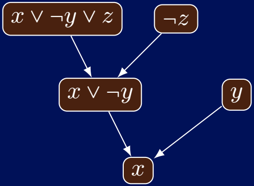
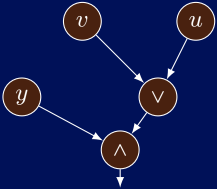
A Few Words About the Proof Complexity





Dmitry Sokolov

SACC 2021
May 27

St Petersburg University

PDMI
RAS

Proof Systems

$L \subseteq \{0,1\}^*$. UNSAT is a language of unsatisfiable boolean CNF formulas.

Proof Systems

$L \subseteq \{0,1\}^*$. UNSAT is a language of unsatisfiable boolean CNF formulas.

Definition[Cook, Reckhow 79]

Proof system for $L \Leftrightarrow$ poly-time algorithm Π: $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}$:

- (completeness) $x \in L \Rightarrow \exists w\ \Pi(x,w) = 1$;
- (soundness) $\exists w\ \Pi(x,w) = 1 \Rightarrow x \in L$.

Length of $|w|$ is the complexity measure.

Proof Systems

$L \subseteq \{0,1\}^*$. UNSAT is a language of unsatisfiable boolean CNF formulas.

Definition[Cook, Reckhow 79]

Proof system for $L \Leftrightarrow$ poly-time algorithm Π: $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}$:

- (completeness) $x \in L \Rightarrow \exists w\ \Pi(x, w) = 1$;
- (soundness) $\exists w\ \Pi(x, w) = 1 \Rightarrow x \in L$.

Length of $|w|$ is the complexity measure.

Cook's Program

Prove superpolynomial lower bounds for stronger and stronger proof systems until the techniques are developed to do it in a general case.
Goal: $\mathbf{NP} \neq \mathbf{coNP}$.

Proof Systems

Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \ldots, D_\ell)$:

Proof Systems

Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \ldots, D_\ell)$:

- $D_i \in \{C_i\}$;

Proof Systems

Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \ldots, D_\ell)$:

- $D_i \in \{C_i\}$;
- $\frac{A \vee x \quad B \vee \bar{x}}{A \vee B}$, $D_i \coloneqq A \vee B$;

Proof Systems

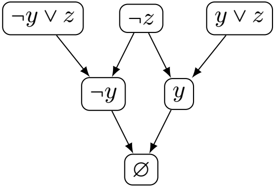
Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \ldots, D_\ell)$:

- $D_i \in \{C_i\}$;
- $\frac{A \vee x \quad B \vee \bar{x}}{A \vee B}$, $D_i \coloneqq A \vee B$;
- $D_\ell = \varnothing$.

Proof Systems

Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \ldots, D_\ell)$:

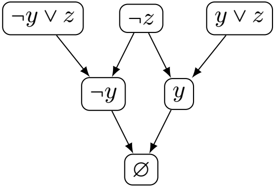
- $D_i \in \{C_i\}$;
- $\frac{A \vee x \quad B \vee \bar{x}}{A \vee B}$, $D_i \coloneqq A \vee B$;
- $D_\ell = \varnothing$.



Proof Systems

Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \ldots, D_\ell)$:

- $D_i \in \{C_i\}$;
- $\frac{A \vee x \quad B \vee \bar{x}}{A \vee B}$, $D_i \coloneqq A \vee B$;
- $D_\ell = \varnothing$.



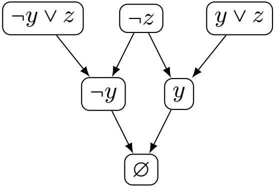
Cutting Planes: proof is a sequence of inequalities over $\mathbb{Z}$ $(p_1 \geq 0, p_2 \geq 0, p_3 \geq 0, \ldots, p_\ell \geq 0)$:

- p_i is an encoding of $C \in \varphi$, $x_k \geq 0$ or $-x_k + 1 \geq 0$;
- $\frac{p_i \quad p_j}{p_k}$, $(p_i \geq 0) \wedge (p_j \geq 0)$ imply $(p_k \geq 0)$ over $\mathbb{Z}^n$;
- $p_\ell = 1$.

Proof Systems

Resolution: proof of $\varphi \coloneqq \bigwedge_i C_i$ is a sequence of clauses $(D_1, D_2, D_3, \dots, D_\ell)$:

- $D_i \in \{C_i\}$;
- $\frac{A \vee x \quad B \vee \bar{x}}{A \vee B}$, $D_i \coloneqq A \vee B$;
- $D_\ell = \varnothing$.



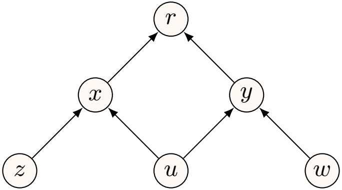
Cutting Planes: proof is a sequence of inequalities over $\mathbb{Z}$ $(p_1 \geq 0, p_2 \geq 0, p_3 \geq 0, \dots, p_\ell \geq 0)$:

- p_i is an encoding of $C \in \varphi$, $x_k \geq 0$ or $-x_k + 1 \geq 0$;
- $\frac{p_i \quad p_j}{p_k}$, $(p_i \geq 0) \wedge (p_j \geq 0)$ imply $(p_k \geq 0)$ over $\mathbb{Z}^n$;
- $p_\ell = 1$.

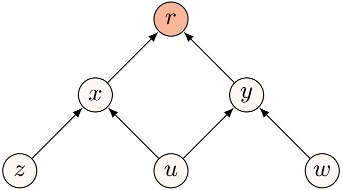
Nullstellensatz: proof of a system of polynomial equalities $f_1 = 0, f_2 = 0, \dots$:

$$\sum_{u=1}^{a} p_u f_u = 1.$$

Pebbing

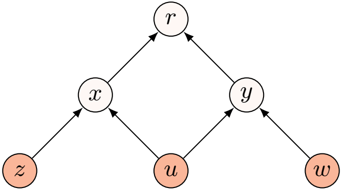


Pebbing



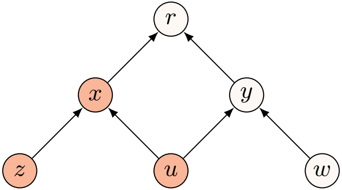
- $(\neg r)$;

Pebbing



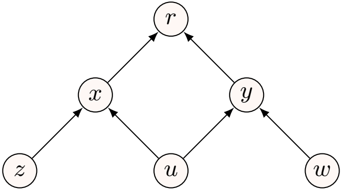
- $(\neg r)$;
- $(z), (u), (w)$;

Pebbing



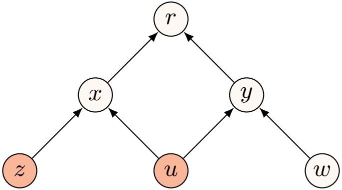
- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \vee \neg u \vee x)$.

Pebbing



- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \lor \neg u \lor x)$.

