Some References

Checking that finite state concurrent programs satisfy their linear specification.
In ACM Symposium PoPL’85, 97–107.

The tableau method for temporal logic: An overview,

The complexity of propositional linear temporal logic.

Fast LTL to Büchi automata translation.
In CAV’01, vol. 2102, Lecture Notes in Computer Science, pp. 53–65.
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

Specification and Verification using Temporal Logics.
In Modern applications of automata theory, IISc Research Monographs 2.
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

Büchi automata

Definition:
A Büchi automaton (BA) is a tuple \( A = (Q, \Sigma, I, T, F) \) where
- \( Q \): finite set of states
- \( \Sigma \): finite set of labels
- \( I \subseteq Q \): set of initial states
- \( T \subseteq Q \times \Sigma \times Q \): set of transitions (non-deterministic)
- \( F \subseteq Q \): set of accepting (repeated, final) states

Run: \( \rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots \) with \((q_i, a_i, q_{i+1}) \in T\) for all \( i \geq 0 \).
\( \rho \) is accepting if \( q_0 \in I \) and \( q_i \in F \) for infinitely many \( i \)'s.

\[ L(A) = \{ a_0 a_1 a_2 \ldots \in \Sigma^\omega \mid \exists \rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots \text{ accepting run} \} \]

A language \( L \subseteq \Sigma^\omega \) is \( \omega \)-regular if it can be accepted by some Büchi automaton.
Büchi automata

Examples:

Infinitely many $a$’s:

Finitely many $a$’s:

Whenever $a$ then later $b$:

Properties

Büchi automata are closed under union, intersection, complement.

- Union: trivial
- Intersection: easy (exercise)
- Complement: difficult

Let $L = \Sigma^* (a \Sigma^{n-1} b \cup b \Sigma^{n-1} a) \Sigma^\omega$

Any non-deterministic Büchi automaton for $\Sigma^\omega \setminus L$ has at least $2^n$ states.

Theorem: Büchi

Let $L \subseteq \Sigma^\omega$ be a language. The following are equivalent:

- $L$ is $\omega$-regular
- $L$ is $\omega$-rational, i.e., $L$ is a finite union of languages of the form $L_1 \cdot L_2^\omega$ where $L_1, L_2 \subseteq \Sigma^\omega$ are rational.
- $L$ is MSO-definable, i.e., there is a sentence $\varphi \in \text{MSO}_{\Sigma} (\leq, <)$ such that $L = L(\varphi) = \{ w \in \Sigma^\omega | w \models \varphi \}$.

Exercises:

1. Construct a BA for $L(\varphi)$ where $\varphi$ is the FO$_\leq (<)$ sentence

   $$(\forall x, (P_a(x) \rightarrow \exists y > x, P_a(y))) \rightarrow (\forall x, (P_b(x) \rightarrow \exists y > x, P_b(y)))$$

2. Given BA for $L_1 \subseteq \Sigma^\omega$ and $L_2 \subseteq \Sigma^\omega$, construct BA for

   $$\text{next}(L_1) = \Sigma \cdot L_1$$
   $$\text{until}(L_1, L_2) = \{ uv \in \Sigma^\omega | u \in \Sigma^+ \land v \in L_2 \land u''v \in L_1 \text{ for all } u', u'' \in \Sigma^+ \text{ with } u = u'u'' \}$$

Generalized Büchi automata

Definition: acceptance on states or on transitions

$A = (Q, \Sigma, I, T, F_1, \ldots, F_n)$ with $F_i \subseteq Q$.

An infinite run $\sigma$ is successful if it visits infinitely often each $F_i$.

$A = (Q, \Sigma, I, T_1, \ldots, T_n)$ with $T_i \subseteq T$.

An infinite run $\sigma$ is successful if it uses infinitely many transitions from each $T_i$.

