Outline

Model checking of CTL

Theorem: MC for CTL

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure and $\varphi \in \mathrm{CTL}$ a formula.
The model checking problem $M \models_\exists \varphi$ is decidable in time $\mathcal{O}(|M| \cdot |\varphi|)$

Proof:

Compute $[\![\varphi]\!] = \{s \in S \mid M, s \models \varphi\}$ by induction on the formula.
The set $[\![\varphi]\!]$ is represented by a boolean array: $L[\varphi][s] = \top$ if $s \in [\![\varphi]\!]$.
The labelling ℓ is encoded in L: for $p \in \mathrm{AP}$ we have $L[p][s] = \top$ if $p \in \ell(s)$.

Model checking of CTL

Definition: procedure semantics(φ)

case $\varphi = \neg\varphi_1$
semantics(φ_1)
$[\![\varphi]\!] := S \setminus [\![\varphi_1]\!]$ — $\mathcal{O}(|S|)$

case $\varphi = \varphi_1 \vee \varphi_2$
semantics(φ_1); semantics(φ_2)
$[\![\varphi]\!] := [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$ — $\mathcal{O}(|S|)$

case $\varphi = EX\varphi_1$
semantics(φ_1)
$[\![\varphi]\!] := \emptyset$ — $\mathcal{O}(|S|)$
for all $(s,t) \in T$ do if $t \in [\![\varphi_1]\!]$ then $[\![\varphi]\!] := [\![\varphi]\!] \cup \{s\}$ — $\mathcal{O}(|T|)$

case $\varphi = AX\varphi_1$
semantics(φ_1)
$[\![\varphi]\!] := S$ — $\mathcal{O}(|S|)$
for all $(s,t) \in T$ do if $t \notin [\![\varphi_1]\!]$ then $[\![\varphi]\!] := [\![\varphi]\!] \setminus \{s\}$ — $\mathcal{O}(|T|)$

Model checking of CTL

Definition: procedure semantics(φ)

case $\varphi = E\varphi_1 \mathsf{U} \varphi_2$ — $\mathcal{O}(|S| + |T|)$
semantics(φ_1); semantics(φ_2)
$L := [\![\varphi_2]\!]$ // the "todo" set L is imlemented with a list — $\mathcal{O}(|S|)$
$Z := [\![\varphi_2]\!]$ // the "result" is computed in the array Z — $\mathcal{O}(|S|)$
while $L \neq \emptyset$ do — $|S|$ times
Invariant: $L \subseteq Z$ and
$[\![\varphi_2]\!] \cup ([\![\varphi_1]\!] \cap T^{-1}(Z \setminus L)) \subseteq Z \subseteq [\![\mathsf{E}\,\varphi_1 \mathsf{U} \varphi_2]\!]$
take $t \in L$; $L := L \setminus \{t\}$ — $\mathcal{O}(1)$
for all $s \in T^{-1}(t)$ do — $|T|$ times
if $s \in [\![\varphi_1]\!] \setminus Z$ then $L := L \cup \{s\}$; $Z := Z \cup \{s\}$ — $\mathcal{O}(1)$
od
$[\![\varphi]\!] := Z$ — $\mathcal{O}(|S|)$

Z is only used to make the invariant clear. It can be replaced by $[\![\varphi]\!]$.

Model checking of CTL

Definition: procedure semantics(φ)

case $\varphi = A\varphi_1 \, \mathsf{U} \, \varphi_2$ — $\mathcal{O}(|S| + |T|)$

semantics(φ_1); semantics(φ_2)

$L := [\![\varphi_2]\!]$ // the "todo" set L is imlemented with a list — $\mathcal{O}(|S|)$

$Z := [\![\varphi_2]\!]$ // the "result" is computed in the array Z — $\mathcal{O}(|S|)$

for all $s \in S$ do $c[s] := |T(s)|$ — $\mathcal{O}(|S|)$

while $L \neq \emptyset$ do — $|S|$ times

Invariant: $L \subseteq Z$ and
$\forall s \in S,\ c[s] = |T(s) \setminus (Z \setminus L)|$ and
$[\![\varphi_2]\!] \cup ([\![\varphi_1]\!] \cap \{s \in S \mid c[s] = 0\}) \subseteq Z \subseteq [\![\mathsf{A}\, \varphi_1 \, \mathsf{U} \, \varphi_2]\!]$

take $t \in L$; $L := L \setminus \{t\}$ — $\mathcal{O}(1)$

for all $s \in T^{-1}(t)$ do — $|T|$ times

$c[s] := c[s] - 1$ — $\mathcal{O}(1)$

if $c[s] = 0 \wedge s \in [\![\varphi_1]\!] \setminus Z$ then $L := L \cup \{s\}$; $Z := Z \cup \{s\}$ — $\mathcal{O}(1)$

od

$[\![\varphi]\!] := Z$ — $\mathcal{O}(|S|)$

Z is only used to make the invariant clear. It can be replaced by $[\![\varphi]\!]$.

Complexity of CTL

Definition: SAT(CTL)

Input: A formula $\varphi \in \mathrm{CTL}$

Question: Existence of a model M and a state s such that $M, s \models \varphi$?

Theorem: Complexity

- The model checking problem for CTL is PTIME-complete.
- The satisfiability problem for CTL is EXPTIME-complete.

Fairness

Example: Fairness

Only fair runs are of interest

- Each process is enabled infinitely often: $\bigwedge_i \mathsf{G}\,\mathsf{F}\,\mathrm{run}_i$
- No process stays ultimately in the critical section: $\bigwedge_i \neg\,\mathsf{F}\,\mathsf{G}\,\mathrm{CS}_i = \bigwedge_i \mathsf{G}\,\mathsf{F}\,\neg \mathrm{CS}_i$

Definition: Fair Kripke structure

$M = (S, T, I, \mathrm{AP}, \ell, F_1, \ldots, F_n)$ with $F_i \subseteq S$.

An infinite run σ is fair if it visits infinitely often each F_i

fair-CTL

Definition: Syntax of fair-CTL

$\varphi ::= \bot \mid p \ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}_f \, \mathsf{X} \, \varphi \mid \mathsf{A}_f \, \mathsf{X} \, \varphi \mid \mathsf{E}_f \, \varphi \, \mathsf{U} \, \varphi \mid \mathsf{A}_f \, \varphi \, \mathsf{U} \, \varphi$

Definition: Semantics as a fragment of CTL*

Let $M = (S, T, I, \mathrm{AP}, \ell, F_1, \ldots, F_n)$ be a fair Kripke structure.

Then, $\mathsf{E}_f\, \varphi = \mathsf{E}(\mathrm{fair} \wedge \varphi)$ and $\mathsf{A}_f\, \varphi = \mathsf{A}(\mathrm{fair} \rightarrow \varphi)$

where $\mathrm{fair} = \bigwedge_i \mathsf{G}\,\mathsf{F}\, F_i$

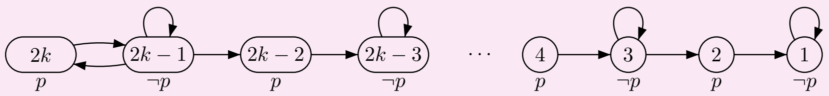
Remark: $\mathsf{A}_f\, \varphi = \neg\, \mathsf{E}_f\, \neg\varphi$

Lemma: fair-CTL cannot be expressed in CTL

fair-CTL

Proof: fair-CTL cannot be expressed in CTL

Consider the Kripke structure M_k defined by:



- $M_k, 2k \models \mathsf{E\,G\,F}\,p$ but $M_k, 2k-2 \not\models \mathsf{E\,G\,F}\,p$
- If $\varphi \in \mathrm{CTL}$ and $|\varphi| \le m \le k$ then

$$M_k, 2k \models \varphi \text{ iff } M_k, 2m \models \varphi$$

$$M_k, 2k-1 \models \varphi \text{ iff } M_k, 2m-1 \models \varphi$$

If the fairness condition is $\ell^{-1}(p)$ then $\mathsf{E}_f\,\top$ cannot be expressed in CTL.

Model checking of fair-CTL

Theorem

The model checking problem for fair-CTL is decidable in time $\mathcal{O}(|M| \cdot |\varphi|)$

Proof: Computation of $\mathrm{FAIR} = \{s \in S \mid M, s \models \mathsf{E}_f\,\top\}$

Compute the SCC of M with Tarjan's algorithm (in time $\mathcal{O}(|M|)$).

Let S' be the union of the (non trivial) SCCs which intersect each F_i.

Then, FAIR is the set of states that can reach S'.

Note that reachability can be computed in linear time.

Model checking of fair-CTL

Proof: Reductions

$\mathsf{E}_f\,\mathsf{X}\,\varphi = \mathsf{E\,X}(\mathrm{FAIR} \wedge \varphi)$ and $\mathsf{E}_f\,\varphi\,\mathsf{U}\,\psi = \mathsf{E}\,\varphi\,\mathsf{U}\,(\mathrm{FAIR} \wedge \psi)$

It remains to deal with $\mathsf{A}_f\,\varphi\,\mathsf{U}\,\psi$.

