Proof Nets and Boolean Circuits

Kazushige Terui

terui@nii.ac.jp

National Institute of Informatics, Tokyo

Motivation (1)

- Proofs-as-Programs (Curry-Howard) correspondence:

Proofs	=	Programs
Cut-Elimination	=	Computation
(Normalization)		

- Usually interested in infinite, uniform, sequential computation such as functional programs.
- Can be extended to finite, nonuniform, parallel computation such as boolean circuits?

Motivation (2)

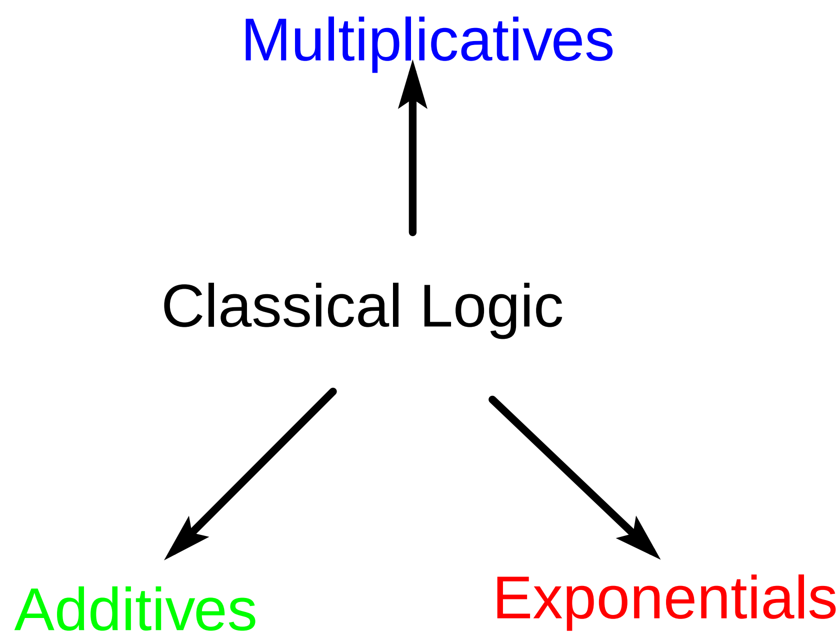
- Our goal:

Proofs	=	Circuits
Cut-Elimination	=	Evaluation

- What system of Proofs?
- Cut-elimination in Classical/Intuitionistic Logics: Non-elementary time. Too much!

Motivation (3)

- Linear Logic (Girard 87): a decomposition of Classical/Intuitionistic Logics:



- MLL: Classical Logic without weakening nor contraction.
- Has a nice parallel syntax: Proof Nets (Girard 87).
- Quadratic time cut-elimination procedure.

Motivation (4)

- Our specific goal: correspondence between MLL proof nets and circuits.
- (Mairson & Terui 2003) Encoding of circuits by linear size MLL proof nets $\Longrightarrow$ P-completeness of cut-elimination in MLL.

  > "Proof nets can represent all finite functions as SIZE-efficient as circuits."

- What about the DEPTH-efficiency? What about the converse?

Parallel computation

- Effective parallelization: Achieve a dramatic speed-up by use of a reasonable number of processors.
- Addition: School method ($O(n)$ time)
  $\Longrightarrow$ Parallel algorithm: (constant time)
- In boolean circuits,

$$\begin{aligned} \text{time} &= \text{depth} \\ \text{num of processors} &= \text{size (num of gates)} \end{aligned}$$

- Fundamental question: Are all feasible algorithms effectively parallelizable?
- $NC = P$ problem: Are all problems in P solvable by polynomial size poly-log depth circuits?

Outline

- Unbounded fan-in boolean circuits.
- Proof nets for MLLu and parallel cut-elimination procedure.
- Simulation of circuits by proof nets
- Simulation of proof nets by circuits
- Proof net complexity classes

Boolean circuits (1)

- An unbounded fan-in circuit with n inputs (and 1 output): a directed acyclic graph made of
  - input nodes $x_1, \ldots, x_n$
  - boolean gates $\neg$, $\bigwedge$, $\bigvee$ (and possibly more).

  (there is a distinguished node for output.)
- $\bigwedge$ and $\bigvee$ may have an arbitrary number of inputs.
- size = the number of gates
- depth = the length of the longest path

Boolean circuits (2)

- A circuit C accepts $w = i_1 \cdots i_n \in \{0,1\}^n$ if

  $C[x_1 := i_1, \ldots, x_n := i_n]$ evaluates to 1.

- C accepts $X \subseteq \{0,1\}^n$ if C accepts $w \Leftrightarrow w \in X$.

Formulas of MLLu

- Formulas:

$\alpha, \alpha^{\perp}$		Literals
$\otimes^n(A_1, \ldots, A_n)$,	$n \geq 2$	n-ary Conjunction
$⅋^n(A_1, \ldots, A_n)$,	$n \geq 2$	n-ary Disjunction

- Negation: defined by

$$(\otimes^n(A_1, \ldots, A_n))^{\perp} \equiv ⅋^n(A_n^{\perp}, \ldots, A_1^{\perp})$$

$$(⅋^n(A_1, \ldots, A_n))^{\perp} \equiv \otimes^n(A_n^{\perp}, \ldots, A_1^{\perp})$$

- Notation:

$$A \otimes B \equiv \otimes^2(A, B) \qquad A ⅋ B \equiv ⅋^2(A, B) \qquad A \multimap B \equiv A^{\perp} ⅋ B$$

Sequent calculus for MLLu

- Sequents: $\vdash \Gamma$, where Γ is a multiset of formulas.
- Inference Rules:

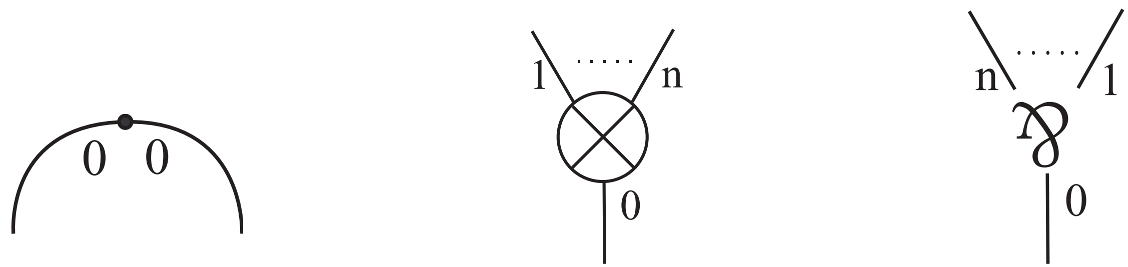
$$\frac{}{\vdash A, A^{\perp}}\ (Axiom) \qquad \frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\ (Cut)$$

$$\frac{\vdash \Gamma_1, A_1 \quad \cdots \quad \vdash \Gamma_n, A_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \otimes^n(A_1, \ldots, A_n)}\ \otimes^n \qquad \frac{\vdash \Gamma, A_1, \ldots, A_n}{\vdash \Gamma, \parr^n(A_1, \ldots, A_n)}\ \parr^n$$

