Reckhow's Theorem

Yuval Filmus

November 2010

1 Introduction

In §5.3.1 of his thesis [2], Reckhow showed that any two Frege systems p-simulate each other. One of the difficulties involves translation of formulas with $\oplus$ into formulas in the De Morgan basis $\neg, \vee, \wedge$ without blowing up the formula size. Naive translation using a representation such as $A \oplus B \leftrightarrow (A \wedge \neg B) \vee (\neg A \wedge B)$ can result in exponentially long formulas. However, a simple argument attributed to Spira, described in §5.3.1.3, shows how to do that with only a polynomial blowup, $O(n^2)$ in case the only 'problematic' connective is $\oplus$. Reckhow's lemma 5.3.1.4.e shows how to 'rebalance' a formula, and is at the heart of his intensional comprehension arguments, which are brought to full fruition in §5.3.1.4.

In this note we will give an account of Reckhow's elegant proof, rephrased as an algorithm to transform an arbitrary proof (over the De Morgan basis) into balanced form with only a polynomial blowup. We do so mainly because the theorem, as such, is not mentioned in Reckhow's thesis, and moreover his arguments are somewhat difficult to follow. The theorem is mentioned and proved in Krajíček's book [1] as Lemma 4.4.14. We offer an alternative exposition.

Before plunging into the details, let us state the desired result.

Theorem 1.1 (Reckhow)**.** *Given a Frege proof of size s proving some formula φ of depth D, we can find another Frege proof of φ having size $s^{O(1)}$, where the size S_ℓ and D_ℓ of each line ℓ of the proof satisfy $D_\ell \leq D + O(\log S_\ell)$.*

In other words, in the new proof, all lines are balanced to the best extent possible given φ.

2 Terminology

Our formulas are expressed in terms of binary $\wedge$ and $\vee$ and unary $\neg$. We also allow the constants 0 and 1 (same as $\bot$ and $\top$). We think of formulas as trees. In our diagrams, a wiggly line between nodes P and Q means that Q is a descendant of P. Also, sometimes a node's name will double as the name of the subtree rooted at the node.

The *logical depth* of a formula is the largest number of alternations of connectives from root to leaf. For example, $(A \wedge B) \wedge (C \wedge D)$ has logical depth 1.

If Q is a descendant of P, we will use $P_{Q=x}$ to mean the subtree P, with its subtree Q replaced by x. Also, $|P|$ will denote the size of P (number of nodes).

We will consider Frege proof systems, which are Hilbert-style proof systems. The exact proof system used is immaterial, as long as it's complete, and its rules are closed under substitution. We will use the phrase *fixed proof* to mean a proof of some basic identity; the exact proof depends on the details of the proof system being used.

3 Lifting

The following two easy lemmas are two basic proof techniques.

The first lemma shows how to lift a proof of $\varphi(x_1, \ldots, x_n)$, where the x_i are variables, to a proof of $\varphi(X_1, \ldots, X_n)$, where the X_i are arbitrary expressions.

Lemma 3.1. *Suppose one can prove a formula φ using l lines, each of size at most s_1 and logical depth at most d_1. Let ψ be obtained from φ by substituting some arbitrary formula for each leaf of φ, the maximum size of a substituted formula being s_2, and their maximum logical depth d_2. Then one can prove ψ using l lines, each of size at most $s_1 s_2$ and logical depth at most $d_1 + d_2$.*

Proof. Replace each instance of a variable with the formula substituted for it. Since in Frege systems both axioms and inference rules are closed under substitution, the result is a valid proof of ψ. □

The second lemma shows how to lift a proof of $X \leftrightarrow Y$ to a proof of $\varphi(X) \leftrightarrow \varphi(Y)$, where $\varphi(\circ)$ is an arbitrary formula with a singled occurrence of a variable.

The proof idea is to use structural induction. For example, suppose that $\varphi(\circ) = (\neg \circ \wedge A) \vee B$. The proof of $\varphi(X) \leftrightarrow \varphi(Y)$ uses the following steps:

- Start from the given $X \leftrightarrow Y$.
- Deduce $(\neg X) \leftrightarrow (\neg Y)$.
- Deduce $(\neg X \wedge A) \leftrightarrow (\neg Y \wedge A)$.
- Conclude $((\neg X) \wedge A) \vee B) \leftrightarrow ((\neg Y) \wedge A) \vee B)$.