Example: Infinitely many $a$’s and infinitely many $b$’s

Theorem:

1. GBA and BA have the same expressive power.
2. Checking whether a BA or GBA has an accepting run is NLOGSPACE-complete.
We identify a Büchi automaton with output

**Definition: SBT (Synchronous letter to letter) Büchi Transducer**

Let $A$ and $B$ be two alphabets. A synchronous Büchi transducer from $A$ to $B$ is a tuple $A = (Q, A, I, T, F, \mu)$ where $(Q, A, I, T, F)$ is a Büchi automaton (input) and $\mu : T \rightarrow B$ is the output function. It computes the relation

$[A] = \{(u, v) \in A^\omega \times B^\omega \mid \exists \rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots \text{ accepting run}
\begin{align*}
\text{with } u &= q_0a_0q_1a_1q_2a_2q_3 \cdots \\
\text{and } v &= \mu(q_0, a_0, q_1)\mu(q_1, a_1, q_2)\mu(q_2, a_2, q_3) \cdots \}
\end{align*}$

If $(Q, A, I, T, F)$ is unambiguous then $[A] : A^\omega \rightarrow B^\omega$ is a (partial) function.

We will also use SGT (synchronous transducers with generalized Büchi acceptance) when the transducers define functions.

**Example: Left shift with $A = B = \{a, b\}$**

![example diagram]

Definition: Composition

Let $A$, $B$, $C$ be alphabets.

Let $A = (Q, A, I, T, (F_i)_i, \mu)$ be an SGT from $A$ to $B$.

Let $A' = (Q', A', I', T', (F'_j)_j, \mu')$ be an SGT from $B$ to $C$.

Then $A \cdot A' = (Q \times Q', A \times I', T''$, $(F_i \times F_j)'_i$, $(Q \times F'_j)_j, \lambda'')$ is defined by:

$\tau'' = (p, p') \xrightarrow{\omega} (q, q') \in T''$ and $\mu''(\tau'') = (b, c)$

iff $\tau = p \xrightarrow{\omega} q \in T$ and $b = \mu(\tau)$ and $\tau' = p' \xrightarrow{\omega} q' \in T'$ and $c = \mu'(\tau')$

$A \cdot A'$ is an SGT from $A$ to $B \times C$.

**Proposition: Product**

We identify $(B \times C)^\omega$ with $B^\omega \times C^\omega$.

1. We have $[A \times A'] = \{(u, v) \mid (u, v) \in [A]$ and $(u, v) \in [A']\}$.

2. If $(Q, A, I, T, (F_i)_i)$ and $(Q', A', I', T', (F'_j)_j)$ are unambiguous then $(Q \times Q', A \times I', T'', (F_i \times F_j)'_i,(Q \times F'_j)_j)$ is also unambiguous.

Then, $\forall u \in A^\omega$ we have $[A \cdot A'](u) = ([A](u), [A'](u))$. 

**Branching Time Specifications**

**Outline**

**Introduction**

**Models**

**Specifications**

- Satisfiability and Model Checking for LTL
  - Büchi automata
    - From LTL to BA
    - Decidability and Complexity
Subalphabets of $\Sigma = 2^{AP}$

**Definition:**
For a propositional formula $\xi$ over $AP$, we let $\Sigma_\xi = \{a \in \Sigma \mid a \models \xi\}$.

For instance, for $p, q \in AP$,
- $\Sigma_p = \{a \in \Sigma \mid p \in a\}$ and $\Sigma_{\neg p} = \Sigma \setminus \Sigma_p$
- $\Sigma_{p \land q} = \Sigma_p \cap \Sigma_q$ and $\Sigma_{p \lor q} = \Sigma_p \cup \Sigma_q$
- $\Sigma_{p \rightarrow q} = \Sigma_p \setminus \Sigma_q$

**Notation:**
In automata, $p \xrightarrow{\Sigma_\xi} q$ stands for the set of transitions $\{p\} \times \Sigma_\xi \times \{q\}$.

To simplify the pictures, we use $p \xrightarrow{\xi} q$ instead of $p \xrightarrow{\Sigma_\xi} q$.

**Example:**
![Diagram](image)

**Lemma:** The input BA is prophetic

For all $u = a_0 a_1 a_2 \cdots \in \Sigma^\omega$, there is a unique accepting run $\rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots$ of $A$ on $u$.