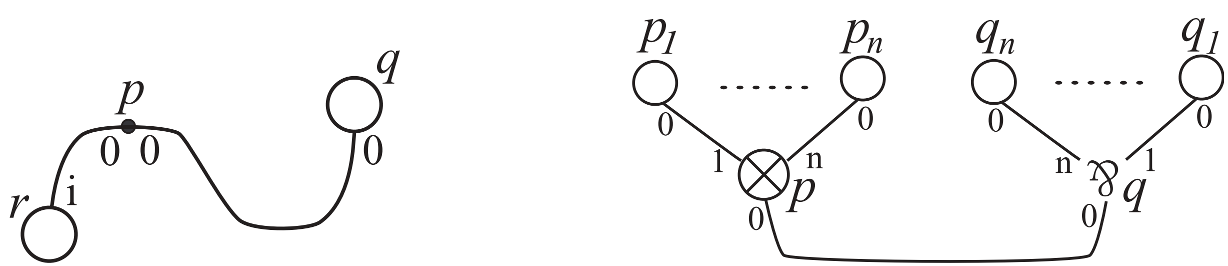
- Exchange is implicit. No weakening, no contraction.

Proof nets for MLLu

- Links:



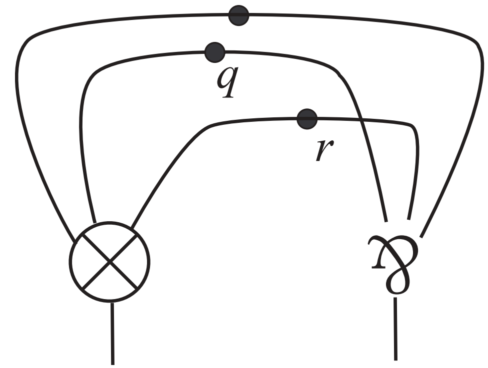
- Each link has several ports. Principal port(s) numbered 0.
- Cut: an edge connecting two principal ports.



- Proof nets are obtained from sequent proofs by extracting their structures (forgetting about formulas).

Example: Negation

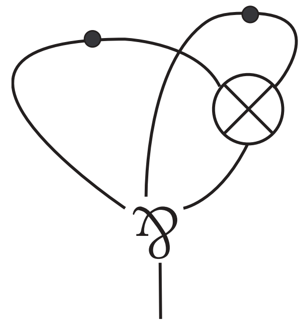
$$\dfrac{\dfrac{\dfrac{}{\vdash \alpha^\perp \parr \alpha^\perp, \alpha \otimes \alpha} \quad \dfrac{}{\vdash \alpha^\perp, \alpha} \quad \dfrac{}{\vdash \alpha^\perp, \alpha}}{\vdash \otimes^3(\alpha^\perp \parr \alpha^\perp, \alpha, \alpha), \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha}}{\vdash \otimes^3(\alpha^\perp \parr \alpha^\perp, \alpha, \alpha), \parr^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha)} \Longrightarrow$$



Example: Booleans

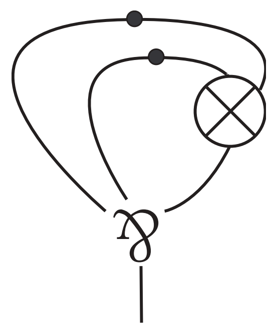
- $\mathbf{B} \equiv \alpha \multimap \alpha \multimap \alpha \otimes \alpha \equiv \parr^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha)$

$$\dfrac{\dfrac{\dfrac{}{\vdash \alpha^\perp, \alpha} \quad \dfrac{}{\vdash \alpha^\perp, \alpha}}{\vdash \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha}\otimes^2}{\vdash \parr^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha)}\parr^3 \quad \Longrightarrow$$



$\equiv \quad \mathsf{b}_1$

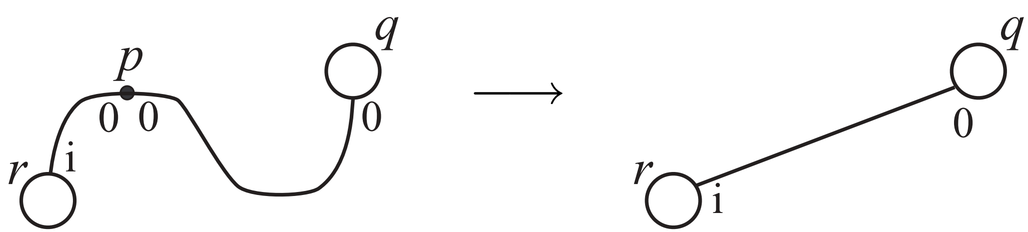
$$\dfrac{\dfrac{\dfrac{}{\vdash \alpha^\perp, \alpha} \quad \dfrac{}{\vdash \alpha^\perp, \alpha}}{\vdash \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha}\otimes^2}{\vdash \parr^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha)}\parr^3 \quad \Longrightarrow$$



