Notre Dame Journal of Formal Logic
Volume 34, Number 2, Spring 1993

BCK and BCI Logics, Condensed Detachment and the 2-Property

J. ROGER HINDLEY

Abstract Some of the main properties of the BCK and BCI logics of implication are summarized, focusing on their connections with their condensed logics and with combinators and lambda-calculus. (A condensed logic is the set of all formulas deducible from the logic's axioms by the condensed detachment rule of Carew Meredith.) A full proof is given of the preservation of the 2- and 1-2-properties by condensed detachment, based on ideas of S. Jaskowski.

1 Introduction and notation

1.0 Introduction This article is a summary of some of the main properties of implicational BCK and BCI logics, focusing on their connections with combinators and their condensed logics. (A condensed logic is the set of all formulas provable from the logic's axioms by Carew Meredith's rule of condensed detachment.)

The material outlined here will not be new; it will mainly be from the work of N. D. Belnap, M. W. Bunder, S. Hirokawa, and R. K. Meyer, but above all from that of S. Jaskowski.

The key definitions will be sketched in Section 1, then BCK-logic will be treated in Section 2 and BCI-logic in Section 3.

A proof of the preservation of the 2- and 1-2-properties by condensed detachment will be given in Section 4. It will be based on ideas from Jaskowski [24]. (A formula has the 2-property when each variable in it occurs exactly twice, and the 1-2 property when each variable in it occurs at most twice.) The first published proof of the preservation of the 2-property is in Belnap [3], which is probably not widely available now.

Most proofs other than the 2- and 1-2-preservation proof will be omitted. Detailed references to the literature will be included, however.

Received December 10, 1990; revised January 15, 1992

No attempt will be made to be comprehensive. BCK and BCI logics are at present under active study by several workers and significant new results have been discovered even in the short time since this paper was first drafted. (See the most recent papers in the References.) In particular, algebras related to BCK and BCI logics are outside the scope of this summary; so are connectives other than implication.

1.1 Propositional logic

Basic notation 1.1.1 *Implicational propositional formulas* will be denoted by "α", "β", "γ", "δ", ... They will contain no constants, only variables (denoted by "a", "b", "c", "d", . . .), and their only connective will be implication, "$\rightarrow$".

Syntactic identity of formulas will be denoted by "$\equiv$".

Each system to be discussed will use some of the following formulas as axioms:

(B)	$(a \rightarrow b) \rightarrow ((c \rightarrow a) \rightarrow (c \rightarrow b))$	(principal type of combinator **B**, see Notation 1.2.1)
(B′)	$(a \rightarrow b) \rightarrow ((b \rightarrow c) \rightarrow (a \rightarrow c))$	(principal type of **B′** or **CB**)
(C)	$(a \rightarrow (b \rightarrow c)) \rightarrow (b \rightarrow (a \rightarrow c))$	(principal type of **C**)
(I)	$a \rightarrow a$	(principal type of **I** or **CKK**)
(K)	$a \rightarrow (b \rightarrow a)$	(principal type of **K**)
(W)	$(a \rightarrow (a \rightarrow b)) \rightarrow (a \rightarrow b)$	(principal type of **W**).

The following two rules will be used:

(i) *Modus Ponens* or *Detachment:* from $\alpha \rightarrow \beta$ and α, deduce β.

(ii) *Substitution:* from α deduce $\sigma(\alpha)$, where σ is any substitution of formulas for variables which do not occur in non-axiom assumptions in the deduction above α.

Definition 1.1.2 For any set $\mathcal{A}$ of formulas, $\mathcal{A}$*-logic* is the set of all formulas (called $\mathcal{A}$*-theorems*) provable from members of $\mathcal{A}$ (called $\mathcal{A}$*-axioms*) by Modus Ponens and Substitution. Iff β is an $\mathcal{A}$-theorem we say

$$\vdash_{\mathcal{A}} \beta,$$

and iff β is deducible from $\mathcal{A}$-axioms and all or some of $\alpha_1, \ldots, \alpha_n$ we say

$$\alpha_1, \ldots, \alpha_n \vdash_{\mathcal{A}} \beta.$$

The special cases $\mathcal{A} = \{(B),(C),(K)\}$ and $\mathcal{A} = \{(B),(C),(I)\}$ are called BCK-*logic* and BCI-*logic*, respectively.