The run $\rho$ satisfies for all $i \geq 0$, $q_i = \begin{cases} 1 & \text{if } u, i \models \rho \\ 2 & \text{if } u, i \models \neg q \land (p U' q) \\ 3 & \text{if } u, i \models \neg (p U' q) \end{cases}$

Semantics of LTL with sequential functions

**Definition:** Semantics of $\varphi \in LTL(\Sigma, S, U)$
Let $\Sigma = 2^{AP}$ and $B = \{0, 1\}$.

Define $[\varphi] : \Sigma^\omega \rightarrow \mathbb{B}$ by $[\varphi](u) = b_0 b_1 b_2 \cdots$ with $b_i = \begin{cases} 1 & \text{if } u, i \models \varphi \\ 0 & \text{otherwise} \end{cases}$

**Example:**
\[
[p U q][\emptyset\{p\}\emptyset\{p\}\{p\}\emptyset\{p\}\emptyset\{p, q\}\emptyset^\omega] = 100111011001
\]
\[
[X p][\emptyset\{p\}\emptyset\{p\}\{p\}\emptyset\{p\}\emptyset\{p, q\}\emptyset^\omega] = 010110011001
\]
\[
[F p][\emptyset\{p\}\emptyset\{p\}\{p\}\emptyset\{p\}\emptyset\{p, q\}\emptyset^\omega] = 111111111001
\]
The aim is to compute $[\varphi]$ with Büchi transducers.

Special cases of Until: Future and Next

**Example:** $F q = T U q$ and $X q = \bot U q$

![Diagram](image)

**Exercise:** Give SBT’s for the following formulae:
- $p U' q$, $F' q$, $G q$, $G' q$, $p R q$, $p R' q$, $p S q$, $p S' q$, $G(p \rightarrow F q)$
From LTL to Büchi automata

Definition: SBT for LTL modalities
- $A_T$ from $\Sigma$ to $B = \{0, 1\}$: $\Sigma/1$
- $A_p$ from $\Sigma$ to $B = \{0, 1\}$: $p/1 \neg p/0$
- $A_\land$ from $B$ to $B$: $0/1 1/0 0,0/0 1,0/1 0,1/1 1,1/1$
- $A_V$ from $B^2$ to $B$: $0$
- $A_\lor$ from $B^2$ to $B$: $0$
- $A_\land$ from $B^2$ to $B$: $0$

Remark

Moreover, for each $\xi \in \text{LTL}(AP, S, U)$ we define inductively an SGBT $A_\xi$ as follows:
- $A_T$ and $A_p$ for $p \in AP$ are already defined
- $A_{\neg \varphi} = A_\varphi \circ A_p$
- $A_{\varphi \land \psi} = A_\varphi \circ (A_p \land A_\psi)$
- $A_{\varphi \land \psi} = A_\varphi \circ (A_p \land A_\psi)$
- $A_{\varphi \lor \psi} = A_\varphi \circ (A_p \land A_\psi)$

Theorem: Correctness of the translation
For each $\xi \in \text{LTL}(AP, S, U)$, we have $[A_\xi] = [\xi]$.

Moreover, the number of states of $A_\xi$ is at most $2^{\ell_S} \cdot 3^{\ell_U}$ where $|\ell_S|$ (resp. $|\ell_U|$) is the number of $S$ (resp. $U$) occurring in $\xi$.

Remark:
- If a subformula $\varphi$ occurs serveral time in $\xi$, we only need one copy of $A_p$.
- We may also use automata for other modalities: $A_X$, $A_U$, ...

Useful simplifications

Reducing the number of temporal subformulae
- $(X \varphi) \land (X \psi) \equiv X(\varphi \land \psi)$
- $(X \varphi) \lor (X \psi) \equiv X(\varphi \lor \psi)$
- $G \varphi \land (G \psi) \equiv G(\varphi \land \psi)$
- $G \varphi \lor (G \psi) \equiv G(\varphi \lor \psi)$
- $(\varphi_1 \land \varphi_2) U \psi \equiv (\varphi_1 \land \varphi_2) U \psi$
- $(\varphi U \psi) \lor (\varphi U \psi_2) \equiv \varphi U (\psi_1 \lor \psi_2)$

Merging equivalent states
Let $A = (Q, \Sigma, I, T, T_1, \ldots, T_n)$ be a GBA and $s_1, s_2 \in Q$. We can merge $s_1$ and $s_2$ if they have the same outgoing transitions:
- $\forall a \in \Sigma, \forall s \in Q$,
- $(s_1, a, s) \in T \iff (s_2, a, s) \in T$
- and $(s_1, a, s) \in T_i \iff (s_2, a, s) \in T_i$ for all $1 \leq i \leq n$. 

From LTL to Büchi automata

Definition: SBT for LTL modalities (cont.)
- $A_U$ from $B^2$ to $B$: $0,1/1 0,0/0 1,0/1 0,1/0 0,0/1$
- $A_S$ from $B^2$ to $B$: $0,0/0 0,1/0 1,0/0 1,1/1 0,1/1 1,0/1$
Other constructions

- Tableau construction. See for instance [13, Wolper 85]
  - : Easy definition, easy proof of correctness
  - : Works both for future and past modalities
- Using Very Weak Alternating Automata [15, Gastin & Oddoux 01].
  - : Very efficient
  - : Only for future modalities
- Using reduction rules [16, Demri & Gastin 10].
  - : Efficient and produces small automata
  - : Can be used by hand on real examples
- The domain is still very active.

Outline

Introduction

Models

Specifications

Satisfiability and Model checking for LTL

- Büchi automata
- From LTL to BA
  - Decidability and Complexity

Branching Time Specifications

Satisfiability for LTL over \((\mathbb{N}, <)\)

Let \(AP\) be the set of atomic propositions and \(\Sigma = 2^{AP}\).

Definition: Satisfiability problem

| Input | A formula \(\varphi \in \text{LTL}(AP, S, U)\) |
| Question | Existence of \(w \in \Sigma^\omega\) and \(i \in \mathbb{N}\) such that \(w, i \models \varphi\). |

Definition: Initial Satisfiability problem

| Input | A formula \(\varphi \in \text{LTL}(AP, S, U)\) |
| Question | Existence of \(w \in \Sigma^\omega\) such that \(w, 0 \models \varphi\). |

Remark: \(\varphi\) is satisfiable iff \(F \varphi\) is initially satisfiable.

Definition: (Initial) validity

\(\varphi\) is valid iff \(\neg \varphi\) is not satisfiable.

Theorem [14, Sistla, Clarke 85], [12, Lichtenstein & Pnueli 85]
The satisfiability problem for LTL is PSPACE-complete.

Model checking for LTL

Definition: Model checking problem

| Input | A Kripke structure \(M = (S, T, I, AP, \ell)\) |
| A formula \(\varphi \in \text{LTL}(AP, S, U)\) |
| Question | Does \(M \models \varphi\) ? |

- Universal MC: \(M \models_\forall \varphi\) if \(\ell(\sigma), 0 \models \varphi\) for all initial infinite run of \(M\).
- Existential MC: \(M \models_\exists \varphi\) if \(\ell(\sigma), 0 \models \varphi\) for some initial infinite run of \(M\).

\(M \models_\forall \varphi\) iff \(M \not\models_\exists \neg \varphi\)

Theorem [14, Sistla, Clarke 85], [12, Lichtenstein & Pnueli 85]
The Model checking problem for LTL is PSPACE-complete.
MC^3(U) \leq_P \text{SAT}(U) \ [14, \ Sistla \ & \ Clarke \ 85]

Let \( M = (S, T, I, AP, \ell) \) be a Kripke structure and \( \varphi \in \text{LTL}(AP, U) \)