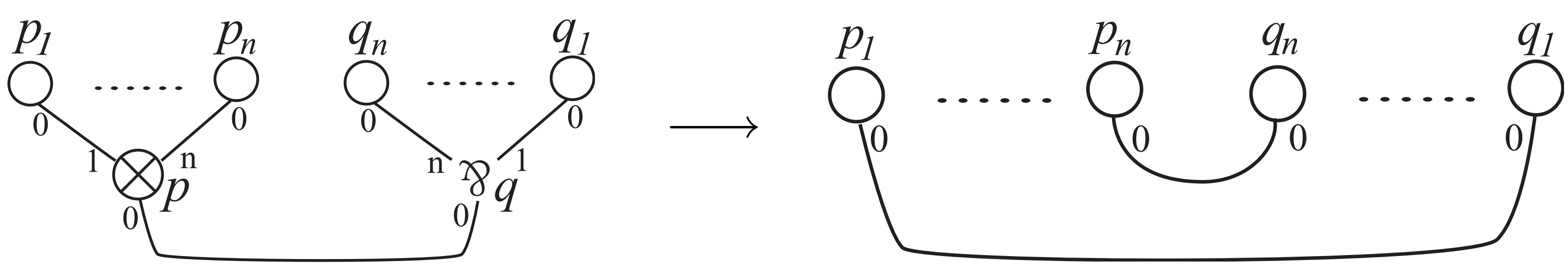
$\equiv \quad \mathsf{b}_0$

Reduction rules

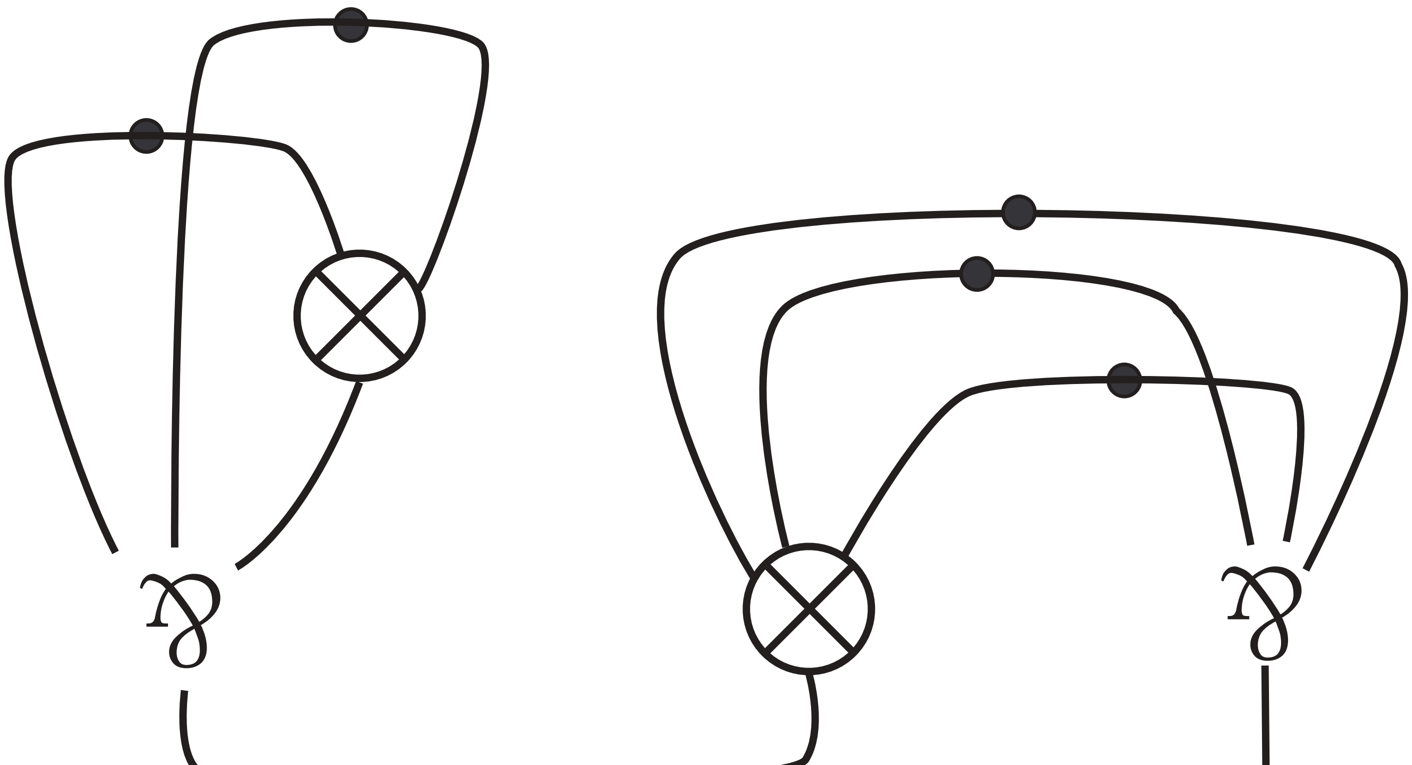
- Axiom reduction:



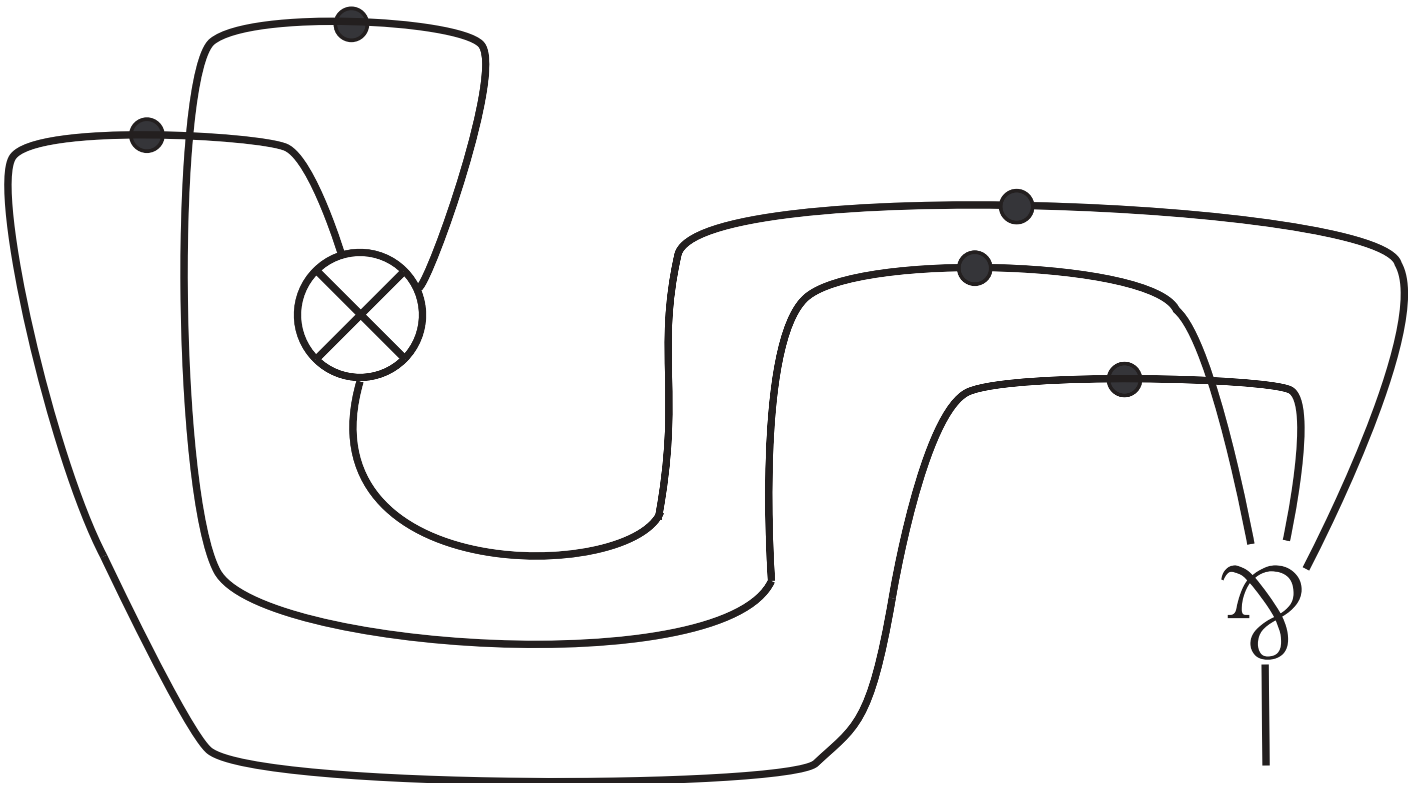
- Multiplicative reduction:



