Propositional Logic for Circuit Classes

Klaus Aehlig, Arnold Beckmann

Comprehension in Propositional Proofs

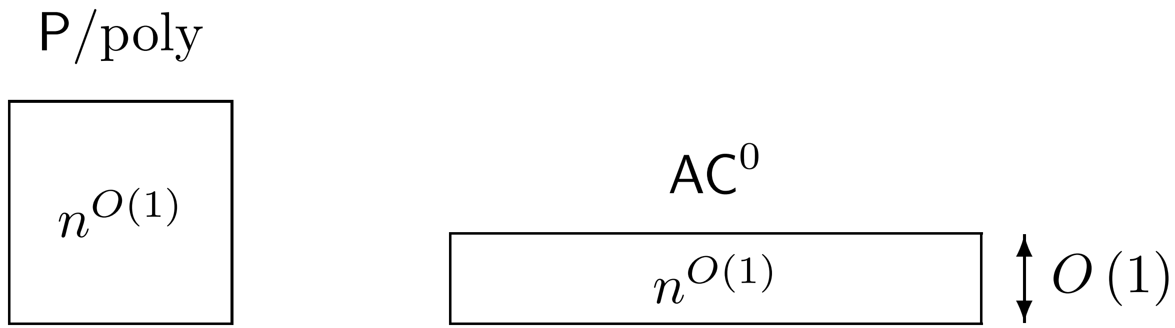
- Extension Rule in Propositional Logic

$$p \leftrightarrow A$$

- introduce several variables $\Rightarrow$ several rules needed

! irrespectively of whether the new variables are interdependent

- But dependencies make a big difference, e.g., for circuits



$\boxed{\textbf{A Calculus for Circuit Complexity}}$

$\therefore$ study *height* instead of *size*
(as it has always been done in traditional proof theory)

- To make the connection precise, need separation
  $\rightsquigarrow$ relativised computation
  Oracle statements:

$$p \leftrightarrow \alpha_k(p_1, \ldots, p_k)$$

- Another goal: generic target for propositional translations
  $\rightsquigarrow$ *quantified* propositional logic

AC^0-**Tait**

- propositional variables: p, q, r, $\ldots$; negated variables: $\bar{p}$, $\bar{q}$, $\bar{r}$, $\ldots$

$$\frac{}{\Gamma, p, \bar{p}}$$

- oracle queries $\alpha_k(\wp_1, \ldots, \wp_k)$ and negated queries $\bar{\alpha}_k(\wp_1, \ldots, \wp_k)$
(the $\overrightarrow{\wp}$ are propositional atoms)

- conjunction: $\bigwedge_k A_1 \ldots A_k$

$$\frac{\ldots \qquad \Gamma, A_i \qquad \ldots}{\Gamma, \bigwedge_k A_1 \ldots A_k} \bigwedge_k$$

- disjunction: $\bigvee_k A_1 \ldots A_k$

$$\frac{\Gamma, A_j}{\Gamma, \bigvee_k A_1 \ldots A_k} \bigvee_k^j$$

AC^0-Tait (cont'd)

- existential $\exists_k \overrightarrow{p} A(\overrightarrow{p})$ and universal $\forall_k \overrightarrow{p} A(\overrightarrow{p})$ quantification

$$\frac{\Gamma, A(\overrightarrow{a})}{\Gamma, \forall_k \overrightarrow{p} A(\overrightarrow{p})} \qquad \frac{\Gamma, A(\overrightarrow{\wp})}{\Gamma, \exists_k \overrightarrow{p} A(\overrightarrow{p})}$$

Note: only atoms as witnesses!

- Negation is a defined concept
  e.g., $\neg p \equiv \bar{p}$, $\neg \exists_k \overrightarrow{p} A(\overrightarrow{p}) \equiv \forall_k \overrightarrow{p} \neg A(\overrightarrow{p})$, ...

- Comprehension

$$\frac{\Gamma, \neg(p_1 \leftrightarrow \varphi_1), \ldots, \neg(p_k \leftrightarrow \varphi_k)}{\Gamma}$$

$\overrightarrow{p}$ *disjoint and new (i.e., not free in* $\overrightarrow{\varphi}$ *or* Γ*)*
$\overrightarrow{\varphi}$ *quantifier free*

AC^0-Tait, Cuts

- Propositional induction

  Given truth values $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$, output q_i if i is the smallest index such that p_i is true.

  can be solved in AC^0.