Each of the deduction steps requires only a fixed proof.

Lemma 3.2. *Let φ, ψ be arbitrary formulas, each of size at most s_1 and logical depth at most d_2. Let $\chi(\circ)$ be an arbitrary formula of size s_2 and logical depth d_2 with a distinguished input. Then one can deduce $\chi(\varphi) \leftrightarrow \chi(\psi)$ from $\varphi \leftrightarrow \psi$ using $O(s_2)$ lines, each of size at most $O(s_1 + s_2)$ and logical depth at most $d_1 + d_2 + O(1)$.*

Proof. Since our proof system is complete, it can derive from $P \leftrightarrow Q$ the formulas $(\neg P) \leftrightarrow (\neg Q)$ and $(P \Diamond R) \leftrightarrow (Q \Diamond R)$, where $\Diamond$ is either $\wedge$ or $\vee$ and R is arbitrary, all with fixed proofs. The lemma follows by structural induction on χ, as follows.

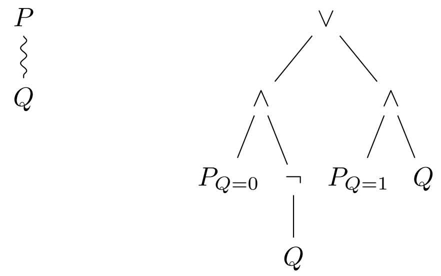
Let $\chi^i(\circ)$ be the unique subformula of χ which contains $\circ$ at depth i; thus $\chi^0(\circ) = \circ$ and $\chi^D = \chi$, where D is the depth of χ. Our starting point is the assumption $\chi^0(\varphi) \leftrightarrow \chi^0(\psi)$. We then derive inductively $\chi^i(\varphi) \leftrightarrow \chi^i(\psi)$ for $i \leq D$.

Since $D \leq s_2$, this requires $O(s_2)$ steps, each consisting of $O(1)$ lines. Since each step consists of a fixed proof, lemma 3.1 shows that the size of all the formulas is $O(|\chi^i(\varphi)| + |\chi^i(\psi)|) = O(s_1 + s_2)$, and that the logical depth is at most $d_1 + d_2 + O(1)$. □

4 Balancing

We will balance formulas using Spira's method. The method is based on the following basic transformation.

Lemma 4.1. *The formula on the left is equivalent to the formula on the right.*



We would like to choose the node Q so that both $P_{Q=0/1}$ and Q are small.

Lemma 4.2. *Every formula P which is not a single node has a subformula Q of size $|P|/3 \leq |Q| \leq 2|P|/3$.*

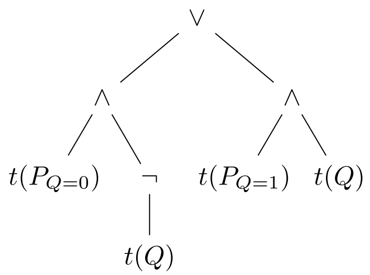
Proof. Starting with P, descend through the tree by always picking a child whose size is at least $|P|/3$, if possible. If not possible, the current node Q, having at most two children, has size at most $2|P|/3$. □

Spira's method applies the transformation 4.1 recursively, always choosing a node according to lemma 4.2. The resulting formula blows up only polynomially, and is balanced.

Definition 4.1. Spira's transformation t, transforming formulas to formulas, is defined recursively as follows:

- If P is an atom, then $t(P) = P$.

- Otherwise, choose a canonical subformula Q satisfying the requirements of lemma 4.2, and then $t(P)$ is given by the following formula:



Lemma 4.3. *Spira's transformation satisfies the following properties, for every formula P:*

(a) P is equivalent to $t(P)$.

(b) $|t(P)| = O(|P|^C)$ for some constant C.

(c) The depth of $t(P)$ is $O(\log |P|)$.