Notation 1.1.3: Variables and substitution The set of all variables occurring in a formula α is called

$$\mathrm{Vars}(\alpha).$$

Let $a_1, \ldots, a_k$ be distinct variables, $\delta_1, \ldots, \delta_k$ be any formulas, and σ be the operation of simultaneously substituting $\delta_1, \ldots, \delta_k$ for $a_1, \ldots, a_k$, respectively; we shall use the notation

$$\sigma = [\delta_1/a_1, \ldots, \delta_k/a_k], \qquad \sigma(\alpha) = [\delta_1/a_1, \ldots, \delta_k/a_k]\alpha.$$

We shall say the *domain* of σ is the set $\{a_1, \ldots, a_k\}$ and the *range* of σ is the set of all variables occurring in $\delta_1, \ldots, \delta_k$. The following notation from the literature will also be used.

(i) σ is an *alphabetic* or *trivial variation* of α (and $\sigma(\alpha)$ is an *alphabetic* or *trivial variant* of α) iff $\{a_1, \ldots, a_k\} \supseteq \mathrm{Vars}(\alpha)$ and $\delta_1, \ldots, \delta_k$ are mutually distinct variables.

(ii) σ is a *purely structural substitution* or a *structuring* relative to α iff $\{a_1, \ldots, a_k\} \supseteq \mathrm{Vars}(\alpha)$ and every variable occurring in $\delta_1, \ldots, \delta_k$ occurs only in one of $\delta_1, \ldots, \delta_k$, say in δ_i, and only occurs once in δ_i.

(iii) σ is a *variables-for-variables substitution* or an *identification* iff $\delta_1, \ldots, \delta_k$ are variables.

(iv) σ is a *unifier* of a pair $\{\alpha_1, \alpha_2\}$ of formulas iff $\sigma(\alpha_1) \equiv \sigma(\alpha_2)$.

(v) σ is a *most general unifier* of $\{\alpha_1, \alpha_2\}$ iff

(a) σ is a unifier of $\{\alpha_1, \alpha_2\}$, and

(b) every other unifier σ' of $\{\alpha_1, \alpha_2\}$ *factors through* σ in the sense that

$$\sigma'(\alpha_1) \equiv \sigma''(\sigma(\alpha_1)), \qquad \sigma'(\alpha_2) \equiv \sigma''(\sigma(\alpha_2))$$

for some σ'' depending on σ'.

Lemma 1.1.4 *For every formula α and substitution σ we can find a purely structural substitution σ_{struc} relative to α and a variables-for-variables substitution σ_{var} such that $\sigma_{\mathrm{var}}(\sigma_{\mathrm{struc}}(\alpha)) \equiv \sigma(\alpha)$. Also the range of σ_{struc} can be made disjoint from any given finite set of variables.*

Proof: ([24] Lemma 2.1) Let $a_1, \ldots, a_k$ be the variables in α and let

$$\sigma(\alpha) \equiv [\delta_1/a_1, \ldots, \delta_k/a_k]\alpha.$$

Construct formulas $\delta_1^*, \ldots, \delta_k^*$ from $\delta_1, \ldots, \delta_k$ by replacing each variable-occurrence in each δ_i by a distinct new variable. Then define

$$\sigma_{\mathrm{struc}} = [\delta_1^*/a_1, \ldots, \delta_k^*/a_k].$$

Definition 1.1.5: The condensed detachment rule (Rule D) This rule was introduced by Carew Meredith in the 1950's and is outlined as follows. (For details, see Hindley and Meredith [17] §2.)

Premises: any pair of formulas $\alpha \to \beta, \gamma$. To find a conclusion, first change γ by alphabetic variation to a γ' with no variables in common with α. Then seek a unifier of $\{\alpha, \gamma'\}$. (If none exists, there is no conclusion.) If a unifier exists, let σ be any most general unifier which satisfies the condition

$$\mathrm{Vars}(\sigma(\alpha)) \cap [\mathrm{Vars}(\beta) - \mathrm{Vars}(\alpha)] = \varnothing.$$

Conclusion: $\sigma(\beta)$.

