**Büchi automata 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 $\mu : T \to B$ is the output function. It computes the relation

$[A] = \{(u, v) \in A^* \times B^* | \exists p = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots \text{ accepting run with } u = a_0 a_1 a_2 \ldots \text{ and } v = \mu(p), \text{ i.e., } v = b_i b_i b_i \ldots \text{ with } b_i = \mu(q_i, a_i, q_{i+1}) \text{ for } i \geq 0\}$

If $(Q, A, I, T, F)$ is unambiguous then $[A] : A^* \to B^*$ is a (partial) function, in which case we also write $[A](u) = v$ for $(u, v) \in [A]$. We will also use SGBT: synchronous transducers with generalized Büchi acceptance.

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

![Diagram of left shift]

**Product of Büchi transducers**

**Definition:** Product

Let $A, B, C$ be alphabets. Let $A = (Q, A, I, T, (F_i)_i, \mu)$ be an SGBT from $A$ to $B$. Let $A' = (Q', A', I', T', (F'_i)_i, \mu')$ be an SGBT from $A$ to $C$. Then $A \times A' = (Q \times Q', A \times A', I \times I', T \times T', (F_i \times F'_i)_i, (Q \times F'_i)_i, \mu \times \mu')$ be an SGBT from $A \times A'$ to $B \times C$.

**Proposition:** Product

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

1. We have $[A \times A'] = \{(u, v, v') | (u, v) \in [A] \text{ and } (u, v') \in [A']\}$.
2. If $(Q, A, I, T, (F_i)_i)$ and $(Q', A', I', T', (F'_i)_i)$ are unambiguous (resp. prophetic) then $(Q \times Q', A \times A', I \times I', T \times T', (F_i \times F'_i)_i, (Q \times F'_i)_i)$ is also unambiguous (resp. prophetic), and
   
   $\forall u \in A^*$ we have $[A \times A'](u) = ([A](u), [A'](u))$.

**Composition of Büchi transducers**

**Definition:** Composition

Let $A, B, C$ be alphabets. Let $A = (Q, A, I, T, (F_i)_i, \mu)$ be an SGBT from $A$ to $B$. Let $A' = (Q', B, I', T', (F'_i)_i, \mu')$ be an SGBT from $B$ to $C$.

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

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

iff

$$\tau = p \xrightarrow{a} q \in T \text{ and } \tau' = p' \xrightarrow{a(\tau')} q' \in T' \text{ and } c = \mu'(\tau')$$

is an SGBT from $A$ to $C$.

**Proposition:** Composition

1. We have $[A \cdot A'] = [A] \cdot [A']$.
2. If $(Q, A, I, T, (F_i)_i)$ and $(Q', B, I', T', (F'_i)_i)$ are unambiguous (resp. prophetic) then $(Q \times Q', A \times A', I \times I', T \times T', (F_i \times F'_i)_i, (Q \times F'_i)_i)$ is also unambiguous (resp. prophetic), and

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

**Subalphabets of $\Sigma = 2^\AP$**

**Definition:**

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

For instance, for $p, q \in \AP$,

$\Sigma_p = \{a \in \Sigma | p \in a\}$ and $\Sigma_{\neg p} = \Sigma \setminus \Sigma_p$

$\Sigma_{p \land q} = \Sigma_p \cap \Sigma_q$ and $\Sigma_{p \lor q} = \Sigma_p \cup \Sigma_q$

$\Sigma_{p \land \neg q} = \Sigma_p \setminus \Sigma_q$. . .

**Notation:**

In automata, $s \xrightarrow{a} s'$ stands for the set of transitions $\{s\} \times \Sigma \times \{s'\}$.

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

**Example:** $G(p \rightarrow F q)$

![Diagram of G(p → F q)]
Semantics of \textit{LTL} with sequential functions

\textbf{Definition:} Semantics of $\varphi \in \text{LTL}(\text{AP}, \text{SU}, \text{SS})$

Let $\Sigma = 2^{\text{AP}}$ and $B = \{0, 1\}$.

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

Example:

$\llbracket p \text{SU} q \rrbracket (\emptyset \{\{q\}\{p\}\{p\}\{q\}\{p, q\}\emptyset^*}) = 1001110110$

$\llbracket X p \rrbracket (\emptyset \{\{q\}\{p\}\{q\}\{p\}\{p, q\}\emptyset^*}) = 0101100110$

$\llbracket F p \rrbracket (\emptyset \{\{q\}\{p\}\{q\}\{p\}\{p, q\}\emptyset^*}) = 1111111110$

The aim is to compute $\llbracket \varphi \rrbracket$ with synchronous Büchi transducers (actually, SGBT).