We have $\mathsf{A}_f\,\varphi\,\mathsf{U}\,\psi = \neg\,\mathsf{E}_f\,\mathsf{G}\,\neg\psi \wedge \neg\,\mathsf{E}_f(\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi))$

Hence, we only need to compute the semantics of $\mathsf{E}_f\,\mathsf{G}\,\varphi$.

Proof: Computation of $\mathsf{E}_f\,\mathsf{G}\,\varphi$

Let M_φ be the restriction of M to $[\![\varphi]\!]_f$.

Compute the SCC of M_φ with Tarjan's algorithm (in linear time).

Let S' be the union of the (non trivial) SCCs of M_φ which intersect each F_i.

Then, $M, s \models \mathsf{E}_f\,\mathsf{G}\,\varphi$ iff $M, s \models \mathsf{E}\,\varphi\,\mathsf{U}\,S'$ iff $M_\varphi, s \models \mathsf{EF}\,S'$.

This is again a reachability problem which can be solved in linear time.

Büchi automata

Definition:

A Büchi automaton (BA) is a tuple $\mathcal{A} = (Q, \Sigma, I, T, F)$ where

- Q: finite set of states
- Σ: 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 $i \ge 0$.

ρ is accepting if $q_0 \in I$ and $q_i \in F$ for infinitely many i's.

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

A language $L \subseteq \Sigma^\omega$ is ω-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:

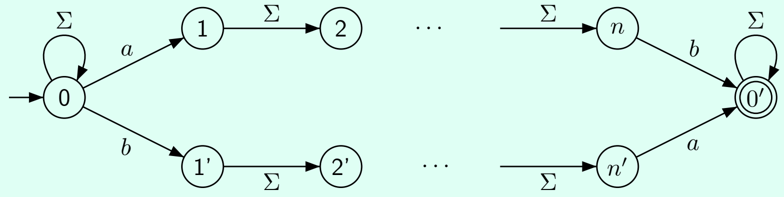
Büchi automata

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.

Büchi automata

Theorem: Büchi

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

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

Exercises:

1. Construct a BA for $\mathcal{L}(\varphi)$ where φ is the $\mathrm{FO}_\Sigma(<)$ sentence

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

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

$$\begin{aligned} \mathrm{next}(L_1) &= \Sigma \cdot L_1 \\ \mathrm{SUntil}(L_1, L_2) &= \{uv \in \Sigma^\omega \mid u \in \Sigma^+ \wedge v \in L_2 \wedge \\ &\qquad u''v \in L_1 \text{ for all } u', u'' \in \Sigma^+ \text{ with } u = u'u''\} \end{aligned}$$

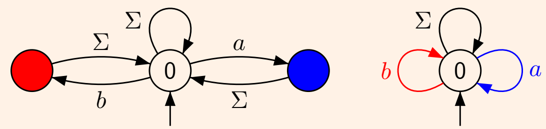
Generalized Büchi automata

Definition: acceptance on states or on transitions

$\mathcal{A} = (Q, \Sigma, I, T, F_1, \ldots, F_n)$ with $F_i \subseteq Q$.
An infinite run σ is successful if it visits infinitely often each F_i.

$\mathcal{A} = (Q, \Sigma, I, T, T_1, \ldots, T_n)$ with $T_i \subseteq T$.
An infinite run σ 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.

Büchi automata with output

Definition: SBT: Synchronous (letter to letter) Büchi transducer

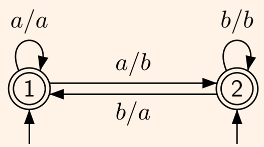
Let A and B be two alphabets.
A synchronous Büchi transducer from A to B is a tuple $\mathcal{A} = (Q, A, I, T, F, \mu)$ where (Q, A, I, T, F) is a Büchi automaton (input) and $\mu : T \to B$ is the output function.
It computes the relation

$$[\![\mathcal{A}]\!] = \{(u, v) \in A^\omega \times B^\omega \mid \exists\, \rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \dots \text{ accepting run with } u = a_0 a_1 a_2 \cdots \text{ and } v = \mu(\tau_0)\mu(\tau_1)\mu(\tau_2)\cdots \text{ where } \tau_i = (q_i, a_i, q_{i+1})\}$$

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

We will also use SGBT: synchronous transducers with generalized Büchi acceptance.