The conclusion is easily shown to be unique modulo alphabetic variation and will be called here

$$\mathrm{D}((\alpha \to \beta), \gamma).$$

Definition 1.1.6: Condensed logics For any set $\mathcal{A}$ of formulas, *condensed $\mathcal{A}$-logic* is the set of all formulas provable from members of $\mathcal{A}$ by Rule D.

(Clearly, condensed $\mathcal{A}$-logic is a subset of $\mathcal{A}$-logic; for further discussion see [17] §7.)

Definition 1.1.7: The 2- and 1-2-properties A formula α has the *2-property* [*1-2-property*] iff each variable in α occurs exactly twice [at most twice].

Theorem 1.1.8: Preservation of 1-2 and 2 *Rule D preserves the* 1-2- *and* 2-*properties. That is, if* $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$ *exists, then*:
(i) *if* α_1 *and* $(\alpha_2 \to \beta_2)$ *have the* 1-2-*property, then so does* $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$;
(ii) *if* α_1 *and* $(\alpha_2 \to \beta_2)$ *have the* 2-*property, then so does* $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$.

Proof: See Section 4, Theorems 4.5 and 4.6.

1.2 Combinators

Notation 1.2.1 Combinatory logic and λ-calculus are formal theories of operators; in each theory there is a set of *terms* (CL-terms or λ-terms) and a *reducibility relation* $\triangleright$. For more details see Hindley and Seldin [18]; Chapter 2 for CL and Chapter 1 for λ-calculus. A short outline of CL is in [17] §§3–4 and a thorough treatment of λ-calculus is in Barendregt [2]. We shall just note some particularly relevant points here.

In both CL and λ-calculus a *combinator* is a term without free variables. In both theories one can define combinators **B**, **B′**, **C**, **I**, **K**, **W** with the reduction properties

$$\mathbf{B}XYZ \triangleright X(YZ), \qquad \mathbf{B}'XYZ \triangleright Y(XZ), \qquad \mathbf{C}XYZ \triangleright XZY,$$

$$\mathbf{I}X \triangleright X, \qquad \mathbf{K}XY \triangleright X, \qquad \mathbf{W}XY \triangleright XYY.$$

The $\triangleright$-relations in CL and λ-calculus do not correspond exactly; reducibility in CL is called *weak reducibility* ($\triangleright_{\mathrm{w}}$) and reducibility in λ-calculus *β-reducibility* ($\triangleright_{\lambda\beta}$).

A weak reducibility relation $\triangleright_{\lambda\mathrm{w}}$ can be defined in λ-calculus, see Hindley [16] §3; it is weaker than $\triangleright_{\lambda\beta}$ in the sense that $X \triangleright_{\lambda\mathrm{w}} Y$ implies $X \triangleright_{\lambda\beta} Y$ but not conversely.

In CL and λ-calculus a (*weak* or *β*-) *normal form* is an irreducible term.

Definition 1.2.2 In CL a **BCK**-*combinator* is any combinator built using occurrences of **B**, **C**, **K** only. Similarly: **BCI**-*combinators*, **BB′I**-*combinators*, etc.

In λ-calculus a **BCK**-λ-*term* is a λ-term T such that for each subterm $\lambda x.M$ of T, the M contains at most one free occurrence of x. A **BCK**-λ-term with no free variables is called a **BCK**-*combinator*.

A **BCI**-λ-*term* is a λ-term T such that for each subterm $\lambda x.M$ of T, the M contains exactly one free occurrence of x. A **BCI**-λ-term with no free variables is called a **BCI**-*combinator*.

There is a precise correspondence between **BCK**-combinators in CL and those in λ-calculus (and similarly for **BCI**); see [16] §1.

Notation 1.2.3: Type assignment *Types* are just formulas as defined in Basic Notation 1.1.1. The variables in types will often be called *type-variables* to contrast with variables that occur in terms.