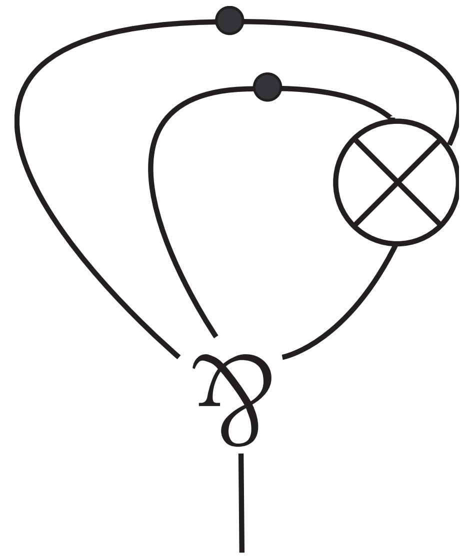
Example: Computing Negation of True



Example: Computing Negation of True



Example: Computing Negation of True



Sequential cut elimination

- Size $|P|$: number of links.

Sequential cut elimination

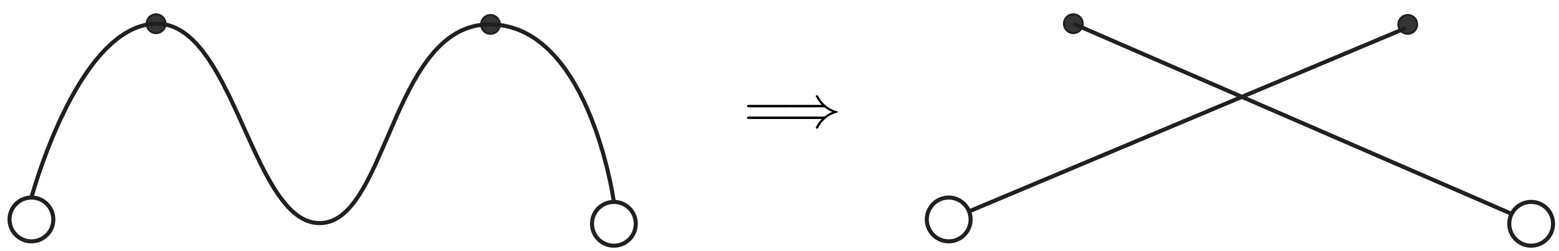
- Size $|P|$: number of links.
- Theorem (Girard 87): Every proof net P reduces to a cut-free proof net in $|P|$ steps.

Sequential cut elimination

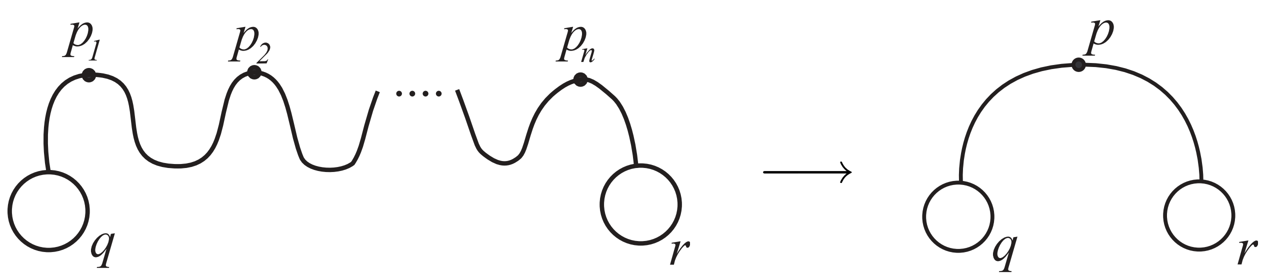
- Size $|P|$: number of links.
- Theorem (Girard 87): Every proof net P reduces to a cut-free proof net in $|P|$ steps.
- Too slow!

Parallel cut elimination (1)

- Applying two axiom reductions in parallel may conflict:

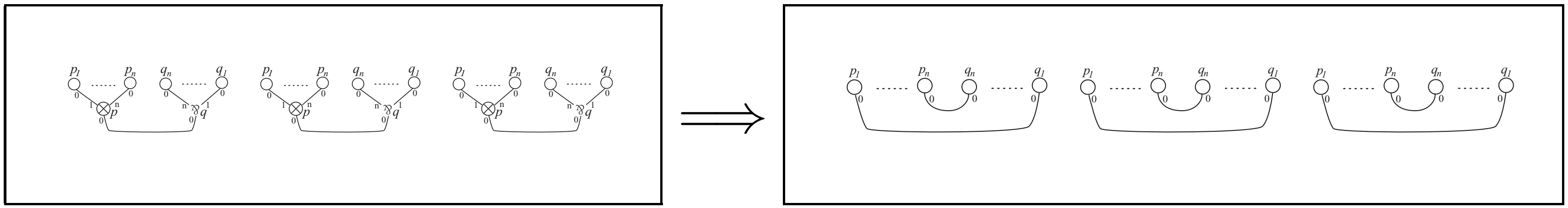


- Global axiom reduction:

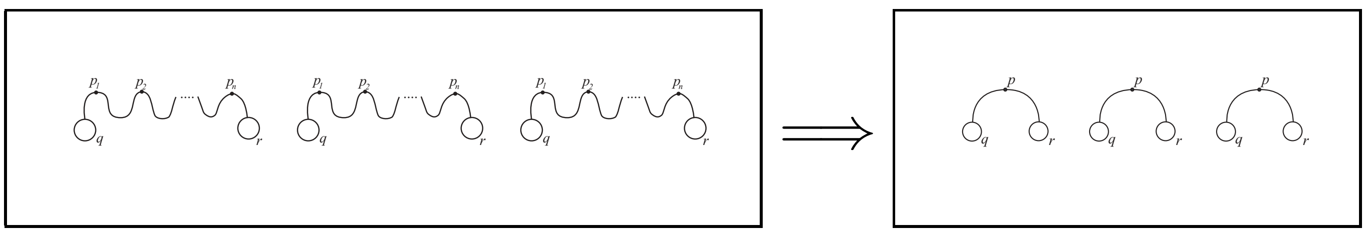


Parallel cut elimination (2)

- Parallel multiplicative reduction:



- Parallel axiom reduction:



- $P_1 \Longrightarrow P_2$ if P_2 is obtained from P_1 either by parallel m-reduction or by parallel a-reduction.

