algorithms

MDPI

Article

Mathematical Foundation of a Functional Implementation of the CNF Algorithm

Francisco Miguel García-Olmedo [1,*,†], Jesús García-Miranda [1,†] and Pedro González-Rodelas [2,*,†]

[1] Departamento de Álgebra, Granada University, 18071 Granada, Spain; jesusgm@ugr.es
[2] Departamento de Matemática Aplicada, Granada University, 18071 Granada, Spain
* Correspondence: folmedo@ugr.es (F.M.G.-O.); prodelas@ugr.es (P.G.-R.)
† These authors contributed equally to this work.

Abstract: The conjunctive normal form (CNF) algorithm is one of the best known and most widely used algorithms in classical logic and its applications. In its algebraic approach, it makes use in a loop of a certain well-defined operation related to the "distributivity" of logical disjunction versus conjunction. For those types of implementations, the loop iteration runs a comparison between formulas to decide when to stop. In this article, we explain how to pre-calculate the exact number of loop iterations, thus avoiding the work involved in the above-mentioned comparison. After that, it is possible to concatenate another loop focused now on the "associativity" of conjunction and disjunction. Also for that loop, we explain how to calculate the optimal number of rounds, so that the decisional comparison phase for stopping can be also avoided.

Keywords: CNF; SAT problem; classical logic; Boolean algebra; Horn clauses; reverse engineering; automatic theorems proving; algorithm implementation; functional programming; Haskell

Citation: García-Olmedo, F.M.; García-Miranda, J.; González-Rodelas, P. Mathematical Foundation of a Functional Implementation of the CNF Algorithm. *Algorithms* **2023**, *16*, 459. https://doi.org/10.3390/a16100459

Academic Editors: Eugene Semenkin, Todor Ganchev, Predrag S. Stanimirovic and Frank Werner

Received: 14 August 2023
Revised: 24 September 2023
Accepted: 24 September 2023
Published: 27 September 2023

1. Introduction

The conjunctive normal form (CNF) is famous in the history of thought because it organizes discourse by normalizing potentially chaotic statements through the conjunction of a series of statements in the form of disjunctions of other simple statements, namely atomic statements, or the negation of them.

CNF has proven to be essential in the treatment of the SAT problem (from "satisfiability", usually abbreviated to SAT), which in turn is at the core of automated theorem proving, thanks to the effectiveness of the resolution rule and what is known about the treatment of Horn clauses (see [1]).

The other great utility of CNF lies in the minimization of Boolean expressions under the condition of being expressed as a product of sums (POS) (see [2,3]). It is clear that the POS criterion is the dual concept of the sum of products (SOP) criterion.

In general, transforming a formula into CNF is the essence of Petrick's method, an algorithm widely used in various fields: cybernetics, economics, linguistics, philosophy, psychology, etc. For an explanation of the algorithm and detailed applications, see [2] (p. 157), plus extensive comments and applications of the CNF algorithm. In [4] (pp. 69–71), we find a brilliant application of Petrick's method to finite Boolean algebra.

Given a propositional formula or a Boolean expression, it is possible to obtain for it an equivalent formula or expression, as the case may be, in CNF by semantic means or by algebraic manipulations. Both procedures are described in [5–7]; however, algebraic manipulation may be faster.

If we focus on the method of syntactic analysis for obtaining CNF, we will consider propositional logic formulas, without loss of generality, in the appropriate language. In this line of thought, the essence of the algorithm is quite simple: after internalizing negation and eliminating double negation, the main task is to replace subformulas of

the form $(\alpha \vee (\beta \wedge \gamma))$ by their equivalent $((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$ (distributivity). For the sake of efficiency, we will use Polish notation here as was introduced by Jan Łukasiewicz in [8] (pp. 33–34). Łukasiewicz selected K (resp. A) to represent conjunction (resp. disjunction). Disjunction was originally called "alternation" by Łukasiewicz (in Polish "alternacja", hence the symbol "A"). The word "disjunction" in Polish is "koniunkcja", hence the symbol "K". Our notation in this logical work, which is classical notation, is based on this Łukasiewicz guideline because it has the great advantage of avoiding parentheses and at the same time being univocal. Furthermore, Polish notation is ideal for transitioning to a functional implementation in the Haskell language given the peculiarities of its syntax (see [9]). Since de functor A stands for disjunction and functor K for conjunction in standard Polish notation, what we are saying is that the application of distributivity translates $\mathrm{A}\,\alpha\,\mathrm{K}\,\beta\gamma$ to $\mathrm{K}\,\mathrm{A}\,\alpha\beta\,\mathrm{A}\,\alpha\gamma$.

With regard to the CNF algorithm, the current state of the art can be found, for example, at [5,6]. To fix ideas, we will focus on [5] (p. 26) and look at Algorithm 2.3, called `TREE-EQ-CNF`. Its pseudocode contains the following snippet:

```
...
{
  do {
    α_orig = α
    ...
  } while (α_orig != α);
  return(α)
}
```

Therefore, we note that the basis of the algorithm is a `while` loop with the sentinel condition $\alpha_{orig} \neq \alpha$, where α_{orig} is a copy of the value of the α at the start of the current round. Therefore, each round of the `while` loop requires a comparison operation between the formula we are transforming in that round and the result of its transformation, stopping the process when there is a match. The basis of this work is to avoid all the comparisons we have just detected, the number of which will depend on the case, and instead to carry out a single inspection of the formula passed as a parameter initially to the coding of the algorithm; that single inspection, ultimately related to "distributivity", will give us the minimum number of rounds for the correct operation of the loop. However, the expression of the resulting formula in the CNF algorithm is not satisfactory until we express it canonically with the functors K and A left-loaded. This has to do with "associativity" and will lead us a second time to the situation where we have to calculate the minimum number of rounds for another `while` loop beforehand; that calculation will be carried out with the function akr defined below in Section 3.

The basis of the classical algorithm, which is essentially the substitution of subformulas by equivalents, does not change in our work, but we will provide a very formal presentation of the substitution process in the algorithm. That presentation will be based on the specialized theory of recursive works, namely the lambda calculus.

Finally, as a result of the work, we provide a functional implementation (https://github.com/ringstellung/CNF, accessed on 23 September 2023) of the algorithm in the Haskell programming language with these and other contributions.

In summary, the sections of this article contain the following. Since the aim of this paper is the manipulation of formulas over a language, in Section 2, we suggest the rigorous definition of language and formula. Four subsets of formulas generated by a non-empty subset of formulas according to appropriate rules are then suggested; essentially they are the support for defining the concept of clause and formula in conjunctive normal forms. The section continues by giving several different concepts of the complexity of a formula. The rest of the section is devoted to defining the concept of semantic equivalence of formulas, according to classical propositional logic; some examples; and the statement of a classical result on distributivity. Negation is not mentioned because it is not relevant to the theoretical framework of the article. Section 3 is devoted to the treatment of the

distributivity that, in a broad sense of the term, classical propositional logic contains between disjunction and conjunction. In that section, the function dak is defined (see Figure 1), which, when applied to formulas, manages to reduce the alternation in them of the connector K over A in a unit; this alternation is measured in each formula by the function alt. The section concludes by showing that the alternation of each formula is precisely the minimum number of applications of dak to it, in order to obtain another equivalent formula in conjunctive normal form. Practical laboratory experience in the field of logical deduction indicates that the associativity of the connectives K and A must be taken into account in order to obtain a canonical form, which, in Polish notation, accumulates these connectives at the beginning of the formula. The rigorous treatment of this matter is the aim of Section 4. In writing it, we were inspired by the structure of Section 3; however, the intrinsic theory is appreciably more complex. It is all based on two "measures" on formulas that essentially express how far the formula is from that canonical form; the maximum of these two values is also important. In that section, we justify the expressive capacity of the aforementioned measures to characterise the membership of the subsets of formulas defined in Section 2. The counterpart for the associativity of the dak function is defined, namely the lasc function (see Figure 2). This time, the application of the lasc function decreases the separation measure of the formula to the canonical form by half; in that sense, it serves to make big steps. A certain natural value based on the logarithm in base two gives the minimum number of applications of lasc to the formula to obtain its desired canonical form. The last section, Section 5, is for the conclusions.

$$x_i \to_{\mathrm{dak}} x_i$$

$$\frac{A\varphi\xi \to_{\mathrm{dak}} \alpha \quad A\psi\xi \to_{\mathrm{dak}} \beta}{AK\varphi\psi\xi \to_{\mathrm{dak}} K\alpha\beta} \qquad \frac{A\xi\varphi \to_{\mathrm{dak}} \alpha \quad A\xi\psi \to_{\mathrm{dak}} \beta}{A\xi K\varphi\psi \to_{\mathrm{dak}} K\alpha\beta}$$

$$\frac{\varphi \to_{\mathrm{dak}} \varphi' \quad \psi \to_{\mathrm{dak}} \psi'}{A\varphi\psi \to_{\mathrm{dak}} A\varphi'\psi'} \qquad \frac{\varphi \to_{\mathrm{dak}} \varphi' \quad \psi \to_{\mathrm{dak}} \psi'}{K\varphi\psi \to_{\mathrm{dak}} K\varphi'\psi'}$$

Figure 1. Axiom and rules for the definition of dak.

$$x_i \to_{\mathrm{lasc}} x_i$$

$$\frac{A\varphi\psi \to_{\mathrm{lasc}} \alpha \quad \xi \to_{\mathrm{lasc}} \beta}{A\varphi A\psi\xi \to_{\mathrm{lasc}} A\alpha\beta} \qquad \frac{K\varphi\psi \to_{\mathrm{lasc}} \alpha \quad \xi \to_{\mathrm{lasc}} \beta}{K\varphi K\psi\xi \to_{\mathrm{lasc}} K\alpha\beta}$$

$$\frac{\varphi \to_{\mathrm{lasc}} \varphi' \quad \psi \to_{\mathrm{lasc}} \psi'}{A\varphi\psi \to_{\mathrm{lasc}} A\varphi'\psi'} \qquad \frac{\varphi \to_{\mathrm{lasc}} \varphi' \quad \psi \to_{\mathrm{lasc}} \psi'}{K\varphi\psi \to_{\mathrm{lasc}} K\varphi'\psi'}$$

Figure 2. Axiom and rules for the definition of lasc.