Types are assigned to certain CL-terms and λ-terms by certain axioms and rules; for details see [18], Chapter 14 for CL and Chapter 15 for λ-calculus. For a term X whose free variables are among $x_1, \ldots, x_n$, the notation

$$x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash X : \beta$$

means that if we assign type α_1 to $x_1, \ldots, \alpha_n$ to x_n (and $x_1, \ldots, x_n$ are distinct), then we can deduce by the type-assignment rules that X receives type β. If we can make such a deduction, we call X *typable* or *stratified.*

It can be shown that each typable term receives a *principal type* (or *principal type-scheme* or *p.t.s.*) of which all its other types are substitution instances. For details see Hindley [15], summarized in [17] §6. The principal types of **B**, **B′**, **C**, **I**, **K**, **W** turn out to be just the formulas (B), (B′), (C), (K), (I), (W) in Basic Notation 1.1.1 above.

It can also be shown that if $x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash X : \beta$ and $x_1, \ldots, x_n$ are exactly the variables free in X, then there is a *principal deduction* giving, say,

$$x_1 : \gamma_1, \ldots, x_n : \gamma_n \vdash X : \delta,$$

such that every other deduction of a type for X from types for $x_1, \ldots, x_n$ is obtainable from the principal one by substituting types for type-variables.

Theorem 1.2.4: Strong normalization *In CL and λ-calculus every typable term X reduces to a normal form. Further, all reductions starting at X are finite.*

Proof: [18] Appendix 2, or other standard texts.

Theorem 1.2.5: Subject reduction *In CL and λ-calculus, types propagate down a reduction; i.e., if*

$$x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash X : \beta$$

and $X \rhd Y$, *then*

$$x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash Y : \beta.$$

Proof: [18] Chs. 14, 15, or other standard texts.

2 BCK-logic

2.1 Basics

Note 2.1.1 Roughly speaking, BCK-logic is the fragment of intuitionist logic in which an assumption α may be used at most once in a proof of $\alpha \to \beta$. The first study of a logic with this restriction seems to have been Fitch [13]. We shall only look at the implicational fragment of BCK-logic here.

Definition 2.1.2 **BCK**-*logic* (strictly speaking, the implicational fragment of BCK-logic) is the set of all formulas (as in Basic Notation 1.1.1) provable by the rules of Modus Ponens and Substitution from the following axioms:

$$\textbf{(B)}\ \ (a \to b) \to ((c \to a) \to (c \to b)),$$

$$\textbf{(C)}\ \ (a \to (b \to c)) \to (b \to (a \to c)),$$

$$\textbf{(K)}\ \ a \to (b \to a).$$

Theorem 2.1.3: Deduction theorem *If $\alpha, \delta_1, \ldots, \delta_n \vdash_{\mathbf{BCK}} \beta$ by a deduction in which α occurs as a non-axiom assumption at most once, then*

$$\delta_1, \ldots, \delta_n \vdash_{\mathbf{BCK}} (\alpha \to \beta).$$

Proof: Straightforward; Cf. [17] Theorem 1.12.

Discussion 2.1.4: Relations to other logics (i) The restriction on the use of α in the Deduction Theorem ties in with Lemma 2.3.3(iii) below which says, in effect, that in BCK-logic an assumption α can be used at most once in a proof of $\alpha \to \beta$. This makes BCK-logic much weaker than intuitionist and classical logics and allows it to serve as the base for a type-free mathematics in the style of the early work of Church and Curry. Church and Curry started their work using stronger logics and found their way blocked by contradictions, but if we weaken the underlying logic by forbidding multiple use of assumptions, the contradictions disappear and we can develop type-free higher-order mathematical systems and prove them consistent. (Some examples are in Fitch [13], Bunder [6], Bunder [7], Bunder and da Costa [11], and Komori [26].)

(ii) For implicational fragments we have

Classical logic $\supset$ *intuitionist logic* $\supset$ BCK-*logic.*

(Intuitionist implicational logic is known to coincide with BCKW-logic.)

(iii) The following two formulas are easily proved in BCK-logic, in fact in condensed BCK-logic (see later).

(I)	$a \to a$	(principal type of combinator **CKK**),
(B′)	$(a \to b) \to ((b \to c) \to (a \to c))$	(principal type of **CB**).

In contrast, the following is not provable in BCK-logic (cf. 2.1.4(i) above):

(W)	$(a \to (a \to b)) \to (a \to b)$	(principal type of **W**).