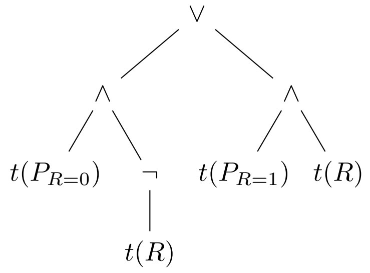
Proof. Spira's construction is applied recursively at most $\log_{3/2} |P|$ levels on a formula P, and so the depth of $t(P)$ is at most $3 \log_{3/2} |P|$. Since every node has arity at most 2, the number of nodes in $t(P)$ is less than $2^{3 \log_{3/2} |P|+1} = O(|P|^{3/ \log_2 \frac{3}{2}})$. □

5 Rebalancing

The main technical tool used to prove intensional comprehension is the operation of *rebalancing*, which replaces the top level subtree used to balance the formula in Spira's construction. The two formulas are clearly equivalent. It is less clear that we can prove this equivalence using a polynomial number of steps, all of which are balanced.

The impatient reader who wants to see the motivation behind rebalancing can peek at the rest of the proof, and return with fresh motivation.

Definition 5.1. Let P be a formula, and R one of its subformulas. The *rebalancing* of P by R, denoted $t(P/R)$, is



Notice that if Q is the node given by lemma 4.2, then $t(P/Q) = t(P)$.

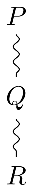
Lemma 5.1. *Let P be a formula, and R an arbitrary descendant of P. There is a Frege proof of $t(P) \leftrightarrow t(P/R)$ of size polynomial in $|P|$, all of whose lines have depth $O(\log |P|)$.*

Proof. The Frege proof is defined recursively. We first define the proof, and then analyze its properties. Let Q denote the node such that $t(P) = t(P/Q)$. If $Q = R$ then there is nothing to prove. Otherwise, Q and R can be related in three different ways:

1. R is a descendant of Q.
2. Q is a descendant of R.
3. Q and R are disjoint as subtrees.

We will deal with these cases separately; the first two will turn out to be virtually the same.

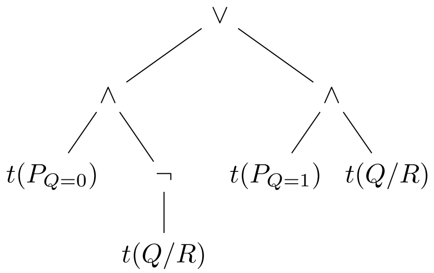
First case: R is a descendant of Q. Pictorially:

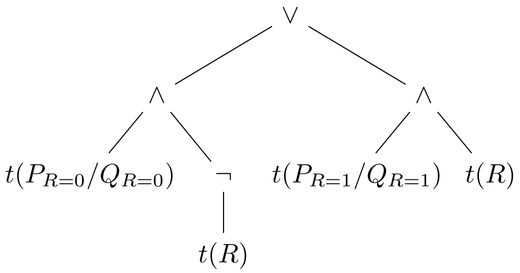


Our starting point is the following three Frege proofs:

1. $t(Q) \leftrightarrow t(Q/R)$.
2. $t(P_{R=0}) \leftrightarrow t(P_{R=0}/Q_{R=0})$.
3. $t(P_{R=1}) \leftrightarrow t(P_{R=1}/Q_{R=1})$.

We obtain these proofs recursively. Using lemma 3.2, we conclude that $t(P)$ is equivalent to the left-hand diagram, and that $t(P/R)$ is equivalent to the right-hand diagram:





If we open up all the rebalanced terms, all the leaves will be labeled by one of the following formulas:

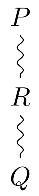
$$t(P_{Q=0}),\ t(P_{Q=1}),\ t(Q_{R=0}),\ t(Q_{R=1}),\ t(R);$$

the only non-obvious case is $t(P_{R=b}/Q_{R=b}) = t(P/Q_{R=b})$.

Notice that the formulas $t(P_{Q=b})$, $t(Q_{R=b})$ and $t(R)$ are completely independent: the first pair defines the stretch between P and Q, the second pair between Q and R, and the last defines R. Thus, the reason that the two formulas pictured above are equivalent doesn't have anything to do with any properties of these five formulas.

If we replace these five formulas by generic variables, we simply obtain two fixed propositional formulas which are equivalent. Since our proof system is complete, their equivalence can be proved using a fixed proof. Applying lemma 3.1, we conclude that there is a Frege proof of the equivalence of the two formulas pictured above using $O(1)$ steps of size $O(|P|)$. This concludes the description of the Frege proof in the first case.

