A Proof of Independence of $\boldsymbol{P} \neq \boldsymbol{NP}$ May Imply Nothing Productive for Realistic Computer Science

Anatol Slissenko
University Paris 12
`slissenko@univ-paris12.fr`

Here are some remarks (mentioned at the end of my memoirs [1]) on logical form of SAT$\notin \boldsymbol{P}$ (that is equivalent to $\boldsymbol{P} \neq \boldsymbol{NP}$) and on what logical independence of this conjecture may mean. It is trivial and well-known that for closed formulas of the form $\forall\, X\, F(X)$, where $F(X)$ has bounded quantifiers (bounded to finite sets, maybe parametric), independence from a formal system (which is consistent and not weaker than arithmetic) implies its validity: if $\forall X F(X)$ is not valid then for some X_0 the formula $F(X_0)$ must be valid, and arithmetic is complete with respect to quantifier bounded closed formulas.

Consider SAT based version of $\boldsymbol{P} \neq \boldsymbol{NP}$:

$$\forall\alpha(\alpha \text{ is an algorithm recognizing SAT} \rightarrow \text{time complexity of } \alpha \text{ is superpolynomial}) \qquad (1)$$

To fix some notations rewrite it as

$$\forall\alpha(C(\alpha) \rightarrow L(\alpha)), \qquad (2)$$

or

$$\forall\alpha(\forall x C'(\alpha) \rightarrow \forall k \exists y L'(\alpha, k, y)) \qquad (3)$$

where

- $C(\alpha) =_{df} \forall x C'(\alpha, x)$,
- $\forall x C'(\alpha, x) =_{df} \forall x \exists m \exists v (T(\alpha, x, v, m) \,\&\, (v = 0 \leftrightarrow x \in SAT))$,
- T is Kleene predicate

$$T(\alpha, x, v, m) \leftrightarrow \text{"}\alpha \text{ finishes its work after exactly } m \text{ steps and outputs v"}\ ,$$

- $L(\alpha) =_{df} \forall k \exists y L'(\alpha, y, k)$,
- $\forall k \exists y L'(\alpha, k, y) =_{df} \forall k \exists y (time_\alpha(y) > |y|^k)$,
- $time_\alpha(y)$ is time complexity of α for argument y.

Clearly,
- the predicate T is polytime,
- the existential quantifiers in $C'(\alpha, x)$ can be bounded: $\exists m \leq 2^{\mathcal{O}(|x|)}\ \exists v \in \{0, 1\}$,
- predicate $time_\alpha(y) > |y|^k$ is polytime,

- formula $x \in SAT$ has bounded quantifiers.

Thus all unbounded quantifiers are explicitly given in the representation (3). One can transform this formula into the prenex form

$$\forall\,\alpha\;\forall\,k\;\exists\,x\;\exists\,y\;\big(C'(\alpha) \rightarrow L'(\alpha,k,y)\big) \tag{4}$$

(or any other one). But there are no visible arguments to bound quantifier on x, as it is actually a universal quantifier in the premise of the correctness of α.

Restrict ourselves to algorithms α whose correctness is provable in a formal system $\mathcal{F}$. To define such algorithms introduce the notation:

$$C_{\mathcal{F}}(\alpha) =_{df} \exists\,D\;\big(D \text{ is a proof in } \mathcal{F} \text{ of } C(\alpha)\big)$$

$$=_{df} \exists\,D\;PROOF_{\mathcal{F}}(D,\alpha).$$

As a restricted version of (3) we take:

$$\forall\,\alpha\;\big(C_{\mathcal{F}}(\alpha) \rightarrow \forall\,k\;\exists\,y\;L'(\alpha,k,y)\big). \tag{5}$$

Now the prenex form

$$\forall\,\alpha\;\forall\,D\;\forall\,k\;\exists\,y\;\big(PROOF_{\mathcal{F}}(D,\alpha) \rightarrow L'(\alpha,k,y)\big). \tag{6}$$

of (5) is more treatable.

Restricting the existential quantifier $\exists k$ in (6) preserves the realistic computational value of the initial assertion. Thus, the formula

$$\forall\,\alpha\;\forall\,D\;\forall\,k\;\exists\,y \le \varphi(\alpha,D,k)\;\big(PROOF_{\mathcal{F}}(D,\alpha) \rightarrow L'(\alpha,k,y)\big) \tag{7}$$

with φ fast growing but provably computable in the formal system, say hyper-exponential, can be considered as a version of $\boldsymbol{P} \neq \boldsymbol{NP}$ adequate to realistic computations, though it is formally weaker than the initial formulation (2). And this version is of the form for which independence implies validity.

So, there could be 2 sources of independence of (2) which does not imply validity (there may be other sources):

(1) algorithms with not provable correctness;

(2) no sequence $\{y_k\}$ providing high complexity, i. e., such that $time_{\alpha}(y_k) > |y_k|^k$, is representable in a provable way. E. g., every such a sequence is very sparse: $length(y_k)$ grows faster than any provably computable total function.

The first source is hard to exploit. It is something like non constructive proof of the existence of algorithms that solve SAT but whose complexity properties are unknown and unprovable in the formal system. The second source, if exploited, gives a result of no interest for realistic computations.

References

[1] A. Slissenko. St.Petersburg/Leningrad (1961-1998): From logic to complexity and further. In C. Calude, editor, *People and Ideas In Theoretical Computer Science*, pages 274–313. Springer-Verlag, 1998.