Pebbing



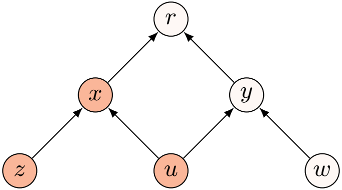
- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \vee \neg u \vee x)$.

u

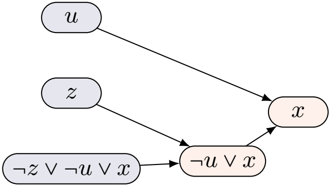
z

$\neg z \vee \neg u \vee x$

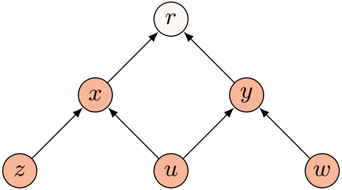
Pebbing



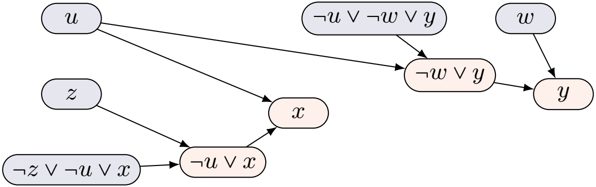
- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \vee \neg u \vee x)$.



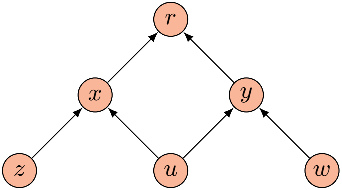
Pebbing



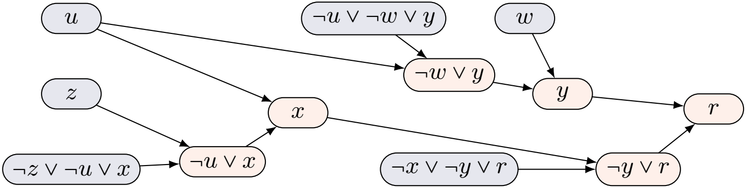
- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \vee \neg u \vee x)$.



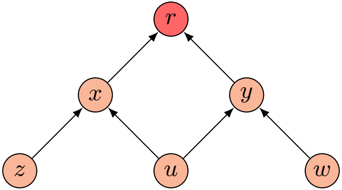
Pebbing



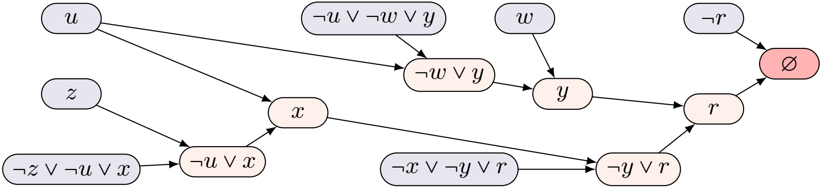
- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \vee \neg u \vee x)$.



Pebbing

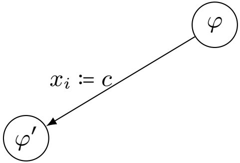


- $(\neg r)$;
- $(z), (u), (w)$;
- $(\neg z \vee \neg u \vee x)$.

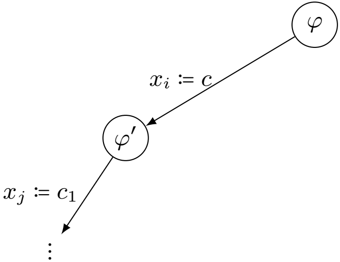


DPLL **Algorithms**

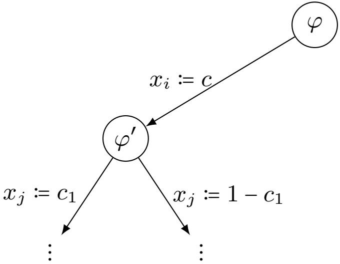
DPLL **Algorithms**



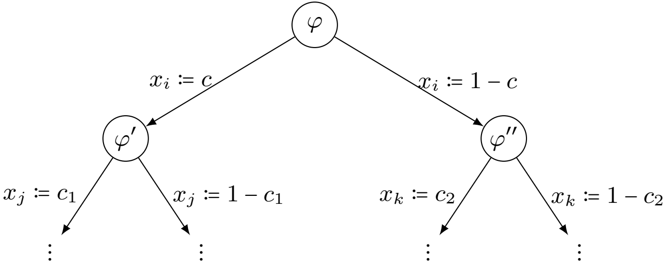
DPLL **Algorithms**



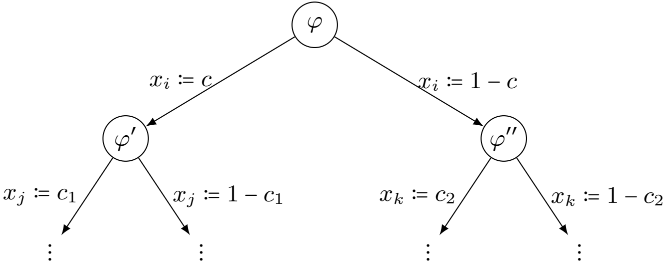
DPLL **Algorithms**



DPLL **Algorithms**

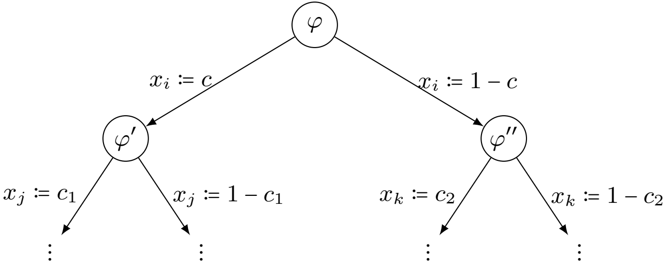


DPLL **Algorithms**



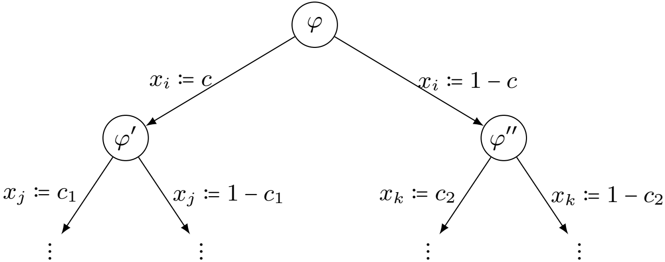
- Heuristic **A** chooses a variable for splitting.

DPLL **Algorithms**



- Heuristic **A** chooses a variable for splitting.
- Heuristic **B** chooses the first value.

DPLL **Algorithms**



- Heuristic **A** chooses a variable for splitting.
- Heuristic **B** chooses the first value.
- Simplification rules: no simplifications!

DPLL and Resolution

Theorem

DPLL algoritm makes t splitting on unsatisfiable CNF formula

$$\varphi \coloneqq \bigwedge_i C_i$$

$\Rightarrow$ there exists a resolution proof of φ of size $2t$.

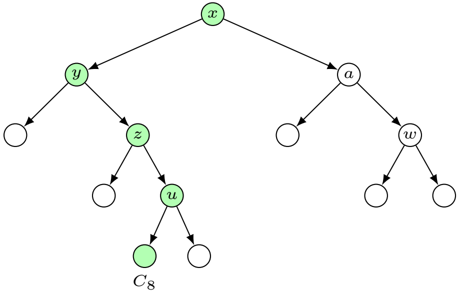
DPLL **and Resolution**

Theorem

DPLL algoritm makes t splitting on unsatisfiable CNF formula

$$\varphi \coloneqq \bigwedge_i C_i$$

$\Rightarrow$ there exists a resolution proof of φ of size $2t$.



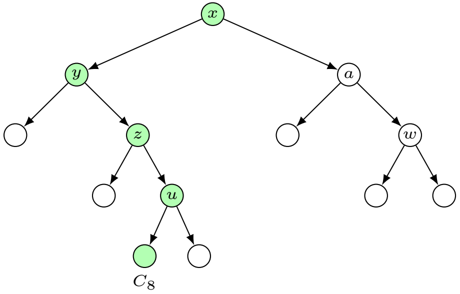
DPLL and Resolution

Theorem

DPLL algoritm makes t splitting on unsatisfiable CNF formula

$$\varphi \coloneqq \bigwedge_i C_i$$

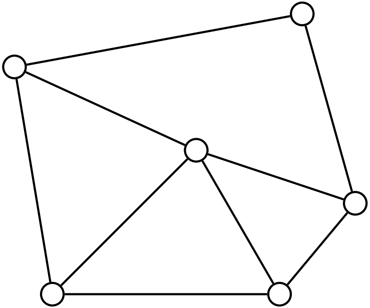
$\Rightarrow$ there exists a resolution proof of φ of size $2t$.

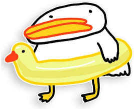


$$\frac{A \vee x \quad B \vee \neg x}{A \vee B} \qquad \frac{A}{A \vee z}$$

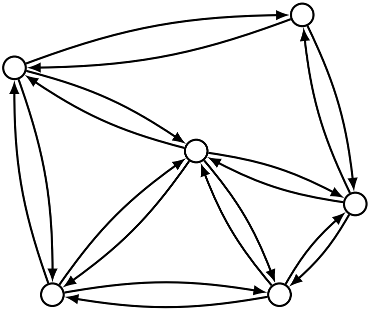
- Node $\Rightarrow$ disjunction of negations of queries.
- $(x \vee \neg y \vee \neg z \vee u)$.

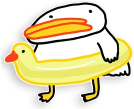
Flow formulas



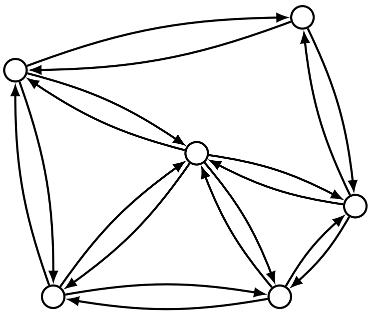


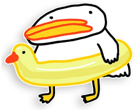
Flow formulas





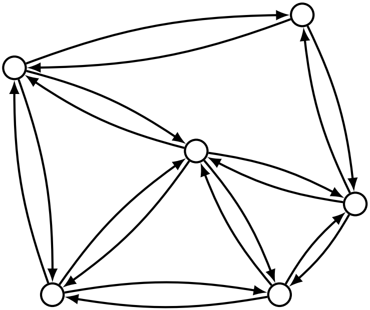
Flow formulas

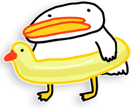




- v: $\sum_{e \in E_v^{\text{in}}} x_e - \sum_{e \in E_v^{\text{out}}} x_e = c(v)$ $(\mathbb{R})$;
- $\sum_v c(v) = 1$ $(\mathbb{R})$;
- graph degree: d.

Flow formulas

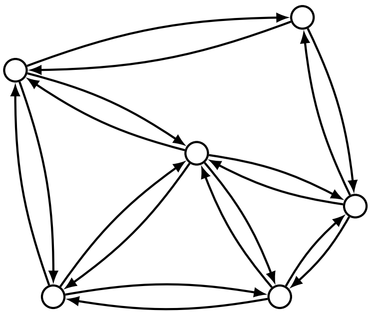


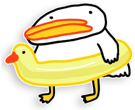


- v: $\sum_{e \in E_v^{\text{in}}} x_e - \sum_{e \in E_v^{\text{out}}} x_e = c(v)$ $(\mathbb{R})$;
- $\sum_v c(v) = 1$ $(\mathbb{R})$;
- graph degree: d.

- There is an efficient Nullstellensatz proof of Flow.
- [Alekhnovich, Razborov 03] If G is an (n, d, α)-expander $\Rightarrow$ any resolution proof has size $2^{\Omega(n)}$.

Flow formulas





- v: $\sum_{e \in E_v^{\text{in}}} x_e - \sum_{e \in E_v^{\text{out}}} x_e = c(v)$ ($\mathbb{R}$);
- $\sum_v c(v) = 1$ ($\mathbb{R}$);
- graph degree: d.

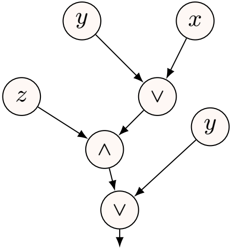
- There is an efficient Nullstellensatz proof of Flow.
- [Alekhnovich, Razborov 03] If G is an (n, d, α)-expander $\Rightarrow$ any resolution proof has size $2^{\Omega(n)}$.

Corollary[Göös, Kamath, Robere, S 19]

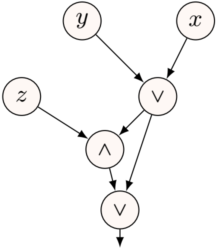
There is a monotone function in $\mathbf{NC}_2$ that cannot be computed by subexponential monotone circuits.