Parallel cut elimination (3)

- What controls the runtime of parallel cut-elimination?
- Depth $d(P)$: Maximal depth of cut formulas in it.
  (P is assumed to be typed by principal types (most general types))
- Depth of formulas:

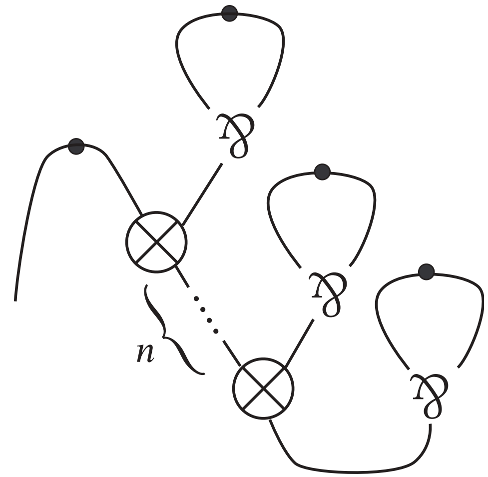
$$
\begin{aligned}
d(\alpha) = d(\alpha^{\perp}) &= 1 \\
d(\otimes^n(A_1, \ldots, A_n)) &= d(\parr^n(A_1, \ldots, A_n)) \\
&= max(d(A_1), \ldots, d(A_n)) + 1
\end{aligned}
$$

Parallel cut elimination (4)

- Theorem: Every proof net P reduces to a normal form in $2 \cdot d(P)$ parallel reduction steps.
- Proof: By applying parallel a-reduction, every cut becomes multiplicative. By applying parallel m-reduction, the depth decreases by 1. $\square$

Limitation on parallelization

- P_n:



- $|P_n|$ and $d(P_n)$ are linear in n.
- Parallel cut-elimination takes almost as long time as sequential cut-elimination.

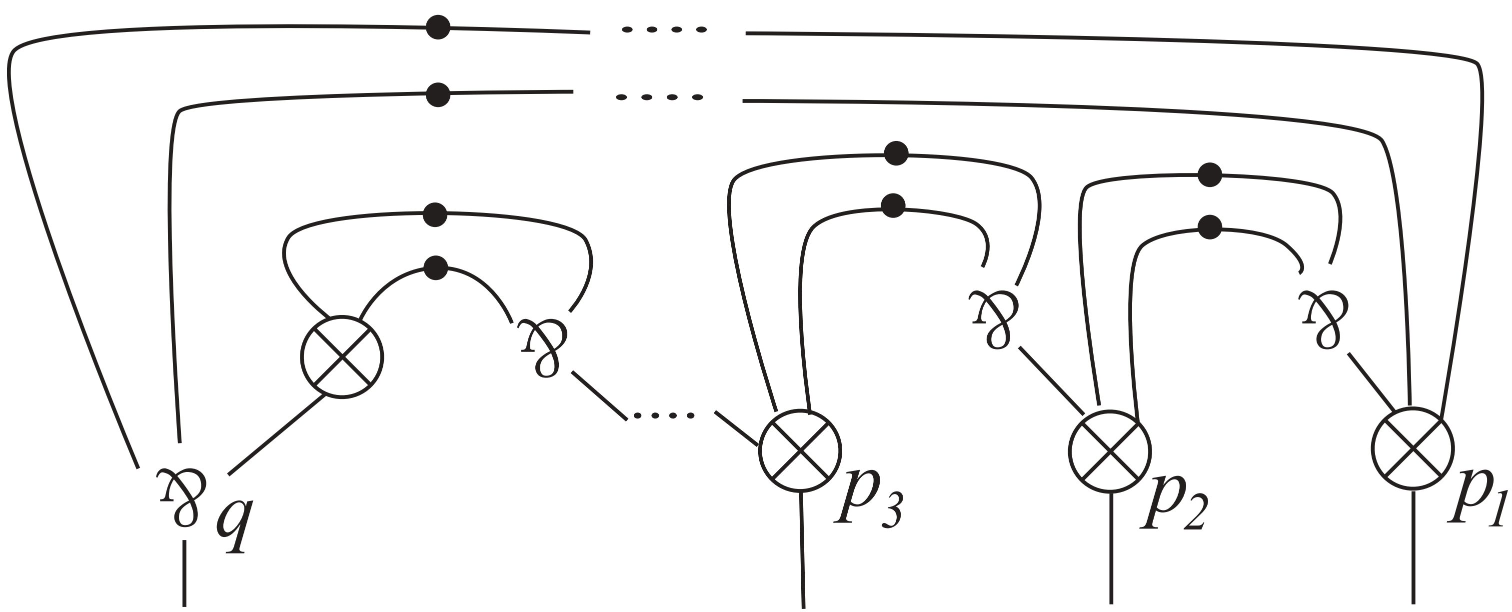
Representing circuits by proof nets

- Idea: Represent

Circuits	by	Proof nets
Boolean values	by	Proof nets (b_1, b_0)
Assignment	by	Cuts
Evaluation	by	Cut elimination

Representation of Parity

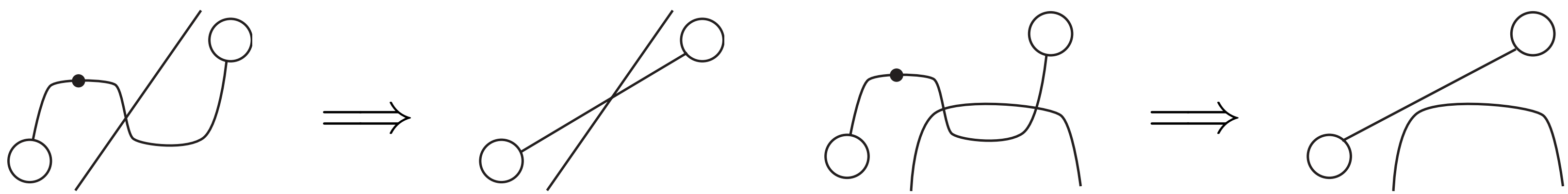
- Parity: $\text{PARITY}^n(x_1,\ldots,x_n) = 1$ iff the sum of $x_1,\ldots,x_n$ is odd.
- CANNOT be represented by (poly-size) circuits of constant depth.



- The conclusion is $\vdash \underbrace{\mathbf{B}^\perp,\ldots,\mathbf{B}^\perp}_{n\ times},\mathbf{B}$

Correctness of parity

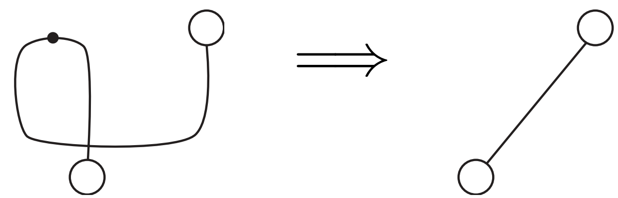
- There are exactly 2 crossings in the drawing.
- Multiplicative reduction does not change the num of crossings.
- Axiom reduction preserves the parity of the num of crossings:



- Therefore, the parity is 0 after cut-elimination.

