Model checking of CTL

Theorem

Let $M = (S, T, I, AP, \ell)$ be a Kripke structure and $\varphi \in CTL$ a formula. The set $[[\varphi]] = \{ s \in S \mid M, s \models \varphi \}$ can be computed in time $O(|M| \cdot |\varphi|)$. Hence, the model checking problem $M \models \exists \varphi$ is decidable in time $O(|M| \cdot |\varphi|)$.

Proof:

Compute $[[\varphi]]$ by induction on the formula.

The set $[[\varphi]]$ is represented by a boolean array: $L[s] = \top$ if $s \in [[\varphi]]$.

For each $t \in S$, the set $T^{-1}(t)$ is represented as a list.

$T^{-1}$ is an array of lists, its size is $|S| + |T|$.

for all $t \in S$ do for all $s \in T^{-1}(t)$ do ... od takes time $O(|S| + |T|)$. 
Model checking of CTL

Definition: function semantics(φ) returns boolean array L

case φ = p ∈ AP
  for all s ∈ S do L[s] := (p ∈ ℓ(s)) od \(O(|S|)\)

case φ = ¬φ₁
  \(L₁ := \) semantics(φ₁)
  for all s ∈ S do L[s] := ¬L₁[s] od \(O(|S|)\)

case φ = φ₁ ∨ φ₂
  \(L₁ := \) semantics(φ₁); \(L₂ := \) semantics(φ₂)
  for all s ∈ S do L[s] := L₁[s] ∨ L₂[s] od \(O(|S|)\)

case φ = EXφ₁
  \(L₁ := \) semantics(φ₁)
  for all s ∈ S do L[s] := ⊤ od \(O(|S|)\)
  for all t ∈ S do if L₁[t] then for all s ∈ T⁻¹(t) do L[s] := ⊤ \(O(|S| + |T|)\)

case φ = AXφ₁
  \(L₁ := \) semantics(φ₁)
  for all s ∈ S do L[s] := ⊤ od \(O(|S|)\)
  for all t ∈ S do if ¬L₁[t] then for all s ∈ T⁻¹(t) do L[s] := ⊥ \(O(|S| + |T|)\)
### Model checking ofCTL

**Definition:** function semantics(ϕ) returns boolean array L

- **Case:** ϕ = E ϕ₁ U ϕ₂
  - \( L₁ := \text{semantics}(ϕ₁); \ L₂ := \text{semantics}(ϕ₂) \)
  - for all \( s ∈ S \) do
    - \( L[s] := L₂[s] \)
    - if \( L₂[s] \) then Todo.add(\( s \)) // Todo is implemented with a stack
- **while** Todo ≠ ∅ do
  -  |S| times
  - **Invariant 1:** \([ϕ₂]\cup\text{Todo} ⊆ L ⊆ [E\ ϕ₁ \ U \ ϕ₂]\)
  - \( t := \text{Todo}.\text{remove}() \)
  -  |T| times
  - for all \( s ∈ T⁻¹(t) \) do
    - if \( L₁[s] ∧ ¬L[s] \) then Todo.add(\( s \)); \( L[s] := \top \)
- **od**
Model checking of CTL

Definition: function semantics(ϕ) returns boolean array L

case ϕ = A ϕ₁ U ϕ₂
    L₁ := semantics(ϕ₁); L₂ := semantics(ϕ₂)
    for all s ∈ S do
        L[s] := L₂[s]
        if L₂[s] then Todo.add(s) // Todo is implemented with a stack
    for all s ∈ S do d[s] := 0
    for all t ∈ S do for all s ∈ T⁻¹(t) do d[s] := d[s] + 1
    while Todo ≠ ∅ do
        Invariant 1: ∀s ∈ S, |T(s)| − d[s] = |T(s) ∩ (L \ Todo)|
        Invariant 2: [[ϕ₂]] ∪ Todo ⊆ L ⊆ [[A ϕ₁ U ϕ₂]]
        t := Todo.remove()
        for all s ∈ T⁻¹(t) do
            d[s] := d[s] − 1
            if L₁[s] ∧ ¬L[s] ∧ d[s] = 0 then Todo.add(s); L[s] := ⊤
        od
Example: Fairness

Only fair runs are of interest

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

Definition: Fair Kripke structure

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

An infinite run $\sigma$ is fair if it visits infinitely often each $F_i$.
**Definition: Syntax of fair-CTL**

\[ \varphi ::= \bot \mid p \ (p \in \text{AP}) \mid \neg \varphi \mid \varphi \lor \varphi \mid E_f \varphi \mid A_f \varphi \mid E_f \varphi U \varphi \mid A_f \varphi U \varphi \]

**Definition: Semantics as a fragment of CTL**

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

Let,

\[ E_f \varphi = E(\text{FairRun} \land \neg \varphi) \quad \text{and} \quad A_f \varphi = A(\text{FairRun} \rightarrow \neg \varphi) \]

where \( \text{FairRun} = \bigwedge_i G F F_i \)

Then,

\[ [\varphi]_f = [\neg \varphi] \]

**Lemma:** \( \text{CTL}_f \) cannot be expressed in \( \text{CTL} \)
The model checking problem for $\text{CTL}_f$ is decidable in time $O(|M| \cdot |\varphi|)$

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

Compute the SCC of $M$ in time $O(|M|)$, e.g., with Tarjan’s algorithm.
Let $S'$ be the union of the (non trivial) SCCs which intersect each $F_i$.
Then, $\text{FairStates}$ is the set of states that can reach $S'$: $\text{FairStates} = \llbracket EF S' \rrbracket$.
Note that reachability can be computed in linear time.

**Proof:** Reductions

\[
E_f X \varphi = E X (\text{FairStates} \land \varphi) \quad \text{and} \quad E_f \varphi U \psi = E \varphi U (\text{FairStates} \land \psi)
\]

It remains to deal with $A_f \varphi U \psi$. We have

\[
A_f \varphi U \psi = \neg E_f G \neg \psi \land \neg E_f (\neg \psi U (\neg \varphi \land \neg \psi))
\]

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

Let $M_\varphi$ be the restriction of $M$ to $\llbracket \varphi \rrbracket_f$. Then,

\[
M, s \models E_f G \varphi \iff M_\varphi, s \models E_f \top.
\]

We apply the above algorithm for $E_f \top$ to $M_\varphi$. 

---

Model checking of $\text{CTL}_f$

**Theorem**

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