Monotone Computations

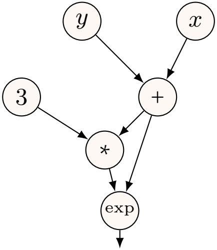
Formulas



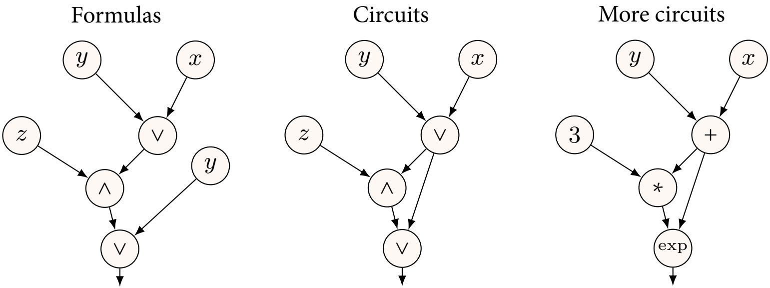
Circuits



More circuits



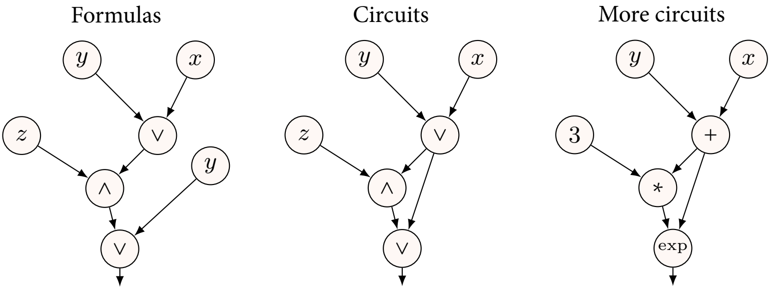
Monotone Computations



Why do we care about lower bounds on monotone computations?

- We can proof something!

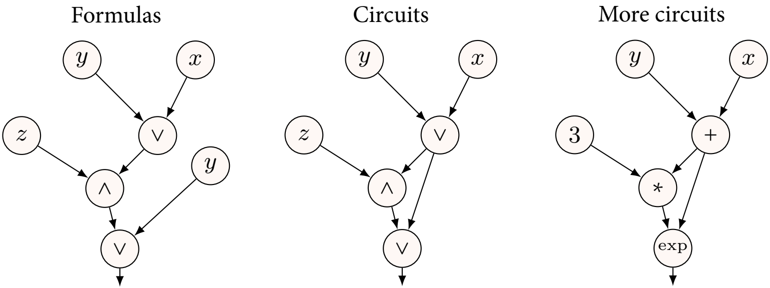
Monotone Computations



Why do we care about lower bounds on monotone computations?

- We can proof something!
- We can control relative error.

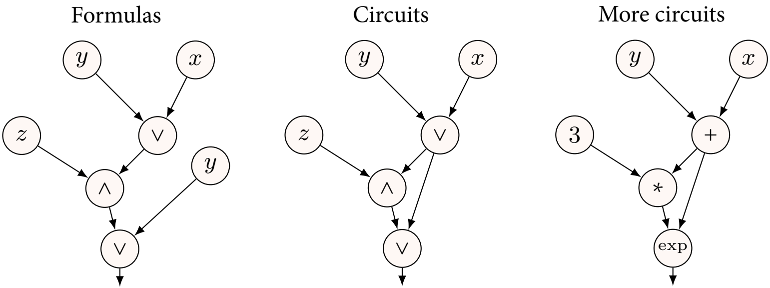
Monotone Computations



Why do we care about lower bounds on monotone computations?

- We can proof something!
- We can control relative error.
- Strong enough lower bounds on monotone circuits ⇒ lower bounds on general circuits.

Monotone Computations



Why do we care about lower bounds on monotone computations?

- We can proof something!
- We can control relative error.
- Strong enough lower bounds on monotone circuits $\Rightarrow$ lower bounds on general circuits.
- Secret sharing/cryptography.

Communication Protocols. $f\colon U \times V \to T$

$$f(x, y) = ?$$

$x \in U$

$y \in V$

Communication Protocols. $f : U \times V \to T$

$$f(x, y) = ?$$

$x \in U$

$r_1 = a(x)$ →

$y \in V$

Communication Protocols. $f\colon U \times V \to T$

$$f(x, y) = ?$$

$x \in U$ $y \in V$

$r_1 = a(x)$ →

$r_2 = b(y, r_1)$ ←

Communication Protocols. $f\colon U \times V \to T$

$$f(x, y) = ?$$

$x \in U$ $\qquad\qquad$ $y \in V$

$$r_1 = a(x) \longrightarrow$$

$$\longleftarrow r_2 = b(y, r_1)$$

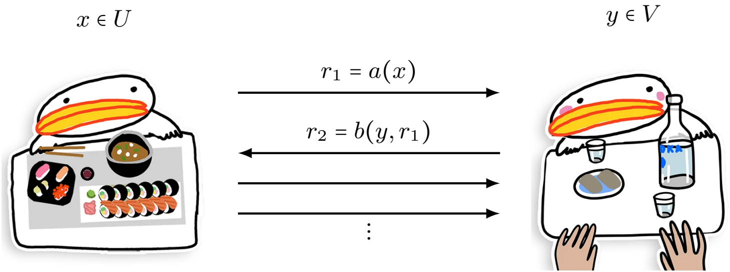
$$\longrightarrow$$

$$\longrightarrow$$

$$\vdots$$

Communication Protocols. $f\colon U \times V \to T$

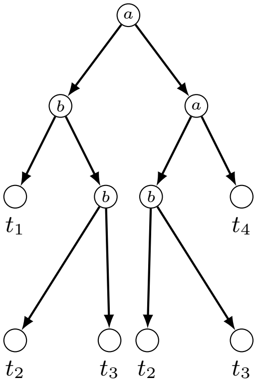
$$f(x, y) = ?$$



- Depth is the number of rounds (in the worst case).
- $\mathrm{D}(f) = \min_{P \in \mathcal{P}} \mathrm{depth}(P)$, where $\mathcal{P}$ is a set of protocols for f.

Protocols and Trees

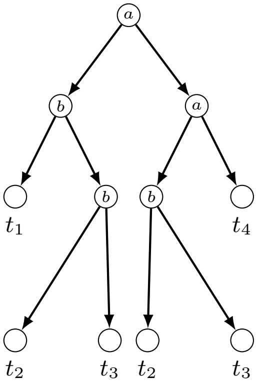
Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:



Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

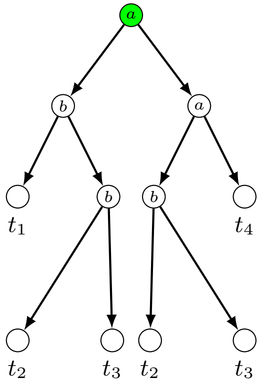
- nodes are marked by players;



Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

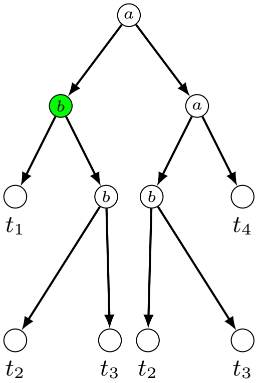
- nodes are marked by players;



Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

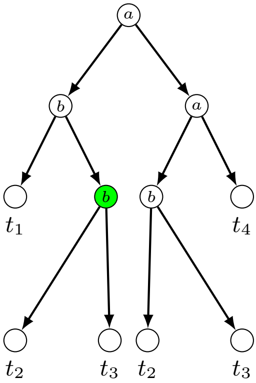
- nodes are marked by players;



Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

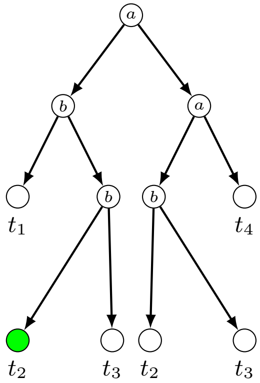
- nodes are marked by players;



Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

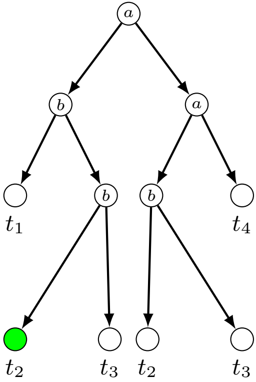
- nodes are marked by players;



Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

- nodes are marked by players;
- leaves by answers.

