Wqos: homework assignment and midterm exam

Jean Goubault-Larrecq
LMF, ENS Paris-Saclay, Université Paris-Saclay

Turn your solution in by October 24th, 2022, in electronic form (LaTeX or scanned), to goubault@lsv.fr

Note: be as clear as possible. If I cannot understand you, this won’t help you obtain a good grade. Specific hints: (1) write, do not scribble; (2) if I use a specific notation, use the same; (3) use the definitions as given in the various questions, and the definitions and results of the lecture notes—if you know an equivalent definition from the literature but the equivalence is not proved in the lecture notes, do not use it (as a last resort, include the proof of equivalence); (4) justify every claim you make, by a proof, by a definition, by a previous question, or by a theorem (preferentially use theorem names, such as “Dickson’s Lemma”, rather than numbers); (5) find the simplest possible proof argument; (6) use theorems, not their proofs.

Also, the final exam may contain sequels to some of the questions asked here—who knows. The various sections are not independent.

1 A wqa on regular languages

Let $\Sigma \defeq \{A_1, \cdots , A_n\}$ be a fixed, finite alphabet.

**Question 1.** For all $u, v, u', v' \in \Sigma^*$, show that $u \preceq_s u'$ and $v \preceq_s v'$ imply $uv \preceq_s u'v'$.

Use the rules defining $\preceq_s$ seen in the lectures.

**Question 2.** Show that, for any two languages $L$ and $L'$, if $L' \neq \emptyset$ then $L (\preceq_s)^b L.L'$.

In general, we have the following:

(i) $L (\preceq_s)^b L + L'$, $L' (\preceq_s)^b L + L'$;

(ii) if $L' \neq \emptyset$ then $L (\preceq_s)^b L.L'$;

(iii) if $L \neq \emptyset$ then $L' (\preceq_s)^b L.L'$;

(iv) if $L_1 (\preceq_s)^b L_1'$ and $L_2 (\preceq_s)^b L_2'$, then $L_1 \cup L_2 (\preceq_s)^b L_1' \cup L_2'$;

(v) if $L_1 (\preceq_s)^b L_1'$ and $L_2 (\preceq_s)^b L_2'$, then $L_1.L_2 (\preceq_s)^b L_1'.L_2'$;
(vi) if \( L (\leq_s) L' \) then \( L^* (\leq_s) L'^* \);
(vii) if \( L (\leq_s) L' \) then \( L (\leq_s) L'^* \).

We have just proved (ii), in Question 2. We will admit the other properties without proof.

A regular expression is a finite tree (a term) over the signature \{ \( A_1, \cdots, A_n \), \( \cdot, +, * \) \}. Those function symbols in this signature have arities: \( A_1, \ldots, A_n \) have arity 0, \( \cdot \) and \( + \) have arity 2, and \( * \) has arity 1. Regular expressions must respect arities, e.g., one can write \( * (E) \) but not \( *(E_1, E_2) \).

The meaning \( m(E) \) of a regular expression \( E \) is defined by:

\[
\begin{align*}
m(A_i) & \overset{\text{def}}{=} A_i \\
m(\cdot(E_1, E_2)) & \overset{\text{def}}{=} m(E_1).m(E_2) & \text{concatenation} \\
m(+E_1, E_2)) & \overset{\text{def}}{=} m(E_1) \cup m(E_2) & \text{union} \\
m(*E)) & \overset{\text{def}}{=} m(E)^* & \text{Kleene star.}
\end{align*}
\]

Notice that \( m(E) \) is never empty.

**Question 3.** Show that the meaning map \( m \) is monotonic, where the set \( \text{Regexp} \) of regular expressions is quasi-ordered by tree embedding \( \leq_T \) (comparing function symbols by equality), and languages (subsets of \( \Sigma^* \)) are quasi-ordered by the Hoare quasi-ordering \( (\leq_s)^\flat \). Cite all the statements you use explicitly (as usual).

**Question 4.** Conclude that \( (\leq_s)^\flat \) is wqo on the collection \( \text{Reg} \) of regular languages over \( \Sigma \). A regular language over \( \Sigma \) is either the empty language \( \emptyset \) or a language of the form \( m(E) \) for some regular expression \( E \).

## 2 Well-quasi-ordering Presburger definable sets

A very practical, decidable fragment of Peano arithmetic is *Presburger arithmetic*. We assume given an infinite denumerable set of variables \( x, y, z, \ldots \). Its formulas are given by:

\[
F, G, \ldots := \sum_{i=1}^{m} a_i x_i \leq b \quad \text{affine inequality}
\]

\[
\begin{align*}
| \top & \quad \text{true} \\
| F \land G & \quad \text{conjunction (and)} \\
| \bot & \quad \text{false} \\
| F \lor G & \quad \text{disjunction (or)} \\
| \neg F & \quad \text{negation (not)} \\
| \exists x. F & \quad \text{existential quantifier} \\
| \forall x. F & \quad \text{universal quantifier}
\end{align*}
\]
where, in an affine inequality, \( m \in \mathbb{N} \), each \( a_i \) and \( b \) are in \( \mathbb{Z} \), and each \( x_i \) is a variable.

The free variables of a Presburger formula \( F \) are the variables that occur in \( F \), provided they are not bound by an enclosing existential or universal quantifier. (If you need a formal definition, ask me.) We will sometimes write \( F(x_1, \ldots, x_n) \) to say that \( F \) is a Presburger formula whose free variables are among \( x_1, \ldots, x_n \).

Given any finite list \( \ell \) of pairwise distinct variables, we write \( \text{dom} \left( x_1, \ldots, x_n \right) \) for the set \( \left\{ x_1, \ldots, x_n \right\} \) and \( |\ell| \) for the length \( n \) of \( \ell \). We define a set of \( |\ell| \)-tuples of natural numbers \( [F]_\ell \) for every Presburger formula \( F \) whose free variables are among \( x_1, \ldots, x_n \).

We define:

\[
\left[ \sum_{i=1}^{m} a_i x_{j_i} \leq b \right]_{\left[x_1, \ldots, x_n\right]} \defeq \left\{ (k_1, \ldots, k_n) \in \mathbb{N}^n \mid \sum_{i=1}^{m} a_i k_{j_i} \leq b \right\}
\]

The notation \( x :: \ell \) is as in OCaml, namely \( x :: \left[ x_1, \ldots, x_n \right] = \left[ x, x_1, \ldots, x_n \right] \).

The idea is that \( [F]_\ell \) is the collection of tuples that make \( F \) true.

Parikh's theorem is as follows.

**Theorem 1 (Parikh)** The following hold.

1. For every context-free language \( L \), \( \Phi(L) \) is Presburger definable.
2. Every Presburger definable set is equal to \( \Phi(L) \), for some regular language \( L \).

The simplest known proof of the difficult part (1) is due to Verma, Seidl and Schwentick [2, Theorem 4], where it is additionally shown that one can compute
a Presburger formula $F_L$ such that $L = \lceil F_L \rceil$ from a context-free grammar that generates $L$, in linear time.

**Question 5.** Using the results of the previous section (plus some additional work), show that the collection $\text{Pres}$ of Presburger definable subsets of $\mathbb{N}^n$ (for $n$ fixed, but arbitrary) is wqo under $\leq^\flat$, where $\leq$ is the usual componentwise ordering on $\mathbb{N}^n$.

### 3 Integral relational automata

An integral relational automaton (or IRA) is a tuple $(V, I, T)$, where $V$ is a finite set of *variables*, $I$ is a closed interval $[a, b]$ with integer bounds (note: “integer”$=$in $\mathbb{Z}$, “natural number”$=$in $\mathbb{N}$), and $T$ is a finite set of *transitions*, defined as follows. A transition is an instruction of the form $g;\ stmt$, where:

- the *guard* $g$ is a Boolean combination of *constraints* of the form $x \leq y$, $x \leq c$ or $c \leq y$ (where $x, y \in V$ and $c \in \mathbb{Z} \cap I$), in short:
  
  
  \[
  g := x \leq y \mid x \leq c \mid c \leq y
  \mid g \land g \mid g \lor g \mid \neg g
  \]

- *stmt* is a *statement*, namely a list of *assignments* of the form $x := y$, $x := c$ or $x := ?$; there is one assignment per variable $x \in V$, $y$ is an arbitrary variable in $V$ (possibly the same as $x$), $c$ is a constant in $\mathbb{Z} \cap I$, and $?$ is a special symbol. The intuitive meaning of $x := ?$ is to assign any integer (in $\mathbb{Z}$) whatsoever, non-deterministically, to $x$; all assignments are done in parallel, not sequentially, so $x := y, y := x$ swaps the values of $x$ and $y$, for example.

A *configuration* of such an IRA is a map $\gamma : V \rightarrow \mathbb{Z}$. Hence the state space is $\mathbb{Z}^V$, the collection of all maps from $V$ to $\mathbb{Z}$.

The notion of a guard $g$ being *true* on $\gamma$, in notation $\gamma \models g$, is defined in the obvious way; e.g., $\gamma \models x \leq c$ if and only if $\gamma(x) \leq c$, and $\gamma \models x \leq y$ if and only if $\gamma(x) \leq \gamma(y)$.

An outcome of $\text{stmt}$ from $\gamma$ is any configuration $\delta$ that maps each $x \in V$ to:

- $\gamma(y)$ if the only assignment to $x$ in $\text{stmt}$ is of the form $x := y$;
- $c$ if that unique assignment is $x := c$;
- any integer whatsoever (possibly outside $I$) if that assignment is $x := ?$.

In the following, we consider a fixed IRA $(V, I, T)$.

One can *fire* a transition $g;\ stmt$ (in $T$) from a configuration $\gamma$ if the guard $g$ is true on $\gamma$, and firing it will go to any configuration obtained as an outcome
of stmt from $\gamma$. We write $\gamma \rightarrow \delta$ if one can fire some transition of $T$ from $\gamma$, going to $\delta$. This defines a transition system $(\mathbb{Z}^V , \rightarrow)$.

Given two configurations $\gamma$ and $\gamma'$, we let $\gamma \leq_{sp} \gamma'$ if and only if:

(a) for every $x \in V$, for every $c \in \mathbb{Z} \cap [a-1, b]$, $\gamma(x) \leq c$ if and only if $\gamma'(x) \leq c$ (we recall that $I = [a, b]$);

(b) for all $x, y \in V$, $\gamma(x) \leq \gamma(y)$ if and only if $\gamma'(x) \leq \gamma'(y)$;

(c) for all $x, y \in V$, if $\gamma(x) > \gamma(y)$ then $\gamma(x) - \gamma(y) \leq \gamma'(x) - \gamma'(y)$.

**Question 6.** Show that $(\mathbb{Z}^V, \leq_{sp})$ is a wqo.

We admit the following lemma without proof. This is a pretty technical result [1], and it would not bring much if I tested you on your capacity to prove it.

**Lemma 1** The relation $\rightarrow$ is strongly monotonic with respect to $\leq_{sp}$: for all configurations $\gamma$, $\gamma'$ and $\delta$ such that $\gamma \leq_{sp} \gamma'$ and $\gamma \rightarrow \delta$, there is a configuration $\delta'$ such that $\delta \leq_{sp} \delta'$ and $\gamma' \rightarrow \delta'$.

