Circuit complexity
Non-uniform model of computation

Boolean circuits

A Boolean circuit is a DAG, where

- Each internal node v is labeled with a Boolean operation $o(v)$ (of small, fixed arity; typically 0, 1 or 2);
- Each internal node v has $\mathrm{arity}(o(v))$ incoming edges;
  - The edges are ordered.
- Each output node has a single incoming edge;
- Input nodes are ordered;
- Output nodes are ordered.

A Boolean circuit C with n inputs and m outputs defines a function $[\![C]\!] : \{0,1\}^n \to \{0,1\}^m$. We mostly consider $m = 1$.

The complexity classes $\mathsf{SIZE}(T)$

- A family of circuits is $\{C_n\}_{n\in\mathbb{N}}$, where C_n is a circuit with n inputs and 1 output.
- The family is of size $T : \mathbb{N} \to \mathbb{N}$ if $\forall n : |C_n| \leq c \cdot T(n)$.
- A language $L \subseteq \{0,1\}^*$ is in $\mathsf{SIZE}(T)$ if exists c and a family of circuits $\{C_n\}_{n\in\mathbb{N}}$ of size $c \cdot T$, such that $L \cap \{0,1\}^n$ is accepted by C_n.
- $\mathsf{P}/poly = \bigcup_{c\in\mathbb{N}} \mathsf{SIZE}(\lambda n.n^c)$.

We call the circuit model of computation non-uniform because computations on input size n can be absolutely different from computations on some other input size n'.

Effects of non-uniformity

- **Exercise.** $L \in \mathsf{SIZE}(\lambda n.n \cdot 2^n)$ for any $L \subseteq \{0,1\}^*$.
- Let $L = \{x \in \{0,1\}^* \mid \mathsf{TM}\ M_{|x|} \text{ stops on input } 1^{|x|}\}$.
  - **Exercise.** L is non-recursive.
  - **Exercise.** $L \in \mathsf{SIZE}(\mathit{const})$

Are families of Boolean circuits a good model for efficient computation?

Uniformly generated circuits

- A family of circuits $\{C_n\}_{n \in \mathbb{N}}$ is uniform if there exists a TM M, such that $M(1^n)$ outputs C_n.
- The family is poly-time uniform if the TM M works in polynomial time.
- The family is log-space uniform if the TM M works in logarithmic space (in n).
  - i.e. the size of C_n is still polynomial in n.

Uniformly generated circuits and class P

Theorem. A language L is accepted by a poly-time / log-space uniform family of circuits iff $L \in$ P.

Corollary. P $\subseteq$ P$/poly$.

Turing machines that take advice

Let $f, g : \mathbb{N} \to \mathbb{N}$. A language L belongs to the class DTIME$(f)/g$, if there exist

- a family of bit-strings $\{\alpha_n\}_{n\in\mathbb{N}}$, where $\alpha_n \in \{0,1\}^{g(n)}$;
- a deterministic Turing machine M working in time $O(f)$ (considering only the first argument)

such that for all $n \in \mathbb{N}$ and $x \in \{0,1\}^n$:

$$x \in L \Leftrightarrow M(x, \alpha_n) \text{ accepts } .$$

Example. If L is unary, then $L \in$ DTIME(*const*)/*const*.

Advice vs. circuits

Theorem. $\mathsf{P}/poly = \bigcup_{c,d\in\mathbb{N}} \mathsf{DTIME}(n^c)/n^d$.

Karp-Lipton theorem

Theorem. If SAT $\in$ P$/poly$ then PH $= \Sigma_2^p$.

Proof. We show that SAT $\in$ P$/poly$ implies Π_2SAT $\in \Sigma_2^p$. Let $\{C_n\}_{n\in\mathbb{N}}$ be the poly-size circuit family recognizing SAT.
Recall that

$$\Pi_2\text{SAT} = \{\langle \vec{u}, \vec{v}, \varphi(\vec{u}, \vec{v})\rangle \,|\, \forall \vec{u} \exists \vec{v} : \varphi(\vec{u}, \vec{v}) = \text{true}\} \ .$$

Also recall that SAT is self-reducible and the existence of $\{C_n\}_{n\in\mathbb{N}}$ implies the existence of poly-size circuit family $\{C'_n\}_{n\in\mathbb{N}}$, such that

$$\varphi \in \{0,1\}^n \text{ is satisfiable} \Rightarrow \varphi(C'_n(\varphi)) = \text{true} \ .$$

Proof

- If we're not worried about size and order, then $\forall u \exists v : \varphi(u, v)$ is equivalent to $\exists f \forall u : \varphi(u, f(u))$.
- Existence of poly-size $\{C'_n\}_{n \in \mathbb{N}}$ allows us to express f in polynomial number of bits.

