TD 4: LTL Model-Checking

1 Synchronous Büchi Transducers

Exercise 1. Give synchronous Büchi transducers for the following formulæ:

1. $\mathsf{SG}\,q$ and $\mathsf{G}\,q$,
2. $p\,\mathsf{SS}\,q$ and $p\,\mathsf{S}\,q$,
3. $\mathsf{G}(p \to \mathsf{F}\,q)$.

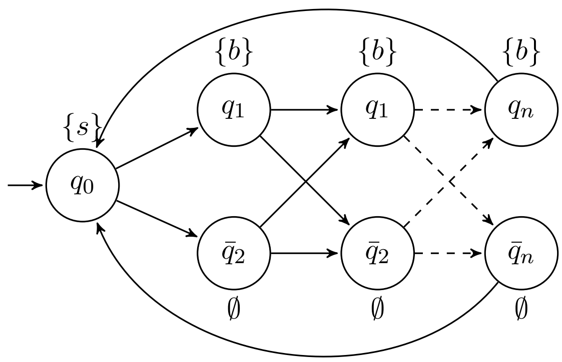
2 Complexity of LTL Model-Checking

Exercise 2 (Complexity of LTL(X))**.** We want to show that LTL(X) existential model checking is NP-complete (instead of PSPACE-complete for the full LTL(SU)).

1. Show that $\mathsf{MC}^\exists(\mathsf{X})$ is in NP.
2. Reduce 3SAT to $\mathsf{MC}^\exists(\mathsf{X})$ in order to prove NP-hardness.

Exercise 3 (Hardness of LTL(X,F))**.** Adapt the proof given during the lecture to show that $\mathsf{MC}^\exists(\mathsf{X},\mathsf{F})$ is PSPACE-hard.

As a preliminary question, consider the following Kripke structure M over AP $= \{s, b\}$:



Any infinite word σ generated by M is in $\big(\{s\}(\{b\}+\emptyset)^n\big)^\omega$, where each segment between two s's can be seen as describing a value from 0 to $2^n - 1$ encoded in binary. Provide an LTL(X,F) formula φ that selects runs ρ where the successive values form the sequence 0, 1, ..., $2^n - 1$, 0, 1, ..., i.e. count modulo 2^n.

Exercise 4 (Stuttering and LTL(U))**.** In the context of a word σ in Σ^ω, *stuttering* denotes the existence of consecutive symbols, like *aaaa* and *bb* in *baaaabb*. Concrete systems tend to stutter, and thus some argue that verification properties should be stutter invariant.

A *stuttering function* $f : \mathbb{N} \to \mathbb{N}_{>0}$ is a function from the positive integers to the strictly positive integers. Let $\sigma = a_0 a_1 \cdots$ be an infinite word of Σ^ω and f a stuttering function, we denote by $\sigma[f]$ the infinite word $a_0^{f(0)} a_1^{f(1)} \cdots$, i.e. where the i-th symbol of σ is repeated $f(i)$ times. A language $L \subseteq \Sigma^\omega$ is *stutter invariant* if, for all words σ in Σ^ω and all stuttering functions f,

$$\sigma \in L \text{ iff } \sigma[f] \in L\ .$$

1. Prove that if φ is a LTL(U) formula, then $L(\varphi)$ is stutter-invariant.
2. A word $\sigma = a_0 a_1 \cdots$ in Σ^ω is *stutter-free* if, for all i in $\mathbb{N}$, either $a_i \neq a_{i+1}$, or $a_i = a_j$ for all $j \geq i$. We note $\mathsf{sf}(L)$ for the set of stutter-free words in a language L.

   Show that, if L and L' are two stutter invariant languages, then $\mathsf{sf}(L) = \mathsf{sf}(L')$ iff $L = L'$.
3. Let φ be a LTL(X, U) formula such that $L(\varphi)$ is stutter invariant. Construct inductively a formula $\tau(\varphi)$ of LTL(U) such that $\mathsf{sf}(L(\varphi)) = \mathsf{sf}(L(\tau(\varphi)))$, and thus such that $L(\varphi) = L(\tau(\varphi))$ according to the previous question. What is the size of $\tau(\varphi)$ (there exists a solution of size $O(|\varphi| \cdot 2^{|\varphi|})$)?

Exercise 5 (Complexity of LTL(U))**.** We want to prove that the model checking and satisfiability problems for LTL(U) formulæ are both PSPACE-complete.

1. Prove that $\mathsf{MC}^\exists(\mathsf{X}, \mathsf{U})$ can be reduced to $\mathsf{MC}^\exists(\mathsf{U})$: given an instance (M, φ) of $\mathsf{MC}^\exists(\mathsf{X}, \mathsf{U})$, construct a stutter-free Kripke structure M' and an LTL(U) formula $\tau'(\varphi)$. *Beware: the τ construction of the previous exercise does not yield a polynomial reduction!*
2. Show that $\mathsf{MC}^\exists(\mathsf{X}, \mathsf{U})$ can be reduced to $\mathsf{SAT}(\mathsf{U})$.