2. Basic Definitions and Preliminary Results

In this section, we list the main definitions of the basic concepts that will be used in the development of this work, as well as the essential results needed in it.

Here, the concept of sentencial or propositional language is that of J.D. Monk in [10] (see also [11,12]), but our functors (in the sense of Jan Łukasiewicz in [8]) will be A and K. Moreover, it is necessary to impose that X, the set of atomic or propositional variables, be a nonfinite numerable set; its elements are notated by the last lower–case letters of the latin alphabet: x, y, z, etc., subindicating them if necessary. We will call $\mathbf{L_{KA}}$ the above

propopositional language and $\mathrm{P}(\mathbf{L_{KA}})$, or simply $\mathrm{P}(\mathbf{L})$, the set of its formulas or sentences. The reader should be thoroughly familiar with the principle of induction for sentences, the construction sequence for sentences, and the unique readability principle. We will also assume the knowledge of the principle of finite induction in its different formulations, as it is exposed, for example, in [6].

The statements of some particular results (lemmas and theorems), in this and subsequent sections, will be given without complete proofs, because some of them are quite evident, and the rest could be carried out using the principle of finite induction, taking into account, in a meticulous and orderly manner, all the possible cases that can occur, in a similar way as it is carried out in the selected proofs included in this paper.

Given a set Δ of formulas of the language $\mathbf{L}$, consider the smallest set of formulas that, containing it, is at the same time closed for the functors A (resp. K); it will be represented by $\mathrm{A}(\Delta)$ (resp. $\mathrm{K}(\Delta)$). Thus, the elements of $\mathrm{K}(\mathrm{A}(X))$ are exactly the set of formulas in *conjunctive normal form*. *Clauses* are exactly the formulas of the smallest set, $\mathrm{al}(X)$, containing X and being closed for the functor A whenever it operates on an element of X to its right. The set $\mathrm{kl}(X)$ (resp. $\mathrm{lcnf}(X)$) is the smallest set containing X (resp. $\mathrm{al}(X)$) and being closed for the functor K whenever it operates on an element of X ($\mathrm{al}(X)$) to its right.

Remark 1. *Note that* $\mathrm{al}(X) \cup \mathrm{kl}(X) \subseteq \mathrm{lcnf}(X)$ *and that* $\mathrm{lcnf}(X) \subseteq \mathrm{K}(\mathrm{A}(X))$.

Definition 1. *For all* $\alpha \in \mathrm{P}(\mathbf{L_{KA}})$ *let* $\mathrm{comp}(\alpha)$ *(complexity), the natural value defined as follows:*

$$\mathrm{comp}(\alpha) = \begin{cases} 0, & \text{if } \alpha \in X, \\ 1 + \mathrm{comp}(\varphi) + \mathrm{comp}(\psi), & \text{if } \alpha \equiv A\varphi\psi \text{ or } \alpha \equiv K\varphi\psi. \end{cases}$$

and $\mathrm{comp_k}(\alpha)$ *(complexity in* K*) defined by*

$$\mathrm{comp_k}(\alpha) = \begin{cases} 0, & \text{if } \alpha \in X, \\ \mathrm{comp_k}(\varphi) + \mathrm{comp_k}(\psi), & \text{if } \alpha \equiv A\varphi\psi, \\ 1 + \mathrm{comp_k}(\varphi) + \mathrm{comp_k}(\psi), & \text{if } \alpha \equiv K\varphi\psi. \end{cases}$$

If necessary, consider $\mathrm{comp_a}(\alpha)$ *as the dual concept of* $\mathrm{comp_k}(\alpha)$.

Consider the semantic consequence $\models$ in the classical sense, as in, for example, Ref. [5]. The formula α and β are *equivalent*, in symbols $\alpha = \beta$, iff by definition $\alpha \models \beta$ and $\beta \models \alpha$. In the following, we will use the symbol $=$ to indicate equivalence and the symbol $\equiv$ to indicate syntactic equality (verbatim equal). It is well known that if $\alpha = \mathrm{A}\,\varphi\psi$ and $\beta = \mathrm{A}\,\varphi\xi$, then $\mathrm{A}\,\varphi\,\mathrm{K}\,\psi\xi = \mathrm{K}\,\alpha\beta$.

Remark 2. *It is well know as a basic theorem of classical logic that if* $\alpha = A\varphi\psi$ *(resp.* $\alpha = K\varphi\psi$*), then* $A\varphi A\psi\xi = A\alpha\xi$ *(resp.* $K\varphi K\psi\xi = K\alpha\xi$*).*

3. Distributivity

The aim now is, given a formula φ in $\mathrm{P}(\mathbf{L})$, to find φ_{cnf} in $K(\mathrm{A}(X))$ such that both are equivalent. This will be the basis of the algorithm, and the first thing to do is to determine how far φ is from $\mathrm{K}(\mathrm{A}(X))$. As we shall see shortly, the expected measure is given by the function *alt*, which indicates the alternation in its formula argument of the symbols K and A from the inner to the outside formula.

Definition 2. *Let* α *be any formula in* $\mathrm{P}(\mathbf{L})$. *The alternation of* α*, in symbols* $\mathrm{alt}(\alpha)$*, is defined by*

$$\mathrm{alt}(\alpha) = \begin{cases} 0, & \text{if } \mathrm{K} \notin \alpha; \\ \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\psi)\}, & \text{if } \alpha \equiv \mathrm{K}\,\varphi\psi; \\ 1 + \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\psi)\}, & \text{if } \alpha \equiv \mathrm{A}\,\varphi\psi \text{ and } \mathrm{K} \in \alpha. \end{cases}$$

In Lemma 1, we characterise the meaning of "belong to the set $\mathrm{K}(\mathrm{A}(X))$" by means of the map alt. As we shall see, the formulas for which $\mathrm{alt}(\alpha) = 0$ are exactly those of the set $\mathrm{K}(\mathrm{A}(X))$.

Lemma 1. *Let $\alpha \in \mathrm{P}(\mathbf{L})$. The following statements are equivalent:*

1. $\mathrm{alt}(\alpha) = 0$.
2. $\alpha \in \mathrm{K}(\mathrm{A}(X))$.

Definition 3 (distributivity). *Consider the following compound rules on binary relationships between formulas of $P(\mathbf{L})$:*

$$x_i \to_{\mathrm{dak}} x_i \tag{1}$$

$$\frac{\mathrm{A}\,\varphi\xi \to_{\mathrm{dak}} \alpha \quad \mathrm{A}\,\psi\xi \to_{\mathrm{dak}} \beta}{\mathrm{A}\,\mathrm{K}\,\varphi\psi\xi \to_{\mathrm{dak}} \mathrm{K}\,\alpha\beta} \tag{2}$$

$$\frac{\mathrm{A}\,\xi\varphi \to_{\mathrm{dak}} \alpha \quad \mathrm{A}\,\xi\psi \to_{\mathrm{dak}} \beta}{\mathrm{A}\,\xi\,\mathrm{K}\,\varphi\psi \to_{\mathrm{dak}} \mathrm{K}\,\alpha\beta} \tag{3}$$

$$\frac{\varphi \to_{\mathrm{dak}} \varphi' \quad \psi \to_{\mathrm{dak}} \psi'}{\mathrm{A}\,\varphi\psi \to_{\mathrm{dak}} \mathrm{A}\,\varphi'\psi'} \tag{4}$$

$$\frac{\varphi \to_{\mathrm{dak}} \varphi' \quad \psi \to_{\mathrm{dak}} \psi'}{\mathrm{K}\,\varphi\psi \to_{\mathrm{dak}} \mathrm{K}\,\varphi'\psi'} \tag{5}$$

where (2)–(4) are applied with the precedence indicated by the order in which they are given. This being so, we define the following application:

$$\mathrm{dak}\colon \mathrm{P}(\mathbf{L}) \longrightarrow \mathrm{P}(\mathbf{L})$$

by

$$\mathrm{dak}(\varphi) \equiv \psi, \textit{ provided that } \varphi \to_{\mathrm{dak}} \psi$$

Remark 3. dak *is an application, because a clear and univocal precedence has been established in the rules on which its definition is based. For example, note that by Rule (2),* $\mathrm{dak}(\mathrm{A}\,\mathrm{K}\,xyz) = \mathrm{K}\,\mathrm{dak}(\mathrm{A}\,xz)\,\mathrm{dak}(\mathrm{A}\,yz)$; dak *is a recursive process with stop in propositional variables due to Rule (1). Moreover, it is clear from Remark 2 that for all $\zeta \in \mathrm{P}(\mathbf{L})$, $\mathrm{dak}(\zeta) = \zeta$ (semantical equality).*

By Lemma 2, $\alpha \in \mathrm{A}(X)$ is a sufficient condition for $\mathrm{dak}(\alpha) \equiv \alpha$, but it is not a necessary condition. Lemma 3, which is a consequence of Lemma 2, gives the necessary and sufficient condition, although this will be fully concluded in Corollary 1. The proof of Lemma 2 is straighforward.

