For 2:

- Let's sps in def'n of H_f we actually include $M;x$ where M accepts x in $\le f(2|x|+1)+c$-many steps, where c is a small constant (see below)
- doesn't change that $H_f \in TIME(O(f))$ at least in cases we care about.

- Now: sps toward a contradiction we can find a t.m T that decides if $w \in H_f$ in $\le f(|w|)$-many steps

- Let S be the machine that, on input w, runs T on $w;w$ and halts in state q_y iff T halts in q_n
- "Clear" there is such an S that halts as such in (time it takes T to halt on $w;w$) + small constant c

$$\le f(|w;w|)+c$$

$$= f(2|w|+1)+c$$

by hypothesis

Now the diagonalization

- Let M_S denote code for S (using our coding for t.m.'s above)

- we ask: is $M_S; M_S \in H_f$?
- to check: run T on input $M_S; M_S$ and see if halt in Q_y or Q_n
  - ↳ if Q_y: means T ran S on input M_S and found S halted in state Q_y (in $\leq f(2|M_S|+1)+c$-many steps)
    - ↳ but this means that S ran T on input $M_S; M_S$ and found it halted in Q_n — contradiction
  - ↳ likewise: if Q_n: means that T ran S on M_S and either
    - (a) S didn't halt by time $f(2|M_S|+1)+c$
    - (b) S did, in state Q_n
  - ↳ but we know on input M_S, S halts in $\leq f(2|M_S|+1)+c$-many steps so it's not (a)
  - ↳ but (b) means S ran $M_S; M_S$ on T and got a Q_y, again a contradiction ✓

<u>Corollary</u>: $P \neq EXP$
- "Clearly" 2^n is proper, so by theorem there is $L \in TIME(c(2^n)) \setminus TIME(2^n)$

- But any $L \in TIME(O(2^n))$ is still in EXP
- Can't have $L \in P$ as this would imply $L \in TIME(2^n)$
  (any poly $p(n)$ is eventually dominated by 2^n) ✓

- Theorem actually shows much finer results: e.g. for any proper poly p there is some L that can be decided in $O(p)$ but not p-time.

Complexity of Presburger Arithmetic

- We showed before:
  $Th(\mathbb{N}, <, S, +, 0)$ ("Presburger Arithmetic" "PrA")
  is decidable

- What is its time complexity?
  i.e. given a sentence φ written in lang of PrA, how long (in general) does it take to decide if $\varphi \in PrA$?

Theorem PrA is EXP-hard, i.e. $\forall c > 0$ for any $L \in TIME(2^{cn})$, there is a poly-time reduction of L to PrA.

Pf: - Sps $L \in TIME(2^{cn})$ for some $c > 0$; alphabet is A
- Let M be a t.m. witnessing $L \in TIME(2^{cn})$; alphabet of M is $B \supseteq A$.
- We describe a poly-time assignment $w \mapsto \varphi_w(x)$ (where $w \in L$, and φ_w is a formula in lang of PrA w/ one free var) such that:

$w \in L$ iff $\exists x\, \varphi_w(x) \in PrA$

- This suffices to prove theorem: deciding membership in PrA must in general be at least as hard as deciding membership in L.

- Idea is: $\varphi_w(x)$ should express "x codes a computation of M on input w of length $\leq 2^{c|w|}$ that terminates in accepting state q_y"

- So fix ~~[illegible]~~ w
- Since M decides if $w \in L$ in $\leq 2^{c|w|}$-many steps, can only see ~~[illegible]~~ cells

(26)

on its tape in the interval $[-2^{c|w|}, 2^{c|w|}]$.
- Let $B = \{b_2, \ldots, b_q\}$ be alphabet of M
- number states of M as $Q_{q+1}, Q_{q+2}, \ldots, Q_{p-1}$ where $Q_{p-1} = Q_y$
- Let $b_0 = \#$ and $b_1 = *$

- As before, we can encode a stage of the computation of M on input w by a sequence:
  $b_{i_0} \ldots b_{i_e} Q_j b_{i_{e+1}} \ldots b_{i_{2^{c|w|+1}}}$