Expressivity of flat proof nets

- Would break down if there were a self-crossing:



- But self-crossing can be avoided.
- As far as proof nets of conclusion $\vdash \underbrace{\mathbf{B}^\perp, \ldots, \mathbf{B}^\perp}_{n\ times}, \mathbf{B}$ are concerned, all what matters is the parity of the num of crossings.
- Theorem: Every proof net with the above conclusion represents either $\textsc{parity}^n$ or $\neg\textsc{parity}^n$.

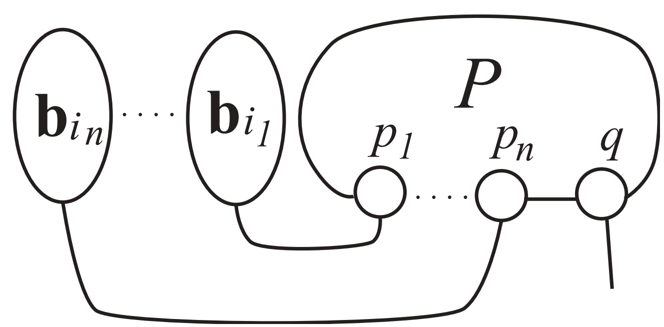
Boolean proof nets

- Boolean proof net: a proof net with conclusion

$$\vdash \mathbf{B}[A_1/\alpha]^{\perp}, \ldots, \mathbf{B}[A_n/\alpha]^{\perp}, \otimes^m(\mathbf{B}, \vec{G})$$

  for some $A_1, \ldots, A_n$ and $\vec{G}$ (garbage).

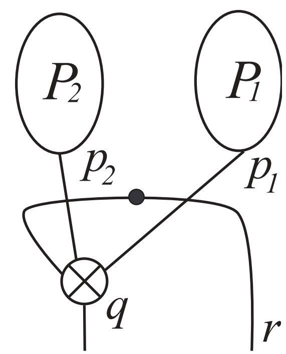
- Given $w = i_1 \cdots i_n \in \{0,1\}^n$, define $P(w)$ to be:



- P accepts $w \in \{0,1\}^n$ if $P(w)$ reduces to $\otimes(\mathrm{b}_1, \vec{P_G})$ for some proof nets $\vec{P_G}$ with conclusions $\vec{G}$.

Conditional and Disjunction

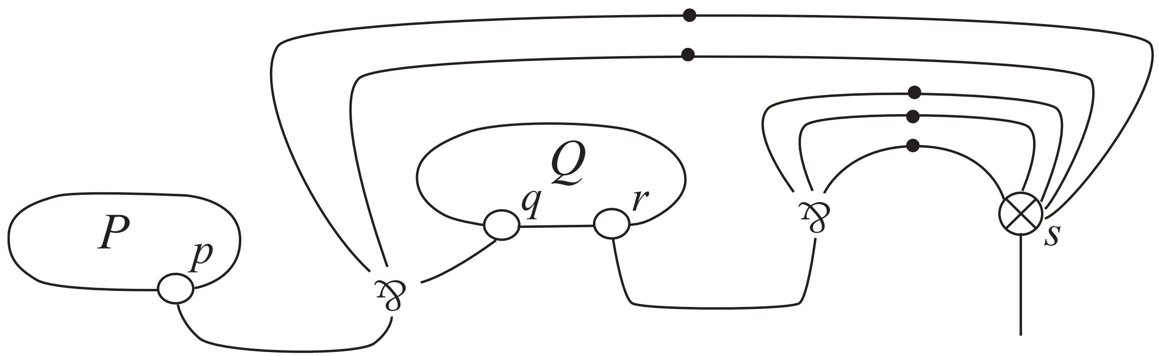
- if q then P_1 else P_2



- Disjunction: or$(p,q) \equiv$ if p then b_1 else q
- Disjunctions of arbitrary arity can be represented by proof nets of constant depth.

Composition

- Composition:



- The depth may increase:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ P_1\\ \vdash \Gamma, \mathbf{B}\end{array}}{\vdash \Gamma[A/\alpha], \mathbf{B}[A/\alpha]} \qquad \begin{array}{c}\vdots\ P_2\\ \vdash \mathbf{B}^{\perp}[A/\alpha], \Delta, \mathbf{B}\end{array}}{\vdash \Gamma[A/\alpha], \Delta, \mathbf{B}}$$

Proof Net for Majority (1)

- $\text{MAJ}^n(x_1 \cdots x_n) = 1$ if at least half of x_i's are 1.
- Let $id, sh : \{0,1\}^{n+1} \longrightarrow \{0,1\}^{n+1}$ be:

$$\begin{aligned} id(i_1 \cdots i_{n+1}) &= i_1 \cdots i_{n+1} \\ sh(i_1 \cdots i_{n+1}) &= i_2 \cdots i_{n+1} i_1 \end{aligned}$$

- F: higher order functional

$$\begin{aligned} F(0) &= id \\ F(1) &= sh \end{aligned}$$

Proof Net for Majority (2)

- $\textsc{maj}^n$ given by

$$\textsc{maj}^n(x_1 \cdots x_n) = FstBit(F(x_1) \circ \cdots \circ F(x_n)(\underbrace{0 \cdots 0}_{n/2}\underbrace{1 \cdots 1}_{n/2+1}))$$

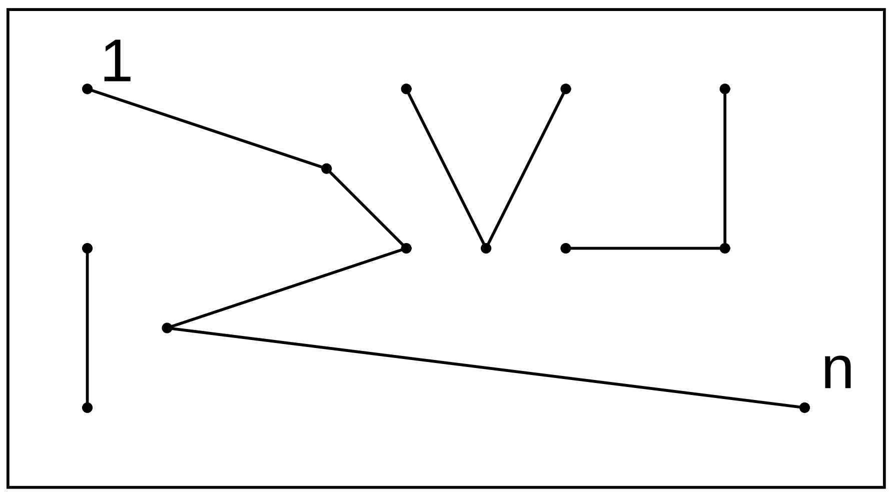
- Example:

$$\begin{aligned}
\textsc{maj}^6(101101) &= FstBit(F(1)F(0)F(1)F(1)F(0)F(1)(0001111)) \\
&= FstBit(sh \circ id \circ sh \circ sh \circ id \circ sh(0001111)) \\
&= FstBit(1110001) \\
&= 1
\end{aligned}$$

- Represented by proof nets of constant depth. (CANNOT be represented by (poly-size) circuits of constant depth.)

