Theorem:

Given a PTA A, we can effectively construct a PTA $\bar{A}$ with $L(\bar{A}) = \overline{L(A)}$.

Proof:

(a) · If $t \notin L(A)$ then player A does not have a winning strategy for $G(A, t)$ from (ε, q_0).

· By positional determinacy, this means player P has a positional winning strategy

$$str: T \times Q^{\leq n} \to D.$$

· We can consider the strategy as a function

$$str: T \to S \text{ with } S = Q^{\leq n} \to D = \{0, \dots, n-1\}.$$

· With S, we define the ranked alphabet

$$\Delta := \Sigma \times S \text{ where } rk(a, s) := rk(a).$$

Observation:

↳ We have $t \notin L(A)$

iff there is t' over $\Delta = \Sigma \times S$ so that

- $proj_1(t') = t$ and
- the S-labels define a positional strategy for P.

↳ More formally, we define the set L' of trees t' over $\Delta = \Sigma \times S$ so that

$$str(w, \bar{q}) := (proj_2(t'(w)))(\bar{q})$$

forms a positional winning strategy for P in $G(A, proj_1(t'))$

Point: · We have $\overline{L(A)} = proj_1(L')$.

· Hence, if L' is PTA-recognizable, then so is $\overline{L(A)}$ by closure under rank-preserving functions

(b) · A tree t' is in L'

iff $str(w, \bar{q}) := (proj_2(t'(w)))(\bar{q})$

is a <u>winning strategy</u> for player P

in $G(A, proj_1(t'))$ from (ε, q_0).

↳ For str to be a winning strategy, we have to consider

<u>every play</u> in $G(A, proj_1(t'))$ from (ε, q_0).

By definition of $G(A, proj_1(t'))$, such a play has the form

$$(\varepsilon, q_0) \rightarrow (\varepsilon, \bar{q}_0) \rightarrow (d_0, q_1) \rightarrow (d_0, \bar{q}_1) \rightarrow \dots$$

where

- $q_i \rightarrow_a \bar{q}_i$ is a transition in A (with $a = proj_1(t'(d_0 \dots d_{i-1}))$)
- $d_i = str(d_0 \dots d_{i-1}, \bar{q}_i)$
- $q_{i+1} = \bar{q}_i(d_i)$ // evaluate vector at component d_i.

The play is won by P if the highest priority that occurs infinitely often in

$$\Omega(q_0)\,\Omega(q_1)\,\Omega(q_2)\dots$$

is <u>odd</u>.

↳ Note that d_i and q_{i+1} are fully determined by strategy str for P.

Hence, <u>different plays</u> only arise from different <u>choices for transitions</u>

$$q_i \rightarrow_a \bar{q}_i \quad \text{by player } A.$$

Hence, to consider every play, we can alternatively consider every sequence of moves

$$z = \bar{q}_0\,\bar{q}_1\,\bar{q}_2 \dots \quad \text{for player } A.$$

Together with str, this sequence z again induces a pseudo play

$$(\varepsilon, q_0) \rightarrow (\varepsilon, \bar{q}_0) \rightarrow \dots \quad \text{as above.}$$

- The pseudo play may actually
  fail to be a play of $G(A, proj_1(t'))$ from (ε, q_0).
  The reason is an invalid transition
  $$q_i \not\to_a \bar{q}_i \quad \text{with } a = proj_1(t'(d_0 \ldots d_{i-1})).$$
  In this case, there is no need to consider
  the sequence $\bar{q}_0 \bar{q}_1 \bar{q}_2 \ldots$.
- If the pseudo play is a play,
  then P has to win.
  This means the highest priority that occurs infinitely often
  has to be odd.

↳ Next we observe that each play defines a path $\pi = d_0 d_1 \ldots$ in t'!
So to consider every play, it is sufficient to consider
- every path $\pi = d_0 d_1 d_2 \ldots$ in t' and
- every sequence of moves $z = \bar{q}_0 \bar{q}_1 \bar{q}_2 \ldots$ for player A.

As before, these two define a pseudo play
$$(\varepsilon, q_0) \to (\varepsilon, \bar{q}_0) \to (d_0, q_1) \to (d_0, \bar{q}_1) \to \ldots$$
Again the pseudo play may fail to be a play
of $G(A, proj_1(t'))$ from (ε, q_0).
There are two reasons:

(W1) We have $q_i \not\to_a \bar{q}_i$ for some transition (as before)

(W2) or the pseudo play does not follow str:
$$d_i \neq str(d_0 \ldots d_{i-1}, \bar{q}_i) \text{ for some } i.$$
If the pseudo play actually is a play, we require

(W3) that P wins (due to the highest priority being odd).

(C) Consider the set of paths $\pi = d_0 d_1 \ldots$ through t'
so that for every sequence of moves $z = \overline{q_0}\,\overline{q_1}\,\overline{q_2} \ldots$
we have $(W1) \vee (W2) \vee (W3)$.

These paths define an <u>ω-regular (word) language L_P</u>.

To be precise, we define the extended alphabet

$$\Delta \times D = (\Sigma \times S) \times D.$$

Then a word

$$(a_0, s_0, d_0)(a_1, s_1, d_1) \ldots \in L_P \subseteq (\Delta \times D)^\omega$$

iff

for all $\overline{q_0}\,\overline{q_1}\,\overline{q_2} \ldots$ where

$\Rightarrow$ $q_i \rightarrow_a \overline{q_i}$ with $a = \mathrm{proj}_1(t'(d_0 \ldots d_{i-1}))$ and   for all $i \in \mathbb{N}$

$\Rightarrow$ $d_i = s_i(\overline{q_i}) = \mathrm{str}(d_0 \ldots d_{i-1}, \overline{q_i}) < \mathrm{rk}(a_i)$

we have

the highest priority that occurs infinitely often

in $\Omega(q_0)\,\Omega(q_1) \ldots$ is odd.

Essentially, this definition of L_P rewrites

$(W1) \vee (W2) \vee (W3)$

to

$(\neg(W1) \wedge \neg(W2)) \rightarrow (W3)$.

(D) Now

$$L' = L_P^t.$$

Moreover, L_P^t is PTA-recognizable
by the lemma from last lecture. $\square$

To sum up the main steps of the proof:

↳ We construct the alphabet $\Delta = \Sigma \times S$ where

$$S = Q^{\leq n} \rightarrow D \text{ with } D = \{0, \ldots, n-1\}.$$

This decorates the trees over Σ with a strategy for P.

↳ Then we construct a deterministic parity (word) automaton (DPA) for
$$L_P \subseteq (\Delta \times D)^\omega.$$
The language requires that the decorating strategy is winning for P.
Let the DPA be
$$\mathcal{A}' = (\Delta \times D, Q_P, q_{0P}, \rightarrow_P, \Omega_P).$$
↳ From $\mathcal{A}'$, we obtain the tree automaton $\overline{\mathcal{A}}^*$ using the construction from the last lecture:
$$\overline{\mathcal{A}}^* = (\Delta, Q_P, q_{0P}, \rightarrow', \Omega_P)$$
where
$$q \rightarrow'_{(a,s)} (q^0, \dots, q^{rk(a,s)-1}) \text{ if } \forall i < rk(a,s) : q \xrightarrow{(a,s,i)} q^i.$$
↳ Finally, we project $\overline{\mathcal{A}}^*$ to the first component
$$\Sigma \text{ of } \Delta = \Sigma \times S$$
and obtain $\overline{\mathcal{A}}$ for $\overline{L(\mathcal{A})}$.