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



Composition of Büchi transducers

Definition: Composition

Let A, B, C be alphabets.
Let $\mathcal{A} = (Q, A, I, T, (F_i)_i, \mu)$ be an SGBT from A to B.
Let $\mathcal{A}' = (Q', B, I', T', (F'_j)_j, \mu')$ be an SGBT from B to C.
Then $\mathcal{A} \cdot \mathcal{A}' = (Q \times Q', A, I \times I', T'', (F_i \times Q')_i, (Q \times F'_j)_j, \mu'')$ is defined by:

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

iff

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

$\mathcal{A} \cdot \mathcal{A}'$ is an SGBT from A to C.
When the transducers define functions, we also denote the composition by $\mathcal{A}' \circ \mathcal{A}$.

Proposition: Composition

1. We have $[\![\mathcal{A} \cdot \mathcal{A}']\!] = [\![\mathcal{A}]\!] \cdot [\![\mathcal{A}']\!]$.
2. If $(Q, A, I, T, (F_i)_i)$ and $(Q', B, I', T', (F'_j)_j)$ are unambiguous then $(Q \times Q', A, I \times I', T'', (F_i \times Q')_i, (Q \times F'_j)_j)$ is also unambiguous.
   Then, $\forall u \in A^\omega$ we have $[\![\mathcal{A}' \circ \mathcal{A}]\!](u) = [\![\mathcal{A}']\!]([\![\mathcal{A}]\!](u))$.

Product of Büchi transducers

Definition: Product

Let A, B, C be alphabets.
Let $\mathcal{A} = (Q, A, I, T, (F_i)_i, \mu)$ be an SGBT from A to B.
Let $\mathcal{A}' = (Q', A, I', T', (F'_j)_j, \mu')$ be an SGBT from A to C.
Then $\mathcal{A} \times \mathcal{A}' = (Q \times Q', A, I \times I', T'', (F_i \times Q')_i, (Q \times F'_j)_j, \mu'')$ is defined by:

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

iff

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

$\mathcal{A} \times \mathcal{A}'$ is an SGBT from A to $B \times C$.

Proposition: Product

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

1. We have $[\![\mathcal{A} \times \mathcal{A}']\!] = \{(u, v, v') \mid (u, v) \in [\![\mathcal{A}]\!] \text{ and } (u, v') \in [\![\mathcal{A}']\!]\}$.
2. If $(Q, A, I, T, (F_i)_i)$ and $(Q', A, I', T', (F'_j)_j)$ are unambiguous then $(Q \times Q', A, I \times I', T'', (F_i \times Q')_i, (Q \times F'_j)_j)$ is also unambiguous.
   Then, $\forall u \in A^\omega$ we have $[\![\mathcal{A} \times \mathcal{A}']\!](u) = ([\![\mathcal{A}]\!](u), [\![\mathcal{A}']\!](u))$.

Subalphabets of $\Sigma = 2^{\mathrm{AP}}$

Definition:

For a propositional formula ξ over AP, we let $\Sigma_\xi = \{a \in \Sigma \mid a \models \xi\}$.
For instance, for $p, q \in \mathrm{AP}$,

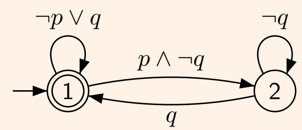
- $\Sigma_p = \{a \in \Sigma \mid p \in a\}$ and $\Sigma_{\neg p} = \Sigma \setminus \Sigma_p$
- $\Sigma_{p \wedge q} = \Sigma_p \cap \Sigma_q$ and $\Sigma_{p \vee q} = \Sigma_p \cup \Sigma_q$
- $\Sigma_{p \wedge \neg q} = \Sigma_p \setminus \Sigma_q$ ...

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:



Semantics of LTL with sequential functions

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

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

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

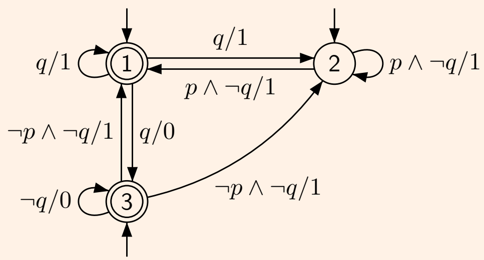
Example:

$$[\![p \,\mathsf{SU}\, q]\!](\emptyset\{q\}\{p\}\emptyset\{p\}\{p\}\{q\}\emptyset\{p\}\{p,q\}\emptyset^\omega) = 10011101100^\omega$$
$$[\![\mathsf{X}\, p]\!](\emptyset\{q\}\{p\}\emptyset\{p\}\{p\}\{q\}\emptyset\{p\}\{p,q\}\emptyset^\omega) = 01011001100^\omega$$
$$[\![\mathsf{F}\, p]\!](\emptyset\{q\}\{p\}\emptyset\{p\}\{p\}\{q\}\emptyset\{p\}\{p,q\}\emptyset^\omega) = 11111111110^\omega$$

The aim is to compute $[\![\varphi]\!]$ with Büchi transducers.