References 2.1.5 A passing mention of BCK-logic is made in Curry and Feys [12] §9F5, p. 338. Some more substantial references are: Meredith and Prior [28] §8 (summarized in Prior [32], p. 316 §12.71); Blok and Pigozzi [5] §5.2.3; Bunder [10]; also the references in 2.1.4(i) above, and all titles containing "BCK" in the references at the end of this paper.

2.2 Characterization of BCK-provability

Theorem 2.2.1: Characterization *A formula α is* BCK-*provable iff α is the result of a variables-for-variables substitution in a classical tautology β which has the 1-2-property.*

Proof: [24] §6.

Corollary 2.2.2 ([24] Lemma 6.1) *Each classical tautology with the* 1-2-*property is* BCK-*provable.*

Corollary 2.2.3 *For formulas α with the* 1-2-*property, classical logic is no stronger than* BCK *logic: i.e.,*

α is a classical tautology $\Leftrightarrow$ *α is provable in intuitionist logic*

$\Leftrightarrow$ *α is provable in* BCK-*logic.*

Proof: BCK $\Rightarrow$ Int $\Rightarrow$ Class, trivially. Class $\Rightarrow$ BCK by Corollary 2.2.2.

Theorem 2.2.4: Decidability of BCK *There is a decision procedure for* **BCK**-*provability which outputs a proof of each provable formula.*

Proof: [24] Theorem 6.5.

2.3 Correspondence between BCK-logic and combinators

Note 2.3.1 The connection between systems of implicational logic and systems of combinators was probably known to Curry since the 1930's. The main features of the correspondence between BCK-logic and combinators will be outlined here. For further details see [17] and [16].

Definition 2.3.2: BCK-abstraction in CL Let x be a term-variable. For each CL-term Y in which x occurs at most once, a CL-term called $\lambda^{abde}x.Y$ is defined thus (algorithm (*abde*) of [12] §6A):

(a) $\lambda^{abde}x.Y \equiv \mathbf{K}Y$	if x does not occur in Y;
(b) $\lambda^{abde}x.x \equiv \mathbf{I}$;	
(d) $\lambda^{abde}x.UV \equiv \mathbf{B}U(\lambda^{abde}x.V)$	if x occurs in V but not in U;
(e) $\lambda^{abde}x.UV \equiv \mathbf{C}(\lambda^{abde}x.U)V$	if x occurs in U but not in V.

Lemma 2.3.3

(i) $\lambda^{abde}x.Y$ *does not contain* x.

(ii) $(\lambda^{abde}x.Y)x \rhd_w Y$.

(iii) *If x occurs in Y more than once, then there is no way to define abstraction with respect to x using only* **B**, **C**, **K**; *i.e., there is no term Y^*, composed only of* **B**, **C**, **K** *and parts of Y other than x, with the property $Y^*x \rhd_w Y$.*

Proof: (i)–(ii) are straightforward, cf. [12] §6A, Theorem 2. For (iii): the reductions for **B**, **C**, **K** in Basic Notation 1.1.1 cannot change the single occurrence of x in Y^*x to multiple occurrences in Y.

Theorem 2.3.4: Abstraction and p.t.s. *Abstraction by algorithm* (*abde*) *preserves principal types. That is, if $x, y_1, \ldots, y_m$ include all the variables free in Y and x occurs at most once in Y, and there is a principal deduction giving*

$$x:\alpha,\ y_1:\delta_1, \ldots, y_m:\delta_m \vdash Y:\beta,$$

then there is a principal deduction giving

$$y_1:\delta_1, \ldots, y_m:\delta_m \vdash \lambda^{abde}x.Y:(\alpha \to \beta).$$

Proof: Straightforward case-checking in the definition of $\lambda^{abde}x.Y$.

Theorem 2.3.5: Typability for BCK *In* CL *and λ-calculus all* **BCK**-*combinators are typable.*

Proof: [16] Corollary 4.1.1.

