Homework 3

Michael Levet

1 Quantifiers

(Recommended) Problem 1. For each quantified statement, do the following:

1. Write the statement in English.
2. Determine whether the statement is true. If the statement is false, give a counter-example.
3. Write the negation of the statement using quantifiers.
4. Write the negation of the statement in English.

(a) $\forall x \in \mathbb{R}, x^2 > 0$.

(b) $\forall x \in \mathbb{R}, \exists n \in \mathbb{Z}^+, x^n \geq 0$.

(c) $\exists a \in \mathbb{R}, \forall y \in \mathbb{R}, ay = y$.

(d) $\forall n \in \mathbb{N}, \exists X \in 2^{\mathbb{N}}, |X| < n$.

(e) $\forall n \in \mathbb{Z}, \exists m \in \mathbb{Z}, m = n + 5$.

(f) $\exists m \in \mathbb{Z}, \forall n \in \mathbb{Z}, m = n + 5$.

Definition 2. Let $f : \mathbb{R} \to \mathbb{R}$. We say that f is *continuous* if for every $x \in \mathbb{R}$ and every $\epsilon > 0$, there exists $\delta > 0$ such that if $|y - x| < \delta$, then $|f(y) - f(x)| < \epsilon$.

Definition 3. We say that $f : \mathbb{R} \to \mathbb{R}$ is *uniformly continuous* if for every $\epsilon > 0$, there exists $\delta > 0$, such that if $x, y \in \mathbb{R}$ satisfy $|y - x| < \delta$, then $|f(y) - f(x)| < \epsilon$.

Remark: Note that in the definition of a continuous function, δ depends on both ϵ and x. That is, there is a δ associated with each pair (x, ϵ). However, if f is uniformly continuous, then the same ϵ works for all x. So δ is only associated with ϵ in the case of uniform continuity.

(Recommended) Problem 4. Negate the definition of a continuous function. Your final answer should be an English sentence without any quantifier symbols.

(Recommended) Problem 5. Negate the definition of a uniformly continuous function. Your final answer should be an English sentence without any quantifier symbols.

2 Polynomial-Time Heirarchy

We recall the definitions of NP and coNP.

Definition 6 (NP)**.** We say that a language $L \in \mathsf{NP}$ if there exists a polynomial $p(\cdot)$ depending only on L and a verifier V such that if $x \in L$, then there exists a string y of length at most $p(|x|)$, such that $V(x, y) = 1$ and V runs in time $\mathcal{O}(p(|x|))$. Here, y is the *certificate*.

Remark: We may write the definition of NP in quantifier notation:

$$x \in L \iff \exists y \text{ s.t. } |y| \leq p(|x|), V(x, y) = 1. \tag{1}$$

This expression is often abbreviated as follows, though (1) is the formalism and intended meaning.

$$x \in L \iff \exists y, V(x, y) = 1.$$

Similarly, coNP is defined as follows.

Definition 7 (coNP)**.** We say that a language $L \in \mathsf{coNP}$ if there exists a polynomial $p(\cdot)$ depending only on L and a verifier V such that if $x \in L$, then for every y of length at most $p(|x|)$, that $V(x, y) = 0$ and V runs in time $\mathcal{O}(p(|x|))$.

Remark: We may write the definition of coNP in quantifier notation:

$$x \in L \iff \forall y \text{ s.t. } |y| \leq p(|x|), V(x, y) = 0. \tag{2}$$

This expression is often abbreviated as follows, though (2) is the formalism and intended meaning.

$$x \in L \iff \forall y, V(x, y) = 0.$$

(Advanced) Problem 8. Show that if $L \in \mathsf{NP}$, then the complement $\overline{L} \in \mathsf{coNP}$.

(Advanced) Problem 9. Show that if $L \in \mathsf{coNP}$, then the complement $\overline{L} \in \mathsf{NP}$.

Our goal now is to generalize NP and coNP. We begin with some motivation. Recall the Independent Set decision problem, which takes as input a graph $G(V, E)$ and integer k, and asks if G has an independent set of size k. Recall that Independent Set is NP-complete. In particular, Independent Set $\in$ NP.

Now consider the Maximum Independent Set problem, which again takes as input a graph $G(V, E)$ and integer k. Here, we ask whether the largest size independent set in G has k vertices. Here, we need to verify a couple conditions:

- G has an independent set of k vertices. This is precisely the condition that Independent Set $\in$ NP.
- G does not have an independent set of $k + 1$ vertices. We note that verifying this second condition is a coNP problem.

So effectively, $(G, k) \in$ Maximum Independent Set $\iff$ there exists a small certificate of one type and no small certificate of another type. This motivates the definition of a new complexity class, which we will call Σ_2^{P}.

Definition 10. We say that a language $L \in \Sigma_2^{\mathsf{P}}$ (pronounced Sigma-2) if there exists a polynomial $p(\cdot)$ depending only on L and a verifier V such that if $\omega \in L$, then there exists a string x of length at most $p(|\omega|)$, such that for all strings y of length at most $p(|\omega|)$, $V(\omega, x, y) = 1$ and V runs in time $\mathcal{O}(p(|\omega|))$.

We may again express the definition of Σ_2^{P} in quantifier notation.

$$\omega \in L \iff \exists x \text{ s.t. } |x| \leq p(|\omega|), \forall y \text{ s.t. } |y| \leq p(|\omega|), M(\omega, x, y) = 1.$$

As we saw with NP and coNP, the quantified expression for Σ_2^{P} is commonly abbreviated as follows.

$$\omega \in L \iff \exists x, \forall y, M(\omega, x, y) = 1.$$

Example 11. We note that Maximum Independent Set $\in \Sigma_2^{\mathsf{P}}$. Here, x is the certificate for the independent set of size k, and y is a vertex set of size $k + 1$. In other words, M checks that x is an independent set of size k and that y is a $(k + 1)$-size vertex set is not an independent set. Note that M itself does not consider all such $(k + 1)$-size vertex sets at once. Rather, the quantifier states that for any given $(k + 1)$-size vertex set y, that M will check that y is not an independent set.

We now turn our attention to some general properties of Σ_2^{P}.

(Advanced) Problem 12. Show that $\mathsf{NP} \subseteq \Sigma_2^{\mathsf{P}}$.

(Advanced) Problem 13. Show that $\mathsf{coNP} \subseteq \Sigma_2^{\mathsf{P}}$.

2.1 Σ_i^{P}

We now turn to generalizing Σ_2^{P}. Here, the subscript 2 indicates that we use two quantifiers. We define Σ_i^{P} to use i quantifiers, starting with $\exists$, and then alternating between $\exists$ and $\forall$. This is formalized as follows.

Definition 14. We say that the language $L \in \Sigma_i^{\mathsf{P}}$ if there exists a polynomial $p(\cdot)$ depending only on L and a verifier V such that:

$$\omega \in L \iff \exists x_1, \forall x_2, \exists x_3, \forall x_4, \ldots, Q_i x_i, V(\omega, x_1, \ldots, x_i) = 1,$$

$|x_j| \leq p(|\omega|)$ for all $1 \leq j \leq i$, and V runs in time $\mathcal{O}(p(|\omega|))$. Note that Q_i indicates a quantifier. In particular, Q_i is an existential quantifier if i is odd and a universal quantifier if i is even.

So for Σ_3^{P}, the abbreviated quantified expression is:

$$\omega \in L \iff \exists x_1, \forall x_2, \exists x_3, V(\omega, x_1, x_2, x_3) = 1.$$

Similarly, for Σ_4^{P}, the abbreviated quantified expression is:

$$\omega \in L \iff \exists x_1, \forall x_2, \exists x_3, \forall x_4, V(\omega, x_1, x_2, x_3, x_4) = 1.$$

Remark: It is also worth noting that $\mathsf{NP} = \Sigma_1^{\mathsf{P}}$.

(Advanced) Problem 15. Show that for each i, $\Sigma_i^{\mathsf{P}} \subseteq \Sigma_{i+1}^{\mathsf{P}}$. Note that this generalizes Problem 12.

2.2 Π_i^{P}

We now turn our attention to generalizing coNP. Note that the quantified expression for coNP is obtained by negating the quantifiers for NP. We define the complexity class $\Pi_i^{\mathsf{P}} := \mathsf{co}\Sigma_i^{\mathsf{P}}$. That is, we negate the quantified expression for Σ_i^{P} to obtain a similar definition regarding alternating quantifiers. However, we begin with a universal quantifier rather than an existential quantifier. This definition is formalized as follows.

Definition 16. We say that the language $L \in \Pi_i^{\mathsf{P}}$ if there exists a polynomial $p(\cdot)$ depending only on L and a verifier V such that if $\omega \in L$

$$\omega \in L \iff \forall x_1, \exists x_2, \forall x_3, \exists x_4, \ldots, Q_i x_i, V(\omega, x_1, \ldots, x_i) = 0,$$

$|x_j| \leq p(|\omega|)$ for all $1 \leq j \leq i$, and V runs in time $\mathcal{O}(p(|\omega|))$. Note that Q_i indicates a quantifier. In particular, Q_i is an existential quantifier if i is even and a universal quantifier if i is odd.

So for Π_3^{P}, the abbreviated quantified expression is:

$$\omega \in L \iff \forall x_1, \exists x_2, \forall x_3, V(\omega, x_1, x_2, x_3) = 0.$$

Similarly, for Π_4^{P}, the abbreviated quantified expression is:

$$\omega \in L \iff \forall x_1, \exists x_2, \forall x_3, \exists x_4, V(\omega, x_1, x_2, x_3) = 0.$$

We now establish the following relations amongst the classes of the polynomial time heirarchy.

(Advanced) Problem 17. Show that $\Pi_i^{\mathsf{P}} \subseteq \Pi_{i+1}^{\mathsf{P}}$.

(Advanced) Problem 18. Show that $\Sigma_i^{\mathsf{P}} \subseteq \Pi_{i+1}^{\mathsf{P}}$.

(Advanced) Problem 19. Show that $\Pi_i^{\mathsf{P}} \subseteq \Sigma_{i+1}^{\mathsf{P}}$.

As a final note, we define the Polynomial-Time Heirarchy formally:

Definition 20. The *Polynomial-Time Heirarchy*, denoted PH, is:

$$\mathsf{PH} = \bigcup_{i \in \mathbb{N}} \Sigma_i^{\mathsf{P}}.$$