Teoremas, prop, etc. (Mi punto de vista de demos)

obs: $L \subseteq NL \subseteq P$

Demo: Si S es construible en espacio, sabemos q' $NSpace(S(n)) \subseteq Space(2^{O(S(n))})$. Tomar $S(n) = \log(n)$.

Proposición 23. PATH $\in$ **NL**.

Demo: Inventa un camino de s a t, almacenando 2 nodas únicamente $(O(2\log(n)))$, la longitud del camino es de a lo sumo $n-1$ $(n = |V|)$.

```
y ← s; m ← 0
mientras m < |V|:
    z ← inventar un valor en {0, ..., |V| − 1}
    si (y, z) ∉ E (para esto, revisa G en la entrada), pasar a q_no
    si no:
        si z = t, pasar a q_sí
        si no:
            y ← z; m ← m + 1
pasar a q_no
```

→ Es útil saberse el algoritmo concreto, pq' es usado algo similar en muchas casas.

Proposición 24. *Si $\mathcal{L} \notin \{\{0,1\}^*, \emptyset\}$ entonces $\mathcal{L}$ es* **NL-hard** *con respecto a* $\leq_p$.

Demo:

Puedo reducir a cualquier $\mathcal{L}' \in NL$ a un $\mathcal{L} \neq$ de los triviales.

Es absurdo pq' puedo computar $\chi_{\mathcal{L}'}$ en tiempo polinomial directamente, ya q' $NL \subseteq P$.

Proposición 25. *Sean f, g computables implícitamente en* **L**. *Entonces $g \circ f$ es computable implícitamente en* **L**.

Demo:

Sale de ir computando M_g y cada vez que requiera un bit de $f(x)$, lo computo con M_f.

$$\left.\begin{array}{l} M_g(\langle x, i\rangle) = g(x)(i) \\ M_f(\langle x, i\rangle) = f(x)(i) \end{array}\right\} \text{Bit } i \text{ de la salida de hacer } g/f \text{ con entrada } x.$$

Teorema de Immerman-Szelepcsényi.

Teorema 21. PATH $\in$ **NL-completo** *y por lo tanto* $\overline{\text{PATH}} \in$ **coNL-completo**.

Demo:

PATH $\in$ NL-HARD: Si $f(x) = \langle G_{N,x}, C_0, C_f\rangle$, con C_0 = Configuración inicial de N con entrada x, C_f = Configuración final.

Cada configuración se codifica con $c.\log|x|$ bits.

Luego,

$x \in \mathcal{L}$ sii N acepta x

sii Existe un camino desde C_0 a C_f en $G_{N,x}$

sii $f(x) \in$ PATH

O sea, básicamente es una especie de Reach, pero implementada con PATH, donde no me guardo el camino q' hago.

Todavía no está todo, hay que ver que f es computable implícitamente en L:

Numeramos todas las configuraciones en orden lexicográfico.

cant. de configuraciones

$G_{N,x}$ es representable con una matriz de adyacencia de $(2^{c \cdot \log|x|})^2$

Luego, $|f(x)| = O((2^{c \cdot \log|x|})^2) = O(|x|^{2c})$

Uso el truquito de obtener la matriz de adyacencia on-the-fly.

O sea, tengo M det. tal q' dada $\langle x, i \rangle$ decide si i corresponde a un bit de la matriz de adyacencia de $G_{N,x}$, o a un bit de la codificación de C_0 o a un bit de C_f.

$\overline{PATH} \in$ coNL-completo:

Si sé que para cualquier $\mathcal{L} \in NL$, $\mathcal{L} \leq_L PATH$, puedo decir q' puedo ver que puedo, para cualquier $\overline{\mathcal{L}} \in coNL$ buscar $\mathcal{L} \in NL$ y reducirlo a PATH, donde su complemento equivaldría a $\overline{\mathcal{L}}$.

$$x \in \overline{\mathcal{L}} \text{ sii } \underbrace{x \notin \mathcal{L} \text{ sii } f(x) \notin PATH \text{ sii}}_{\text{Esto es computable implícitamente en } L\text{, por lo cual:}} f(x) \in \overline{PATH}$$

$$\overline{\mathcal{L}} \leq_L \overline{PATH}$$

^^