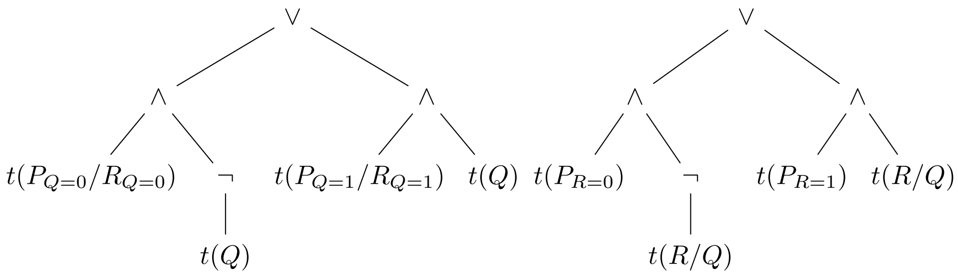
Second case: Q is a descendant of R. Pictorially:



This case is very similar to the first case. We first prove the following equivalences recursively:

1. $t(R) \leftrightarrow t(R/Q)$.
2. $t(P_{Q=0}) \leftrightarrow t(P_{Q=0}/R_{Q=0})$.
3. $t(P_{Q=1}) \leftrightarrow t(P_{Q=1}/R_{Q=1})$.

Using lemma 3.2, we lift these up to show that $t(P)$ and $t(P/R)$ are equivalent to the following formulas:

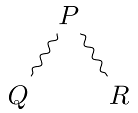


If we open up all the rebalanced terms, all the leaves will be labeled by one of the following formulas:

$$t(P_{Q=0}),\ t(P_{Q=1}),\ t(Q_{R=0}),\ t(Q_{R=1}),\ t(R).$$

Note that this list is the same as in the first case. As in the that case, since these two formulas are equivalent, we can prove this with a constant number of lines of size $O(|P|)$.

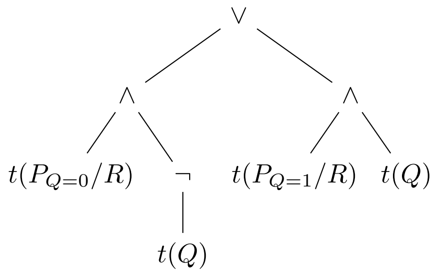
Third case: Q and R are disjoint subtrees. Pictorially:

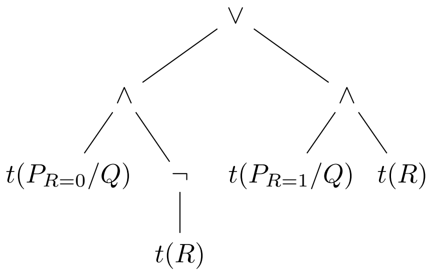


This time we have to prove four equivalences recursively:

1. $t(P_{Q=0}) \leftrightarrow t(P_{Q=0}/R)$.
2. $t(P_{Q=1}) \leftrightarrow t(P_{Q=1}/R)$.
3. $t(P_{R=0}) \leftrightarrow t(P_{R=0}/Q)$.
4. $t(P_{R=1}) \leftrightarrow t(P_{R=1}/Q)$.

Using lemma 3.2, we lift these rebalanced terms into the following equivalents of $t(P)$ and $t(P/R)$:





If we open up all the rebalanced terms, all the leaves will be labeled by one of the following formulas:

$$t(P_{Q=0,R=0}),\ t(P_{Q=0,R=1}),\ t(P_{Q=1,R=0}),\ t(P_{Q=1,R=1}),\ t(Q),\ t(R).$$

The derivation is completed as in the previous two cases.

Analysis of the resulting proof. A quick consideration of the proof shows that at all points, all nodes beyond some constant depth are expressed in the form $t(\cdot)$, and so all lines in the proof are balanced. Moreover, each such line contains at most a constant number of terms of the form $t(S)$, where S is always a subformula of P. Thus all the lines have depth $O(\log|P|)$ and size polynomial in $|P|$. It remains to show that the number of lines is also polynomial in $|P|$.