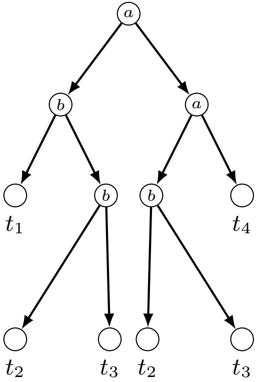


Protocols and Trees

Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

- nodes are marked by players;
- leaves by answers.

Size of protocol is a size of the tree.
$\text{Size}(f) = \min_{P \in \mathcal{P}} \text{Size}(P)$.



Protocols and Trees

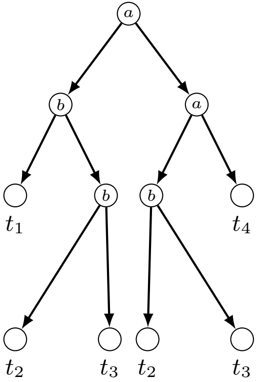
Alice gets $u \in U$, Bob gets $v \in V$. Protocol is a tree:

- nodes are marked by players;
- leaves by answers.

Size of protocol is a size of the tree.
$\mathrm{Size}(f) = \min_{P \in \mathcal{P}} \mathrm{Size}(P)$.

Lemma

$\mathrm{D}(f) = \Omega(\log(\mathrm{Size}(f)))$.



KW Relation [Karchmer, Wigderson 90]

Let $U, V \subseteq \{0,1\}^n$ and $U \cap V = \varnothing$.

KW:

- Alice gets $u \in U$, Bob gets $v \in V$;
- goal: find i such that $u_i \neq v_i$.

KW Relation [Karchmer, Wigderson 90]

Let $U, V \subseteq \{0,1\}^n$ and $U \cap V = \emptyset$.

KW:

- Alice gets $u \in U$, Bob gets $v \in V$;
- goal: find i such that $u_i \neq v_i$.

Monotone version KW^{m}:

- goal: find i such that $u_i = 1 \wedge v_i = 0$.

KW Relation [Karchmer, Wigderson 90]

Let $U, V \subseteq \{0,1\}^n$ and $U \cap V = \varnothing$.

KW:

- Alice gets $u \in U$, Bob gets $v \in V$;
- goal: find i such that $u_i \neq v_i$.

Monotone version KW^{m}:

- goal: find i such that $u_i = 1 \wedge v_i = 0$.

Theorem[Karchmer, Wigderson 90]

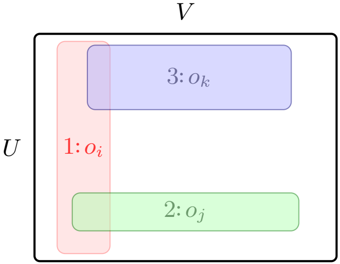
Monotone formula for a function f of size S $\Leftrightarrow$ communication protocol for KW^{m} KW of size S, where $U := f^{-1}(1), V := f^{-1}(0)$.

KW^{m} is a "Complete Relation"

- $\mathcal{S} \subseteq U \times V \times \mathcal{O}$;
- define $F_{\mathcal{S}}\colon \{0,1\}^m \to \{0,1\}$ such that $\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(S)$.

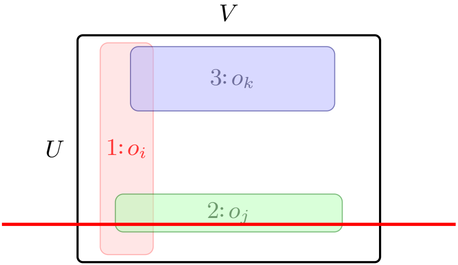
KW^m is a “Complete Relation”

- $\mathcal{S} \subseteq U \times V \times \mathcal{O}$;
- define $F_\mathcal{S}\colon \{0,1\}^m \to \{0,1\}$ such that $\mathrm{D}(\mathsf{KW}^\mathsf{m}_{F_\mathcal{S}}) = \mathrm{D}(\mathcal{S})$.



$\mathsf{KW^m}$ is a "Complete Relation"

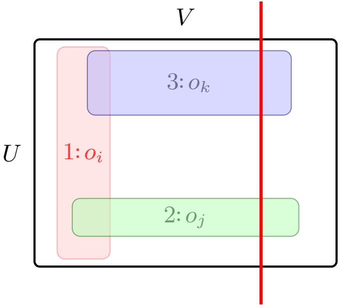
- $\mathcal{S} \subseteq U \times V \times \mathcal{O}$;
- define $F_{\mathcal{S}}\colon \{0,1\}^m \to \{0,1\}$ such that $\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(\mathcal{S})$.



$F_{\mathcal{S}}(1,1,0,\dots) \coloneqq 1$

KW^{m} is a "Complete Relation"

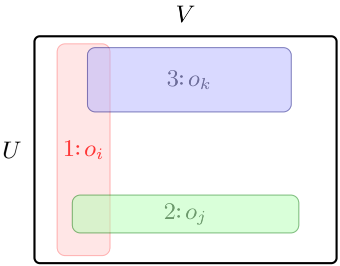
- $\mathcal{S} \subseteq U \times V \times \mathcal{O}$;
- define $F_{\mathcal{S}} \colon \{0,1\}^m \to \{0,1\}$ such that $\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(S)$.



$$F_{\mathcal{S}}(1,1,0,\dots) \coloneqq 1, \quad F_{\mathcal{S}}(1,0,0,\dots) \coloneqq 0$$

KW^{m} is a "Complete Relation"

- $\mathcal{S} \subseteq U \times V \times \mathcal{O}$;
- define $F_{\mathcal{S}}\colon \{0,1\}^m \to \{0,1\}$ such that $\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(S)$.



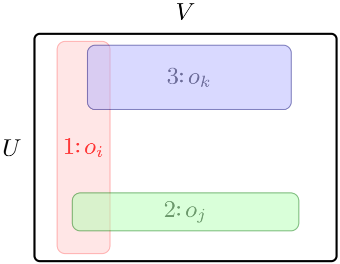
$$F_{\mathcal{S}}(1,1,0,\dots) \coloneqq 1, \quad F_{\mathcal{S}}(1,0,0,\dots) \coloneqq 0$$

Lemma

$\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(S)$.

