Last time: $\underline{p}$-PWPP and PWPP

$\{0,1\}^{n+p} \to \{0,1\}^n \leq \{0,1\}^{n+1} \to \{0,1\}^n$

$(\geq)$ $\quad (x_1, \ldots, x_p) \longmapsto$

$C(x_1) \| C(x_2) \| \cdots \| C(x_p)$

✓ mapping $\{0,1\}^{np} \to \{0,1\}^{np+p}$ bits

✓ reduction

✓ p constant

✗ $p = p(n)$ $\qquad \underbrace{\{0,1\}^{\underline{n+p(n)}} \to \{0,1\}^{np(n)+p(n)}}_{\text{not } p(n)\text{-PWPP}}$

$n+p(n) \longmapsto n+p(n)+p(n+p(n))$

$PWPP^{p(n)}$: $\{0,1\}^{p(n)} \to \{0,1\}^n \quad p(n) > n$

✓ (trivial) $PWPP^{p(n)} < PWPP$

$PWPP \leq PWPP^{p(n)}$

Given: $C: \{0,1\}^{n+1} \to \{0,1\}^n$

Construct $C_{p(n)}: \{0,1\}^{p(n)} \to \{0,1\}^n$

$$\underbrace{C(\ldots C(C(C}_{\substack{p(n) \text{ applications} \\ -n}}(\underbrace{C(x_1, \ldots x_{n+1})}_{n\text{-bits}}, x_{n+2}), x_{n+3}) \ldots), x_{p(n)})$$

For $(x_1, \ldots, x_{p(n)}) \neq (y_1, \ldots, y_{p(n)})$ that output same for $C_{p(n)}$

Look at first time a nested C had same output

$\Rightarrow$ collision of C $\square$

Back to graph problems

<u>Attempt</u> Given $d = \text{poly}(n)$, C representing G, (undirected) and v of deg 1, find another.

Let's add enforcements of acyclicity.

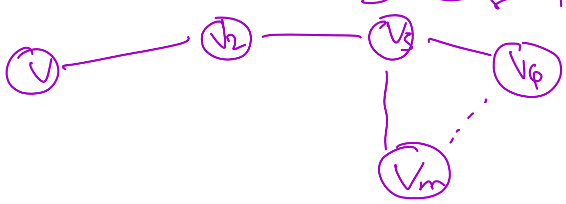
1) G undirected. Each $u \in V$ has exactly 1 or 2 neighbors.
How to enforce this syntactically: set <u>$d = 2$</u>.
(recall $(u,v) \in E$ if $C(u,i) = v$, $C(v,j) = u$ for $i, j \in [2]$)

$\Rightarrow$ <u>Def</u> END-OF-UNDIRECTED-LINE (EOUL)

<u>Def</u> PPA is the class of search problems/relations reducible to EOUL.
parity applications

Why acyclic?
- have v w/ deg 1

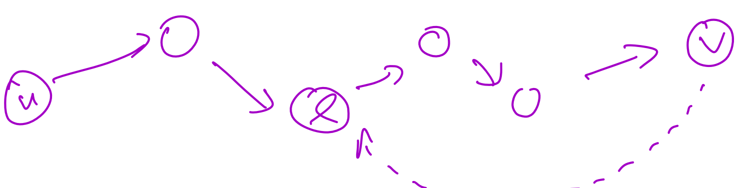
3-edges

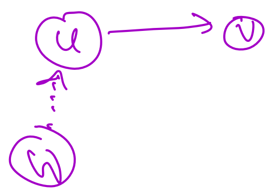


2) G is <u>directed</u> and each $u \in V$ has at most <u>one in-edge</u> and <u>one out-edge</u>

Recall (in convention used prev.)

Given $u \in V$ with <u>only an out-edge</u> (source) $C(u,i) = v$
Find $v \in V$ with no out-edge (sink) means





(would have >1 in-edge

so no cycles possible

Problem: no way to check how many in-edges a vertex has.

Idea: Allow as solutions a proof that either u has an in-edge ($v \in V$ st $C(v)=u$) or v that has more than 1 in-edge ($\exists w \neq x$, $C(w)=C(x)=v$)

Prop This "graph" problem is equivalent to PIGEONHOLE

Exercise.

Idea Enforce one in/out edge using an additional circuit for predecessors.

Def END-OF-LINE (EOL): given successor circuit $S:\{0,1\}^n \to \{0,1\}^n$, and predecessor circuit $P:\{0,1\}^n \to \{0,1\}^n$ representing a directed graph, and a source $v \in \{0,1\}^n$ st $S(v) \neq v$ (out-edge) and $P(S(v)) = v$ but $P(v)=v$ (same convention for none-edge). Find a sink (u st it has no successor/no out-edge but $P(u) \neq u$) and $S(P(u))=u$

from previous convention

$(u,v) \in E$ iff $S(u)=v$, $P(v)=u$

u: sink

- $S(u)=u$
- $S(u)=w$ but $P(w) \neq u$ (they don't agree so this edge doesn't exist)

Remark This is a directed version of EOUL.

