Outline

Introduction

Models

Temporal Specifications

* Satisfiability and Model Checking
  * CTL
  * Fair CTL
  * Buchi automata
  * From LTL to BA
  * LTL
  * CTL

More on Temporal Specifications

---

Model checking of CTL

Definition: procedure semantics(φ)

```plaintext
case φ = ¬φ₁
  semantics(φ₁)
  [φ] := S \ {φ₁}
  O(|S|)
case φ = φ₁ ∨ φ₂
  semantics(φ₁); semantics(φ₂)
  [φ] := [φ₁] ∪ [φ₂]
  O(|S|)
case φ = EXφ₁
  semantics(φ₁)
  [φ₁] := ∅
  for all t ∈ [φ₁] do for all s ∈ T⁻¹(t) do [φ] := [φ] ∪ {s}
  O(|S|)
  O(|T|)
case φ = AXφ₁
  semantics(φ₁)
  [φ₁] := S
  for all t ∈ [φ₁] do for all s ∈ T⁻¹(t) do [φ] := [φ] \ {s}
  O(|S|)
  O(|T|)
```

---

Model checking of CTL

Theorem

Let M = (S, T, I, AP, ϵ) be a Kripke structure and φ ∈ CTL a formula.
The model checking problem M |= φ is decidable in time O(|M| ⋅ |φ|)

Proof:

Compute [φ] = {s ∈ S | M, s |= φ} by induction on the formula.
The set [φ] is represented by a boolean array: L[s][φ] = T if s ∈ [φ].
The labelling ϵ is encoded in L: for p ∈ AP we have L[s][p] = T if p ∈ ϵ(s).
For each t ∈ S, the set T⁻¹(t) is represented as a list.
for all t ∈ S do for all s ∈ T⁻¹(t) do ... od takes time O(|T|).

Good is only used to make the invariant clear. It can be replaced by [φ].
Model checking of \(\text{CTL}\)

Definition: procedure semantics(\(\varphi\))

\[
\text{case } \varphi = A\varphi_1 \cup \varphi_2 \quad \text{O}(|S| + |T|)
\]

semantics(\(\varphi_1\)); semantics(\(\varphi_2\))

Todo := \([\varphi_2]\) // the "todo" set is implemented with a list

\[
\text{Good := } [\varphi_2] \quad \text{O}(|S|)
\]

for all \(s \in S\) do \(c[s] := |T(s)|\)

\[
\text{while } \text{Todo} \neq \emptyset \text{ do } \text{O}(|S| \text{ times})
\]

Invariant 1: \([\varphi_2] \cup \text{Todo} \subseteq \text{Good} \subseteq [A\varphi_1 \cup \varphi_2]\) and

Invariant 2: \(\forall s \in S, \ c[s] = |T(s)| \setminus (\text{Good} \setminus \text{Todo})\) and

Invariant 3: \([\varphi_1] \cap \{s \in S \mid c[s] = 0\} \subseteq \text{Good}\)

\[
\begin{align*}
\text{take } t & \in \text{Todo}; \text{Todo} := \text{Todo} \setminus \{t\} \quad \text{O}(1) \\
\text{for all } s & \in T^{-1}(t) \text{ do } \quad \text{O}(1) \\
& \quad \text{c}[s] := c[s] - 1 \\
\text{if } & c[s] = 0 \land s \in [\varphi_1] \setminus \text{Good} \text{ then } \\
& \quad \text{Todo} := \text{Todo} \cup \{s\}; \text{Good} := \text{Good} \cup \{s\} \quad \text{O}(1)
\end{align*}
\]

\[
\text{od}
\]

\[
\{\varphi\} := \text{Good} \quad \text{O}(|S|)
\]

Good is only used to make the invariant clear. It can be replaced by \([\varphi]\).