$\langle \vec{u}, \vec{v}, \varphi(\vec{u}, \vec{v}) \rangle \in \Pi_2\mathsf{SAT} \Leftrightarrow \exists C' \in \{0,1\}^{poly(|\varphi|)} \forall \vec{u} : \varphi(\vec{u}, C'(\varphi(\vec{u}, \vec{\cdot}))) =$ true

The problem in the right is in Σ_2^p.

Size hierarchy

Exercise. Most functions require large circuits.
More precisely: for a large enough n, all but exponentially small fraction of functions $\{0,1\}^n \to \{0,1\}$ requires a circuit of size at least $2^n/(10n)$.

Theorem. If $n < T(n) < T'(n) < 2^n/(100n)$ and $T \log T \in o(T')$, then SIZE$(T) \subsetneq$ SIZE(T').

Proof

- For every ℓ, there is a function $f_\ell : \{0,1\}^\ell \to \{0,1\}$ not computable by a circuit of size $2^\ell/(10\ell)$.

- f_ℓ is computable by a circuit of size $2^\ell \cdot (10\ell)$.

- Let $s : \mathbb{N} \to \mathbb{N}$ be such that
  - $2^{s(n)} \cdot (10s(n)) \leq T'(n)$
  - $2^{s(n)}/(10s(n)) \geq T(n)$

- Let $g : \{0,1\}^* \to \{0,1\}$ be the following

$$g(x) = f_{s(|x|)}(lsb_{s(|x|)}(x)) \quad .$$

then $g \in \mathsf{SIZE}(T') \backslash \mathsf{SIZE}(T)$.

The class NC

Definition. $L \in \mathsf{NC}_j$ (Nick's class, after Nicholas John Pippenger) if it can be decided by a log-space uniform family of circuits with depth $O(\lambda n. \log^j n)$.

$\mathsf{NC} = \bigcup_{j \in \mathbb{N}} \mathsf{NC}_j$.

The class NC is the model for efficient parallel computation.

- A problem is efficiently parallelizable if it can be computed with a polynomial number of processors in polylogarithmic time.
    - This statement is somewhat less agreeable than "P models efficient computation".

Graph reachability is in NC_2

(Boolean) matrix multiplication is in NC_1. Hence transitive closure is in NC_2.

Reducibility in NC

Theorem. If $L \leq_{\mathrm{m}}^{\mathrm{L}} L'$ and $L' \in$ NC then $L \in$ NC.

(If $L \leq_{\mathrm{m}}^{\mathrm{L}} L'$ and $L' \in \mathsf{NC}_i$ then $L \in \mathsf{NC}_{\max(i,2)}$)

P-completeness

- A problem in P is P-complete if every other problem in P is log-space reducible to it.
- The P-complete problems are the hardest to parallelize.

