Branching time models & verification

Cătălin Dima

LACL, Université Paris 12

Temporal properties

- Safety, termination, mutual exclusion – LTL.
- Liveness, reactiveness, responsiveness, infinitely repeated behaviors – LTL.
- Available choices, strategies, adversial situations?

CNIL

Tout utilisateur peut demander le retrait de ses données...

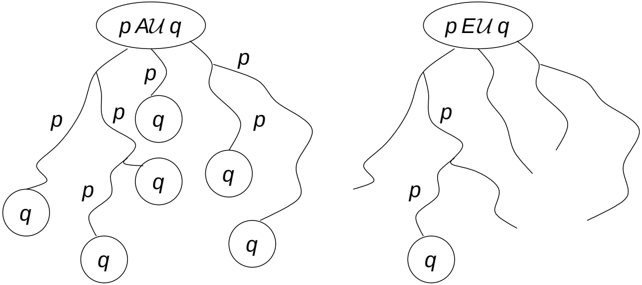
- How do we interpret peut?
  - $p =$ demander le retrait...
  - Then formula = $\Box p$??
  - NO!

Strategy to win a game

Black has a strategy to put the game in a situation from which White king will never get close to Black pawn.

- Not specifiable in LTL either!

Branching time



CTL syntax

- Computational tree logic:

$$\phi ::= p \mid \phi \wedge \phi \mid \neg\phi \mid A\bigcirc\phi \mid \phi\, A\mathcal{U}\, \phi \mid \phi\, E\mathcal{U}\, \phi$$

  - $p \in AP$, set of atomic propositions.

- The usual abbreviations:

$$E\Diamond\phi = \text{true}\, E\mathcal{U}\, \phi \qquad A\Box\phi = \neg E\Diamond\neg\phi$$

$$A\Diamond\phi = \text{true}\, A\mathcal{U}\, \phi \qquad E\Box\phi = \neg A\Diamond\neg\phi$$

$$E\bigcirc\phi = \neg A\bigcirc\phi$$

Semantics

- *AP*-labeled trees $t : \mathbb{N}^* \rightharpoonup 2^{AP}$.
- "States" for interpreting CTL operators = positions in the tree: $x \in \mathrm{supp}(t)$.

$$
\begin{array}{ll}
(t,x) \models p & \text{if } p \in t(x) \\
(t,x) \models \phi_1 \wedge \phi_2 & \text{if } (t,x) \models \phi_j \text{ for both } j = 1,2 \\
(t,x) \models \neg\phi & \text{if } (t,x) \not\models \phi \\
(t,x) \models A\bigcirc\phi & \text{if for all } i \in \mathbb{N} \text{ with } xi \in \mathrm{supp}(t), (t,xi) \models \phi \\
(t,x) \models \phi_1 A\mathcal{U} \phi_2 & \text{if for any infinite path } (x_k)_{k\geq 1} \text{ in } t \text{ with } x_1 = x \\
& \quad \text{there exists } k_0 \geq 1 \text{ with } (t,x_{k_0}) \models \phi_2 \\
& \quad \text{and } (t,x_j) \models \phi_1 \text{ for all } 1 \leq j \leq k_0 - 1 \\
(t,x) \models \phi_1 E\mathcal{U} \phi_2 & \text{if there exists a finite path } (x_j)_{1\leq j\leq k_0} \text{ in } t \text{ with } x_1 = x, \\
& \quad (t,x_{k_0}) \models \phi_2 \text{ and } (t,x_j) \models \phi_1 \text{ for all } 1 \leq j \leq k_0 - 1
\end{array}
$$

Property specification

CNIL

Tout utilisateur peut demander le retrait de ses données...

- How do we interpret peut?
  - $p =$ demander le retrait...
  - $A\Box E\Diamond p$!

Strategy to win a game

Black has a strategy to put the game in a situation from which White king will never get close to Black pawn.

- $q =$ White king never gets close to Black pawn.
- $E\Diamond A\Box q$!

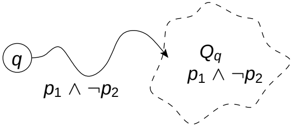
The model-checking problem

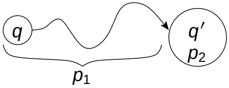
- Given a CTL formula ϕ and a finitely presentable model M, does $M \models \phi$ hold?
  - Finitely presentable tree = Büchi automaton over AP.
  - The tree = the unfolding of $\mathcal{A}$.
- State labeling algorithm:
  - Given formula ϕ, **split** Q into Q_ϕ and $Q_{\neg\phi}$
  - Structural induction on the syntactic tree of ϕ.
  - Add a new propositional symbol p_ϕ for each analyzed ϕ.
  - Label Q_ϕ with p_ϕ and do not label $Q_{\neg\phi}$ with p_ϕ.

CTL model-checking (2)

- For $\phi = A\bigcirc p$

$$Q_{A\bigcirc p} = \{q \in Q \mid \forall q' \in \delta(q), p \in \pi(q')\}$$
$$Q_{\neg A\bigcirc p} = \{q \in Q \mid \exists q' \in \delta(q), p \notin \pi(q')\}$$

- $\phi = p_1 \, A\mathcal{U} \, p_2$
  - $Q_{\neg(p_1 \, A\mathcal{U} \, p_2)}$ contains q iff $\exists Q_q \subseteq Q$ strongly connected s.t.:
    - ★ 

  - $Q_{p_1 \, A\mathcal{U} \, p_2} = Q \setminus Q_{\neg(p_1 \, A\mathcal{U} \, p_2)}$.
- $\phi = p_1 \, E\mathcal{U} \, p_2$
  - $Q_{p_1 \, E\mathcal{U} \, p_2}$ contains q iff $\exists q' \in Q$ s.t.:
    - ★ 

  - $Q_{\neg(p_1 \, E\mathcal{U} \, p_2)} = Q \setminus Q_{p_1 \, E\mathcal{U} \, p_2}$.

Fixpoints

$$E\Diamond p \equiv ...?$$
$$A\Diamond p \equiv ...?$$
$$E\Box p \equiv ...?$$
$$A\Box p \equiv ...?$$
$$p\, A\mathcal{U}\, q \equiv q \vee \big(p \wedge A\bigcirc(p\, A\mathcal{U}\, q)\big)$$
$$p\, E\mathcal{U}\, q \equiv ...?$$

- Which is a μX and which is a νX?

Tree automata

- Büchi automata only give one set of options.
- A formula may incorporate several sets of options!
- Tree automata: $(Q, \Sigma, \delta, Q_0, R)$ where
  - $\delta \subseteq Q \times \mathcal{P}(Q)$.
- Accepts trees.
  - Each run has to satisfy the requirement R (repeated sets like Büchi).
- Emptiness, intersection, complementation (harder than Büchi!).