Let $\ell(P, R)$ denote the number of lines required to prove that $t(P) \leftrightarrow t(P/R)$. The three different cases correspond to the following recurrence relations:

1. $\ell(P,R) = \ell(Q,R) + \ell(P_{R=0}, Q_{R=0}) + \ell(P_{R=1}, Q_{R=1}) + O(|P|)$.
2. $\ell(P,R) = \ell(R,Q) + \ell(P_{Q=0}, R_{Q=0}) + \ell(P_{Q=1}, R_{Q=1}) + O(|P|)$.
3. $\ell(P,R) = \ell(P_{Q=0}, R) + \ell(P_{Q=1}, R) + \ell(P_{R=0}, Q) + \ell(P_{R=1}, Q) + O(|P|)$.

The base case is $\ell(P,Q) = 0$ if Q is the node singled out by lemma 4.2.

In order to bound $\ell(P,R)$, we consider the value $L(n,m)$, which denotes $\max \ell(P,R)$ over all formulas P and R satisfying $|P| \le n$ and $n - m \le |R| \le m$. Plugging in the previous inequalities, we get the following recurrence relation:

$$L(n,m) = O(n) + \\ \max\left[L(\tfrac{2}{3}n, \tfrac{2}{3}n) + 2L(m, \tfrac{2}{3}n), L(m, \tfrac{2}{3}n) + 2(\tfrac{2}{3}n, \tfrac{2}{3}n), 2L(\tfrac{2}{3}n, \tfrac{2}{3}n) + 2L(m, \tfrac{2}{3}n)\right].$$

If we substitute the recurrence relation into $L(m, \frac{2}{3}n)$, we find out that

$$L(n,n) = O(n) + 10L(\tfrac{2}{3}n, \tfrac{2}{3}n),$$

The solution of this recurrence relation is $L(n,n) = O(n^{\log_{3/2} 10})$. We conclude that the number of lines in the proof is polynomial in $|P|$. □

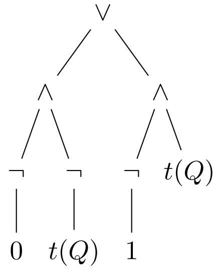
6 Intensional comprehension

Using lemma 5.1, we can prove lemmas expressing *intensional comprehension*. These lemmas show that we can express the meanings of connectives (*comprehend* them) using *internal* means, i.e. with balanced proofs.

Comprehension of $\neg$ is easiest.

Lemma 6.1. *For every formula Q, there is a Frege proof of $t(\neg Q) \leftrightarrow \neg t(Q)$ consisting of $|Q|^{O(1)}$ lines, each of depth $O(\log |Q|)$.*

Proof. Let $P = \neg Q$. Using lemma 5.1, we can prove $t(P) \leftrightarrow t(P/Q)$. The formula $t(P/Q)$, written out, is



It is straightforward to prove that this is equivalent to $\neg t(Q)$. □

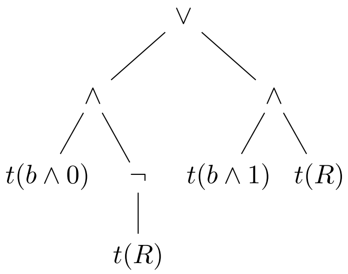
In order to prove comprehension of the binary operators, we must proceed in two steps.

Lemma 6.2. *For every two formulas Q, R, there is a Frege proof of $t(Q \wedge R) \leftrightarrow t(Q) \wedge t(R)$ consisting of $(|Q| + |R|)^{O(1)}$ lines, each of depth $O(\log(|Q| + |R|))$.*

Proof. Let $P = Q \wedge R$. Before we can prove the actual statement, we need to prove two auxiliary results:

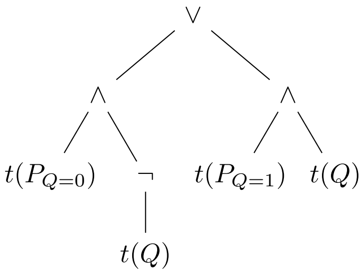
1. $t(P_{Q=0}) \leftrightarrow 0$.
2. $t(P_{Q=1}) \leftrightarrow t(R)$.