CVP and variations

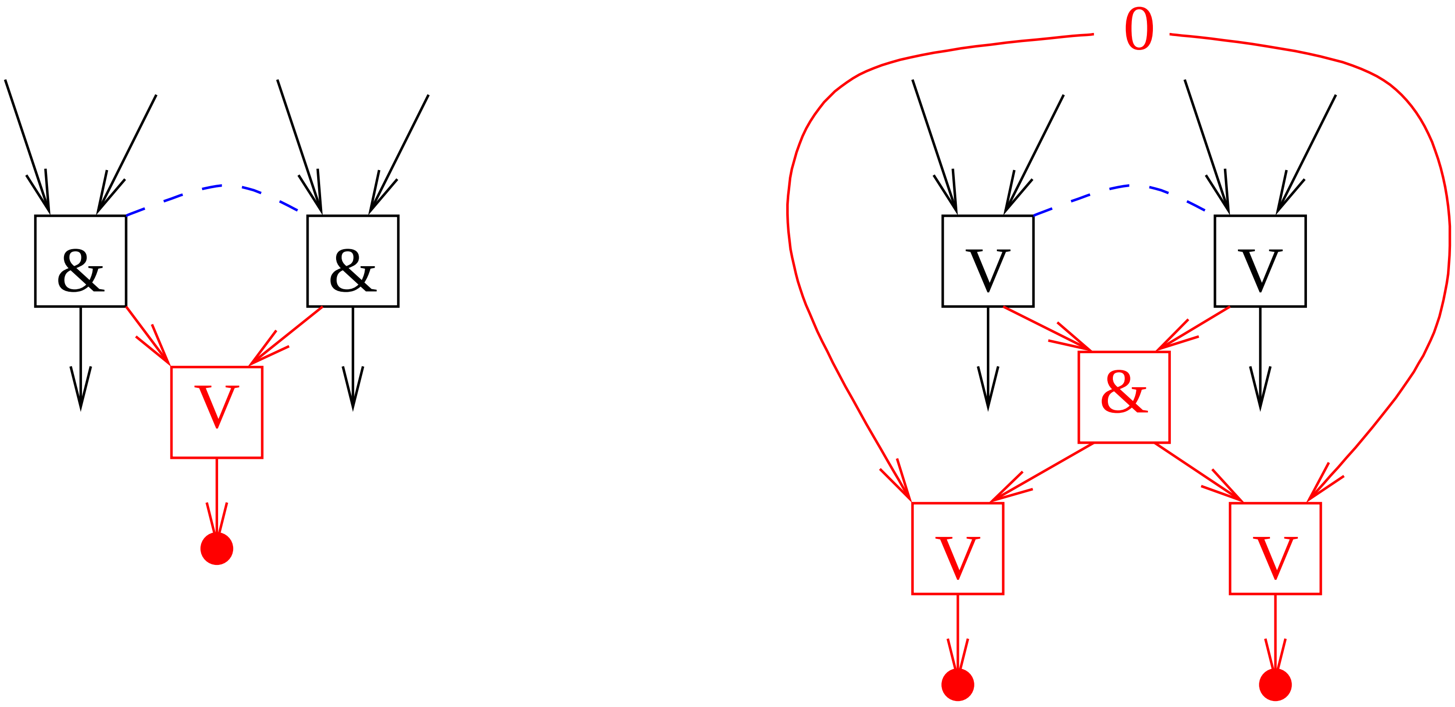
- Given M, x and 1^n. Does M accept x in at most n steps?
- Given a Boolean circuit $\{0,1\}^n \to \{0,1\}^m$, an n-bit string and $i \in \{1,\ldots,m\}$. What is the i-th output of the circuit on that string? (Circuit Value Problem — CVP)
- Same as previous, but circuit may contain only AND- and OR-nodes. (Monotone CVP — MCVP).
- Same as previous, but the fan-in and fan-out of each internal node is 2, inputs go to and outputs come from OR-gates, and AND-s and OR-s **a**lternate on each path from input gate to output gate (AM2CVP).
    - Fan-out of input nodes is also 2.

MCVP $\leq^{\mathrm{L}}_{\mathrm{m}}$ AM2CVP

- Input node → OR-node with fan-in 2; two new input nodes with fan-out 1.
- AND-node giving output → AND to OR to output.
- Fan-out > 2 → fan-out tree of OR-nodes with fan-in 1 and fan-out 2.
- AND → AND goes to AND → OR → AND. Same for OR-s.
- OR-node with single input gets and extra input node as input.
- AND-node with single input gets other input from OR-node, that gets input from two input nodes.
- Replicate the circuit as follows: . . .

Replication

- Double all nodes, except inputs. In this way, inputs get fan-out 2.
- Introduce fan-out to AND- and OR-nodes as follows.



NANDCVP / NORCVP

- NANDCVP — like CVP, but all gates are NAND-s, and the fan-out of inputs and internal gates must be 2.
- NORCVP — same, but all gates are NOR-s.

Theorem. AM2CVP $\leq_{\mathrm{m}}^{\mathrm{L}}$ NANDCVP.

Proof. Complement all inputs and turn all gates to NAND-s.

Exercise. How about NORCVP?

