Problem 1: Big languages in $\mathbf{P}/\text{poly}$

(a) We saw in the lecture that

$$\text{UHALT} = \{1^n : n = \langle \alpha, x \rangle, M_\alpha(x) \text{ halts} \}$$

is in $\mathbf{P}/\text{poly}$. Show that $\text{UHALT}$ is undecidable.

Note: There are two ways to do it (or more). One is by reducing to $\text{HALT}$, the other is by redoing the proof of undecidability of $\text{HALT}$. The reference solution will use the first way, but the second way is fine as well.

Solution. We show it by reduction to $\text{HALT}$. Assume that $\text{UHALT}$ is decidable. That is, there is a TM $M$ (not necessarily polynomial-time) such that $\forall x. M(x) = 1 \iff x \in \text{UHALT}$.

Let $M'(\alpha, x)$ do the following: Set $n := \langle \alpha, x \rangle$, run $b := M(1^n)$, return $b$. Note that $M'$ may take much more runtime than $M$ because it runs $M$ with an exponentially larger input. But that does not matter here, because we are not concerned with the runtime of the TMs.

We have for all $\alpha, x$: $M'(\alpha, x) = 1$ iff $M(1^n) = 1$ with $n := \langle \alpha, x \rangle$ iff $1^n \in \text{UHALT}$ iff $M_\alpha(x)$ halts iff $\langle \alpha, x \rangle \in \text{HALT}$. Thus $M'$ decides $\text{HALT}$. Hence $\text{HALT}$ is decidable. But we know from the lecture that $\text{HALT}$ is undecidable. Thus the assumption that $\text{UHALT}$ is decidable was wrong. Thus $\text{UHALT}$ is undecidable. 

(b) Show that there is a decidable language $L$ such that $L \in \mathbf{P}/\text{poly}$ but $L \notin \mathbf{P}$.

Hint: Show that there exists a decidable language with $L' \notin \text{DTIME}(2^{2|x|^2})$. Construct the unary variant of $L'$: $L := \{1^n : n = x, x \in L' \}$. Show that $L$ is decidable, show that $L \in \mathbf{P}/\text{poly}$ (think of $\text{UHALT}$), and that $L \notin \mathbf{P}$ (assume that $L \in \mathbf{P}$, and from that construct a TM deciding $L'$ in too little time).

Solution. The time hierarchy theorem implies that there exists a language $L' \notin \text{DTIME}(2^{2|x|^2})$ but $L' \in \text{DTIME}(T')$ for some larger $T'$. In particular, $L'$ is decidable.
Let $L$ be the unary variant of $L'$. That is,

$$L := \{1^n : n = x, x \in L'\}.$$  

Then $L$ is decidable: given $1^n$, we simply run $L'(n)$.

Also we have that $L \in \text{P/poly}$. This is analogous to showing that UHALT $\in \text{P/poly}$: Let $C_n(x) := x_1 \wedge \cdots \wedge x_n$ iff $n \in L'$, and $C_n(x) = 0$ otherwise. Then $C_n$ form a polynomial-size circuit family that decides $L$.

Assume that $L \in \text{P}$. Then there is a polynomial $p$ and a TM $M$ that decides $L$ in time $p(|x|)$. We construct the Turing machine $M'$: Given input $x$, it runs $b := M(1^n)$ with $n := x$ and returns $b$. Since $M$ decides $L$, $M'$ decides $L'$.

The running time of $M'$ is: $O(p(2^{|x|}))$ because it runs $M$ with an input $1^n$ of lengths $2^{|x|}$. Since $p$ is a polynomial, $O(p(2^{|x|})) = O((2^{|x|}c) \subseteq O(2^{|x|})$ for some constant $c > 0$. Thus $M'$ runs in time $O(2^{|x|})$ and decides $L'$, in contradiction to the assumption that $L' \notin \text{DTIME}(2^{|x|})$.

Problem 2: Circuit lower bounds: Parity

Let

$$\text{PARITY} := \{x : x \text{ has an odd number of 1's}\}.$$  

(Or expressed differently: $x \in \text{PARITY}$ iff $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$.)

We will show a lower bound on the circuit complexity of PARITY. Namely, we show that constant-depth circuits cannot compute PARITY.

For this, we first establish a bit of notation:

- The **height** of a node $\nu$ in a circuit $C$ is the longest path from $\nu$ to any leaf. That is, leaves (variables) have height 0. And an inner node has the maximum height of its children, plus 1.

- The **depth** of a circuit is the height of its root.

- Given a polynomial $p$ over the real numbers with $n$ variables, and a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we say that $p$ **computes** $f$ iff $\forall x_1, \ldots, x_n \in \{0, 1\}. p(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$. (Note: we do not care what $p$ evaluates to when it gets inputs different from 0, 1.)

