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

Problem Set 6
discussion session: 4th of December 2012

Automata, Games, and Verification

---

1. **LTL & Determininistic Automata** (Group G06, discussion session 12:00 with Felix Klein)
   - Compare the expressive power of linear-time temporal logic and deterministic Büchi automata.
   - Compare the expressive power of linear-time temporal logic and deterministic co-Büchi automata.

2. **Temporal Operators** (Group G04, discussion session 12:00 with Hazem Torfah)

   Show that $\{\neg, \wedge, \mathrm{X}, \mathcal{W}\}$ is an operator basis for LTL, i.e., that you can express every LTL formula ψ as an equivalent LTL formula ψ', in which apart from atomic propositions, only the operators $\neg$, $\wedge$, X and $\mathcal{W}$ are used.

3. **LTL, QPTL & S1S** (Group G12, discussion session 12:20 with Hazem Torfah)

   Let $\mathsf{AP} = \{q, p, r\}$. Given some word $w = w_0 w_1 w_2 \ldots \in (2^{\mathsf{AP}})^\omega$, for every $a \in \mathsf{AP}$, we denote $w|_a = (w_0 \cap \{a\})(w_1 \cap \{a\})(w_2 \cap \{a\}) \ldots$ and $w(i,j) = w_i w_{i+1} \ldots w_j$ for every $i, j \in \mathbb{N}$ with $i \leq j$.

   Given some finite word $w = w_0 w_1 \ldots w_n$, we define $f : (2^{\mathsf{AP}}) \to \mathbb{N}$ to denote the number represented by w in binary (with the least significant bit first), where we treat the letter $\emptyset$ as 0 and every other letter in 2^{AP} as 1, i.e., $f(\epsilon) = 0$ and:

   $$f(w_0 w_1 \ldots w_n) = \begin{cases} f(w(1,n)) \cdot 2 & \text{if } w_0 = \emptyset \\ f(w(1,n)) \cdot 2 + 1 & \text{if } w_0 \neq \emptyset \end{cases}$$

   Represent the following language L as LTL, QPTL and S1S formulas.

   $$L = \{w \in (2^{\mathsf{AP}})^\omega \mid \forall j \in \mathbb{N} : f(w|_r(0,j)) = f(w|_p(0,j)) + f(w|_q(0,j))\}$$

4. **S1S and LTL** (Group G03, discussion session 12:40 with Hazem Torfah)

   Decide for each of the languages over $2^{\{p,q\}}$ described below if they can be defined in S1S and/or LTL. Justify your answer in each case by either providing a formula or an argument why the language is not definable.

   a) $L_1 = \{\alpha \mid p \in \alpha(0), p \notin \alpha(i) \text{ for all } i \geq 1\}$;
   b) $L_2 = \{\alpha \mid p \in \alpha(i) \text{ for exactly two different } i \in \omega\}$;
   c) $L_3 = \{\alpha \mid \ |\{i \in \omega \,|\, p \in \alpha(i)\}| \text{ is finite and even}\}$;
   d) $L_4 = \{\alpha \mid \ |\{i \in \omega \,|\, p \in \alpha(i)\}| \text{ and } |\{i \in \omega \,|\, q \in \alpha(i)\}| \text{ are finite and equal}\}$.

5. **Efficient Determinization – Safra's Construction** (challenge question)

   In the lecture we proved McNaughton's Theorem in two steps. Show that the following construction can be used to turn a nondeterministic Büchi automaton $\mathcal{A} = (S, I, T, F)$ directly into a deterministic Muller automaton $\mathcal{M} = (S', I', T', \mathcal{F}')$:

   A tree is called a Safra tree iff

   a) Each node of the tree is labeled with a set of states, called the macrostates of the node.

b) The macrostates of brother nodes are disjoint.

c) The union of the sets of brother macrostates is a proper subset of the macrostates of their parent node.

d) Each node has a unique name in $\{1, \ldots, n\}$ for some $n \in \omega$.

e) A (possibly empty) subset of the nodes nodes are marked !.

S' is the set of Safra trees such that

a) the root node is named 1,

b) the root node has a macrostate $M \subseteq S$ which forms a subset of S, and

c) the set of names is $\{1, \ldots, 2|S|\}$.

The initial state s'_0 is the Safra tree with a single (unmarked) node with macrostate I and name 1.

The transition is performed in six steps:

a) All marks ! are removed.

b) For every node with macrostate M, a new son with macrostate
$M' = pr_3(T \cap M \times \{\sigma\} \times F)$, is created. The new nodes get fresh names (in a predefined fashion).

c) For every old node with macrostate M, the macrostate is updated to
$M' = pr_3(T \cap M \times \{\sigma\} \times S)$.

d) (horizontal merge): For every node with macrostate M and $s \in M$, remove s from the macrostate of all younger brothers and their descendants .

e) Remove all nodes with empty macrostate.

f) (vertical merge): For every node whose macrostate equals the union of the macrostates of its sons:

   i. delete its descendants , and

   ii. mark n with !.

$\mathcal{F} = \{F' \subseteq S' \mid \exists i \in 1, \ldots, 2|S|$ s.t. a node named i is in all Safra trees in F', and
the node named i is marked ! in some Safra tree in $F'\}$.

Hints: For $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$ you can use Königs Lemma in the same way as we used it for the semi-determinization in the lecture.

For $\alpha \in \mathcal{L}(\mathcal{A}) \Rightarrow \alpha \in \mathcal{L}(\mathcal{A}')$, fix an accepting run $r = s_0 s_1 s_2 \ldots$ of $\mathcal{A}$ for α, and consider the run $r' = s'_0 s'_1 s'_2 \ldots$ of $\mathcal{A}'$ for α. Let $\pi = p_0 p_1 p_2 \ldots$ be the sequence of paths of nodes s.t. p_i is the sequence of names of nodes of the Safra tree s'_i whose macrostates contain s_i (naturally they always form a path in s'_i). Does π stabilize in some sense, or is there a useful limit operation you can exploit?

Remark: Even (or: especially) if you do not solve the challenge question, try the construction on the following semi-deterministic Büchi automaton: