Languages, automata and computation II Tutorial 11 – Ordinal numbers, register automata, orbit-finite sets

Winter semester 2024/2025

Ordinal numbers

Ordinal numbers generalize natural numbers and can be seen as representations of well-founded total orders. For a set A ordered by $\leq$ we denote by $\mathrm{tp}(A, \leq)$ (or just $\mathrm{tp}(A)$) the ordinal number representing this relation. Two well-founded total orders $(A, \leq_A)$, $(B, \leq_B)$ are isomorphic if and only if $\mathrm{tp}(A) = \mathrm{tp}(B)$. A number $n \in \mathbb{N}$ represents a totally ordered finite set with n elements, i.e. $\mathrm{tp}(A) = n$ if $|A| = n$. By ω we denote the ordinal type of the set of natural numbers with the usual order relation, i.e. $\mathrm{tp}(\mathbb{N}) = \omega$. The class of ordinal numbers is denoted by **ON**. They have precise set theoretic definition which we will not mention it here.

Let $(A, \leq)$, $(B, \leq)$ be well-founded total orders represented by respectively $\alpha, \beta \in \mathbf{ON}$. We write $\alpha \leq \beta$ if can $(A, \leq)$ be embedded into $(B, \leq)$. We also define the following operations:

- $\alpha + \beta = \mathrm{tp}(\{0\} \times A \cup \{1\} \times B, \preceq)$, where $(i, x) \preceq (j, y)$ if $i < j$ or ($i = j$ and $x \leq y$)
- $\alpha \cdot \beta = \mathrm{tp}(B \times A, \preceq_{lex})$
- $\alpha^\beta = \mathrm{tp}(F(A, B), \trianglelefteq)$, where $F(A, B)$ is the set of all functions $B \to A$ with finite support and $f \trianglelefteq g$ if $f = g$ or $f(\xi) < g(\xi)$, where $\xi = \max\{\eta \in B : f(\eta) \neq g(\eta)\}$

Consecutive ordinal numbers are: $0, 1, 2, \ldots, \omega, \omega{+}1, \ldots, \omega{\cdot}2, \ldots, \omega^2, \ldots, \omega^\omega, \ldots$

Exercise 1. Prove that for $\alpha, \beta, \gamma \in \mathbf{ON}$

1. $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$
2. $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$
3. $\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$
4. $\alpha^{\beta+\gamma} = \alpha^\beta \cdot \alpha^\gamma$
5. $(\alpha^\beta)^\gamma = \alpha^{\beta \cdot \gamma}$

Exercise 2. Show the following equalities/inequalities:

1. $1 + \omega = \omega < \omega + 1$
2. $2 \cdot \omega = \omega < \omega \cdot 2$
3. $2^\omega = 3^\omega = \omega$

Register automata

Let $(\mathbb{A}, =)$ be the set of data values with equality as the only relation (one can consider other data structures). An *nondeterministic register automaton (NRA)* is a tuple $A = (\Sigma, R, L, I, F, \Delta)$ where Σ is an orbit-finite *input alphabet* (often $\Sigma = \mathbb{A}$), R is a finite set of registers, L is a finite set of *locations*, $I, F \subseteq L$ are *initial*, resp., *final* locations, and $\Delta \subseteq (L \times \mathbb{A}^R) \times \Sigma \times (L \times \mathbb{A}^R)$ is an equivariant set of *transitions* (thus orbit-finite). A *state* of a register automaton is a location together with register valuation, i.e. $Q = L \times \mathbb{A}^R$.

Exercise 3. Show that following languages are recognizable by register automata:

1. the first data value is the same as the last data value;
2. some data value appears twice;
3. the last data value appears again;
4. every three consecutive data values are pairwise distinct.

Exercise 4. Show that the class of languages recognizable by NRA is not closed under complement.

Exercise 5. Show that a nondeterministic register automaton can recognize the language "some data value appears twice", but a deterministic one cannot.

Exercise 6. Call a nondeterministic register automaton *guessing* if there exists a transition $t \in \Delta$ such that some data value in the target state appears neither in the source state nor in the input. Give an example of a language that needs guessing to be recognized.

A corollary of the above two exercises is that:

$$\text{deterministic} \subsetneq \text{nondeterministic without guessing} \subsetneq \text{nondeterministic}$$

Exercise 7. Prove the following (non)-closure properties of languages of finite data words recognised by NRA and its deterministic variant.

	$\cup$	$\cap$	$\_^R$	$\Sigma^* \setminus \_$
Deterministic NRA	✓	✓	×	✓
Nondeterministic NRA	✓	✓	✓	×

Exercise 8. The complexity of the emptiness problem for nondeterministic register automata depends on how the size $|A|$ of the input automaton is measured. Show that emptiness is:

- PSPACE-complete if $|A|$ is the number of locations and registers;

- NP-complete if $|A|$ is the number of reachable orbits of states;
- polynomial time if $|A|$ is the number of orbits of transitions.

Exercise 9. Consider the set of atoms $(\mathbb{Q}, \leq)$. Show that there exists a language closed under permutation of atoms (not only automorphism of $\mathbb{Q}$) that is recognizable by an NRA over $(\mathbb{Q}, \leq)$ without guessing, but is not recognizable by an NRA over $(\mathbb{Q}, =)$ without guessing.

Sets with atoms

In this tutorial we explore sets with atoms. Fix a countable set A and consider the relational structure $\mathbb{A} = (A, =)$ over the signature consisting only of equality. Elements of $\mathbb{A}$ are called *atoms* and automorphisms α of this structure are the bijections of A. A *set with atoms* is any set built from set constructors and atoms from $\mathbb{A}$. For an automorphism α of $\mathbb{A}$ and a set with atoms x we extend α to x as $\alpha(x) = \{\alpha(y) \mid y \in x\}$. An automorphism α *fixes* a set with atoms x if $\alpha(x) = x$. Fix a tuple $\bar{a} \in A^n$. An *$\bar{a}$-automorphism* is an automorphism fixing $\bar{a}$. A set with atoms x is *supported by* $\bar{a}$ if x is fixed by every $\bar{a}$-automorphism, and it is *equivariant* if it is supported by the empty set. A set with atoms is *finitely supported* if it is supported by some finite tuple of atoms, and it is *legal* if it is finitely supported and all its elements are legal (this is a recursive definition).

The *orbit* of element x is the set $\mathsf{orbit}(x) = \{\alpha(x) \mid \text{automorphism } \alpha\}$ of elements which can be obtained by applying some automorphism to x. Thus an equivariant set x is a union of orbits (of its elements), and an equivariant set is *orbit finite* if this union is finite. The set of orbits of a set x is $\mathsf{Orbits}(x) = \{\mathsf{orbit}(y) \mid y \in x\}$ (this is a partition of x). For instance, for given $a, b \in \mathbb{A}$ with $a \neq b$, we have $\mathsf{orbit}(a, a) = \big\{(c, d) \in \mathbb{A}^2 \mid c = d\big\}$ and $\mathsf{orbit}(a, b) = \big\{(c, d) \in \mathbb{A}^2 \mid c \neq d\big\}$. Since there are no more orbits in $\mathbb{A}^2$, we have $\mathsf{Orbits}(\mathbb{A}^2) = \{\mathsf{orbit}(a, a), \mathsf{orbit}(a, b)\}$ and $\mathbb{A}^2$ is the union of two orbits.

Exercise 10. Fix the equality atoms $(\mathbb{A}, =)$. For each of the following sets with atoms decide whether they are 1) legal, 2) equivariant, 3) orbit finite.

1. $\mathbb{A}^2$.
2. $\mathbb{A}^* := \mathbb{A}^0 \cup \mathbb{A}^1 \cup \mathbb{A}^2 \cup \cdots$.
3. $\mathbb{A}^\omega := \{a_1 a_2 \cdots \mid a_1, a_2, \cdots \in \mathbb{A}\}$.
4. $2^{\mathbb{A}} := \{B \mid B \subseteq \mathbb{A}\}$ (powerset).
5. $2^{\mathbb{A}}_{\text{fin}} := \{B \mid B \subseteq \mathbb{A}, B \text{ finite}\}$ (finite powerset).
6. $2^{\mathbb{A}}_{\text{fs}} := \{B \mid B \subseteq \mathbb{A}, B \text{ finitely supported}\}$ (finitely supported powerset).
7. $2^{\mathbb{A}}_{\text{eq}} := \{B \mid B \subseteq \mathbb{A}, B \text{ equivariant}\}$ (equivariant powerset).

Exercise 11. An atom structure $\mathbb{A}$ is called *oligomorphic* if $\mathbb{A}^n$ is orbit-finite for every $n \in \mathbb{N}$. Are the following atom structures oligomorphic?

1. $(\mathbb{N}, =)$.
2. $(\mathbb{Z}, \leq)$.

3. $(\mathbb{Q}, \leq)$.
4. $(\mathbb{Q}, +)$.