- We say $p$ computes a node $\nu$ in a circuit $C$ if $p$ computes the function $f$ evaluated by the node $\nu$. (That is, for any assignment to the variables $x_1, \ldots, x_n$ of $C$, the node $\nu$ has a well-defined value $\in \{0, 1\}$, so the node evaluates some function $f$ of $x_1, \ldots, x_n$. We want $p$ to compute that function.)

- The **degree** of a multivariate polynomial $p$ is the largest sum of the exponents in any monomial. (E.g., $6x_1x_2x_3^2 + x_2x_5$ has degree 4 from the first monomial.)
• We call a polynomial \textit{multilinear} if no variable occurs with an exponent greater than 1. (That is, $6x_1x_2x_3 + x_2x_5$ is multilinear, but $6x_1x_2x_3^2 + x_2x_5$ is not multilinear.)

We now develop a proof that \textsc{parity} cannot be computed by constant-depth circuits:

(a) Show: For each leaf $\nu$ of a circuit $C$, there is a polynomial $p$ of degree $\leq 1$ that computes $\nu$.

\textbf{Solution.} Since $\nu$ is a leaf, $\nu$ is a variable, say $x_i$. Then $\nu$ evaluates the function $f(x_1,\ldots,x_n) := x_i$. Then the polynomial $p := x_i$ computes $f$ and therefore $p$ computes $\nu$. And $p$ has degree 1.

(b) Show: For each node $\nu$ of height $h$ in a circuit $C$, there is a polynomial of degree $2^h$ that computes $\nu$.

\textbf{Hint:} Induction over the height, using (a). Express each AND/OR/NOT node as a polynomial of its children’s values, and compute the degree of the polynomial when plugging in the inputs. Remember that all nodes have fan-in at most 2.

\textbf{Solution.} For $h = 0$, this follows directly from (a). We do induction over $h$, assume it holds for all heights smaller than $h$. We show the statement for height $h$. Fix a node $\nu$ of height $h+1$. Then all children $\nu_i$ of $\nu$ have height smaller than $h$. Thus by induction hypothesis, there are polynomials $p_i$ of degree $\leq 2^{h-1}$ such that $p_i$ computes $f_i$ where $f_i$ is the function computed by $\nu_i$. Let $f$ be the function computed by $\nu$. We abbreviate $x_1,\ldots,x_n$ as $\bar{x}$. We distinguish three cases, depending on whether $\nu$ is AND/OR/NOT:

- $\nu =$ AND: For all $\bar{x} \in \{0,1\}^n$, $f(\bar{x}) = f_1(\bar{x}) \land f_2(\bar{x}) = f_1(\bar{x}) \cdot f_2(\bar{x}) = p_1 \cdot p_2 =: p$. Thus $p$ computes $f$ and therefore $p$ computes $\nu$. And $p$ has degree $\leq 2^{h-1} + 2^{h-1} = 2^h$ because when multiplying polynomials, the degrees add up.
- $\nu =$ OR: For all $\bar{x} \in \{0,1\}^n$, $f(\bar{x}) = f_1(\bar{x}) \lor f_2(\bar{x}) = f_1(\bar{x}) + f_2(\bar{x}) - f_1(\bar{x}) \cdot f_2(\bar{x}) = p_1 + p_2 - p_1 p_2 =: p$. Thus $p$ computes $\nu$. And $p$ has degree $\leq 2^{h-1} + 2^{h-1} = 2^h$.
- $\nu =$ NOT: For all $\bar{x} \in \{0,1\}^n$, $f(\bar{x}) = \neg f_1(\bar{x}) = 1 - f_1(\bar{x}) = 1 - p_1 =: p$. Thus $p$ computes $\nu$. And $p$ has degree $\leq 2^{h-1} \leq 2^h$.

Thus in each case, there is a polynomial $p$ of degree $\leq 2^h$ that computes $\nu$.

(c) \textbf{[Bonus points, tricky]} Let $f_n(x_1,\ldots,x_n) := x_1 \oplus \cdots \oplus x_n$. Assume that $p_n$ is a multilinear polynomial that computes $f_n$. Show that $p_n$ contains the monomial $\alpha x_1 x_2 \ldots x_n$ (with some coefficient $\alpha \neq 0$).

\textbf{Hint:} Show it for $n = 1$ first. Then do induction. Express the polynomial $p_n$ as $p_n = x_n q + r$. Relate the polynomials $r$, $1 - (q + r)$, $1 - (q + r) + r$, and finally $(1 - q)/2$ to $f_{n-1}$. Show that $q$ contains $\alpha x_1 \ldots x_{n-1}$ using induction hypothesis.
Solution. For \( n = 1 \), \( f_n = x_1 \). Since \( p_n \) is a multilinear polynomial in one variable, it must be of the form \( ax + b \). Only with \( a = 1 \) and \( b = 0 \) we have that \( p_n(x_1) = f_n(x_1) \) for all \( x_1 \in \{0, 1\} \). Thus for \( n = 1 \), \( p_n = x_1 \) and thus contains the monomial \( x_1 \). So the statement holds for \( n = 1 \).