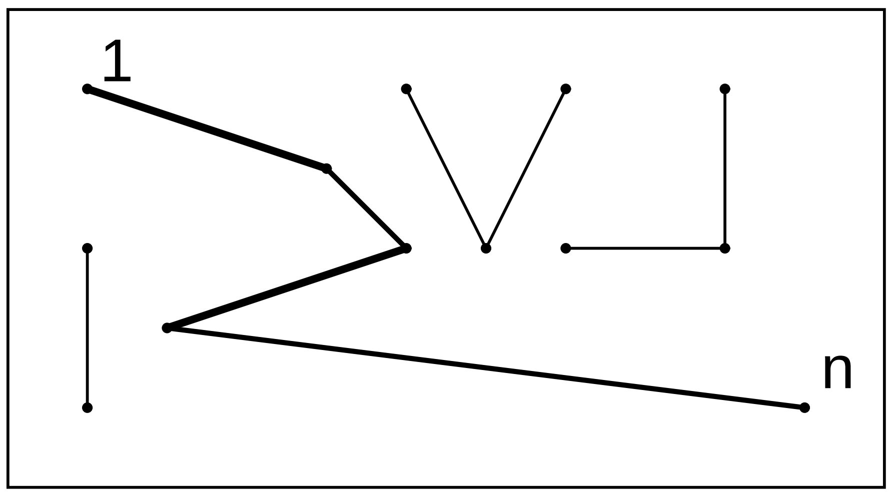
St-connectivity$_2$

- **Input:** an undirected graph of degree 2 (i.e., nonbranching graph) with vertices $\{1, \ldots, n\}$. Assume it is coded by a $n \times n$ boolean matrix.
- **Output:** is 1 if vertices 1 and n are connected.



St-connectivity$_2$

- Input: an undirected graph of degree 2 (i.e., nonbranching graph) with vertices $\{1, \ldots, n\}$. Assume it is coded by a $n \times n$ boolean matrix.
- Output: is 1 if vertices 1 and n are connected.



- Can simulate MAJ in constant depth.
- Represented by proof nets of constant depth.

From Circuits to Proof nets

- Theorem: For every circuit C of size s and depth d (possibly equipped with $st\textsc{conn}_2$), there is a boolean proof net P_C of size $O(s^5)$ and depth $O(d)$ which accepts the same set as C.

From Circuits to Proof nets

- Theorem: For every circuit C of size s and depth d (possibly equipped with $st\textsc{conn}_2$), there is a boolean proof net P_C of size $O(s^5)$ and depth $O(d)$ which accepts the same set as C.

$$\boxed{\begin{array}{c}\text{Circuit } C\\ \text{size } s\text{, depth } d\end{array}} \quad\Longrightarrow\quad \boxed{\begin{array}{c}\text{Proof Net } P_C\\ \text{size } O(s^5)\text{, depth } O(d)\end{array}}$$

From Circuits to Proof nets

- Theorem: For every circuit C of size s and depth d (possibly equipped with $st\textsc{conn}_2$), there is a boolean proof net P_C of size $O(s^5)$ and depth $O(d)$ which accepts the same set as C.

$$\boxed{\begin{array}{c}\text{Circuit } C\\ \text{size } s\text{, depth } d\end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c}\text{Proof Net } P_C\\ \text{size } O(s^5)\text{, depth } O(d)\end{array}}$$

- Proof: $\neg$, $\bigwedge^n$, $\bigvee^n$, $st\textsc{conn}_2^n$ represented by a proof net of size $O(n^4)$ and of constant depth. Composition increases the depth linearly.

Representing proof nets by circuits

- Idea: Represent

A proof net P	by	a set of boolean values
One-step reduction	by	constant depth circuit
$2 \cdot d(P)$ reduction steps	by	$O(d)$-depth circuit

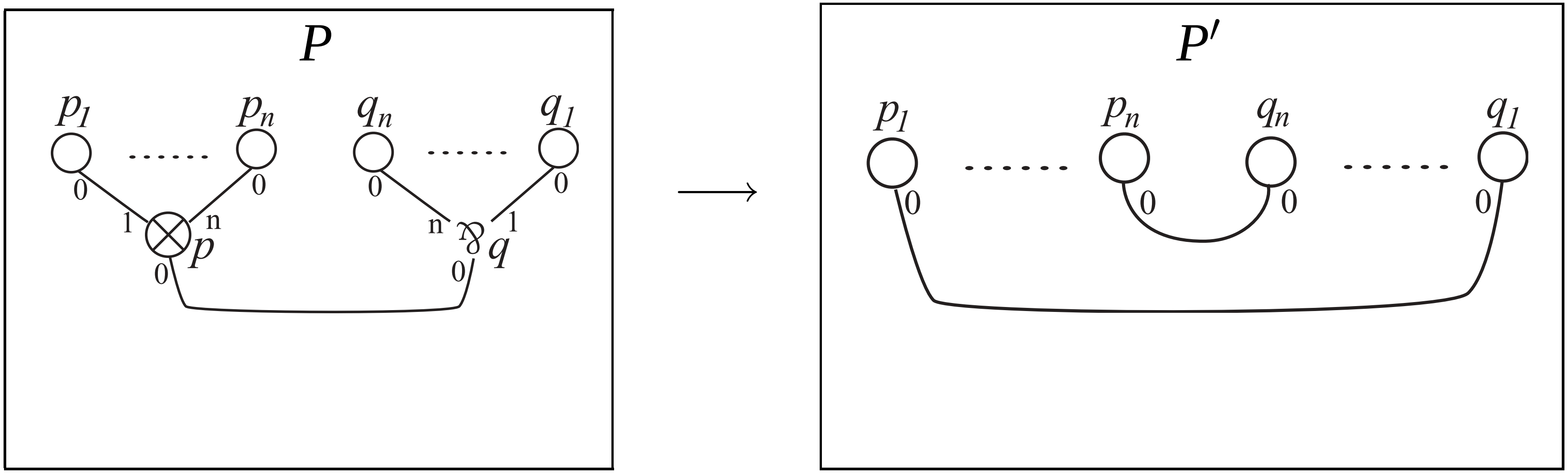
Coding proof nets by boolean values

- P : a proof net with links $\subseteq L$.
- $Conf(P)$: consists of the following boolean values:
  For $p, q \in L$,

$$\begin{array}{lcl} alive(p) & \Leftrightarrow & p \text{ is a link of } P \\ sort(p,s) & \Leftrightarrow & \text{link } p \text{ is of sort } s \in \{\otimes, \invamp, \bullet\} \\ edge(p,i,q,j) & \Leftrightarrow & \text{there is an edge between port } i \text{ of} \\ & & \text{link } p \text{ and port } j \text{ of link } q \end{array}$$

- Build a circuit C s.t.
  if $P_1 \Longrightarrow P_2$ then C computes $Conf(P_2)$ from $Conf(P_1)$.