Introduce new atomic propositions: \( AP_S = \{ at_s \mid s \in S \} \)
Define \( AP' = AP \uplus AP_S \quad \Sigma' = 2^{AP'} \quad \pi : \Sigma' \to \Sigma' \) by \( \pi(a) = a \cap AP \).
Let \( w \in \Sigma' \). We have \( w \models \varphi \iff \pi(w) \models \varphi \)

Define \( \psi_M \in \text{LTL}(AP', X, F') \) of size \( O(|M|^2) \) by
\[
\psi_M = \left( \bigwedge_{s \in I} \neg at_s \right) \wedge G' \left( \bigwedge_{s \in S} \left( \bigwedge_{t \in T(s)} \neg at_t \wedge \bigwedge_{t \in T(s)} p \wedge \bigwedge_{t \in T(s)} \neg p \wedge \bigwedge_{t \in T(s)} X at_t \right) \right)
\]
Let \( w = a_0a_1a_2 \cdots \in \Sigma' \). Then, \( w \models \psi_M \iff \) there exists an initial infinite run \( \sigma \)
M such that \( \pi(w) = \ell(\sigma) \) and \( a_i \in AP_S = \{ at_s \} \) for all \( i \geq 0 \).
Therefore, \( M \models \varphi \) \iff \( \psi_M \wedge \varphi \) is satisfiable
\( M \models \neg \varphi \) \iff \( \psi_M \wedge \neg \varphi \) is not satisfiable
Remark: we also have \( \text{MC}^3(X, F') \leq_P \text{SAT}(X, F') \).

QBF \leq_P \text{MC}^3(U') \ [14, \ Sistla \ & \ Clarke \ 85]

Let \( \gamma = Q_1x_1 \cdots Q_nx_n \gamma' \) with \( \gamma' = \bigwedge_{1 \leq i \leq m} \bigvee_{1 \leq j \leq k_i} a_{ij} \)
\( Q_i \in \{ \forall, \exists \} \) and \( a_{ij} \in \{ x_1, \neg x_1, \ldots, x_n, \neg x_n \} \).

Consider the KS \( M \):

QBF Quantified Boolean Formulae

Definition: QBF
Input: A formula \( \gamma = Q_1x_1 \cdots Q_nx_n \gamma' \) with \( \gamma' = \bigwedge_{1 \leq i \leq m} \bigvee_{1 \leq j \leq k_i} a_{ij} \)
\( Q_i \in \{ \forall, \exists \} \) and \( a_{ij} \in \{ x_1, \neg x_1, \ldots, x_n, \neg x_n \} \).
Question: Is \( \gamma \) valid?

Definition:
An assignment of the variables \( \{ x_1, \ldots, x_n \} \) is a word \( v = v_1 \cdots v_n \in \{ 0, 1 \}^n \).
We write \( v[i] \) for the prefix of length \( i \).
Let \( V \subseteq \{ 0, 1 \}^n \) be a set of assignments.
- \( V \) is valid (for \( \gamma' \) if \( v \models \gamma' \) for all \( v \in V \),
- \( V \) is closed (for \( \gamma' \) if \( \forall v \in V, \forall 1 \leq i \leq n \) s.t. \( Q_i = \forall \),
\[ \exists v' \in V \text{ s.t. } v'[i-1] = v'[i] \text{ and } \{ v, v' \} = \{ 0, 1 \} \].

Proposition:
\( \gamma \) is valid \iff \( \exists V \subseteq \{ 0, 1 \}^n \) s.t. \( V \) is nonempty valid and closed

Complexity of LTL

Theorem: Complexity of LTL
The following problems are PSPACE-complete:
- \( \text{SAT}(LTL(S, U)) \), \( \text{MC}^3(LTL(S, U)) \)
- \( \text{SAT}(LTL(X, F')) \), \( \text{MC}^3(LTL(X, F')) \)
- \( \text{SAT}(LTL(U')) \), \( \text{MC}^3(LTL(U')) \)
- The restriction of the above problems to a unique propositional variable

The following problems are NP-complete:
- \( \text{SAT}(LTL(F')) \), \( \text{MC}^3(LTL(F')) \)