigma, i \models \alpha \mathsf{U} \beta$ iff for some $k : i \leq k : \sigma, k \models \beta$ and
  $\forall j : i \leq j < k : \sigma, j \models \alpha$

Some known results

A formula is satisfiable if it holds in some model at the beginning.

Theorem 1 (Sistla,Clarke) *Satisfiability for* $LTL[\mathsf{X},\mathsf{U}]$ *is in* PSPACE.

Corollary 2 *Satisfiability for* $LTL^{bin}[\mathsf{X}^m,\mathsf{U}]$ *is in* EXPSPACE.

Theorem 3 (SC; Alur,Henzinger) *Satisfiability for* $LTL[\mathsf{X},\mathsf{F}]$ *is* PSPACE*-hard, and for* $LTL^{bin}[\mathsf{X}^m,\mathsf{F}]$ *is* EXPSPACE*-hard.*

Theorem 4 (SC) *Satisfiability for* $LTL[\mathsf{F}]$ *is* NP*-complete.*

Modelchecking

- Results are also known for the modelchecking question of these logics, where a finite transition system describes all the word models.
- We ignore modelchecking in this presentation and stick to satisfiability.

Expressiveness

Some results which show that some of these fragments relate to other logics.

Theorem 5 (Kamp; Gabbay,Pnueli,Shelah,Stavi) $LTL[\mathsf{X}, \mathsf{U}]$ *has the same expressiveness as first-order logic (with monadic predicates) on linear orders.*

Theorem 6 (Etessami,Vardi,Wilke) $LTL[\mathsf{F}, \mathsf{P}]$ *(with the past modality as well) has the same expressiveness as two-variable first-order logic (with monadic predicates) on word models.*

Modulo counting extensions of LTL

$$\delta ::= \#\alpha \mid \delta_1 + \delta_2 \mid \delta_1 - \delta_2 \mid c\delta,\ c \in \mathcal{N}$$

$$\phi ::= \delta \equiv r \text{ mod } q,\ q \in \mathcal{N}, 0 \leq r < q$$

$$\alpha ::= p \in Prop \mid \phi \mid \neg\, \alpha \mid \alpha\ \vee\ \beta \mid \mathsf{X}\alpha \mid \mathsf{F}\alpha \mid \alpha\ \mathsf{U}\ \beta$$

- $\sigma, i \models \#\alpha \equiv r$ mod q iff $(\Sigma_{j=1}^{i} : \sigma, j \models \alpha) \equiv r$ mod q
- We will use the "length" ℓ to abbreviate $\#true$.
- Notice that ℓ evaluates at the index i to the (unbounded) value i, but the modulo counting bounds it syntactically.

More results

The logic in the previous slide is called $LTL[\mathsf{X}, \mathsf{U}]{+}MOD$.

Theorem 7 (Baziramwabo,McKenzie,Thérien) *$LTL[\mathsf{X}, \mathsf{U}]{+}MOD$ has the same expressiveness as $FO{+}MOD$ on finite words.*

The complexity is preserved for the unary version.

Theorem 8 (Wolper; Serre) *Satisfiability of $LTL^{un}[\mathsf{X}, \mathsf{U}]{+}MOD$ is in* PSPACE.

Corollary 9 *Satisfiability of $LTL^{bin}[\mathsf{X}, \mathsf{U}]{+}MOD$ is in* EXPSPACE.

Modulo counting is not that weak

CORRIDOR TILING: Given a finite set of tile types, relations which say when a tile can be to the right of a tile, and when a tile can be below a tile, a number $n > 2$, a top row of n tiles and a bottom row of n tiles, is there a tiling with n columns from the top row to the bottom row?

Theorem 10 *Satisfiability of* $LTL^{un}[\mathsf{F}]$+MOD *is* PSPACE*-hard.*

Corridor tiling is coded in this logic. We use length $\ell \equiv 0 \text{ mod } n$ to go down a column, and $\#p \equiv 0 \text{ mod } 2$ to code alternate rows and columns. We also use the first n primes to encode modulo constraints on large numbers in unary.

...but is succinct

Theorem 11 *Satisfiability of* $LTL^{bin}[\mathsf{X},\mathsf{U}]$+$MOD$ *is in* PSPACE*.*

We discuss the proof over the next few slides.

- First observe that if formulae with moduli q_1 and q_2 occur within the given formula α whose satisfiability is being checked, we consider them using the larger modulus $lcm(q_1, q_2)$, which is a polynomial blowup.
- Hence we can without loss of generality consider only a single modulus q as occurring in α.