- (Here: the last index is $2^{c|w|+1} = 2^{c|w|} \cdot 2$ since we want to record entry on every cell from $-2^{c|w|}$ to $2^{c|w|}$ — but want to use natural numbers as indices)
- using $* = b_1$ for blank spaces
- using $\# = b_0$ as separator, can encode a sequence of stages too
- using natural numbers i to encode the symbols b_i and j to encode the states Q_j, a computation of M (i.e. sequence of computation stages) can be represented by a natural number in base p

- Notice: a computation of length $2^{c|w|}$, w/ stages rep'd by sequences

of length $2^{c|w|+1}$, is rep'd by a sequence (of symbols) of length $2^{c|w|}$.
$(2^{c|w|+1} + 1)$ (extra +1 for the #'s)
which is (say) $< 2^{2c|w|+2}$

Notice: - if the sequence is
$(\dots) \# (\dots) \# \dots \# (\dots)$ (+)
where each $(\dots)$ is a stage of the computation, then writing (+) as $a_0 a_1 \dots a_N$ $(N < 2^{2c|w|+2})$, the number x encoding this sequence is
$a_0 + a_1 p + \dots + a_N p^N$ $(a_i < p \ \forall i)$
which is $\leq N \cdot p^{N+1}$
which is $\leq 2^{2^{d|w|}}$ for some large enough (but fixed) d, depending only on c and p.

- Now: want a fmla $\varphi_w(x)$ s.t. $w \to \varphi_w(x)$ is poly time (in $|w|$) and for $x \in \mathbb{N}$ we have:
$(\mathbb{N}, <, S, +, 0) \models \varphi_w(x)$ iff $x < 2^{2^{d|w|}}$ and x codes comp'n of M on input w that halts in state q_y.

- How to recognize if x is such a code (will be unique if exists - THE code)

- let $n = |w|$
- $\varphi_w(x)$ should assert
  "x, viewed in base p, codes a sequence $(\dots)\#(\dots)\#\dots\#(\dots)$
  where: - string is length $\le N$ (i.e. $x < 2^{2^{dn}}$)
  - are exactly 2^{cn}-many $(\dots)$'s
  - each $(\dots)$ is (a code for) a stage of comp'n of M that follows preceding one using one of M's instructions
  (- if we halt before 2^{cn}th-stage - consider all subsequent stages a copy of prev. one)
  - first $(\dots)$ is q_0 (symbols in w)(some blank spaces)
  - last $(\dots)$ is of form $b_{\mathcal{U}} \dots q_y \dots b_{2^{cn+1}}$
  (i.e. we terminate in q_y)"

- Can we express above in lang of PrA?
- Write $\Phi_p(x,i)$ for ith digit of x written in base p, i.e. if $x = a_0 + a_1 p + \dots + a_i p^i + \dots + a_m p^m$ $(a_i < p \ \forall i)$
then $\Phi_p(x,i) = a_i$

- Leave it to you to convince yourself above can be expressed in lang of PrA (in a poly-in-n length fmla) as long as we can define the following relations:

$y < 2^n$ (write $A_n(y)$ for this)

$x < 2^{2^n}$ (write as $B_n(x)$)

$\Phi_p(x,y) = z$

- e.g. can express "$x < 2^{2^{dn}}$ and every $(2^{cn+1}+2)$nd digit of x (viewed in base p) is # (i.e. 0)" by the formula:

$$B_{dn}(x) \wedge \forall i\, (A(i) \Rightarrow \Phi_p(x, i(2^{cn+1}+2)) = 0)$$

- Also leave to you to convince self that the relations A, B, Φ_p can be defined using following relations:

$(x < 2^{2^n}) \wedge (xy = z)$ (write as $M_n(x,y,z)$)

$(x, z < 2^{2^n}) \wedge (y^x = z)$ (denote $E_n(x,y,z)$)

- Note: Defining M_n, E_n in PrA is something like defining multiplication and exponentiation — something we know we can't do
- difference is the boundedness: can define $\cdot$ and exp below a given bound (in this case 2^{2^n})