Theorem 2.3.6: Subject-conversion for BCK

(i) *If X and Y are* **BCK**-*combinators and* $X \rhd Y$ *by weak reduction in* CL *or* λ-*calculus, then X and Y receive exactly the same types and hence have the same principal types.*

(ii) *In* CL *and* λ-*calculus, all reductions of a* **BCK**-*combinator terminate.*

(iii) *In* CL *the principal type of a* **BCK**-*combinator is the same as that of its weak normal form.*

Proof:

(i) In λ-calculus this is [16] Theorem 3.8. In CL, note that every type of X is a type of Y by Theorem 1.2.5. To prove the converse, note that by Theorem 2.3.5 all **BCK**-combinators are typable, so any subterms of X that are cancelled in the reduction $X \rhd_w Y$ are typable. Hence we can apply the subject-expansion theorem in [12] §9C4.

(ii) Reducing a **BCK**-combinator does not duplicate any parts of terms, so terms shorten as they are reduced.

(iii) By (i).

Warning 2.3.7 In λ-calculus Theorem 2.3.6(i) and (iii) fail for $\rhd_{\lambda\beta}$, and the principal type of a **BCK**-combinator may differ from that of its β-normal form; see D. Meredith's counter-example in [16] §3.4.

Note 2.3.8 The study of **BCK**-combinators and their types is by no means complete; for further results see especially the papers by Bunder, Hirokawa, and Komori in the References.

2.4 Condensed BCK-logic

Definition 2.4.1 Condensed BCK-*logic* is the set of all formulas provable from (B), (C), and (K) by Rule D. These formulas are called *condensed* BCK-*theorems.*

Theorem 2.4.2 *The condensed* BCK-*theorems are exactly the principal types of* BCK-*combinators.*

Proof: [17] Theorem 6.7.

Theorem 2.4.3 *If α is a condensed* BCK-*theorem, then so is every formula obtained from α by a purely structural substitution.*

Proof: Meyer and Bunder [30] show that this holds for every logic that extends BB′I-logic, and BCK extends BB′I by 2.1.4(iii).

Theorem 2.4.4: The BCK-1-2 theorem *Every condensed* BCK-*theorem has the* 1-2-*property.*

Proof: Axioms (B), (C), (K) clearly have the 1-2-property. And by Theorem 1.1.8(i), Rule D preserves it.

Theorem 2.4.5: Uniqueness (Hirokawa [19]) *In λ-calculus, for each type α there is at most one* **BCK**-*combinator in β-normal form whose principal type is α.*

2.5 *Relation between BCK and condensed BCK*

Note 2.5.1 Every BCK-theorem is a substitution instance of a condensed BCK-theorem (because all the types of a combinator are instances of its principal type). But further, we have the following.

Theorem 2.5.2 *Every* BCK-*theorem is obtainable from a condensed* BCK-*theorem by a variables-for-variables substitution.*

Proof: By Theorem 2.4.3 and Lemma 1.1.4. (Another proof is in Bunder [10].)

Warning 2.5.3 It is tempting to conjecture that a BCK-theorem is a condensed BCK-theorem iff it has the 1-2-property. But this is false; for example, the formula

$$\alpha \equiv a \to (a \to (b \to b))$$

has the 1-2-property and is a BCK-theorem, being a type of $\mathbf{K(KI)}$, but it is not a condensed BCK-theorem. (If α was a p.t.s. of a **BCK**-combinator X, then it would be a p.t.s. of the β-normal form of X. But this is impossible, because by Ben-Yelles [4] Chapter 3 the only β-normal form with type α is $\lambda xyz.z$, for which this type is not a p.t.s.)

Theorem 2.5.4: Incompleteness of D *Rule* D *is not complete for* BCK-*logic, i.e., there are* BCK-*theorems that are not condensed* BCK-*theorems.*

Proof: By 2.5.3 above.

3 *BCI-logic*

3.1 *Definition and basics*

Note 3.1.1 BCI-logic seems to have been first studied in the 1950's by Carew Meredith along with BCK-logic. Roughly speaking, a proof of $\alpha \to \beta$ is a BCI-proof iff the assumption α is used exactly once in it. Here we shall be concerned with implication only.