! standard calculi would require a tree of cuts of height $\log(n)$

$\rightsquigarrow$ Add a new rule ("multi-cut")

$$\frac{\ldots \qquad \Gamma, \Delta_i \qquad \ldots}{\Gamma}$$

with the $\overrightarrow{\Delta}$ *quantifier-free, and* $\overrightarrow{\Delta} \vdash \emptyset$ *by cuts only*

- More complex cuts (Γ, C and $\Gamma, \neg C$, ergo Γ) are admissible and can be eliminated at the usual costs

$\boxed{\textbf{Iteration}}$

- quite a powerful calculus—how to get (lower) height bounds?

! height corresponds to sequential time

$\leadsto$ what is an inherently sequential principle?

- When iterating a function 0, $f(0)$, $f(f(0))$, $f(f(f(0)))$ ...
  the evaluations of f have to be done one after another

  ...provided the domain/range of f is big enough!

The Iteration Principle

- Consider $f\colon [2^n] \to [2^n]$ *(domain of size n bits)*
- Bit-graph: code the i'th bit of $f(a)$ by $\alpha_{n+\log(n)}(i, a)$
  (with n bits for a and $\log(n)$ bits for i)
- How to express $f^\ell(0) = b$ for $\ell \gg n$, e.g., $\ell \in [2^n]$?

$\rightsquigarrow$ Add another oracle to check the answers!
Use $\alpha_{2n}(\ell, b)$ to stand for $f^\ell(0) = b$

- Iteration principle $\Phi_{n,\ell}$

$$\exists_{4n} \overrightarrow{p}\,\overrightarrow{p}\,'\overrightarrow{q}\,\overrightarrow{q}\,'[\text{ ``}f^\ell(0) = \overrightarrow{p}\text{''} \vee \neg\text{``}f^0(0) = 0\text{''}$$
$$\vee(\text{``}\overrightarrow{q}\,' = \overrightarrow{q} + 1\text{''} \ \wedge\ \text{``}f^{\overrightarrow{q}}(0) = \overrightarrow{p}\text{''} \ \wedge$$
$$\text{``}f(\overrightarrow{p}\,) = \overrightarrow{p}\,'\text{''} \ \wedge\ \neg\text{``}f^{\overrightarrow{q}\,'}(0) = \overrightarrow{p}\,'\text{''})\ ]$$

$\boxed{\textbf{Boundedness}}$

- Assume $\vdash^h \Phi_{n,\ell}$. Want to show $\ell \le h$.

$\rightsquigarrow$ find a path through the proof with all sequents of the form $\Phi_{n,\ell}, \Delta$ with Δ quantifier-free and false

- On this path reveal f only a little bit

  α contains exponentially many bits of information!

$\rightsquigarrow$ consider *partial* function $f\colon [2^n] \rightharpoonup [2^n]$

- f is ℓ-sequential, if for some $k \le \ell$

$$0, f(0), f^2(0), \ldots, f^k(0)$$

  are defined but $f^k(0) \notin \mathsf{dom}(f)$.

Extending Partial Functions

- Keep f still s-sequential after having followed a path for s steps
- If $f(a)$ is defined, this fixes $\alpha_{n+\log(n)}(i, a)$ in the obvious way.
- ... have to fix "$f^b(0) = c$" as well
- Recall: ... $f^k(0) \notin \mathsf{dom}(f)$
  so values in the domain are "forbidden" for future extensions!

$\therefore$ can set "$f^b(0) = c$" to false, if $c \in \mathsf{dom}(f)$ and $f^b(0)$ undefined
in particular, $c \in \mathsf{dom}(f)$ forces "$f^b(0) = c$" to have a truth value

- To extend $\mathsf{dom}(f)$ by M, just pick $a \notin M \cup \mathsf{dom}(f)$ and set $f'(x) = a$ for the new x
  the new f' is then $s + 1$ sequential and compatible to the "$f^b(0) = c$" already fixed

Circuits and Iteration

- Can show that a circuit of height h can be evaluated
  ...by a proof of height $h + O(1)$
  just do comprehension level by level!
- There is a circuit of height h that computes $f^h(0)$

... so knowing that this circuit can be evaluated implies the iteration principle

! By a constant height proof!