Synchronous Büchi transducer for $p \,\mathsf{SU}\, q$

Example: An SBT for $[\![p \,\mathsf{SU}\, q]\!]$



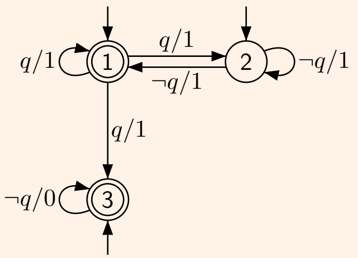
Lemma: The input BA is prophetic

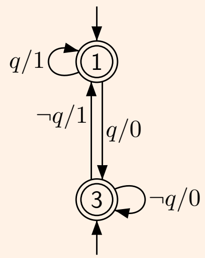
For all $u = a_0 a_1 a_2 \cdots \in \Sigma^\omega$,
there is a unique accepting run $\rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots$ of $\mathcal{A}$ on u.

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

Special cases of Until: Future and Next

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



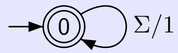


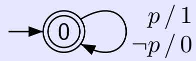
Exercise: Give SBT's for the following formulae:

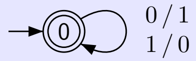
$p \,\mathsf{U}\, q$, $\mathsf{G}\, q$, $p \,\mathsf{R}\, q$, $p \,\mathsf{SR}\, q$, $p \,\mathsf{S}\, q$, $p \,\mathsf{SS}\, q$, $\mathsf{G}(p \to \mathsf{F}\, q)$.

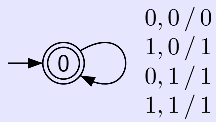
From LTL to Büchi automata

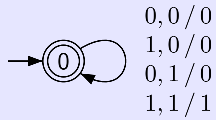
Definition: SBT for LTL modalities

- $\mathcal{A}_\top$ from Σ to $\mathbb{B} = \{0,1\}$: 

- $\mathcal{A}_p$ from Σ to $\mathbb{B} = \{0,1\}$: 

- $\mathcal{A}_\neg$ from $\mathbb{B}$ to $\mathbb{B}$: 

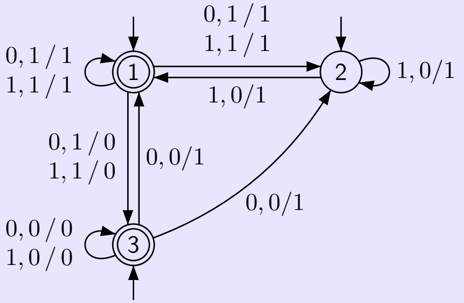
- $\mathcal{A}_\vee$ from $\mathbb{B}^2$ to $\mathbb{B}$: 

- $\mathcal{A}_\wedge$ from $\mathbb{B}^2$ to $\mathbb{B}$: 

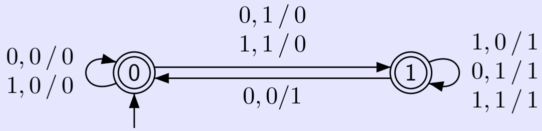
From LTL to Büchi automata

Definition: SBT for LTL modalities (cont.)

- $\mathcal{A}_{\mathsf{SU}}$ from $\mathbb{B}^2$ to $\mathbb{B}$:



- $\mathcal{A}_{\mathsf{SS}}$ from $\mathbb{B}^2$ to $\mathbb{B}$:



From LTL to Büchi automata

Definition: Translation from LTL to SGBT

For each $\xi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{SU}, \mathsf{SS})$ we define inductively an SGBT $\mathcal{A}_\xi$ as follows:

- $\mathcal{A}_\top$ and $\mathcal{A}_p$ for $p \in \mathrm{AP}$ are already defined
- $\mathcal{A}_{\neg\varphi} = \mathcal{A}_\neg \circ \mathcal{A}_\varphi$
- $\mathcal{A}_{\varphi\vee\psi} = \mathcal{A}_\vee \circ (\mathcal{A}_\varphi \times \mathcal{A}_\psi)$
- $\mathcal{A}_{\varphi \mathsf{SS} \psi} = \mathcal{A}_{\mathsf{SS}} \circ (\mathcal{A}_\varphi \times \mathcal{A}_\psi)$
- $\mathcal{A}_{\varphi \mathsf{SU} \psi} = \mathcal{A}_{\mathsf{SU}} \circ (\mathcal{A}_\varphi \times \mathcal{A}_\psi)$

Theorem: Correctness of the translation

For each $\xi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{SU}, \mathsf{SS})$, we have $[\![\mathcal{A}_\xi]\!] = [\![\xi]\!]$.