We consider the following extended Presburger arithmetic. The formulae are the same as with Presburger arithmetic, but variables range over $\mathbb{Z}$, not $\mathbb{N}$. The definition of the semantics $\llbracket F \rrbracket^Z_{\ell}$ of extended Presburger formula is just like $\llbracket F \rrbracket_{\ell}$, replacing all the occurrences of $\mathbb{N}$ by $\mathbb{Z}$. We will admit the following.

**Theorem 2** The following problem is decidable.

**INPUT:** an extended Presburger formula $F(x_1, \ldots, x_n)$, the list $\ell \overset{\text{def}}{=} [x_1, \ldots, x_n]$, and an $n$-tuple of integers $(k_1, \ldots, k_n) \in \mathbb{Z}^n$.

**QUESTION:** $(k_1, \ldots, k_n) \in \llbracket F \rrbracket^Z_{\ell}$?

This can be shown by reduction to ordinary Presburger arithmetic, or directly.

We fix a numbering of the variables in $V$, say $V = \{x_1, \ldots, x_n\}$, where $x_1$, $\ldots$, $x_n$ are pairwise distinct. We equate the configurations $\gamma \in \mathbb{Z}^V$ with the $n$-tuples $(\gamma(x_1), \ldots, \gamma(x_n))$. We fix $\ell \overset{\text{def}}{=} [x_1, \ldots, x_n]$. A set of configurations is defined by an extended Presburger formula $F(x_1, \ldots, x_n)$ if and only if it is equal to $\llbracket F \rrbracket^Z_{\ell}$, and a set of configurations is definable if it is defined by some extended Presburger formula $F(x_1, \ldots, x_n)$.

**Question 7.** Let $U$ be a definable set of configurations, defined by some extended Presburger formula $F$. Show that $\text{Pre}(U)$ is definable too, and that one can compute a formula, which we will call $\text{Pre}(F)$, which defines it.

**Question 8.** Show that, given an extended Presburger formula $F$ defining a set $\llbracket F \rrbracket^Z_{\ell}$ that is upwards-closed with respect to $\leq_{sp}$, one can compute an extended Presburger formula $\text{Pre}^*(F)$ that defines the set $\text{Pre}^*(\llbracket F \rrbracket^Z_{\ell})$ of iterated predecessors of $\llbracket F \rrbracket^Z_{\ell}$.
Question 9. Show that the coverability problem for IRAs is decidable. The quasi-ordering we take is $\leq_{sp}$.

Question 10. The classical procedure for solving the coverability problem would represent upwards-closed sets by their finite basis. In solving the previous question, you should have gotten an idea how to convert a finite basis $A$ into an extended Presburger formula $Up(A)$ defining $\uparrow A$. (Upward closure is taken with respect to $\leq_{sp}$.) Show that one can also compute a finite basis of $\left\lceil F \right\rceil_{\ell}$ from an extended Presburger formula, under the assumption that $\left\lceil F \right\rceil_{\ell}$ is upwards-closed under $\leq_{sp}$. The algorithm I have in mind is terribly inefficient, uses Theorem 2 as a black box, and does not require any knowledge of arithmetic.

4 A minimal bad sequence argument

In the lectures, we have seen that if $(D, \leq)$ is wqo, then so is $(D^\circ, \leq_{\circ})$, the collection of finite multisets over $D$, using a pretty simple argument: the Parikh mapping $\Phi$ is monotonic and surjective from $D^*$ onto $D^\circ$, and $(D^*, \leq_{\circ})$ is wqo by Higman’s Lemma. The purpose of this section is to test whether you master the minimal bad sequence argument instead.

Question 11. Let $(D, \leq)$ be a wqo. Give a direct minimal bad sequence based proof of the fact that $(D^\circ, \leq_{\circ})$ is wqo. Your proof should not use Higman’s Lemma, or Kruskal’s Theorem. You should explicitly use the rules $(\emptyset)$, $(add)$ and $(inc)$ of the lecture notes in order to reason on $\leq_{\circ}$.

References