Definition 3.1.2 BCI-*logic* (or strictly speaking, its implicational fragment) is the set of all formulas provable by Modus Ponens and Substitution (see Basic Notation 1.1.1) from the axioms

$$\textbf{(B)} \quad (a \to b) \to ((c \to a) \to (c \to b)),$$

$$\textbf{(C)} \quad (a \to (b \to c)) \to (b \to (a \to c)),$$

$$\textbf{(I)} \quad a \to a.$$

Theorem 3.1.3: Deduction theorem *If $\alpha, \delta_1, \ldots, \delta_n \vdash_{\mathrm{BCI}} \beta$ by a deduction in which α occurs as a non-axiom assumption exactly once, then*

$$\delta_1, \ldots, \delta_n \vdash_{\mathrm{BCI}} (\alpha \to \beta).$$

Proof: Cf. [17] Theorem 1.12.

Discussion 3.1.4: Relations to other logics (i) This theorem ties in with Lemma 3.3.3(iv) below which says in effect that in BCI-logic a premise α cannot be used more than once in a proof of $\alpha \to \beta$ (and must be used at least once). The condition that α must be used at least once implies that BCI-logic is a restriction of Church's relevance logic $\mathbf{R}_{\to}$ whose axioms are **(B)**, **(C)**, **(I)**, **(W)**. (See Anderson and Belnap [1] Chapter 1 §3; the theorems of $\mathbf{R}_{\to}$ are exactly the types of λ**I**-combinators.)

(ii) By 2.1.4(iii) the formula (I) is a BCK-theorem and (B′) is a BCI-theorem, so

BCK-*logic* ⊃ BCI-*logic* ⊃ BB′I-*logic*.

(iii) The system called Linear Logic developed in the 1980's by J-Y. Girard as part of his analysis of the concept of computation contains an implication concept satisfying the condition that a premise α must be used exactly once in a proof of $\alpha \to \beta$. Implicational BCI-logic is in fact a fragment of Linear Logic, though a very small fragment.

References 3.1.5 The earliest published results on BCI-logic are in [28] §7, where they are attributed to work of Meredith done in 1956 and presumably not published then. They are summarized in [32] p. 316 §12.7. For later work see Meredith [29], Blok and Pigozzi [5] §5.2.3, and the references in the present Section 3.

3.2 Characterization of BCI-provability

Theorem 3.2.1: Characterization *A formula α is* BCI-*provable iff α is the result of a variables-for-variables substitution in a formula β which is a* C′-*tautology* (*see below*) *with the* 2-*property.*

Proof: Jaskowski [24] §6.

Definition 3.2.2 ([24] §3) A C′-*tautology* is a formula that takes the values T or $\frac{1}{2}$ but not F, for each valuation of its variables, when "→" is interpreted by the following matrix:

→	T	½	F
T	T	F	F
½	T	½	F
F	T	T	T

Theorem 3.2.3: C′ and Classical Tautologies *A formula α with the* 1-2-*property is a* C′-*tautology iff*

(i) *α is a classical tautology with the* 2-*property, and*

(ii) *α is not the result of a variables-for-variables substitution in a classical tautology without the* 2-*property.*

Proof: [24] Theorem 6.3.

Claim 3.2.4 *No variable occurs just once in a* C′-*tautology, so every* C′-*tautology with the* 1-2-*property has the* 2-*property.*

Proof: (M. W. Bunder, informal communication.) Let v occur only once in α. By [24] Lemma 3.4, if v is given the value T and all other variables the value $\frac{1}{2}$, then α gets the value F or T according as the position of v is negative or positive in α. The following dual lemma is also easy to prove: if v is given the value F and all other variables the value $\frac{1}{2}$, then α gets the value F or T according as the position of v is positive or negative in α. By these two lemmas, α is not a C′-tautology.

Warning 3.2.5 The use of C′-tautologies in the Characterization Theorem 3.2.1 could not be replaced by that of classical tautologies, as not every classical tautology with the 2-property is a BCI-theorem.

Proof: Consider the formula

$$a \to (a \to (b \to b)).$$

If this was a BCI-theorem, it would be a type in λ-calculus of a **BCI**-combinator X, and hence of its β-normal form X^*. But the only β-normal form with this type is $\lambda xyz.z$ (see Warning 2.5.3), and this is not a **BCI**-combinator.