Moreover, the number of states of $\mathcal{A}_\xi$ is at most $2^{|\xi|_{\mathsf{SS}}} \cdot 3^{|\xi|_{\mathsf{SU}}}$
where $|\xi|_{\mathsf{SS}}$ (resp. $|\xi|_{\mathsf{SU}}$) is the number of SS (resp. SU) occurring in ξ.

Remark:

- If a subformula φ occurs serveral times in ξ, we only need one copy of $\mathcal{A}_\varphi$.
- We may also use automata for other modalities: $\mathcal{A}_{\mathsf{X}}$, $\mathcal{A}_{\mathsf{U}}$, A_{F}, ...

Useful simplifications

Reducing the number of temporal subformulae

$$(\mathsf{X}\,\varphi) \wedge (\mathsf{X}\,\psi) \equiv \mathsf{X}(\varphi \wedge \psi) \qquad (\mathsf{X}\,\varphi)\,\mathsf{U}\,(\mathsf{X}\,\psi) \equiv \mathsf{X}(\varphi\,\mathsf{U}\,\psi) \equiv \varphi\,\mathsf{SU}\,\psi$$
$$(\mathsf{G}\,\varphi) \wedge (\mathsf{G}\,\psi) \equiv \mathsf{G}(\varphi \wedge \psi) \qquad \mathsf{G}\,\mathsf{F}\,\varphi \vee \mathsf{G}\,\mathsf{F}\,\psi \equiv \mathsf{G}\,\mathsf{F}(\varphi \vee \psi)$$
$$(\varphi_1\,\mathsf{U}\,\psi) \wedge (\varphi_2\,\mathsf{U}\,\psi) \equiv (\varphi_1 \wedge \varphi_2)\,\mathsf{U}\,\psi \qquad (\varphi\,\mathsf{U}\,\psi_1) \vee (\varphi\,\mathsf{U}\,\psi_2) \equiv \varphi\,\mathsf{U}\,(\psi_1 \vee \psi_2)$$

Merging equivalent states

Let $\mathcal{A} = (Q, \Sigma, I, T, T_1, \ldots, T_n, \mu)$ be an SGBT and $s_1, s_2 \in Q$.
We can merge s_1 and s_2 if they have the same outgoing transitions:
$\forall a \in \Sigma$, $\forall s \in Q$,

$$\begin{array}{lrcll} & (s_1, a, s) \in T & \Longleftrightarrow & (s_2, a, s) \in T & \\ \text{and} & (s_1, a, s) \in T_i & \Longleftrightarrow & (s_2, a, s) \in T_i & \text{for all } 1 \le i \le n \\ \text{and} & \mu(s_1, a, s) & = & \mu(s_2, a, s) & \end{array}$$

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.

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

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

Definition: Satisfiability problem

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

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 $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{SU}, \mathsf{SS})$

Question: Existence of $w \in \Sigma^\omega$ such that $w, 0 \models \varphi$.

Remark: φ is satisfiable iff $\mathsf{F}\,\varphi$ is *initially* satisfiable.

Definition: (Initial) validity

φ 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 $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{SU}, \mathsf{SS})$

Question: Does $M \models \varphi$?

- Universal MC: $M \models_\forall \varphi$ 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.

$$M \models_\forall \varphi \quad \text{iff} \quad M \not\models_\exists \neg\varphi$$

Theorem [10, Sistla, Clarke 85], [9, Lichtenstein & Pnueli 85]

The Model checking problem for LTL is PSPACE-complete

$\mathrm{MC}^\exists(\mathsf{SU}) \leq_P \mathrm{SAT}(\mathsf{SU})$ [10, Sistla & Clarke 85]

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

Introduce new atomic propositions: $\mathrm{AP}_S = \{\mathrm{at}_s \mid s \in S\}$
Define $\mathrm{AP}' = \mathrm{AP} \uplus \mathrm{AP}_S$ $\quad \Sigma' = 2^{\mathrm{AP}'}$ $\quad \pi : \Sigma'^\omega \to \Sigma^\omega$ by $\pi(a) = a \cap \mathrm{AP}$.