Synchronous Büchi transducer for $p \text{SU} q$

Example: An SBT for $[p \text{SU} q]$

Lemma: The input BA is prophetic

For all $u = a_0 a_1 a_2 \cdots \in \Sigma^*$, there is a unique final run $\rho = s_0, a_0, s_1, a_1, s_2, a_2, a_3, \ldots$ of $A$ on $u$.

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

Hence, the SBT computes $[p \text{SU} q]$.

Synchronous Büchi transducer for $p \text{U} q$

Example: An SBT for $[p \text{U} q]$

The automaton is prophetic (same input BA as for $p \text{SU} q$). This SBT computes $[p \text{U} q]$.

Special cases of Until: Future and Next

Example: $F q = \top \text{U} q$ and $X q = \bot \text{SU} q$

Exercise: Give SBT’s for the following formulae:

$SF q, SG q, p SR q, p SS q, Y q, G q, p R q, p S q, G(p \to F q)$.
From LTL to Büchi automata

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

Remark: where
- For each
- Theorem: Correctness of the translation

From LTL to Büchi automata

Definition: SBT for LTL modalities (cont.)
- $A_{SU}$ from $B^2$ to $B$:
  - Prophetic
  - Deterministic
  - Not prophetic

From LTL to Büchi automata

Definition: Translation from LTL to SGBT
For each $\xi \in \text{LTL}(\text{AP}, \text{SU}, \text{SS})$ we define inductively an SGBT $A_\xi$ as follows:
- $A_T$ and $A_p$ for $p \in \text{AP}$ are already defined
- $A_\land = A_\land \circ A_p$
- $A_\lor\lor\lor = A_\lor \circ (A_\land \times A_p)$
- $A_{\text{ASS}} = A_{\text{SS}} \circ (A_\land \times A_p)$
- $A_{\text{SUS}} = A_{\text{SU}} \circ (A_\land \times A_p)$

Theorem: Correctness of the translation
For each $\xi \in \text{LTL}(\text{AP}, \text{SU}, \text{SS})$, we have $[A_\xi] = [\xi]$ and $A_\xi$ is unambiguous.
Moreover, the number of states of $A_\xi$ is at most $2^{\#\text{SS}} \cdot 3^{\#\text{SU}}$
the number of acceptance conditions is $|\xi|_{\text{SS}}$
where $|\xi|_{\text{SS}}$ (resp. $|\xi|_{\text{SU}}$) is the number of SS (resp. SU) occurring in $\xi$.

Remark:
- If a subformula $\varphi$ occurs several times in $\xi$, we only need one copy of $A_\varphi$.
- We may also use automata for other modalities: $A_\forall$ (2 states), $A_\forall$, ...

Useful simplifications

Reducing the number of temporal subformulae
- $(X \varphi) \land (X \psi) \equiv X(\varphi \land \psi)$
- $(X \varphi) \land (G \psi) \equiv (X \varphi) \land (G \psi)$
- $GF \varphi \land GF \psi \equiv GF(\varphi \land \psi)$
- $(\varphi_1 \land \varphi_2) \land (\varphi_3 \land \varphi_4) \equiv (\varphi_1 \land \varphi_2) \land (\varphi_3 \land \varphi_4)$
- $(\varphi_1 \land \varphi_3) \lor (\varphi_2 \land \varphi_4) \equiv (\varphi_1 \lor \varphi_2) \lor (\varphi_3 \lor \varphi_4)$

Merging equivalent states
Let $A = (Q, \Sigma, I, T, (F_i)_i, \mu)$ be an SGBT and $s_1, s_2 \in Q$.
We can merge $s_1$ and $s_2$ if they satisfy the same final conditions:
- $s_1 \in F_i \iff s_2 \in F_i$ for all $i$
and they have the same outgoing transitions: $\forall a \in \Sigma, \forall s \in Q$,
- $\tau_1 = (s_1, a, s) \in T \iff \tau_2 = (s_2, a, s) \in T$ and $\mu(\tau_1) = \mu(\tau_2)$
Other constructions

- Tableau construction. See for instance [15, Wolper 85]
  + : Easy definition, easy proof of correctness
  + : Works both for future and past modalities
  − : Inefficient without strong optimizations
- Using Very Weak Alternating Automata [16, Gastin & Oddoux 01].
  + : Very efficient
  − : Only for future modalities

Online tool: http://www.lsv.ens-cachan.fr/~gastin/ltl2ba/

- Using reduction rules [6, Demri & Gastin 10].
  + : Efficient and produces small automata
  + : Can be used by hand on real examples
  − : Only for future modalities

The domain is still very active.

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

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

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

Definition: Satisfiability problem

Input: A formula \(\varphi \in \text{LTL}(\mathrm{AP}, \text{SU, SS})\)

Question: Existence of \(w \in \Sigma^*\) and \(i \in \mathbb{N}\) such that \(w, i \models \varphi\).