Complexity of \(\text{CTL}\)

Definition: SAT(\(\text{CTL}\))

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

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

Theorem: Complexity

- The model checking problem for \(\text{CTL}\) is PTIME-complete.
- The satisifiability problem for \(\text{CTL}\) is EXPTIME-complete.

fair \(\text{CTL}\)

Definition: Syntax of fair-\(\text{CTL}\)

\[
\varphi ::= \bot \mid p (p \in \text{AP}) \mid \neg \varphi \mid \varphi \lor \varphi \mid E\varphi \mid A\varphi \mid E\varphi \cup \varphi \mid A\varphi \cup \varphi
\]

Definition: Semantics as a fragment of \(\text{CTL}^+\)

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

Then, \(E\varphi = E(\text{fair} \land \varphi)\) and \(A\varphi = A(\text{fair} \rightarrow \varphi)\)

where \(\text{fair} = \bigwedge_i GF F_i\)

Lemma: \(\text{CTL}_f\) cannot be expressed in \(\text{CTL}\)
**fair CTL**

Proof: $CTL_f$ cannot be expressed in CTL

Consider the Kripke structure $M_k$ defined by:

\[
\begin{array}{ccccccc}
2k & \cdots & 2k - 3 & \cdots & 1 \\
p & \cdots & p & \cdots & p \\
\end{array}
\]

- $M_k, 2k \models EGFp$ but $M_k, 2k - 2 \not\models EGFp$
- If $\varphi \in CTL$ and $|\varphi| \leq m \leq k$ then $M_k, 2k \models \varphi$ iff $M_k, 2m \models \varphi$

Hence, we only need to compute the semantics of $E_f \top$.

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

**Model checking of CTL$_f$**

Theorem
The model checking problem for $CTL_f$ is decidable in time $O(|M| \cdot |\varphi|)$

Proof: Computation of $Fair = \{ s \in S \mid M, s \models E_f \top \}$

Compute the SCC of $M$ with Tarjan’s algorithm (in time $O(|M|)$). 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.

**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 final (repeated) 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 initial if $q_0 \in I$.

$\rho$ is final (successful) if $q_i \in F$ for infinitely many $i$’s.

$\rho$ is accepting if it is both initial and final.

$L(A) = \{ a_0a_1a_2 \cdots \in \Sigma^\omega \mid \exists \rho = q_0, a_0, q_1, a_1, q_2, a_2, q_3, \ldots \text{ accepting run} \}$

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

Proof: $E_f \top$ can be computed in linear time.

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

Compute the SCC of $M_\varphi$ with Tarjan’s algorithm (in linear time).
Let $S'$ be the union of the (non trivial) SCCs of $M_\varphi$ which intersect each $F_i$.
Then, $M, s \models E_f \varphi$ iff $M, s \models E \varphi U S'$ iff $M_\varphi, s \models EF S'$.
This is again a reachability problem which can be solved in linear time.
**Büchi automata**

**Examples:**

- Infinitely many $a$’s:
  
  Whenever $a$ then later $b$:

- Finitely many $a$’s:

**Properties**

Büchi automata are closed under union, intersection, complement.

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

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

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

---

**Büchi automata**

**Theorem:** Büchi

Let $L \subseteq \Sigma^*$ 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 $L_1 \cdot L_2^*$ where $L_1, L_2 \subseteq \Sigma^*$ are rational.
- $L$ is MSO-definable, i.e., there is a sentence $\varphi \in \text{MSO}_\Sigma(<)$ such that $L = L(\varphi) = \{ w \in \Sigma^* \mid w \models \varphi \}$.

**Exercises:**

1. Construct a BA for $L(\varphi)$ where $\varphi$ is the FO$_\Sigma(<)$ 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 $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$, construct BA for

   $\text{next}(L_1) = \Sigma : L_1$

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

**Generalized Büchi automata**

**Definition:** final condition 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 final (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 final 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.
Unambiguous or prophetic Büchi automata

Definition: Unambiguous Büchi automata
A BA or GBA $\mathcal{A}$ is unambiguous if every word has at most one accepting run in $\mathcal{A}$.

Definition: Prophetic Büchi automata
A BA or GBA $\mathcal{A}$ is prophetic if every word has exactly one final run in $\mathcal{A}$.

Examples: UBA and PBA
- Finitely many $a$’s.
- $G(a \rightarrow F b)$ with $\Sigma = \{a, b, c\}$.

Theorem: Prophetic Büchi automata (Carton-Michel 2003)
Every $\omega$-regular language can be accepted by a prophetic BA.