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

Define $\psi_M \in \mathrm{LTL}(\mathrm{AP}', \mathsf{X}, \mathsf{F})$ of size $\mathcal{O}(|M|^2)$ by

$$\psi_M = \left(\bigvee_{s \in I} \mathrm{at}_s\right) \wedge \mathsf{G}\left(\bigvee_{s \in S}\left(\mathrm{at}_s \wedge \bigwedge_{t \neq s} \neg\mathrm{at}_t \wedge \bigwedge_{p \in \ell(s)} p \wedge \bigwedge_{p \notin \ell(s)} \neg p \wedge \bigvee_{t \in T(s)} \mathsf{X}\,\mathrm{at}_t\right)\right)$$

Let $w = a_0 a_1 a_2 \cdots \in \Sigma'^\omega$. Then, $w \models \psi_M$ iff there exists an initial infinite run $\sigma = s_0 s_1 s_2 \cdots$ of M such that $\ell(\sigma) = \pi(w)$ and $a_i \cap \mathrm{AP}_S = \{\mathrm{at}_{s_i}\}$ for all $i \geq 0$.

Therefore, $M \models_\exists \varphi$ iff $\psi_M \wedge \varphi$ is satisfiable
$M \models_\forall \varphi$ iff $\psi_M \wedge \neg\varphi$ is not satisfiable

Remark: we also have $\mathrm{MC}^\exists(\mathsf{X}, \mathsf{F}) \leq_P \mathrm{SAT}(\mathsf{X}, \mathsf{F})$.

QBF Quantified Boolean Formulae

Definition: QBF

Input: A formula $\gamma = Q_1 x_1 \cdots Q_n x_n \gamma'$ 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 γ 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 γ') if $v \models \gamma'$ for all $v \in V$,
- V is closed (for γ) 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 $\{v_i, v'_i\} = \{0,1\}$.

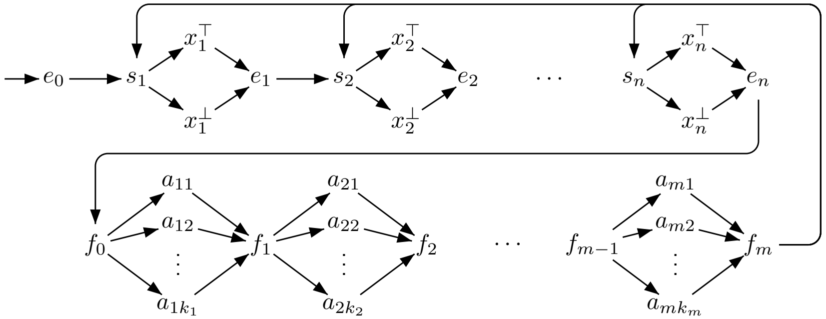
Proposition:

γ is valid iff $\exists V \subseteq \{0,1\}^n$ s.t. V is nonempty valid and closed

$\mathrm{QBF} \leq_P \mathrm{MC}^{\exists}(\mathsf{U})$ [10, Sistla & Clarke 85]

Let $\gamma = Q_1 x_1 \cdots Q_n x_n \bigwedge_{1 \leq i \leq m} \bigvee_{1 \leq j \leq k_i} a_{ij}$ with $Q_i \in \{\forall, \exists\}$ and a_{ij} literals.

Consider the KS M:



Let $\psi_{ij} = \begin{cases} \mathsf{G}(x_k^\perp \to s_k \,\mathsf{R}\, \neg a_{ij}) & \text{if } a_{ij} = x_k \\ \mathsf{G}(x_k^\top \to s_k \,\mathsf{R}\, \neg a_{ij}) & \text{if } a_{ij} = \neg x_k \end{cases}$ and $\psi = \bigwedge_{i,j} \psi_{ij}$.

Let $\varphi_j = \mathsf{G}(e_{j-1} \to (\neg s_{j-1} \,\mathsf{U}\, x_j^\top) \wedge (\neg s_{j-1} \,\mathsf{U}\, x_j^\perp))$ and $\varphi = \bigwedge_{j \mid Q_j = \forall} \varphi_j$.

Then, γ is valid iff $M \models_\exists \psi \wedge \varphi$.