Lemma 2. *Let α be a formula in $\mathrm{P}(\mathbf{L})$. If $\alpha \in \mathrm{A}(X)$, then $\mathrm{dak}(\alpha) \equiv \alpha$.*

Lemma 3. *Let $\alpha \in \mathrm{P}(\mathbf{L})$. If $\alpha \in \mathrm{K}(\mathrm{A}(X))$ then $\mathrm{dak}(\alpha) \equiv \alpha$.*

Proof. Let us assume that $\alpha \in \mathrm{K}(\mathrm{A}(X))$ and $\mathrm{comp}(\alpha) = n$. Reasoning by induction on $\mathrm{comp}(\alpha)$ we will show that $\mathrm{dak}(\alpha) \equiv \alpha$. As an induction hypothesis, suppose that the implication is true for any formula $\beta \in \mathrm{K}(\mathrm{A}(X))$ such that $\mathrm{comp}(\beta) < n$. If $\alpha \in \mathrm{K}(\mathrm{A}(X))$, then two situations are possible:

1. $\alpha \in \mathrm{A}(X)$; in this case $\mathrm{dak}(\alpha) \equiv \alpha$ is what Lemma 2 states.

2. There exist formulas φ and ψ in $\mathrm{K}(\mathrm{A}(X))$ of complexities lower than those of α such that $\alpha \equiv K\varphi\psi$. For the calculation of $\mathrm{dak}(\alpha)$, it is only possible to start with Rule (5):

$$\begin{aligned} \mathrm{dak}(\alpha) &\equiv \mathrm{dak}(K\varphi\psi) & \\ &\equiv K\,\mathrm{dak}(\varphi)\,\mathrm{dak}(\psi) & \text{by Rule (5)} \\ &\equiv K\varphi\psi & \text{induc. hyp.} \\ &\equiv \alpha & \end{aligned}$$

□

As for Theorem 1, in essence its meaning is that in applying dak to a given formula not in $\mathrm{K}(\mathrm{A}(X))$, say α, according to the "measure" alt, the result is closer to $\mathrm{K}(\mathrm{A}(X))$ than α.

Theorem 1. *For all* $\alpha \in \mathrm{P}(\mathbf{L})$,

$$\mathrm{alt}(\mathrm{dak}(\alpha)) = \begin{cases} 0, & \text{if } \alpha \in \mathrm{K}(\mathrm{A}(X)); \\ \mathrm{alt}(\alpha) - 1, & \text{otherwise.} \end{cases} \tag{6}$$

Proof. The proof is by induction on the complexity of the formula α. Let α be a formula in $\mathrm{P}(\mathbf{L})$ such that $\mathrm{comp}(\alpha) = n$. Suppose, as an induction hypothesis, that (6) holds for any formula β such that $\mathrm{comp}(\beta) < n$. Several cases are possible:

1. $\alpha \equiv x \in X$; in this case, $\mathrm{dak}(\alpha) \equiv x \in X$ and since $X \subseteq \mathrm{K}(\mathrm{A}(X))$, we deduce according to Lemma 1 that $\mathrm{alt}(\alpha) = 0$, which proves the result in this case.
2. $\alpha \equiv AK\varphi\psi\xi$; therefore, $\alpha \notin \mathrm{K}(\mathrm{A}(X))$ and

$$\begin{aligned} \mathrm{alt}(\alpha) &= 1 + \max\{\mathrm{alt}(K\varphi\psi), \mathrm{alt}(\xi)\} \\ &= 1 + \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\psi), \mathrm{alt}(\xi)\} \end{aligned} \tag{7}$$

On the other hand, $\mathrm{dak}(\alpha) \equiv K\,\mathrm{dak}(A\varphi\xi)\,\mathrm{dak}(A\psi\xi)$ so that

$$\mathrm{alt}(\mathrm{dak}(\alpha)) = \max\{\mathrm{alt}(\mathrm{dak}(A\varphi\xi)), \mathrm{alt}(\mathrm{dak}(A\psi\xi))\} \tag{8}$$

For short, we will call β to $A\varphi\xi$ and γ to $A\psi\xi$. Let us bear in mind the following:

(a) $K \in \beta$; then $\mathrm{alt}(\beta) = 1 + \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\xi)\}$ and $\beta \notin \mathrm{K}(\mathrm{A}(X))$. Since $\mathrm{comp}(\beta) < \mathrm{comp}(\alpha)$, the induction hypothesis allows us to establish that:

$$\begin{aligned} \mathrm{alt}(\mathrm{dak}(\beta)) &= \mathrm{alt}(\beta) - 1 & \\ &= 1 + \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\xi)\} - 1 & (9) \\ &= \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\xi)\} & (10) \end{aligned}$$

(b) $K \notin \beta$; then $\varphi, \xi, \beta \in \mathrm{A}(X)$. According to Lemma 2, then $\mathrm{dak}(\beta) \equiv \beta$ and, according to Lemma 1,

$$\begin{aligned} \mathrm{alt}(\mathrm{dak}(\beta)) = \mathrm{alt}(\beta) &= 0 & (11) \\ \mathrm{alt}(\varphi) &= 0 & (12) \\ \mathrm{alt}(\xi) &= 0 & (13) \end{aligned}$$

We will now analyse equality (8) on a case-by-case basis:

(a) $K \in \beta$ and $K \in \gamma$; then

$$\begin{aligned} \mathrm{alt}(\mathrm{dak}(\alpha)) &= \max\{\mathrm{alt}(\mathrm{dak}(\beta)), \mathrm{alt}(\mathrm{dak}(\gamma))\} & \text{by (8)} \\ &= \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\psi), \mathrm{alt}(\xi)\} & \text{by (10)} \\ &= \mathrm{alt}(\alpha) - 1 & \end{aligned}$$

(b) $K \notin \beta$ and $K \in \gamma$; then

$$\begin{aligned}
\mathrm{alt}(\mathrm{dak}(\alpha)) &= \max\{\mathrm{alt}(\mathrm{dak}(\beta)), \mathrm{alt}(\mathrm{dak}(\gamma))\} && \text{by (8)}\\
&= \max\{0, \mathrm{alt}(\mathrm{dak}(\gamma))\} && \text{by (11)}\\
&= \mathrm{alt}(\mathrm{dak}(\gamma))\\
&= \max\{\mathrm{alt}(\psi), \mathrm{alt}(\xi)\} && \text{by (10)}\\
&= \mathrm{alt}(\psi) && \text{by (13)}\\
&= \max\{0, \mathrm{alt}(\psi), 0\}\\
&= \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\psi), \mathrm{alt}(\xi)\} && \text{by (11) and (13)}\\
&= \mathrm{alt}(\alpha) - 1 && \text{by (7)}
\end{aligned}$$

(c) $K \in \beta$ and $K \notin \gamma$; this situation is treated as the case in paragraph 2b.

(d) $K \notin \beta$ and $K \notin \gamma$; in this case $\beta, \gamma \in \mathrm{A}(X)$ and $\alpha \in \mathrm{K}(\mathrm{A}(X))$. By what Lemma 3 states, $\mathrm{dak}(\alpha) \equiv \alpha$ and, as Lemma 1 states, $\mathrm{alt}(\alpha) = 0$; so $\mathrm{alt}(\mathrm{dak}(\alpha)) = 0$.

