Prof. Bernd Finkbeiner, Ph.D.
Markus Rabe, M.Sc.
Hazem Torfah, B.Sc.
Felix Klein, B.Sc.

Problem Set 13
No discussion session

Automata, Games, and Verification

---

1. **Alternating tree automata - part one**

   Describe alternating parity tree automata for the following tree languages:

   a) $L_1 = \{(T, \tau) \mid T \subseteq \{0, \dots, 3\}^*, \tau : T \to 2^{\{a,b,c\}}$, whenever for a tree node t in (T, τ) we have $c \in \tau(t)$, then (1) there exists a branch in the tree on which a is contained infinitely often in the label of the nodes and the branch contains t, and (2) there exists a branch in the tree on which b is contained only finitely often in the label of the nodes and the branch contains $t\}$.

   b) $L_2 = \{(T, \tau) \mid T \subseteq \{0, \dots, 1\}^*, \tau : T \to \{a, b, c\}$, for every node t in the tree and $x \in \{a, b, c\}$, if there is some $t' \in \{0, 1\}^*$ with $t1t' \in T$ and $\tau(t1t') = x$, then there also exists some $t'' \in \{0, 1\}^*$ with $t0t'' \in T$ and $\tau(t0t'') = x\}$

2. **Alternating tree automata - part two**

   Let a deterministic parity word automaton $\mathcal{A} = (S, I, T, c)$ over some alphabet Σ be given, and let $k = |\Sigma|$. Take for granted that all words in the language of $\mathcal{A}$ start with the letter $a \in \Sigma$. Construct an alternating parity tree automaton over Σ-labeled trees that accepts precisely the trees over the set of directions $\mathcal{D} = \{0, \dots, k-1\}$ for which the set of its infinite branches represents (by their label sequences) precisely the set of words accepted by $\mathcal{A}$.

   More formally, we search for an alternating tree automaton $\mathcal{A}'$ over the set of directions $\mathcal{D} = \{0, \dots, k-1\}$ such that $\mathcal{A}'$ accepts precisely the Σ-labeled $\mathcal{D}$-trees (T, τ) for which $\{\tau(\epsilon)\tau(t_0)\tau(t_0t_1)\tau(t_0t_1t_2)\dots \mid t_0t_1t_2\dots \in \mathcal{D}^\omega \wedge \forall i \in \mathbb{N} : t_0t_1\dots t_i \in T\}$ is the set of words accepted by $\mathcal{A}$.

   Provide a procedure to construct such an automaton $\mathcal{A}'$ from $\mathcal{A}$. Is it possible that $\mathcal{A}'$ has an empty language even though $\mathcal{A}$ does not?

3. **CTL$^+$**

   Consider the following fragment, called CTL$^+$, of CTL*, which extends CTL by allowing Boolean operators in path formulas:

   - State formulas f:
   $$f ::= AP \mid \neg f \mid f \vee g \mid A\varphi \mid E\varphi$$
   - Path formulas φ:
   $$\varphi ::= \neg\varphi \mid \varphi \vee \psi \mid Gf \mid Ff \mid f\,U g \mid Xf$$

   (Note: CTL* extends CTL$^+$ by allowing to use state formulas f as one more alternative in the definition of path formulas φ.)

   a) Provide, if they exist, equivalent CTL and LTL formulas for the CTL$^+$ formulas $A(F\,a \wedge G\,b)$ and $A(X\,a \wedge \neg(a\,U\,(G\,b)))$.

   b) Compare the expressive power of CTL$^+$ with the expressive power of CTL and LTL.