KW^{m} is a "Complete Relation"

- $\mathcal{S} \subseteq U \times V \times \mathcal{O}$;
- define $F_{\mathcal{S}}\colon \{0,1\}^m \to \{0,1\}$ such that $\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(\mathcal{S})$.



$$F_{\mathcal{S}}(1,1,0,\dots) \coloneqq 1, \quad F_{\mathcal{S}}(1,0,0,\dots) \coloneqq 0$$

Lemma

$\mathrm{D}(\mathsf{KW}^{\mathsf{m}}_{F_{\mathcal{S}}}) = \mathrm{D}(\mathcal{S})$.

Search$_\varphi$ [Lovász, Naor, Newman, Wigderson et al. 94]

$\varphi(z) \coloneqq \bigwedge_{i=1}^{m} C_i$ is unsatisfiable CNF formula.

Search_φ [Lovász, Naor, Newman, Wigderson et al. 94]

$\varphi(z) \coloneqq \bigwedge_{i=1}^{m} C_i$ is unsatisfiable CNF formula.

$\mathsf{Search}_\varphi \subseteq \{0,1\}^n \times [m]$:

- $(\alpha, i) \in \mathsf{Search}_\varphi \Leftrightarrow C_i(\alpha) = 0$.

Search_φ [Lovász, Naor, Newman, Wigderson et al. 94]

$\varphi(z) \coloneqq \bigwedge_{i=1}^{m} C_i$ is unsatisfiable CNF formula.

$\mathsf{Search}_\varphi \subseteq \{0,1\}^n \times [m]$:

- $(\alpha, i) \in \mathsf{Search}_\varphi \Leftrightarrow C_i(\alpha) = 0$.

Communication version:

- "gadget" $g \colon X \times Y \to \{0,1\}$;
- $\mathsf{Ind} \colon [k] \times \{0,1\}^k \to \{0,1\}$, $\mathsf{Ind}(x, y) = y_x$.

Search_φ [Lovász, Naor, Newman, Wigderson et al. 94]

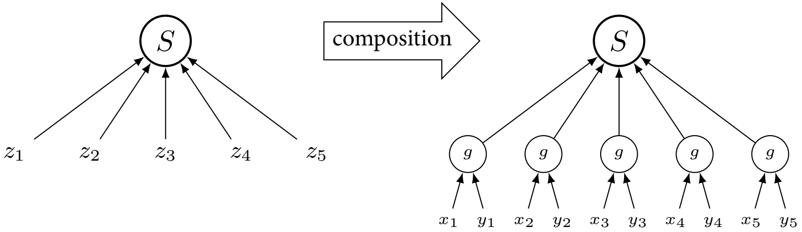
$\varphi(z) \coloneqq \bigwedge_{i=1}^{m} C_i$ is unsatisfiable CNF formula.

$\mathsf{Search}_\varphi \subseteq \{0,1\}^n \times [m]$:

- $(\alpha, i) \in \mathsf{Search}_\varphi \Leftrightarrow C_i(\alpha) = 0$.

Communication version:

- "gadget" $g \colon X \times Y \to \{0,1\}$;
- $\mathsf{Ind} \colon [k] \times \{0,1\}^k \to \{0,1\}$, $\mathsf{Ind}(x,y) = y_x$.



$\mathsf{Search}_\varphi \circ g \equiv \mathsf{Search}_{\varphi \circ g}$.

Lifting 

Theorem[Raz, McKenzie 99; Göös, Pitassi, Watson 16]

Resolution depth of φ is at least $d \Rightarrow \mathrm{D}(\mathsf{Search}_\varphi \circ \mathsf{Ind}_m) \geq n^{\mathcal{O}(d)}$, where $m := \mathsf{poly}(n)$. $\mathrm{D}(\mathsf{Search}_\varphi \circ \mathsf{Ind}_m) \approx \mathrm{D}(\mathsf{Ind}) \cdot \text{res-depth}(\varphi)$.

Corollary: lower bound on monotone formulas $2^{n^{\varepsilon}}$.

Lifting 

Theorem[Raz, McKenzie 99; Göös, Pitassi, Watson 16]

Resolution depth of φ is at least $d \Rightarrow \mathrm{D}(\mathsf{Search}_\varphi \circ \mathsf{Ind}_m) \geq n^{\mathcal{O}(d)}$, where $m \coloneqq \mathsf{poly}(n)$. $\mathrm{D}(\mathsf{Search}_\varphi \circ \mathsf{Ind}_m) \approx \mathrm{D}(\mathsf{Ind}) \cdot \text{res-depth}(\varphi)$.

Corollary: lower bound on monotone formulas 2^{n^ε}.

Theorem[Garg, Göös, Kamath, S 18]

Resolution size φ at least $S \Rightarrow$ size of dag-like protocols for $\mathsf{Search}_\varphi \circ \mathsf{Ind}_m$ at least $\Omega(S)$, where $m \coloneqq \mathsf{poly}(n)$.

Corollary: lower bound on monotone circuits 2^{n^ε}.

Lifting

Theorem[Raz, McKenzie 99; Göös, Pitassi, Watson 16]

Resolution depth of φ is at least $d \Rightarrow \mathrm{D}(\mathsf{Search}_\varphi \circ \mathsf{Ind}_m) \geq n^{\mathcal{O}(d)}$, where $m \coloneqq \mathsf{poly}(n)$. $\mathrm{D}(\mathsf{Search}_\varphi \circ \mathsf{Ind}_m) \approx \mathrm{D}(\mathsf{Ind}) \cdot \text{res-depth}(\varphi)$.

Corollary: lower bound on monotone formulas 2^{n^ε}.

Theorem[Garg, Göös, Kamath, S 18]

Resolution size φ at least $S \Rightarrow$ size of dag-like protocols for $\mathsf{Search}_\varphi \circ \mathsf{Ind}_m$ at least $\Omega(S)$, where $m \coloneqq \mathsf{poly}(n)$.

Corollary: lower bound on monotone circuits 2^{n^ε}.

Theorem[Pitassi, Robere 16; Robere, Pitassi 18, informal]

Nullstellensatz $\Leftrightarrow$ algebraic tiling for $\mathsf{Search}_\varphi \circ g$.

Easy Function?

$f\colon \{0,1\}^{2n^3} \to \{0,1\}$

- Enumerate equalities $z_i \oplus z_j \oplus z_k = c$ (at most $2n^3$);
- $x_i = 1 \Leftrightarrow$ add the equality to the system;
- $f(x) = 1 \Leftrightarrow$ system is unsatisfiable.

Easy Function?