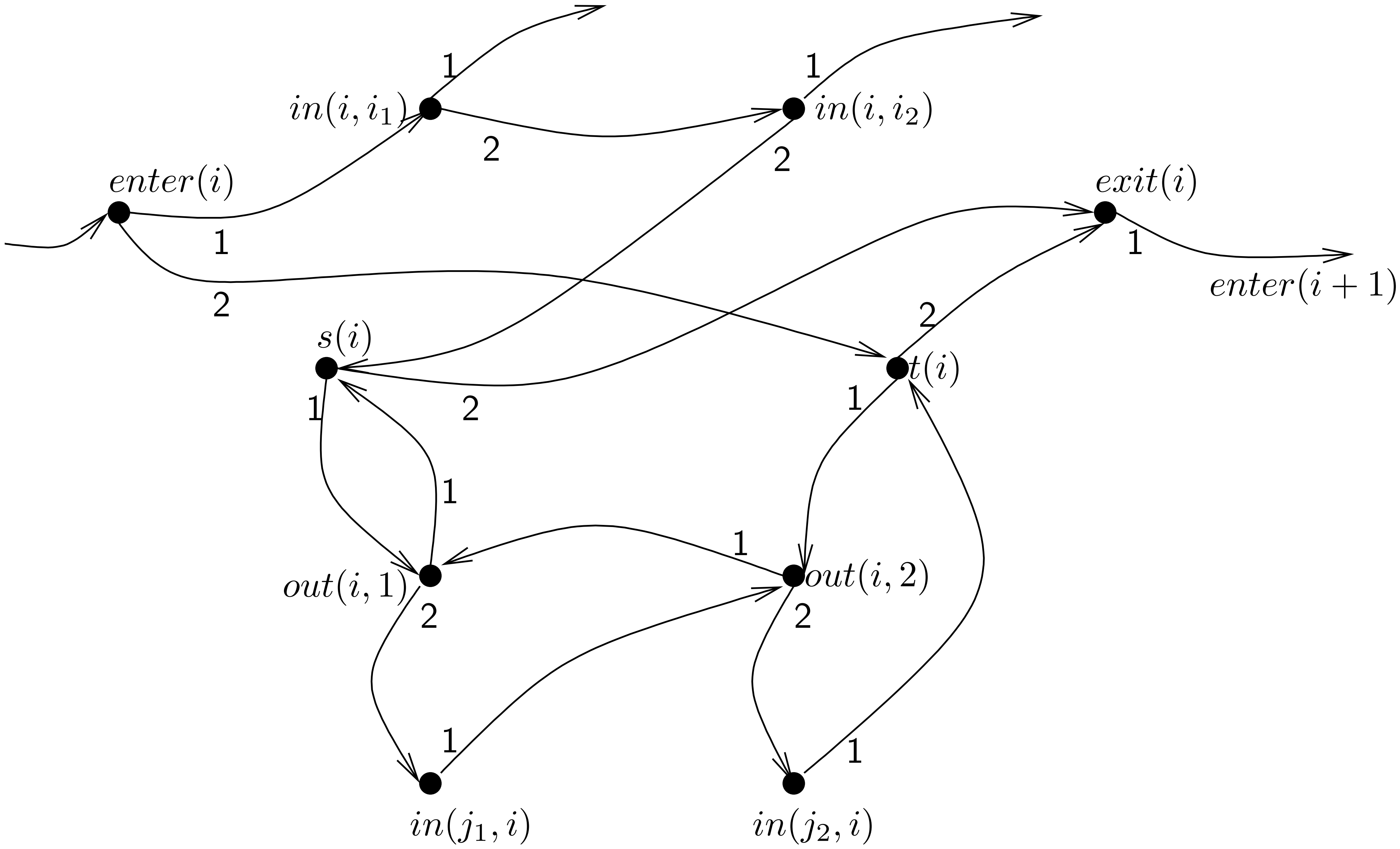
DFS

- Given a directed graph G, where
  - the outputs of each node are ordered;
  - the nodes are also ordered,

  and two vertices u and v. Is u visited before v in the depth-first traversal of G?

Theorem. NORCVP $\leq^{\mathrm{L}}_{\mathrm{m}}$ DFS.

Proof. Let C be a circuit consisting only of NOR-gates with fan-in and fan-out 2. Assume its nodes are topologically sorted (all our reductions provide or preserve this).
Let i be a node with inputs i_1, i_2 and outputs j_1, j_2. A fragment of the graph is:



1
1
$in(i, i_1)$
$in(i, i_2)$
2
2
$enter(i)$
$exit(i)$
1
1
$enter(i+1)$
2
2
$s(i)$
$t(i)$
1
2
1
1
1
1
$out(i, 1)$
$out(i, 2)$
2
2
1
1
$in(j_1, i)$
$in(j_2, i)$

Lemmas

Lemma. At the moment the traversal is about to enter $enter(i)$:

- $s(i)$, $t(i)$, $out(i,1)$, $out(i,2)$, $exit(i)$, $in(j_1,i)$, $in(j_2,i)$ are untraversed.
- $in(i,i_k)$ is traversed iff the node i_k evaluates true in the circuit.
- Nodes pointed to by the first outedges of $in(i,i_k)$ are traversed.