Complexity of LTL

Theorem: Complexity of LTL

The following problems are PSPACE-complete:

- $\mathrm{SAT}(\mathrm{LTL}(\mathsf{SU}, \mathsf{SS}))$, $\mathrm{MC}^{\forall}(\mathrm{LTL}(\mathsf{SU}, \mathsf{SS}))$, $\mathrm{MC}^{\exists}(\mathrm{LTL}(\mathsf{SU}, \mathsf{SS}))$
- $\mathrm{SAT}(\mathrm{LTL}(\mathsf{X}, \mathsf{F}))$, $\mathrm{MC}^{\forall}(\mathrm{LTL}(\mathsf{X}, \mathsf{F}))$, $\mathrm{MC}^{\exists}(\mathrm{LTL}(\mathsf{X}, \mathsf{F}))$
- $\mathrm{SAT}(\mathrm{LTL}(\mathsf{U}))$, $\mathrm{MC}^{\forall}(\mathrm{LTL}(\mathsf{U}))$, $\mathrm{MC}^{\exists}(\mathrm{LTL}(\mathsf{U}))$
- The restriction of the above problems to a unique propositional variable

The following problems are NP-complete:

- $\mathrm{SAT}(\mathrm{LTL}(\mathsf{F}))$, $\mathrm{MC}^{\exists}(\mathrm{LTL}(\mathsf{F}))$

Complexity of CTL^*

Theorem

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

Proof:

PSPACE-hardness: follows from $\mathrm{LTL} \subseteq \mathrm{CTL}^*$.

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

$\mathrm{MC}^{\exists}_{\mathrm{CTL}^*}$ in PSPACE

Proof:

For $\psi \in \mathrm{LTL}$, let $\mathrm{MC}^{\exists}_{\mathrm{LTL}}(M, t, \psi)$ be the function which computes in polynomial space whether $M, t \models_\exists \psi$, i.e., if $M, t \models \mathsf{E}\,\psi$.

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure, $s \in S$ and $\varphi \in \mathrm{CTL}^*$.
Replacing $\mathsf{A}\,\psi$ by $\neg\,\mathsf{E}\,\neg\psi$ we assume φ only contains the existential path quantifier.

$\mathrm{MC}^{\exists}_{\mathrm{CTL}^*}(M, s, \varphi)$

- If E does not occur in φ then return $\mathrm{MC}^{\exists}_{\mathrm{LTL}}(M, s, \varphi)$ fi
- Let $\mathsf{E}\,\psi$ be a subformula of φ with $\psi \in \mathrm{LTL}$
- Let e_ψ be a new propositional variable
- Define $\ell' : S \to 2^{\mathrm{AP}'}$ with $\mathrm{AP}' = \mathrm{AP} \uplus \{e_\psi\}$ by
  - $\ell'(t) \cap \mathrm{AP} = \ell(t)$ and $e_\psi \in \ell'(t)$ iff $\mathrm{MC}^{\exists}_{\mathrm{LTL}}(M, t, \psi)$
- Let $M' = (S, T, I, \mathrm{AP}', \ell')$
- Let $\varphi' = \varphi[e_\psi / \mathsf{E}\,\psi]$ be obtained from φ by replacing each $\mathsf{E}\,\psi$ by e_ψ
- Return $\mathrm{MC}^{\exists}_{\mathrm{CTL}^*}(M', s, \varphi')$

Satisfiability for CTL^*

Definition: Satisfiability problem for CTL^*

Input: A formula $\varphi \in \mathrm{CTL}^*$

Question: Existence of a model M, a run σ, a position i such that $M, \sigma, i \models \varphi$?

Definition: Initial Satisfiability problem for CTL^*

Input: A formula $\varphi \in \mathrm{CTL}^*$

Question: Existence of a model M and a run σ such that $M, \sigma, 0 \models \varphi$?

Theorem

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

Some References

[9] O. Lichtenstein and A. Pnueli.
Checking that finite state concurrent programs satisfy their linear specification.
In *ACM Symposium PoPL'85*, 97–107.

[15] P. Wolper.
The tableau method for temporal logic: An overview,
Logique et Analyse. **110–111**, 119–136, (1985).

[10] A. Sistla and E. Clarke.
The complexity of propositional linear temporal logic.
Journal of the Association for Computing Machinery. **32** (3), 733–749, (1985).

[16] P. Gastin and D. Oddoux.
Fast LTL to Büchi automata translation.
In *CAV'01*, vol. 2102, *Lecture Notes in Computer Science*, pp. 53–65.
Springer, (2001).
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

[6] S. Demri and P. Gastin.
Specification and Verification using Temporal Logics.
In Modern applications of automata theory, IISc Research Monographs 2.
World Scientific, 2012.
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php