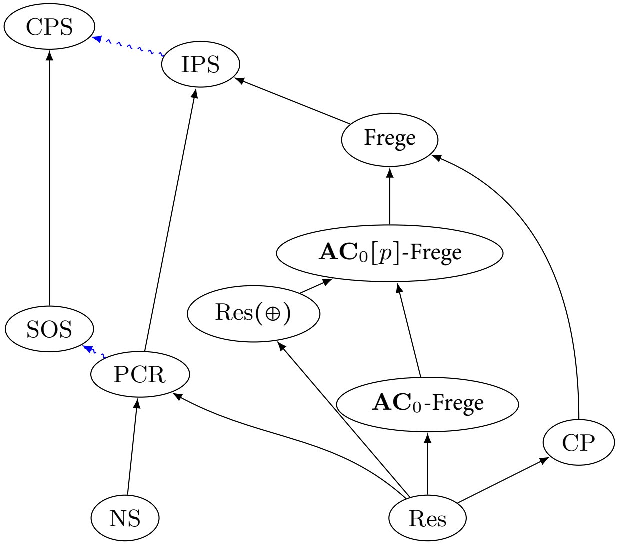
$f\colon \{0,1\}^{2n^3} \to \{0,1\}$

- Enumerate equalities $z_i \oplus z_j \oplus z_k = c$ (at most $2n^3$);
- $x_i = 1 \Leftrightarrow$ add the equality to the system;
- $f(x) = 1 \Leftrightarrow$ system is unsatisfiable.

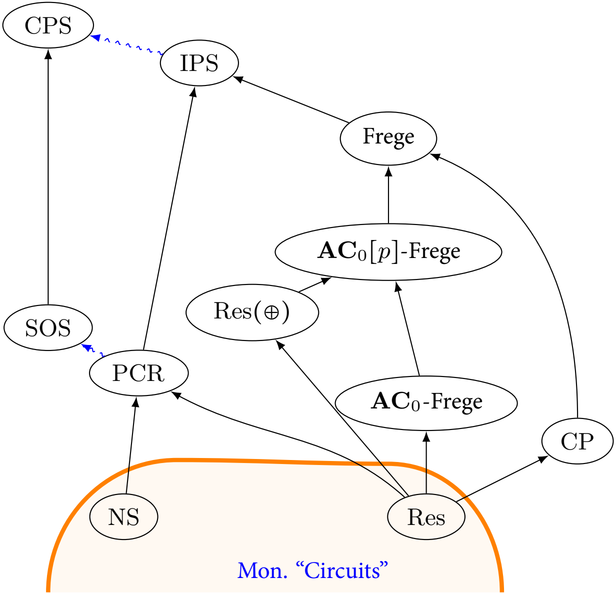
Facts about f:

- $f \in \mathbf{NC}^2$;
- F_{Flow} can be embedded into f (since there is an efficient NS proof of Flow!);
- there is no small monotone circuit for f (since there is no efficient proofs in resolution of Flow + lifting Theorem).

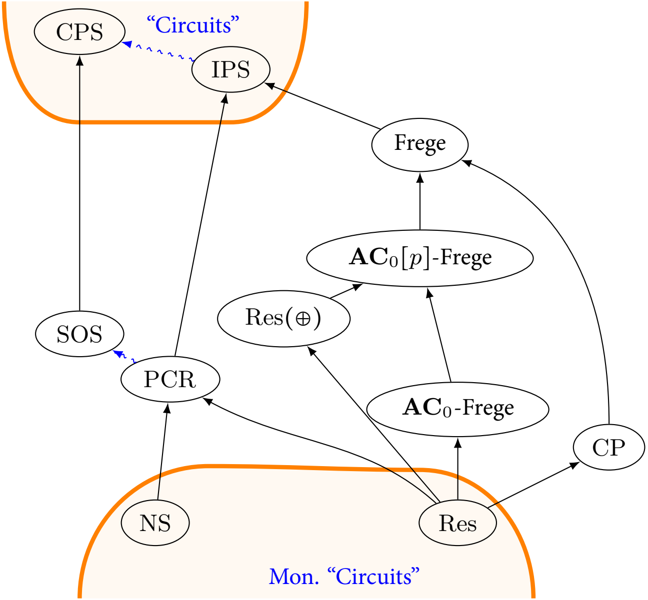
Hierarchy


CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
Res(⊕)
SOS
PCR
$\mathbf{AC}_0$-Frege
CP
NS
Res

Hierarchy

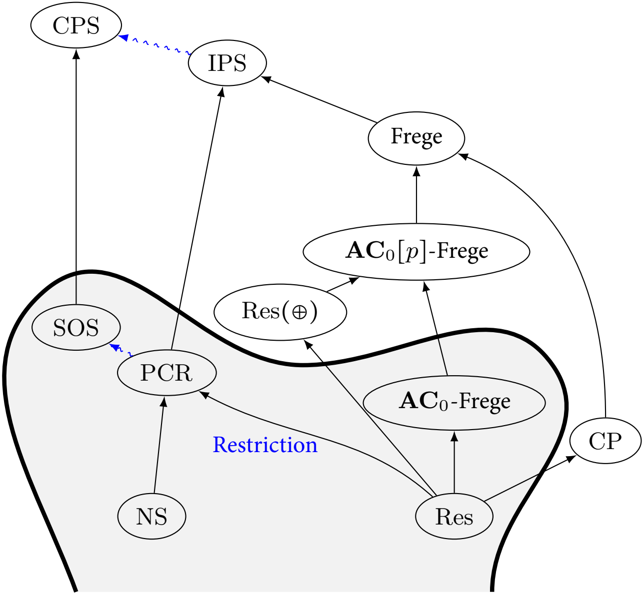

CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
Res(⊕)
SOS
PCR
$\mathbf{AC}_0$-Frege
CP
NS
Res
Mon. "Circuits"

Hierarchy

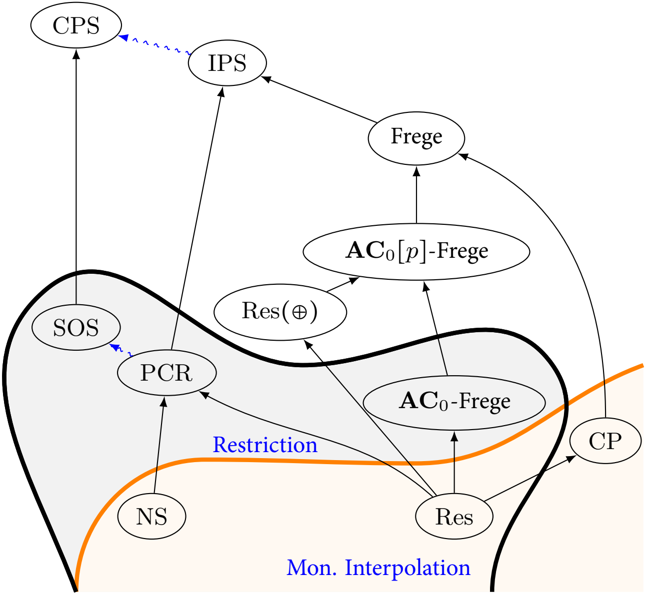


"Circuits"
CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
Res(⊕)
SOS
PCR
$\mathbf{AC}_0$-Frege
CP
NS
Res
Mon. "Circuits"