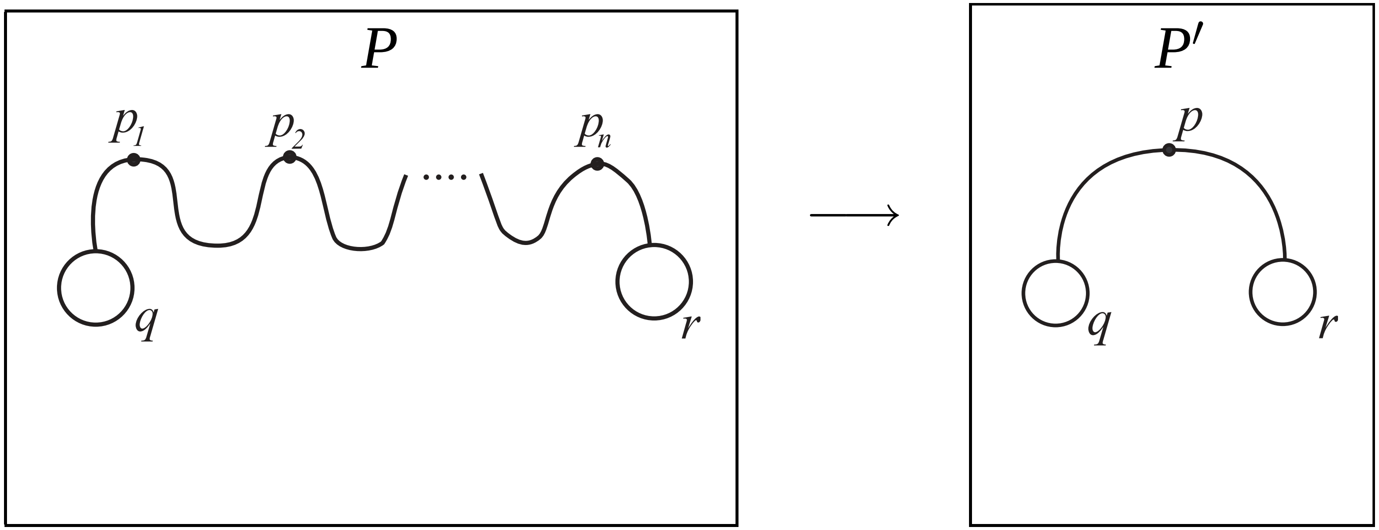
Multiplicative Reduction



- Easily implemented by a constant depth circuit: E.g,

$$
\begin{aligned}
edge'(p_i,0,q_i,0) \quad &= \quad edge(p_i,0,q_i,0) \vee \\
& \bigvee_{p,q \in L} \left(edge(p,0,q,0) \wedge \bigvee_j edge(p_i,0,p,j) \wedge edge(q_i,0,q,j) \right)
\end{aligned}
$$

Global Axiom Reduction



- Naive attempt to build a constant depth circuit leads to exponential size.
- Use $st\textsc{conn}_2$ gates.
- Apply to the axiom-cut subgraph of P, that is of degree 2.

From Proof nets to Circuits

- Theorem: For every boolean proof net P of size s and depth d, there is a circuit C (with $st\textsc{conn}_2$ gates) of size $O(s^4)$ and depth $O(d)$ which accepts the same set as P.

From Proof nets to Circuits

- Theorem: For every boolean proof net P of size s and depth d, there is a circuit C (with $st\text{CONN}_2$ gates) of size $O(s^4)$ and depth $O(d)$ which accepts the same set as P.

$$\boxed{\begin{array}{c}\text{Proof Net } P\\ \text{size } s\text{, depth } d\end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c}\text{Circuit } C_P\\ \text{size } O(s^4)\text{, depth } O(d)\end{array}}$$

From Proof nets to Circuits

- Theorem: For every boolean proof net P of size s and depth d, there is a circuit C (with $st\textsc{conn}_2$ gates) of size $O(s^4)$ and depth $O(d)$ which accepts the same set as P.

$$\boxed{\begin{array}{c}\text{Proof Net } P\\ \text{size } s\text{, depth } d\end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c}\text{Circuit } C_P\\ \text{size } O(s^4)\text{, depth } O(d)\end{array}}$$

- Proof: Cut-elimination of P requires of $2 \cdot d$ steps. Each step simulated by a constant depth circuit (with $st\textsc{conn}_2$ gates). Initialization and acceptance checking also by constant depth circuits.

Proof net complexity classes (1)

- For $X \subseteq \{0,1\}^*$,

  $X \in AC^i(\mathscr{F}) \overset{def}{\iff}$ X is accepted by a polynomial size $\log^i$-depth family of unbounded fan-in circuits with additional gates from $\mathscr{F}$.

  $X \in APN^i \overset{def}{\iff}$ X is accepted by a polynomial size $\log^i$-depth family of proof nets.

- Theorem: For every $i \geq 0$,

$$APN^i = AC^i(st\textsc{conn}_2).$$

Proof net complexity classes (2)

- $APN = \bigcup_i APN^i$.
- Theorem $APN = NC$.
- Proof: $AC^i \subseteq AC^i(st\textsc{conn}_2) = APN^i \subseteq AC^{i+1}$ and $NC = \bigcup_i AC^i$.
- $P/poly$: nonuniform version of P.
- Theorem $P/poly =$ the class of languages $X \subseteq \{0,1\}^*$ accepted by polynomial size families of proof nets.
- Note $AC^0(st\textsc{conn}_2) = L/poly$ (nonuniform logspace). Hence,

$$AC^0 \subsetneq NC^1 \subseteq L/poly = APN^0 \subseteq AC^1$$

Conclusion

- Extended the Proofs-as-Programs paradigm to a finite, nonuniform, parallel setting.
- Characterized proof net complexity by circuit complexity.
- Proof nets represent ”higher order gates.”

Future Work

- $NC^{i+1} \subseteq APN^{i}$?
- Comparison with other approaches to proofs-circuits correspondence (Propositional Proof Systems and Bounded Arithmetic).
- Study the bounded fan-in case.
- Implicit parallel complexity.