Lemma. If node i evaluates to true then $s(i)$ is traversed before $t(i)$. If node i evaluates to false then $t(i)$ is traversed before $s(i)$.

Initialization

- There are nodes $in(k, i)$ for each input node i and each node k that receives input from i.
- All true inputs are chained together and traversed first.
- I.e. when we reach $enter(k)$, the node $in(k, i)$ has been traversed iff input i is true.

Finally, we ask whether $s(output)$ is traversed before or after $t(output)$.

Max-Flow parity (MAXFLOW$\oplus$)

Given a directed graph G with source node s and sink node t. Also given integer capacity $c(e) \geq 0$ for each edge e. What is the parity of maximum flow from s to t?

Theorem. AM2CVP $\leq^{\mathrm{L}}_{\mathrm{m}}$ MAXFLOW$\oplus$.

Alternating reachability (AGAP)

- Given a directed graph G where each node is labeled with $\forall$ or $\exists$, and two nodes u, v. Does $apath(u, v)$ hold, where
    - $apath(x, x)$ holds for all nodes x.
    - If x is labeled with $\exists$, then $apath(x, y)$ if exists z, such that $x \to z$ and $apath(z, y)$.
    - If x is labeled with $\forall$, then $apath(x, y)$ if for all z, where $x \to z$, we have $apath(z, y)$.

Theorem. MCVP $\leq_{\mathrm{m}}^{\mathrm{L}}$ AGAP.

Proof

- Given: circuit C, inputs x, output z.
- Introduce two constant nodes 0 and 1; use those instead of the inputs.
- Let AND-nodes be labeled with $\forall$ and OR-nodes with $\exists$.
- Reverse all edges.
- $apath(z, 1)$?

HORNSAT

- Recall: variable, literal (positive or negative).
- A Horn clause is $l_1 \vee l_2 \vee \cdots \vee l_n$, where at most one literal is positive.
- A Horn formula is a conjunction of Horn clauses.

HORNSAT — the set of all satisfiable Horn formulas.

Theorem. AGAP $\leq_{\mathrm{m}}^{\mathrm{L}}$ HORNSAT.

Unit resolution (UNIT)

- If $C = l_1 \vee \cdots \vee l_m$ is a disjunct and $l = \neg l_j$, then the unit resolution of C and l gives

$$l_1 \vee \cdots \vee l_{j-1} \vee l_{j+1} \vee \cdots \vee l_m \quad .$$

Given a set of disjuncts (of size ≥ 3). Can the empty disjunct be derived from those using unit resolution?

Theorem. CVP $\leq^{\mathrm{L}}_{\mathrm{m}}$ UNIT.

Proof. The gates $v_i \equiv v_j \wedge v_k$, $v_i \equiv v_j \vee v_k$ and $v_i \equiv v_j$ can all be represented as conjunctions of disjuncts of size at most 3. True inputs are represented as unit clauses.

Generability (GEN)

Given a set X, subset $S \subseteq X$, element $x \in X$ and a binary operation $\bullet : X \times X \to X$ (given as table). Can x be generated from S using $\bullet$?

Theorem. UNIT $\leq_{\mathrm{m}}^{\mathrm{L}}$ GEN.

Proof. A set of disjuncts D is given.

- Let X be the set of all subdisjuncts of D, plus an extra element $\perp$.
- Let $S = D$.
- Let $\bullet$ be unit resolution (result is $\perp$, if nonapplicable).
- Let x be the empty disjunct.

Context-free parsing (CFPARSE)

Given a context-free grammar $G = (N, T, P, s)$ and a word w. Does $w \in L(G)$?

Theorem. GEN $\leq_{\mathrm{m}}^{\mathrm{L}}$ CFPARSE.

Proof. $(X, S, \bullet, x)$ is given. Let $N = X$, $T = \{a\}$, $s = x$. Let P contain the following productions

- $r \to st$, where $s \bullet t = r$;
- $r \to \varepsilon$ for $r \in S$.

Let w be the empty word.