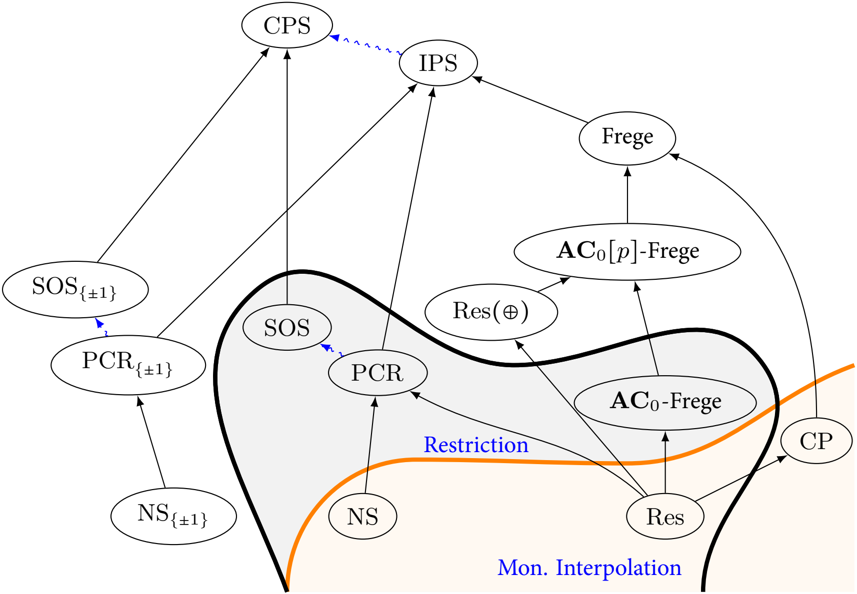
Hierarchy


CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
Res(⊕)
SOS
PCR
$\mathbf{AC}_0$-Frege
Restriction
CP
NS
Res

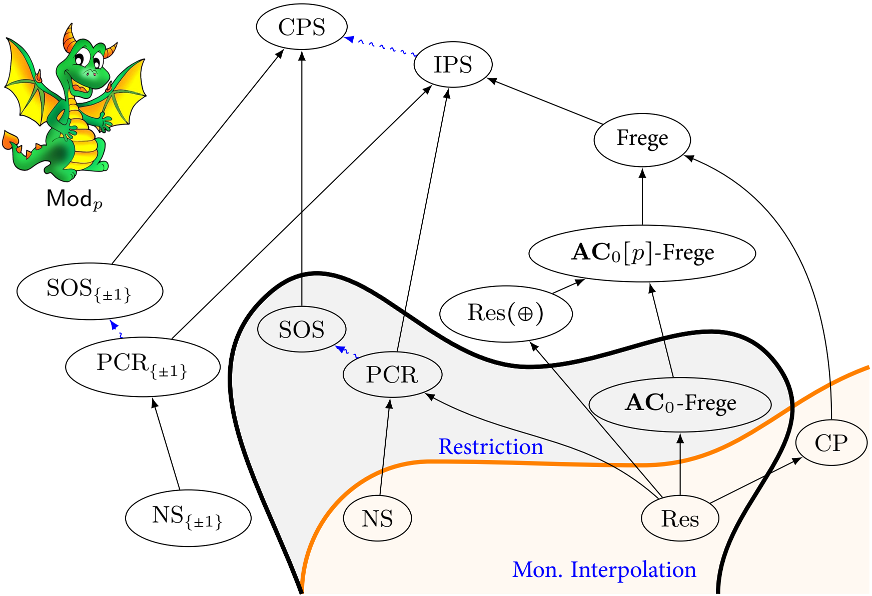
Hierarchy


CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
Res(⊕)
SOS
PCR
$\mathbf{AC}_0$-Frege
Restriction
CP
NS
Res
Mon. Interpolation

Hierarchy

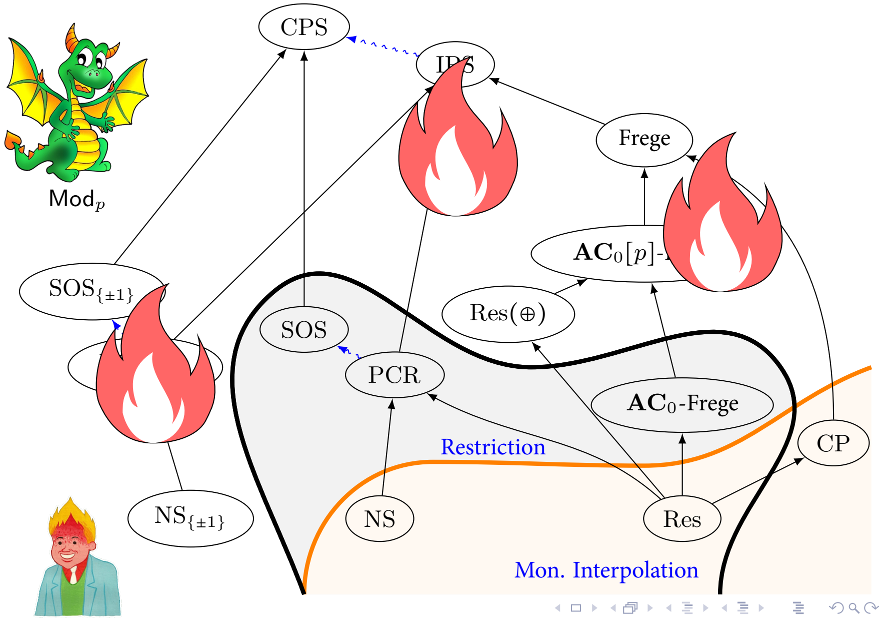

CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
$\text{SOS}_{\{\pm1\}}$
$\text{Res}(\oplus)$
SOS
$\text{PCR}_{\{\pm1\}}$
PCR
$\mathbf{AC}_0$-Frege
Restriction
CP
$\text{NS}_{\{\pm1\}}$
NS
Res
Mon. Interpolation

Hierarchy



Mod$_p$
CPS
IPS
Frege
$\mathbf{AC}_0[p]$-Frege
SOS$_{\{\pm1\}}$
Res(⊕)
SOS
PCR$_{\{\pm1\}}$
PCR
$\mathbf{AC}_0$-Frege
Restriction
CP
NS$_{\{\pm1\}}$
NS
Res
Mon. Interpolation

Hierarchy



CPS
IPS
Frege
Mod$_p$
$\mathbf{AC}_0[p]$-
SOS$_{\{\pm1\}}$
Res(⊕)
SOS
PCR
$\mathbf{AC}_0$-Frege
Restriction
CP
NS$_{\{\pm1\}}$
NS
Res
Mon. Interpolation