3. $\alpha \equiv A\xi K\varphi\psi$; this situation is treated as the case in paragraph 2.
4. $\alpha \equiv A\varphi\psi$; neither φ nor ψ begin with K but $K \in \alpha$; without loss of generality, suppose that $\mathrm{alt}(\psi) \leq \mathrm{alt}(\varphi)$, whence $K \in \varphi$ and φ begins with A, i.e., $\varphi \notin \mathrm{K}(\mathrm{A}(X))$. Then $\mathrm{alt}(\alpha) = 1 + \mathrm{alt}(\varphi)$ and

$$\begin{aligned}
\mathrm{alt}(\mathrm{dak}(\alpha)) &= \mathrm{alt}(A\,\mathrm{dak}(\varphi)\,\mathrm{dak}(\psi)) && \text{by Rule (4)}\\
&= 1 + \max\{\mathrm{alt}(\mathrm{dak}(\varphi)), \mathrm{alt}(\mathrm{dak}(\psi))\}\\
&= 1 + \mathrm{alt}(\mathrm{dak}(\varphi))\\
&= 1 + \mathrm{alt}(\varphi) - 1 && \text{hyp. induc. and conditions of } \varphi\\
&= \mathrm{alt}(\varphi)\\
&= \mathrm{alt}(\alpha) - 1
\end{aligned}$$

5. $\alpha \equiv K\varphi\psi$; without loss of generality, suppose that $\mathrm{alt}(\psi) \leq \mathrm{alt}(\varphi)$. If $\mathrm{alt}(\varphi) = 0$, then $\mathrm{alt}(\psi) = 0$, $\varphi, \psi, \alpha \in \mathrm{K}(\mathrm{A}(X))$ and so $\mathrm{alt}(\alpha) = 0$ (see Lemma 1). If $\mathrm{alt}(\varphi) \neq 0$, i.e., $\varphi \notin \mathrm{K}(\mathrm{A}(X))$, then $\alpha \notin \mathrm{K}(\mathrm{A}(X))$. Thus,

$$\begin{aligned}
\mathrm{alt}(\mathrm{dak}(\alpha)) &= \mathrm{alt}(K\,\mathrm{dak}(\varphi)\,\mathrm{dak}(\psi)) && \text{by Rule (5)}\\
&= \max\{\mathrm{alt}(\mathrm{dak}(\varphi)), \mathrm{alt}(\mathrm{dak}(\psi))\} && \text{def. of alt}\\
&= \mathrm{alt}(\mathrm{dak}(\varphi))\\
&= \mathrm{alt}(\varphi) - 1 && \text{induc. hyp. and conditions of } \varphi\\
&= \max\{\mathrm{alt}(\varphi), \mathrm{alt}(\psi)\} - 1\\
&= \mathrm{alt}(\alpha) - 1 && \text{def. of alt}
\end{aligned}$$

□

As a consequence of Theorem 1, it follows that the sufficient condition of Lemma 3 is also a necessary condition.

Corollary 1. *Let α be any formula in $\mathrm{P}(\mathbf{L})$. The following statements are equivalent:*

1. $\mathrm{dak}(\alpha) \equiv \alpha$.
2. $\alpha \in \mathrm{K}(\mathrm{A}(X))$.
3. $\mathrm{alt}(\alpha) = 0$.

Given any formula α in $\mathrm{P}(\mathbf{L})$, we now know that it is possible to obtain from it another in conjunctive normal form by iterated application of the dak function. Moreover, the minimum number or iterations required is exactly $\mathrm{alt}(\alpha)$. By Remark 3, we know that this other formula in conjunctive normal form is logically equivalent to α, the input formula.

Corollary 2. *For all* $\alpha \in \mathrm{P}(\mathbf{L})$*, the natural number* $\mathrm{alt}(\alpha)$ *is the smallest natural number* m *satisfying* $\mathrm{dak}^m(\alpha) \in \mathrm{K}(\mathrm{A}(X))$.

Proof. The proof is by induction on n according to the predicate $Q(n)$ of the literal content:

If $\alpha \in \mathrm{P}(\mathbf{L})$ and $n = \mathrm{alt}(\alpha)$, then n is the smallest natural m satisfying $\mathrm{dak}^m(\alpha) \in \mathrm{K}(\mathrm{A}(X))$.

The reasoning is as follows:

- $n = 0$; if $\alpha \in \mathrm{P}(\mathbf{L})$ and $0 = \mathrm{alt}(\alpha)$, then by Corollary 1, we know that $\alpha \in \mathrm{K}(\mathrm{A}(X))$, i.e., $\mathrm{dak}^0(\alpha) \in \mathrm{K}(\mathrm{A}(X))$, since dak^0 is the identity map. Since 0 is the smallest natural number, the set of natural numbers smaller than it is empty, from which we conclude the assertion.
- Suppose that $0 < n$, that $Q(n-1)$ is true, and that $\alpha \in \mathrm{P}(\mathbf{L})$ is fixed but arbitrary under the condition that $n = \mathrm{alt}(\alpha)$. As we know from Theorem 1 and Corollary 1, it holds that

$$\mathrm{alt}(\mathrm{dak}(\alpha)) = \mathrm{alt}(\alpha) - 1 = n - 1 \tag{14}$$

By (14) and the induction hypothesis, we have, in particular, that

$$\mathrm{dak}^n(\alpha) = \mathrm{dak}^{n-1}(\mathrm{dak}(\alpha)) \in \mathrm{K}(\mathrm{A}(X))$$

On the other hand, let m be a natural number, such that $m < n$. Three cases can occur:

* $m = n - 1$; then:

$$\begin{aligned} \mathrm{alt}(\mathrm{dak}^{n-1}(\alpha)) &= \mathrm{alt}(\alpha) - n + 1 \\ &= n - n + 1 \\ &= 1 \end{aligned}$$

so (see Corollary 1) $\mathrm{dak}^{n-1}(\alpha) \notin \mathrm{K}(\mathrm{A}(X))$.

* $0 < m < n - 1$; by Theorem 1, we know that $\mathrm{alt}(\mathrm{dak}(\alpha)) = n - 1$, and since $m - 1 < m < n - 1$, by the induction hypothesis, we have

$$\mathrm{dak}^m(\alpha) = \mathrm{dak}^{m-1}(\mathrm{dak}(\alpha)) \notin \mathrm{K}(\mathrm{A}(X))$$

* $m = 0$; $\mathrm{dak}^0(\alpha) = \alpha$, and since $\mathrm{alt}(\alpha) = n > 0$, we deduce that $\mathrm{dak}^0(\alpha) \notin \mathrm{K}(\mathrm{A}(X))$.

By the principle of finite induction, we deduce that $Q(n)$ is true for any natural number n. Since the function alt can be applied to any formula, the result is true. □

4. Associativity

By iterating dak from any formula, we obtain, as we have seen, a formula equivalent to it that is in conjunctive normal form. However, for certain formulas in conjunctive normal form, there are several others also in conjunctive normal form that are equivalent to it, but such that they are all distinct from each other. In this section, we intend to provide an algorithm to select among all those formulas, one of which we will consider in canonical form. For practical reasons, we will consider the set $\mathrm{lcnf}(X)$, the set of formulas in conjunctive left normal form, as the one that gathers exactly all the formulas in canonical form.

Definition 4. *Let* $\mathrm{ar}\colon \mathrm{P}(\mathbf{L}) \longrightarrow \mathbb{Z}$ *be defined as follows:*

$$\mathrm{ar}(\alpha) = \begin{cases} -1, & \text{if } \alpha \in X; \\ \max\{\mathrm{ar}(\varphi), \mathrm{ar}(\psi)\}, & \text{if } \alpha = \mathrm{K}\,\varphi\psi; \\ \max\{\mathrm{ar}(\varphi), 1 + \mathrm{ar}(\psi)\}, & \text{if } \alpha = \mathrm{A}\,\varphi\psi. \end{cases}$$

and let $\mathrm{kr}\colon \mathrm{P}(\mathbf{L}) \longrightarrow \mathbb{Z}$ *be defined as follows:*

$$\mathrm{kr}(\alpha) = \begin{cases} -1, & \text{if } \alpha \in X; \\ \max\{\mathrm{kr}(\varphi), 1 + \mathrm{kr}(\psi)\}, & \text{if } \alpha = \mathrm{K}\,\varphi\psi; \\ \max\{\mathrm{kr}(\varphi), \mathrm{kr}(\psi)\}, & \text{if } \alpha = \mathrm{A}\,\varphi\psi. \end{cases}$$

Also let $\mathrm{akr}\colon \mathrm{P}(\mathbf{L}) \longrightarrow \mathbb{Z}$ *be defined as follows:*

$$\mathrm{akr}(\alpha) = \max\{\mathrm{ar}(\alpha), \mathrm{kr}(\alpha)\}$$

Remark 4. *The maps* ar *and* kr *have the properties given in Lemma 4. This Lemma characterises the elements of* $\mathrm{K}(X)$, $\mathrm{A}(X)$, *and* X. *The particular assignation in both functions of the value* -1 *to the elements of* X *is just in order to adjust the final computations accordingly.*

Lemma 4. *For all* $\alpha \in \mathrm{P}(\mathbf{L})$*:*

1. $\alpha \in \mathrm{K}(X)$ *if, and only if,* $\mathrm{ar}(\alpha) = -1$.
2. $\alpha \in \mathrm{A}(X)$ *if, and only if,* $\mathrm{kr}(\alpha) = -1$.
3. $\alpha \in X$ *if, and only if,* $\mathrm{kr}(\alpha) = -1 = \mathrm{ar}(\alpha)$.
4. $\mathrm{K}(X) \cap \mathrm{A}(X) = X$.