Closure and atoms

- Now once a formula $\delta \equiv r \text{ mod } q$ enters the Fischer-Ladner closure of the given α, the entire set $\delta_q = \{\delta \equiv r \text{ mod } q \mid 0 \leq r < q\}$ has to be included in the closure of α. This is exponential in q and hence exponential in the size of α.
- Although the closure of α is now exponential in the size of α, we can change the definition of an atom (maximal consistent subset) so that exactly one of the formulas in δ_q is in an atom and its existence implicitly implies the negation of the others. The number of atoms continues to be exponential in α.

The formula automaton

- Hence there is a finite automaton M_α of size exponential in α which accepts the language of models of α. Each of its states can be represented in polynomial space. So also its transition relation.
- By representing the modulus in binary, a state can be updated along a transition relation using polynomial space.
- Now one can guess and verify an accepting path in polynomial space.

Still weaker logics

When counting formulae in $LTL[\mathsf{X}, \mathsf{U}]$+$MOD$ are only restricted to using ℓ (that is, $\#true$), we call the resulting logic $LTL[\mathsf{X}, \mathsf{U}]$+$LEN$.

Theorem 12 *Satisfiability of* $LTL^{bin}[\mathsf{F}]$+LEN *is in* Σ_3^P*.*

Notice that there need be no polynomial-sized model. The smallest model for a formula α can be exponential in the size of α, because of the binary notation.

Proof idea for the Σ_3^P bound

- We can divide the "requirements" for which witnesses are required into future requirements of the form $\mathsf{F}\alpha$ and modulo requirements of the form $\delta \equiv r \ \mathsf{mod}\ q$. A shorter representation of the model consists of the witness points for the future requirements and (representations of) blocks of length at most $O(q)$ between them during which modulo requirements are satisfied.
- Assuming that the last problem can be solved by making calls to a Π_2^P oracle, satisfiability can be checked in NP, that is, we have a Σ_3^P procedure.

Block satisfiability

- Let LEN be the restriction of the logic to word models where only boolean and length counting properties are allowed.
- BLOCKSAT: Given a LEN formula α and a natural number n in binary, is there a word model of size n of the formula $\mathsf{G}\alpha$?
- BLOCKVAL: Is the formula $\mathsf{F}\alpha$ valid over word models of size n?

Block validity

Lemma 13 BlockVal *can be checked in* Σ_2^P.

Proof Massage the formula α and guess the position where the massaged formula should be propositionally valid, which is in coNP.

Hence BlockSat is in Π_2^P.
Hence LTL^{bin}[F]+LEN has an NP procedure making calls to a Π_2^P oracle and is therefore in Σ_3^P.

A lower bound

Theorem 14 *Satisfiability of* $LTL^{un}[\mathsf{F}]+LEN$ *is* Σ_3^P*-hard.*

- The proof is by reduction from QBF with three levels of alternation: let $\beta = \exists x_1, \ldots, x_k \, \forall y_1, \ldots, y_l \, \exists z_1, \ldots, z_m \, B$.
- Consider the first k prime numbers $q_1, \ldots, q_k$. Replace the x_i's in B above by $\mathsf{FG}(\ell \equiv 0 \bmod q_i)$.
- Then take the next l prime numbers $p_1, \ldots, p_l$. Replace the y_j's in B by $\ell \equiv 0 \bmod p_j$.
- Add a conjunct $\mathsf{F}(\bigwedge_{j=1}^{l} \ell \equiv 0 \bmod p_j)$. Call the resulting formula γ. There is a polynomial blowup in constructing this formula.

Lemma 15 β *is satisfiable iff there is a word model for* $\mathsf{G}\gamma$.

Discussion

- When LTL is extended with threshold counting, the specification of the threshold in succinct notation leads to an exponential blowup.
- When LTL is extended with modulo counting, it does not matter if the specification of the moduli is in succinct notation.
- Is this just something ad hoc, or is there a more abstract way of understanding what is going on?

How far does this go?

- We have also considered LTL extended with a generalized quantifier corresponding to the symmetric group $S_n, n \geq 2$. (Also present in the papers by Baziramwabo,McKenzie,Thérien and by Serre.)
- The definitions are ugly. Refer to the papers.
- In this case we can use succinct notation based on the generators which is linear in n rather than on the elements of the group (which are exponential in n). Again there is no blowup and satisfiability is in PSPACE.

Question

This leads us to raise the following question.

- Can one think of other families of automata, where a "standard" enumeration of their states and transitions can be represented in logarithmic notation, and for which the PSPACE bound will continue to hold?