Definition: Initial Satisfiability problem

Input: A formula \(\varphi \in \text{LTL}(\mathrm{AP}, \text{SU, SS})\)

Question: Existence of \(w \in \Sigma^*\) 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 [10, Sistla, Clarke 85], [9, Lichtenstein & Pnueli 85]
The satisﬁability problem for LTL is PSPACE-complete.

Model checking for LTL

Definition: Model checking problem

Input: A Kripke structure \(M = (S, T, I, \mathrm{AP}, \ell)\)
A formula \(\varphi \in \text{LTL}(\mathrm{AP}, \text{SU, SS})\)

Question: Does \(M \models \varphi\) ?

Universal MC: \(M \models \varphi\) if \(\ell(\sigma), 0 \models \varphi\) for all initial infinite runs of \(M\).

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

\[M \models \varphi\] iff \[M \models \neg \exists \neg \varphi\]

Theorem [10, Sistla, Clarke 85], [9, Lichtenstein & Pnueli 85]
The Model checking problem for LTL is PSPACE-complete.
**MC$^3$(SU) $\leq_P$ SAT(SU)**

[10, Sistla & Clarke 85]

Let $M = (S, T, I, AP, \ell)$ be a Kripke structure and $\varphi \in $ LTL(AP, SU)

Introduce new atomic propositions: $AP_S = \{ at_s \mid s \in S \}$

Define $AP' = AP \uplus AP_S$, $\Sigma' = 2^{AP'}$, $\pi: \Sigma' \rightarrow \Sigma$ by $\pi(a) = a \cap AP$.

Let $w \in \Sigma^\omega$. We have $w \models \varphi$ iff $\pi(w) \models \varphi$.

Define $\psi_M \in \text{LTL}(AP', X, F)$ of size $O(|M|)$ by

$$
\psi_M = \left( \bigwedge_{s \in S} at_s \right) \land \left( \bigwedge_{t \in T(s)} \neg at_t \land \bigwedge_{p \in \ell(t)} p \land \bigwedge_{p \in \ell(t')} \neg p \land \bigvee_{t \in T} X at_t \right).
$$

Let $w = a_0a_1a_2\cdots \in \Sigma^\omega$. Then, $w \models \psi_M$ iff there exists an initial infinite run $\sigma = s_0, s_1, s_2, \ldots$ of $M$ such that $\pi(w) = \ell(\sigma)$ and $a_i \cap AP_S = \{ at_s \}$ for all $i \geq 0$.

Therefore, $M \models \varphi$ iff $\psi_M \land \varphi$ is initially satisfiable.

$M \models \neg \varphi$ iff $\psi_M \land \neg \varphi$ is not initially satisfiable.

Remark: we also have $MC^3(SU) \leq_P$ SAT(SU).

**QBF Quantified Boolean Formulae**

**Definition:** QBF

**Input:** A formula $\gamma = Q_1x_1 \cdots Q_nx_n \gamma'$ with $\gamma' = \bigwedge_{1 \leq i \leq \ell} \bigvee_{1 \leq j \leq k_i} a_{ij}$ (CNF)

$Q_i \in \{ \forall, \exists \}$ 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. $Q_i = \forall$, $\exists v' \in V$ s.t. $v[i-1] = v'[i-1]$ and $v'[i] = 1 - v_i$.

**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:
- SAT(LTL(SU, SS)), MC$^3$(LTL(SU, SS)), MC$^5$(LTL(SU, SS))
- SAT(LTL(X, F)), MC$^3$(LTL(X, F)), MC$^5$(LTL(X, F))
- SAT(LTL(U)), MC$^3$(LTL(U)), MC$^5$(LTL(U))
- The restriction of the above problems to a unique propositional variable

The following problems are NP-complete:
- SAT(LTL(F)), MC$^3$(LTL(F))
**Complexity of CTL^***

**Definition: Syntax of the Computation Tree Logic CTL^***

\[ \varphi ::= \bot \mid p \mid (p \in AP) \mid \neg \varphi \mid \varphi \lor \psi \mid X \varphi \mid \varphi U \psi \mid E \varphi \mid A \varphi \]

**Theorem**

The model checking problem for CTL^* is PSPACE-complete

**Proof:**

PSPACE-hardness: follows from LTL \( \subseteq \) CTL^*.

PSPACE-easiness: reduction to LTL-model checking by inductive eliminations of path quantifications.

---

**Satisfiability for CTL^***

**Definition: SAT(CTL^*)**

**Input:** A formula \( \varphi \in \text{CTL}^* \)

**Question:** Existence of a model \( M \) and a run \( \sigma \) such that \( M, \sigma, 0 \models \varphi \)

**Theorem**

The satisfiability problem for CTL^* is 2-EXPTIME-complete