Proof. Let us prove statement 1. First we will reason by induction according to the complexity of α and according to the predicate $Q(n)$ of the literal content:

$$\text{“for all } \alpha \in \mathrm{P}(\mathbf{L}), \text{ if } \alpha \in \mathrm{K}(X) \text{ and } \mathrm{comp}(\alpha) = n, \text{ then } \mathrm{ar}(\alpha) = -1\text{”}$$

Suppose, as an induction hypothesis, that n is a natural number and that for any natural number k such that $k < n$, $Q(k)$ holds. We have the following cases:

- $n = 0$; then let —as the only case of interest— $\alpha \equiv x \in X$. By Definition 4, $\mathrm{ar}(\alpha) = -1$, so $Q(0)$ is true.
- $n > 0$; if $\alpha \in \mathrm{K}(X)$ and $n > 0$, there must exist $\varphi, \psi \in \mathrm{K}(X)$ such that $\alpha \equiv K\varphi\psi$. Then

$$\begin{aligned} \mathrm{ar}(\alpha) &= \max\{\mathrm{ar}(\varphi), \mathrm{ar}(\psi)\} && \text{Definition 4} \\ &= \max\{-1, -1\} && \text{induc. hyp.} \\ &= -1 \end{aligned}$$

so $Q(n)$ is true in this case.

By the *second principle of finite induction*, for any natural number n is true $Q(n)$ and hence the implication. Reciprocally, let us now consider the predicate $Q(n)$:

$$\text{“for all } \alpha \in \mathrm{P}(\mathbf{L}), \text{ if } \mathrm{comp}(\alpha) = n \text{ and } \mathrm{ar}(\alpha) = -1, \text{ then } \alpha \in \mathrm{K}(X)\text{”}$$

Suppose, as an induction hypothesis, that n is a natural number and that for any natural number k, such that $k < n$, $Q(k)$ holds. We have the following cases:

- $n = 0$; then let —as the only case of interest— $\alpha \equiv x \in X$. Since $X \subseteq \mathrm{K}(X)$ it follows that $Q(0)$ is true.
- $n > 0$; let $\alpha \in \mathrm{P}(\mathbf{L})$ such that $\mathrm{comp}(\alpha) = n$ and $\mathrm{ar}(\alpha) = -1$. In principle, the following are possible:
  - there exist $\varphi, \psi \in \mathrm{P}(\mathbf{L})$, such that $\alpha \equiv K\varphi\psi$; then

$$\begin{aligned} -1 &= \mathrm{ar}(\alpha) \\ &= \max\{\mathrm{ar}(\varphi), \mathrm{ar}(\psi)\} \\ &\Rightarrow \mathrm{ar}(\varphi) = -1 = \mathrm{ar}(\psi) \\ &\Rightarrow \varphi, \psi \in \mathrm{K}(X) && \text{induc. hyp.} \\ &\Rightarrow \alpha \in \mathrm{K}(X) \end{aligned}$$

– there exist $\varphi, \psi \in \mathrm{P}(\mathbf{L})$, such that $\alpha \equiv A\varphi\psi$; then

$$\begin{aligned} -1 &= \mathrm{ar}(\alpha) \\ &= \max\{\mathrm{ar}(\varphi), 1 + \mathrm{ar}(\psi)\} \\ &\geq 0 \\ &\Rightarrow \text{therefore this case is not possible} \end{aligned}$$

so that $Q(n)$ is true.

By the *second principle of finite induction*, for any natural number n, $Q(n)$ holds and hence the implication. Statement 2 can be proved with the same scheme as above. Suppose now that $\mathrm{kr}(\alpha) = -1 = \mathrm{ar}(\alpha)$. Since $\mathrm{ar}(\alpha) = -1$, we have that $\alpha \in \mathrm{K}(X)$ and if there exist $\varphi, \psi \in \mathrm{P}(\mathbf{L})$, such that $\alpha \equiv K\varphi\psi$, then one would have

$$\begin{aligned} -1 &= \mathrm{kr}(\alpha) && \text{hypothesis} \\ &= \max\{\mathrm{kr}(\varphi), 1 + \mathrm{kr}(\psi)\} && \text{Definition 4} \\ &\geq 0 \end{aligned}$$

which is absurd, so $\alpha \in X$. The reciprocal statement is obviously true and it follows that $\mathrm{K}(X) \cap \mathrm{A}(X) = X$. □

The proof of Lemma 5 is straightforward from Lemma 4.

Lemma 5. *For all* $\alpha \in \mathrm{P}(\mathbf{L})$*:*

1. *If* $\alpha \in \mathrm{K}(X) \setminus X$ *then* $0 \leq \mathrm{kr}(\alpha)$.
2. *If* $\alpha \in \mathrm{A}(X) \setminus X$ *then* $0 \leq \mathrm{ar}(\alpha)$.
3. $\mathrm{ar}(\alpha) = -1$ *and* $\mathrm{kr}(\alpha) = -1$ *if, and only if,* $\alpha \in X$.

Remark 5. *The respective reciprocal statements of the first two sentences of Lemma 5 are not true. Indeed,* $\mathrm{kr}(AKxyx) = 0$ *(resp.* $\mathrm{ar}(KAxyx) = 0$*), and yet* $AKxyx \notin \mathrm{K}(X) \setminus X$ *(resp.* $KAxyx \notin \mathrm{A}(X) \setminus X$*).*

Lemma 6 characterises the elements of $\mathrm{kl}(X) \setminus X$ and $\mathrm{al}(X) \setminus X$. Its proof can be carried out by induction by making a careful distinction of cases.

Lemma 6. *For all* $\alpha \in \mathrm{P}(\mathbf{L})$*,*

1. $\mathrm{ar}(\alpha) = -1$ *and* $\mathrm{kr}(\alpha) = 0$ *if, and only if,* $\alpha \in \mathrm{kl}(X) \setminus X$.
2. $\mathrm{ar}(\alpha) = 0$ *and* $\mathrm{kr}(\alpha) = -1$ *if, and only if,* $\alpha \in \mathrm{al}(X) \setminus X$.

Lemma 7. *For all* $\alpha \in \mathrm{K}(\mathrm{A}(X))$*, the following statements are equivalent:*

1. $\mathrm{ar}(\alpha) = 0$ *and* $\mathrm{kr}(\alpha) = 0$.
2. $\alpha \in \mathrm{lcnf}(X) \setminus (\mathrm{al}(X) \cup \mathrm{kl}(X))$.

Remark 6. *The formula* $\alpha \equiv AKxyx$ *satisfies* $\mathrm{ar}(\alpha) = 0 = \mathrm{kr}(\alpha)$*, but* $\alpha \notin \mathrm{lcnf}(X)$*; hence the need for the restriction in the statement of Lemma 7.*

By means of akr, the above technical lemmas make it possible to characterise in Theorem 2 the set $\mathrm{lcnf}(X)$ of formulas in left conjunctive normal form. Note how $\leq$ appears in the statement, again highlighting the subtle role played by -1 in the definition of akr.

Theorem 2. *For all* $\alpha \in \mathrm{K}(\mathrm{A}(X))$*, the following statements are equivalent:*

1. $\mathrm{akr}(\alpha) \leq 0$.
2. $\alpha \in \mathrm{lcnf}(X)$.

Proof. Let us first show that statement 1 is a sufficient condition for 2. to be fulfilled. The following cases are possible:

1. $\mathrm{ar}(\alpha) = -1 = \mathrm{kr}(\alpha)$; as stated in Lemma 5, $\alpha \in X$.
2. $\mathrm{ar}(\alpha) = -1$ and $\mathrm{kr}(\alpha) = 0$; as stated in Lemma 6, $\alpha \in \mathrm{kl}(X) \setminus X$.
3. $\mathrm{ar}(\alpha) = 0$ and $\mathrm{kr}(\alpha) = -1$; as stated in Lemma 6, $\alpha \in \mathrm{al}(X) \setminus X$.
4. $\mathrm{ar}(\alpha) = 0$ and $\mathrm{kr}(\alpha) = 0$; as stated in Lemma 7, $\alpha \in \mathrm{lcnf}(X) \setminus (\mathrm{al}(X) \cup \mathrm{kl}(X))$.

and hence, $\alpha \in \mathrm{lcnf}(X)$. However, 1 is a necessary condition for 2, which follows as a consequence of the aforementioned lemmas. □

Now everything is ready to carry out the accumulation of the functors A and K on the left side of the formula without changing its logical meaning. This task will be carried out by the function lasc, defined in Definition 5, by means of convenient iterations. The lasc function is the classical one, but formulated here univocally in a novel recursive way via rules.