END-OF-LINE $\in$ PPA

EOUL

Pf $S, P \longrightarrow C(x,\{1,2\})$

$S(u)=w$

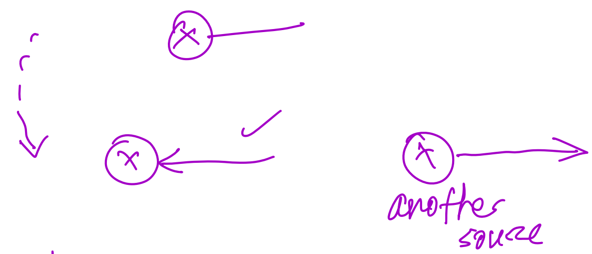
$P(w) \neq u$

$\Rightarrow$ no edge $u \to w$

$C(x,1) := S(x)$ $\qquad$ $S(w) = u$

$C(x,2) := P(x)$ $\qquad$ $P(u) = w$

Proof not complete, sketch (issue: the PPA solver returns a node of deg 1, $\neq v$.



Def PPAD (directed graphs) is the class of problems that reduce to EOL.

Prop PPAD $\subseteq$ PPA $\cap$ PPP

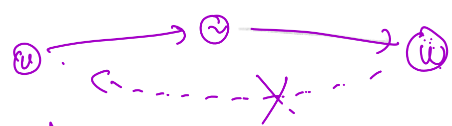
PPAD $\subseteq$ PPA (showed)

PPAD $\subseteq$ PPP from previous argument (only have a successor circuit, allow "violations" of degree as solutions)

OPEN: PPA $\cap$ PPP $\overset{?}{=}$ PPAD

(black-box separations)

3) An "ordering" on vertices (edges can only increase) would enforce no cycles.



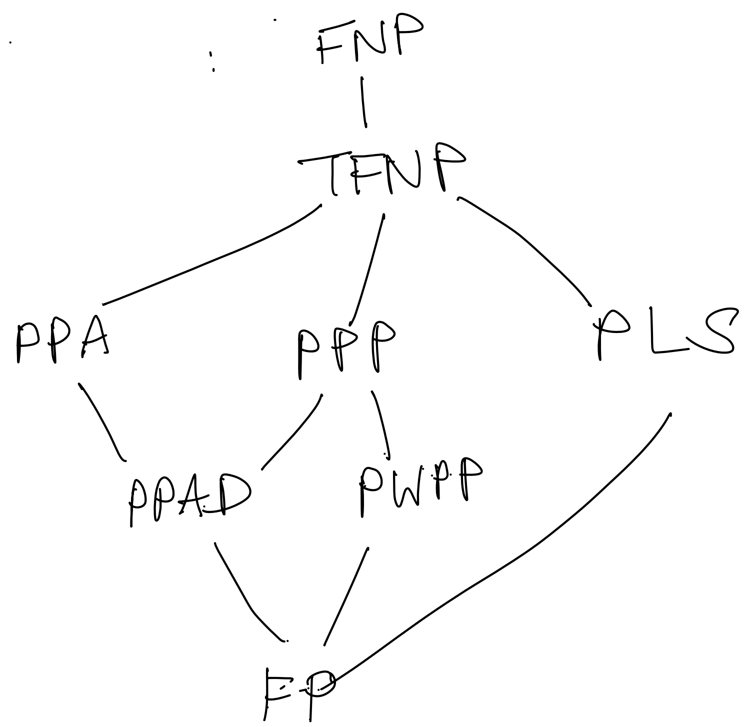
Def (ITER) Directed G rep. by C st $C(0^n) \neq 0^n$ and C must increase: u has an out-edge only if $C(u) > u$

Find a sink, ie v st v has no out-edge ($C(v) \leq v$)

(local search)

Def PLS is the class of problems reducing to ITER.

Picture so far



(The following was written on blackboard:)

Def: (ODD-DEGREE) Given a graph G represented by $C: \{0,1\}^n \times [d] \to \{0,1\}^n$ with $d = \mathrm{poly}(n)$, and an odd-degree u, find another odd-degree v

both names are used in literature

Def: (LONELY / PAIRING): Given a circuit $C: \{0,1\}^n \to \{0,1\}^n$, we say $a, b \in \{0,1\}^n$ are paired (aka "matched") if $a \neq b$ and $C(a) = b$ and $C(b) = a$. Given such a C and $u \in \{0,1\}^n$ such that $C(u) = u$ (i.e., u is unpaired), find another unpaired $w \neq u$

→ i.e., either $C(w) = w$
or $C(C(w)) \neq w$.

Note: sometimes people use $u = 0^n$ (as in Yuriko's lecture notes). These formulations are equivalent (can be reduced to each other)

Theorem: ODD-DEGREE and PAIRING/LONELY are both PPA Complete.

(proof omitted)