These follow by rebalancing $P_{Q=b}$ on R, resulting in the following formulas:

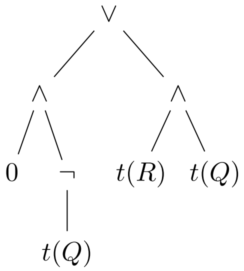


The two auxiliary results can be proved easily, since $t(b \wedge 0)$ and $t(b \wedge 1)$ are just constant formulas.

Next, we rebalance the actual formula P on Q:



Using lemma 3.2, we can substitute the auxiliary results to get



The lemma now easily follows. □

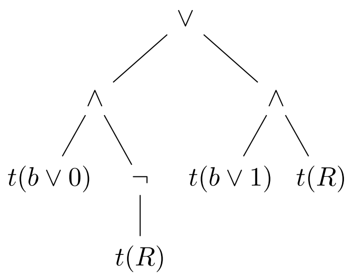
Comprehension of the other binary operator is very similar.

Lemma 6.3. *For every two formulas Q, R, there is a Frege proof of $t(Q \vee R) \leftrightarrow t(Q) \vee t(R)$ consisting of $(|Q| + |R|)^{O(1)}$ lines, each of depth $O(\log(|Q| + |R|))$.*

Proof. Let $P = Q \vee R$. First, we prove two auxiliary results:

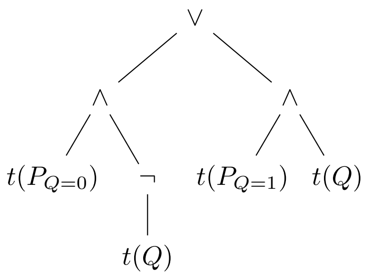
1. $t(P_{Q=0}) \leftrightarrow t(R)$.
2. $t(P_{Q=1}) \leftrightarrow 1$.

These follow by rebalancing $P_{Q=b}$ on R:

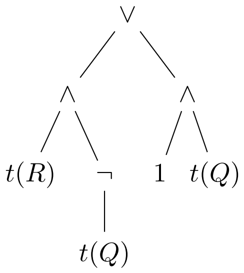


The auxiliary results follow easily.

Next, we rebalance the actual formula P on Q:



Substituting the auxiliary results, this leads to



The lemma now easily follows. □

Combining the preceding lemmas, we obtain the following general form of internal comprehension.

Lemma 6.4. *Let P be an arbitrary formula of depth D with n parameters. For every n formulas Q_i, Frege proves $t(P(Q_1, \ldots, Q_n)) \leftrightarrow P(t(Q_1), \ldots, t(Q_n))$ with $M^{O(1)}$ steps, each of depth $D + O(\log M)$, where $M = |P| + \sum |Q_i|$.*

Proof. By structural induction, using lemmas 6.1, 6.2 and 6.3. □

7 Conclusion of the proof

The main idea of the proof of Theorem 1.1 is to replace each formula by its balanced form. All the formulas become balanced, yet the proof is no longer valid. Also, instead of proving the original formula, we prove its balanced form. We address the former issue first.

Lemma 7.1. *Consider a rule of the Frege proof system, which given the premises $P_j(x_1, \ldots, x_n)$ concludes $Q(x_1, \ldots, x_n)$ (if there are no premises, then this is an axiom). For every n formulas R_i, Frege concludes $t(Q(R_1, \ldots, R_n))$ from $t(P_j(R_1, \ldots, R_n))$ with $M^{O(1)}$ steps, each of depth $O(\log M)$, where $M = \sum |P_j| + |Q| + \sum |R_i|$.*

Proof. Given each $t(P_j(R_1, \ldots, R_n))$, use lemma 6.4 to conclude $P_j(t(R_1), \ldots, t(R_n))$. Then use the original rule to conclude $Q(t(R_1), \ldots, t(R_n))$. Finally, use lemma 6.4 again to conclude $t(Q(R_1), \ldots, Q(R_n))$. □

We are almost done.

Proof of Theorem 1.1. Replace each axiom and each derivation rule of the original proof with an application of lemma 7.1 to get a proof of $t(\varphi)$. In order to conclude φ, use lemma 6.4 with the variables of φ as Q_i. □

References

[1] Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory.* Cambridge University Press, New York, NY, USA, 1995.

[2] Robert A. Reckhow. *On the Lengths of Proofs in the Propositional Calculus.* PhD thesis, University of Toronto, Department of Computer Science, 1976.