Definition 5 (left associativity). *Consider the following rules:*

$$x_i \to_{\mathrm{lasc}} x_i \tag{15}$$

$$\frac{A\varphi\psi \to_{\mathrm{lasc}} \alpha \quad \xi \to_{\mathrm{lasc}} \beta}{A\varphi A\psi\xi \to_{\mathrm{lasc}} A\alpha\beta} \tag{16}$$

$$\frac{K\varphi\psi \to_{\mathrm{lasc}} \alpha \quad \xi \to_{\mathrm{lasc}} \beta}{K\varphi K\psi\xi \to_{\mathrm{lasc}} K\alpha\beta} \tag{17}$$

$$\frac{\varphi \to_{\mathrm{lasc}} \varphi' \quad \psi \to_{\mathrm{lasc}} \psi'}{A\varphi\psi \to_{\mathrm{lasc}} A\varphi'\psi'} \tag{18}$$

$$\frac{\varphi \to_{\mathrm{lasc}} \varphi' \quad \psi \to_{\mathrm{lasc}} \psi'}{K\varphi\psi \to_{\mathrm{lasc}} K\varphi'\psi'} \tag{19}$$

where (16)–(19) shall be applied with the priority from highest to lowest according to the order given. Les us now define the map $\mathrm{lasc}\colon P(\mathbf{L}) \longrightarrow P(\mathbf{L})$ *by* $\mathrm{lasc}(\varphi) \equiv \psi$ *if, and only if,* $\varphi \to_{\mathrm{lasc}} \psi$.

It is clear that for any formula φ, $\mathrm{lasc}(\varphi)$ is equivalent to φ (see Remark 2); therefore, lasc does not alter the logical meaning of the formulas by acting on them, although it does eventually alter their syntax. What is stated in Lemma 8 is obviously true.

Lemma 8. *For all* $\alpha \in \mathrm{P}(\mathbf{L})$,

1. *If* $\alpha \in A(X)$ *then* $\mathrm{lasc}(\alpha) \in A(x)$.
2. *If* $\alpha \in K(X)$ *then* $\mathrm{lasc}(\alpha) \in K(x)$.

Lemma 9. *For all* $\alpha \in A(X)$,

$$\mathrm{ar}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor.$$

Proof. The proof is by induction on the complexity of α. □

Remark 7. *It is also evident that for all* $\alpha \in A(X)$

$$\mathrm{kr}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor = \left\lfloor \frac{-1}{2} \right\rfloor = -1.$$

As we can see, the reductive role of lasc on $\mathrm{akr}(\alpha)$ is very powerful when applying lasc to the formula α; as we can see, it is such that divides the "complexity of the situation" by 2, which inevitably invokes the logarithm in base 2. The information provided by Theorem 3 is crucial in this section, so we will give a detailed demonstration of it.

Theorem 3. *For all* $\alpha \in \mathrm{K}(\mathrm{A}(X))$*:*

1. $\mathrm{ar}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor$
2. $\mathrm{kr}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor$
3. $\mathrm{akr}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{akr}(\alpha)}{2} \right\rfloor$

Proof. To prove 1, we will reason by induction about the complexity of α using the predicate $Q(n)$ of the literal content:

$$\text{“for all } \alpha, \text{ if } \alpha \in K(A(X)) \text{ and } \mathrm{comp_k}(\alpha) = n, \text{ then } \mathrm{ar}(\mathrm{lasc}(\alpha)) = \left\lfloor \tfrac{\mathrm{ar}(\alpha)}{2} \right\rfloor\text{”}$$

Suppose, as an induction hypothesis, that n is a natural number and that for any natural number k such that $k < n$ $Q(k)$ holds. We have the following cases:

- $n = 0$; must be $\alpha \in \mathrm{A}(X)$, the formula for which is $\mathrm{ar}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor$, as set out in Lemma 9.
- $n > 0$; let —as the only case of interest— $\alpha \equiv K\varphi\rho$ for certain $\varphi, \rho \in \mathrm{K}(\mathrm{A}(X))$. Let us distinguish the following cases:
  - $\rho \in A(X)$; then:

$$\begin{aligned}
\mathrm{ar}(\mathrm{lasc}(\alpha)) &= \mathrm{ar}(\mathrm{lasc}(K\varphi\rho)) = \mathrm{ar}(K\,\mathrm{lasc}(\varphi)\,\mathrm{lasc}(\rho)) \\
&= \max\{\mathrm{ar}(\mathrm{lasc}(\varphi)), \mathrm{ar}(\mathrm{lasc}(\rho))\} \\
&= \max\left\{ \left\lfloor \frac{\mathrm{ar}(\varphi)}{2} \right\rfloor, \left\lfloor \frac{\mathrm{ar}(\rho)}{2} \right\rfloor \right\} && \text{induc. hyp. and Lemma 9} \\
&= \left\lfloor \frac{\max\{\mathrm{ar}(\varphi), \mathrm{ar}(\rho)\}}{2} \right\rfloor = \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor
\end{aligned}$$

  - $\rho \notin A(X)$; then $\alpha \equiv K\varphi K\psi\xi$ for certain $\psi, \xi \in \mathrm{K}(\mathrm{A}(X))$. In this case,

$$\begin{aligned}
\mathrm{ar}(\mathrm{lasc}(\alpha)) &= \mathrm{ar}(\mathrm{lasc}(K\varphi K\psi\xi)) = \mathrm{ar}(K\,\mathrm{lasc}(K\varphi\psi)\,\mathrm{lasc}(\xi)) \\
&= \max\{\mathrm{ar}(\mathrm{lasc}(K\varphi\psi)), \mathrm{ar}(\mathrm{lasc}(\xi))\} \\
&= \max\left\{ \left\lfloor \frac{\mathrm{ar}(K\varphi\psi)}{2} \right\rfloor, \left\lfloor \frac{\mathrm{ar}(\xi)}{2} \right\rfloor \right\} && \text{induc. hyp} \\
&= \max\left\{ \left\lfloor \frac{\max\{\mathrm{ar}(\varphi), \mathrm{ar}(\psi)\}}{2} \right\rfloor, \left\lfloor \frac{\mathrm{ar}(\xi)}{2} \right\rfloor \right\} \\
&= \max\left\{ \left\lfloor \frac{\mathrm{ar}(\varphi)}{2} \right\rfloor, \left\lfloor \frac{\mathrm{ar}(\psi)}{2} \right\rfloor, \left\lfloor \frac{\mathrm{ar}(\xi)}{2} \right\rfloor \right\} \\
&= \left\lfloor \frac{\max\{\mathrm{ar}(\varphi), \mathrm{ar}(\psi), \mathrm{ar}(\xi)\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\max\{\mathrm{ar}(\varphi), \max\{\mathrm{ar}(\psi), \mathrm{ar}(\xi)\}\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\max\{\mathrm{ar}(\varphi), \mathrm{ar}(K\psi\xi)\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\mathrm{ar}(K\varphi K\psi\xi)}{2} \right\rfloor \\
&= \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor
\end{aligned}$$

By the *second principle of finite induction*, for every natural number n, $Q(n)$ holds, hence the validity of statement 1. To prove 2, let us reason by induction about the complexity of α according to the predicate $Q(n)$ of the literal content:

"for all α, if $\alpha \in K(A(X))$ and $\mathrm{comp_k}(\alpha) = n$, then $\mathrm{kr}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor$"

Suppose, as an induction hypothesis, that n is a natural number and that for any natural number k such that $k < n$, $Q(k)$ holds. We have the following cases:

- $n = 0$; must be $\alpha \in \mathrm{A}(X)$, the formula for which is $\mathrm{kr}(\mathrm{lasc}(\alpha)) = \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor$, as set out in Remark 7.
- $n > 0$; let —as the only case of interest— $\alpha \equiv K\varphi\rho$ for certain $\varphi, \rho \in \mathrm{K}(\mathrm{A}(X))$. Let us distinguish the following cases:
  - $\rho \in A(X)$; then

$$\begin{aligned}
\mathrm{kr}(\mathrm{lasc}(\alpha)) &= \mathrm{kr}(\mathrm{lasc}(K\varphi\rho)) \\
&= \mathrm{kr}(K\,\mathrm{lasc}(\varphi)\,\mathrm{lasc}(\rho)) \\
&= \max\{\mathrm{kr}(\mathrm{lasc}(\varphi)), 1 + \mathrm{kr}(\mathrm{lasc}(\rho))\} \\
&= \max\left\{\left\lfloor \frac{\mathrm{kr}(\varphi)}{2} \right\rfloor, 0\right\} && \text{induc. hyp. and Lemma 4} \\
&= \left\lfloor \frac{\max\{\mathrm{kr}(\varphi), 0\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\max\{\mathrm{kr}(\varphi), 1 + \mathrm{kr}(\rho)\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\mathrm{kr}(K\varphi\rho)}{2} \right\rfloor \\
&= \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor
\end{aligned}$$