Assume the statement holds for \( n - 1 \). We show it for \( n \geq 2 \). Since \( p_n \) is multilinear, it does not contain \( x_n \) with an exponent greater than 1. Hence \( p_n \) can be written as \( p_n = x_n q + r \) where \( q, r \) are multilinear polynomials over \( x_1, \ldots, x_{n-1} \). For all \( x_1, \ldots, x_{n-1} \in \{0, 1\} \), we have

\[
f_{n-1}(x_1, \ldots, x_{n-1}) = f_n(x_1, \ldots, x_{n-1}, 0) \overset{(1)}{=} p_n(x_1, \ldots, x_{n-1}, 0) = 0 \cdot q + r = r. \tag{1}
\]

Here \( (\ast) \) uses that \( p_n \) computes \( f_n \). Hence \( r \) computes \( f_{n-1} \).

We further have

\[
f_{n-1}(x_1, \ldots, x_{n-1}) = 1 - f_n(x_1, \ldots, x_{n-1}, 1) \overset{(\ast)}{=} 1 - p_n(x_1, \ldots, x_{n-1}, 1) = 1 - (1 \cdot q + r) = 1 - (q + r). \tag{2}
\]

Here \( (\ast) \) uses that \( p_n \) computes \( f_n \). Hence \( 1 - (q + r) \) computes \( f_{n-1} \).

Thus \( 1 - (q + r) + r \) computes \( 2f_{n-1} \). Since \( 1 - (q + r) + r = 1 - q \), we have that \( 1 - q \) computes \( 2f_{n-1} \) and thus \( (1 - q)/2 \) computes \( f_{n-1} \). By induction hypothesis, this implies that \( (1 - q)/2 \) contains the monomial \( \alpha' x_1 \ldots x_{n-1} \) for some \( \alpha' \neq 0 \).

Thus \( q \) contains the monomial \( \alpha x_1 \ldots x_{n-1} \) with \( \alpha := -2 \alpha' \neq 0 \). Thus \( x_n q \) contains \( \alpha x_1 \ldots x_n \). And hence \( p_n = x_n q + r \) contains \( \alpha x_1 \ldots x_n \). \( r \) cannot cancel out that monomial, because \( r \) contains only the variables \( x_1, \ldots, x_{n-1} \).

Since \( p_n \) was an arbitrary multilinear polynomial computing \( f_n \), the statement follows.

(d) Let \( f_n(x_1, \ldots, x_n) := x_1 \oplus \cdots \oplus x_n \). Assume that \( p_n \) is a polynomial that computes \( f_n \). Show that \( p_n \) has degree at least \( n \).

Hint: Transform \( p_n \) into a multilinear polynomial by removing all exponents from \( p_n \). Show that the resulting polynomial still computes \( f_n \). Then use (\ref{eq:monomial1}).

Solution. Let \( \hat{p}_n \) be the polynomial that results from replacing all \( x_i^e \) by \( x_i \) for all exponents \( e > 1 \). Then \( \hat{p}_n \) is multilinear. Also, the degree of \( p_n \) is at least as large as that of \( \hat{p}_n \). And for all \( x_1, \ldots, x_n \in \{0, 1\} \) we have \( \hat{p}_n = p_n \). (Because \( x^e = x \) for \( x \in \{0, 1\} \).) Thus, since \( p_n \) computes \( f_n \), \( \hat{p}_n \) computes \( f_n \). By (\ref{eq:monomial1}), \( \hat{p}_n \) then contains the monomial \( \alpha x_1 \ldots x_n \). Thus \( \hat{p}_n \) has degree at least \( n \). Thus \( p_n \) has degree at least \( n \).

(e) Show that no circuit \( C_n \) of depth \( d < \log n \) with \( n \) variables decides \textsc{parity}.

Hint: Use (\ref{eq:monomial1}) and (\ref{eq:monomial2}).

\(^1\)The last two steps are the reason why we cannot do the same proof for polynomials over \( \text{GF}(2) \) instead of the reals. Over \( \text{GF}(2) \), \((1 - q)/2\) is undefined and \( 2 \alpha' = 0 \). In fact, over \( \text{GF}(2) \) there is a polynomial of degree 1 that computes \( f_n \), namely \( p_n := x_1 + \cdots + x_n \).
**Solution.** Assume that $C_n$ decides \textsc{Parity}. That is, $C_n$ computes $f_n(x_1, \ldots, x_n) := x_1 \oplus \cdots \oplus x_n$. By (b), there is a polynomial $p_n$ that computes the root node of $C_n$, and $p_n$ has degree at most $2^d$. (Since the root node has height $d$.) Since the root node evaluates $f_n$, $p_n$ computes $f_n$. By (d), this means that $p_n$ has degree at least $n$. Thus $2^d \geq n$, hence $d \geq \log n$. \hfill \textit{.solution}