  - $\rho \notin A(X)$; then $\alpha \equiv K\varphi K\psi\xi$ for certain $\psi, \xi \in \mathrm{K}(\mathrm{A}(X))$. In this case,

$$\begin{aligned}
\mathrm{kr}(\mathrm{lasc}(\alpha)) &= \mathrm{kr}(\mathrm{lasc}(K\varphi K\psi\xi)) = \mathrm{kr}(K\,\mathrm{lasc}(K\varphi\psi)\,\mathrm{lasc}(\xi)) \\
&= \max\{\mathrm{kr}(\mathrm{lasc}(K\varphi\psi)), 1 + \mathrm{kr}(\mathrm{lasc}(\xi))\} \\
&= \max\left\{\left\lfloor \frac{\mathrm{kr}(K\varphi\psi)}{2} \right\rfloor, 1 + \left\lfloor \frac{\mathrm{kr}(\xi)}{2} \right\rfloor\right\} && \text{induc. hyp.} \\
&= \max\left\{\left\lfloor \frac{\max\{\mathrm{kr}(\varphi), 1 + \mathrm{kr}(\psi)\}}{2} \right\rfloor, \left\lfloor \frac{2 + \mathrm{kr}(\xi)}{2} \right\rfloor\right\} \\
&= \left\lfloor \frac{\max\{\max\{\mathrm{kr}(\varphi), 1 + \mathrm{kr}(\psi)\}, 2 + \mathrm{kr}(\xi)\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\max\{\mathrm{kr}(\varphi), \max\{1 + \mathrm{kr}(\psi), 2 + \mathrm{kr}(\xi)\}\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\max\{\mathrm{kr}(\varphi), 1 + \max\{\mathrm{kr}(\psi), 1 + \mathrm{kr}(\xi)\}\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\max\{\mathrm{kr}(\varphi), 1 + \mathrm{kr}(K\psi\xi)\}}{2} \right\rfloor \\
&= \left\lfloor \frac{\mathrm{kr}(K\varphi K\psi\xi)}{2} \right\rfloor \\
&= \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor
\end{aligned}$$

By the *second principle of finite induction*, for every natural number n, $Q(n)$ holds, hence the validity of statement 2. Statement 3 is immediate from statements 1 and 2, given that

$$\max\left\{\left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor, \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor\right\} = \left\lfloor \frac{\max\{\mathrm{ar}(\alpha), \mathrm{kr}(\alpha)\}}{2} \right\rfloor = \left\lfloor \frac{\mathrm{akr}(\alpha)}{2} \right\rfloor$$

□

Lemma 10. *Let $\alpha \in \mathrm{A}(X)$. The following statements are equivalent:*

1. $\alpha \in \mathrm{al}(X)$
2. $\mathrm{lasc}(\alpha) \equiv \alpha$
3. $\mathrm{ar}(\alpha) \leq 0$

Proof. To show that statement 1 implies statement 2, we will reason by induction about the complexity of α according to the predicate $Q(n)$ of the literal content:

$$\text{“for all } \alpha, \text{ if } \alpha \in \mathrm{al}(X) \text{ and } \mathrm{comp}(\alpha) = n \text{ then } \mathrm{lasc}(\alpha) \equiv \alpha\text{”}$$

Suppose, as an induction hypothesis, that n is a natural number and that for any natural number k such that $k < n$, $Q(k)$ holds. We distinguish the following cases:

- $n = 0$; must be $\alpha \equiv x \in X$ and then $\mathrm{lasc}(\alpha) \equiv \mathrm{lasc}(x) \equiv x \equiv \alpha$.
- $n > 0$; let —as the only case of interest— $\alpha \equiv A\varphi x$, where $\varphi \in \mathrm{al}(X)$ and $x \in X$. Then

$$\begin{aligned} \mathrm{lasc}(\alpha) &\equiv \mathrm{lasc}(A\varphi x) \\ &\equiv A\,\mathrm{lasc}(\varphi)\,\mathrm{lasc}(x) \\ &\equiv A\varphi x && \text{induc. hyp. and Definition 5} \\ &\equiv \alpha \end{aligned}$$

hence, $Q(n)$ holds.

By the *second principle of finite induction*, for every natural number n $Q(n)$ holds, hence the validity of statement 2. Let us now suppose that statement 2 is true, i.e., that $\alpha \in \mathrm{A}(X)$ and that $\mathrm{lasc}(\alpha) \equiv \alpha$; then, one has

$$\begin{aligned} \mathrm{ar}(\alpha) &= \mathrm{ar}(\mathrm{lasc}(\alpha)) \\ &= \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor && \text{(Lemma 9)} \end{aligned}$$

from which we deduce that $\mathrm{ar}(\alpha) \in \{-1, 0\}$, i.e., that statement 3 holds. Finally, suppose that $\alpha \in \mathrm{A}(X)$ and that $\mathrm{ar}(\alpha) \leq 0$. The following cases are possible (note that, according to statement 2 of Lemma 4, necessarily $\mathrm{kr}(\alpha) = -1$):

1. $\mathrm{ar}(\alpha) = -1$ and $\mathrm{kr}(\alpha) = -1$; then $\alpha \in X \subseteq \mathrm{al}(X)$ (see Lemma 5).
2. $\mathrm{ar}(\alpha) = 0$ and $\mathrm{kr}(\alpha) = -1$; then $\alpha \in \mathrm{al}(X) \setminus X$ (see Lemma 6).

This proves that under the assumption of statement 3. the fact $\alpha \in \mathrm{al}(X)$ is satisfied, as we sought to prove. □

In Theorem 4, the effects of alt and lasc are finally combined to characterise the formulas in $\mathrm{lcnf}(X)$.

Theorem 4. *For all $\alpha \in \mathrm{P}(\mathbf{L})$, the following statements are equivalent:*

1. $\alpha \in \mathrm{lcnf}(X)$.
2. $\mathrm{dak}(\alpha) \equiv \alpha$ *and* $\mathrm{lasc}(\alpha) \equiv \alpha$.
3. $\mathrm{alt}(\alpha) = 0$ *and* $\mathrm{akr}(\alpha) \leq 0$.

Proof. Let α be any formula in $\mathrm{P}(\mathbf{L})$. Assume what statement 1 states, i.e., that $\alpha \in \mathrm{lcnf}(X)$. We will reason by induction about the complexity of α according to the predicate $Q(n)$ of the literal content:

$$\text{“for all } \alpha, \text{ if } \alpha \in \mathrm{lcnf}(X) \text{ and } \mathrm{comp}(\alpha) = n, \text{ then } \mathrm{dak}(\alpha) = \alpha \text{ and } \mathrm{lasc}(\alpha) = \alpha\text{”}$$

Suppose, as an induction hypothesis, that n is a natural number and that for any natural number k such that $k < n$, $Q(k)$ holds. We distinguish the following cases:

- $n = 0$; must be $\alpha \equiv x \in X$ and then $\mathrm{dak}(\alpha) \equiv \mathrm{dak}(x) \equiv x \equiv \alpha$ and similarly, $\mathrm{lasc}(\alpha) \equiv \alpha$. It follows that $Q(0)$ is true.
- $n > 0$; let—as the only case of interest— $\alpha \equiv K\varphi\psi$, where $\varphi \in \mathrm{lcnf}(X)$ and $\psi \in \mathrm{al}(X)$. Then

$$\begin{aligned}
\mathrm{dak}(\alpha) &\equiv \mathrm{dak}(K\varphi\psi) && \\
&\equiv K\,\mathrm{dak}(\varphi)\,\mathrm{dak}(\psi) && \\
&\equiv K\varphi\,\mathrm{dak}(\psi) && \text{induc. hyp.} \\
&\equiv K\varphi\psi && \text{Remark 1 and induc. hyp.} \\
&\equiv \alpha && \\
\mathrm{lasc}(\alpha) &\equiv \mathrm{lasc}(K\varphi\psi) && \\
&\equiv K\,\mathrm{lasc}(\varphi)\,\mathrm{lasc}(\psi) && \\
&\equiv K\varphi\,\mathrm{lasc}(\psi) && \text{induc. hyp.} \\
&\equiv K\varphi\psi && \text{Lemma 10} \\
&\equiv \alpha &&
\end{aligned}$$

so we know that $Q(n)$ is true.

By the *second principle of finite induction*, for every natural number n is true $Q(n)$, hence the validity of assertion 2. If we now assume that 2 is true, that $\mathrm{alt}(\alpha) = 0$ and that $\alpha \in \mathrm{K}(\mathrm{A}(X))$ is ensured by the Corollary 1. In particular, as a consequence of Theorem 3, we have

$$\begin{aligned}
\mathrm{ar}(\alpha) = \mathrm{ar}(\mathrm{lasc}(\alpha)) &= \left\lfloor \frac{\mathrm{ar}(\alpha)}{2} \right\rfloor \\
\mathrm{kr}(\alpha) = \mathrm{kr}(\mathrm{lasc}(\alpha)) &= \left\lfloor \frac{\mathrm{kr}(\alpha)}{2} \right\rfloor
\end{aligned}$$

and therefore, $\mathrm{ar}(\alpha) \leq 0$ and $\mathrm{kr}(\alpha) \leq 0$; thus, we have proved 3. Let us finally assume 3 to be true and show that $\alpha \in \mathrm{lcnf}(X)$. Since $\mathrm{alt}(\alpha) = 0$ and again using Corollary 1, we know that $\alpha \in \mathrm{K}(\mathrm{A}(X))$. According to Theorem 2, since $\mathrm{akr}(\alpha) \leq 0$, $\alpha \in \mathrm{lcnf}(X)$ must necessarily hold and this is what statement 1 establishes. □

Definition 6. *For all* $\alpha \in \mathrm{K}(\mathrm{A}(X))$, $\mathrm{hre}(\alpha)$ *is the natural number defined by the equality:*

$$\mathrm{hre}(\alpha) = \begin{cases} 0, & \text{if } \mathrm{akr}(\alpha) \leq 0, \\ \lfloor log_2(\mathrm{akr}(\alpha)) \rfloor + 1, & \text{otherwise.} \end{cases}$$

Remark 8. *Understanding the evaluation of the expressions from a "lazy" point of view, the following equality should be accepted:*

$$\mathrm{hre}(\alpha) = (1 - \chi_{\{-1,0\}}(\mathrm{akr}(\alpha)))(\lfloor log_2(\mathrm{akr}(\alpha)) \rfloor + 1)$$

where, of course, $\chi_{\{-1,0\}}$ *is the characteristic function on the set* $\{-1, 0\}$. *Let it also be noted that in the case where for the formula* α *one has* $0 < \mathrm{akr}(\alpha)$, *then* $\mathrm{hre}(\alpha)$ *is the number of digits in the (single) binary expression of* $\mathrm{akr}(\alpha)$ *when it is greater than* 0, *and* 0 *otherwise.*

Finally, the next corollary, Corollary 3, informs us that for any formula α in $\mathrm{K}(\mathrm{A}(X))$, $\mathrm{lasc}^{\mathrm{hre}(\alpha)}(\alpha)$ is a formula in left conjunctive normal form (equivalent to α, of course) and that per iteration of lasc, the number $\mathrm{hre}(\alpha)$ of iterations is the smallest number of those that achieve it. It is based on how many times the function $\lfloor \frac{x}{2} \rfloor$ must be iterated to obtain zero, in order to express x in binary form. This gives us an estimate of the complexity of

our algorithm: it is logarithmic, which is fantastic news. The corresponding result, with its complete proof, is given just below. The demonstration of the corollary is simple if we rely on this observation, and it can be carried out by inductive reasoning based on a careful distinction of cases.

Corollary 3. *For all* $\alpha \in \mathrm{K}(\mathrm{A}(X))$, $\mathrm{hre}(\alpha)$ *is the smallest of the natural numbers* m *satisfying* $\mathrm{lasc}^m(\alpha) \in \mathrm{lcnf}(X)$.

5. Conclusions

Algorithm 1 summarises the content of the article. The structure of the work suggests using `for` loops instead of `while` loops, and we have done so. The following comments are appropriate:

- As we commented before, according to Definition 3, Definition 5, and Remark 2, the result of the two `for` loops in Algorithm 1 is a formula equivalent to the one that each of the loops took as a starting point. Thus, in each application of the algorithm, the result obtained retains the semantic meaning of the form it took as initial data.
- Each `for` loop in Algorithm 1 acts if and only if a change in the formula is necessary, i.e., if $m > 0$. In this case, the pre-calculated length of each of the two loops (respectively, alt, according to Defintion 2, and akr, acording to Definition 4) is exactly the minimum necessary to obtain the intended purpose of the loop. This extreme is guaranteed by Corollary 2 and Corollary 3. In short, under this approach, the length of the two `for` loops is optimal.
- The algorithm achieves what it sets out to do, as stated in Corollary 2 and Corollary 3.
- In each of the `for` loops of Algorithm 1, the comparison between the result formula of each round and that of the previous one, which was carried out in the classical formulation of the algorithm (reproduced in summary and pseudocode in Section 1) in order to decide whether the work has been completed, has disappeared. In its place, a single operation has appeared for each loop, before its start, consisting of the traversal and reading of a single formula, namely the loop data formula. In the general case, a priori this is an improvement over the classical formulation of the CNF algorithm, which in pseudocode we summarily reproduced in Section 1. Very unfavorable examples can be consulted in the readme.md of our GitHub repository https://github.com/ringstellung/CNF (accessed on 23 September 2023), and you can see how it is confronted by our Haskell implementation.
- Definition 3 and Definition 5 are an unpublished presentation of the well-known core of the CNF algorithm. We believe we have included, with great concreteness, the concept of "recursive substitution" inspired by the usual style of lambda calculus. We believe we have found the appropriate language to be able to express in a very precise way, eliminating all ambiguity, a process usually described with certain formal licenses.

We have no doubt that other improvements in the efficiency of the CNF algorithm are possible, although we have only dealt with one here. In the future, we will investigate how to combine all of them. For this, we will build on some of those already introduced in the Haskell code of our GitHub repository.

The content of this article will be used in the future to act in the field of the classical SAT problem, e.g., to optimally prepare the application of the Davis–Putnam algorithm. We intend to compare this improved algorithm with that of deduction by sequents.

Algorithm 1 CNF simplified algorithm for formulas in $\mathrm{P}(\mathbf{L})$

Require: $\alpha \in \mathrm{P}(\mathbf{L})$
Ensure: $\beta \in \mathrm{lcnf}(X)$
  procedure CNFsa(α)
    $m \leftarrow \mathrm{alt}(\alpha)$
    for $1 \leq n \leq m$ **do**
      $\alpha \leftarrow \mathrm{dak}(\alpha)$
    end for
    $m \leftarrow \mathrm{hre}(\alpha)$
    for $1 \leq n \leq m$ **do**
      $\alpha \leftarrow \mathrm{lasc}(\alpha)$
    end for
    return α
  end procedure

In order to show the feasibility and effectiveness of the theoretical advances presented in this article, we have elaborated a functional implementation for the Haskell language. In it, we also give a preview of the implementation of the Davis–Putnam algorithm. The code can be consulted at https://github.com/ringstellung/CNF (accessed on 23 September 2023). This justifies the fundamental objective of the work, which is set out in its title.

Author Contributions: The authors of this article contributed equally to formal analysis, investigation, methodology, project administration, writing—original draft and writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: In order to facilitate the article's comprehension and leveraging of the presented results and to allow the immediate utilization of the algorithmic improvements by researchers on the topic, we have made the code of the corresponding Haskell implemented programs public through a GitHub repository: https://github.com/ringstellung/CNF (accessed on 23 September 2023).

Acknowledgments: The authors would like to thank the staff of the Algorithms' journal, in particular its editing assistants, for all the encouragement and support during the process of preparing and submitting the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNF	conjunctive normal form
DNF	disjunctive normal form
PNF	prenex normal form
POS	product of sums
SOP	sum of products
SAT	Boolean satisfiability
iff	if and only if
def.	definition
induc hyp.	induction hypothesis
resp.	respectively
p.	page
pp.	pages

References

1. Tseitin, G.S. On the Complexity of Derivation in Propositional Calculus. In *Automation of Reasoning. 2 Classical Papers on Computational Logic 1967–1970*; Bolç, L., Bundy, A., Siekmann, J., Sloman, A., Eds.; Springer: Berlin/Heidelberg, Germany, 1983; pp. 466–483.
2. Hill , F.J.; Peterson, G.R. *Introduction to Switching Theory and Logical Design*; John Wiley & Sons: Hoboken, NJ, USA, 1981.
3. Whitesitt, J.E. *Boolean Algebra and Its Applications*; Addison-Wesley Publishing Company: Boston, MA, USA, 1962.
4. Gorbátov, V.A. *Fundamentos de la Matemática Discreta*; Mir: Moscow, Russia, 1988.
5. Büning, H.K.; Lettmann, T. *Propositional Logic: Deduction and Algorithms*; Cambridge University Press: Cambridge, UK, 1999.
6. Gill, A. *Applied Algebra for the Computer Sciences*; Prentice-Hall: Hoboken, NJ, USA, 1976.
7. Jackson, P.; Sheridan, D. Clause form Conversions for Boolean Circuits. In *SAT 2004, LNCS 3542*; Hoos, H.H., Mitchell, D.G., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 183–198.
8. Łukasiewicz, J. *Elements of Mathematical Logic*; Pergamon Press: Oxford, UK, 1966.
9. Lipovača, M. *Learn You a Haskell for Great Good*; No Starch Press: San Francisco, CA, USA, 2011.
10. Monk, J.D. *Mathematical Logic*; Springer: Berlin/Heidelberg, Germany, 1976.
11. Bourbaki, N. *Théorie des Ensembles*; Eléments de Mathématique; Hermann: Paris, France, 1977.
12. Kunen, K. *The Foundations of Mathematics*; Mathematical Logic and Foundations; College Publications: Norcross, GA, USA, 2009; Volume 19.