Theorem 3.2.6: Axiomatization of C′ *The following are equivalent:*

(i) α *is a* C′*-tautology*;

(ii) α *is provable in the implicational fragment* $\mathbf{RM}_{\to}$ *of the logic* **RM** *of* [1] §8.15;

(iii) α *is provable by Modus Ponens and Substitution from the axioms* (B), (C), (I), (W) *and the following*:

$$\textbf{(M)} \quad a \to (a \to a) \qquad (\textit{the "mingle" axiom}),$$

$$(*) \quad ((((a \to b) \to b) \to a) \to c) \to (((((b \to a) \to a) \to b) \to c) \to c).$$

Proof: (i) ⇔ (ii): [1] §29.3.2 (by R. K. Meyer).

(ii) ⇒ (iii): By Meyer and Parks [31], $\mathbf{RM}_{\to}$ is axiomatizable by (B′), (W), (∗) and

$$\textbf{(CI)} \qquad a \to ((a \to b) \to b).$$

But (CI) is the p.t.s. of **CI** and (B′) is the p.t.s. of **CB**, so (B′), (W), (∗), and (CI) are provable from (B), (C), (I), (W), (M), (∗).

(iii) ⇒ (ii): (B), (C), (I), and (M) are easily seen to be C′-tautologies; hence, since (i) ⇔ (ii), they are $\mathbf{R}_{\to}$-theorems. Also (W) and (∗) are $\mathbf{R}_{\to}$-theorems by the result in Meyer and Parks [31] quoted above.

Theorem 3.2.7: Decidability of BCI *There is a decision procedure for* BCI-*provability which outputs a proof of each provable formula.*

Proof: [24] Theorem 6.5.

3.3 Corresponding combinators

Note 3.3.1 A formula is BCI-provable iff it is a type of a **BCI**-combinator. (See [17] and [16] for details.)

Definition 3.3.2: BCI-abstraction in CL Let x be a term-variable. For each CL-term Y in which x occurs exactly once, a CL-term called $\lambda^{bde}x.Y$ is defined by algorithm (bde) of [12] §6A thus.

(b) $\lambda^{bde}x.x \equiv \mathbf{I}$;
(d) $\lambda^{bde}x.UV \equiv \mathbf{B}U(\lambda^{bde}x.V)$ if x occurs in V but not in U;
(e) $\lambda^{bde}x.UV \equiv \mathbf{C}(\lambda^{bde}x.U)V$ if x occurs in U but not in V.

Lemma 3.3.3

(i) *$\lambda^{bde}x.Y$ does not contain x;*
(ii) *$\lambda^{bde}x.Y$ is defined iff x occurs in Y exactly once;*
(iii) *Further, if x does not occur in Y exactly once, then there is no way to define abstraction with respect to x using only* **B**, **C**, **I**; *i.e., there is no term Y^*, composed only of* **B**, **C**, **I** *and parts of Y other than x, with the property that $Y^*x \rhd_w Y$.*

Proof:

(i)–(ii) Cf. [12] §6A Theorem 2.
(iii) The reductions for **B**, **C**, **I** in Basic Notation 1.1.1 cannot cancel or duplicate the single occurrence of x in Y^*x.

Theorem 3.3.4: Abstraction and p.t.s. *Abstraction by algorithm (bde) preserves principal types. That is, if $x, y_1, \ldots, y_m$ include all the variables free in Y and x occurs exactly once in Y, and there is a principal deduction giving*

$$x:\alpha,\ y_1:\delta_1, \ldots, y_m:\delta_m \vdash Y:\beta,$$

then there is a principal deduction giving

$$y_1:\delta_1, \ldots, y_m:\delta_m \vdash \lambda^{bde}x.Y:(\alpha \to \beta).$$

Proof: Special case of Theorem 2.3.4.

Theorem 3.3.5: Typability for BCI ([16] §5) *In CL and λ-calculus all* **BCI**-*combinators are typable.*

Theorem 3.3.6: Subject-conversion for BCI

(i) *Let X and Y be* **BCI**-*combinators. If $X \rhd_W Y$ in* CL *or $X \rhd_{\lambda\beta} Y$ in λ-calculus, then X and Y receive exactly the same types and hence have the same p.t.s.*
(ii) *In* CL *and λ-calculus, all reductions of a* **BCI**-*combinator terminate.*
(iii) *The p.t.s. of a* **BCI**-*combinator is the same as that of its normal form (weak normal form in* CL, *β-normal form in λ-calculus).*

Proof:

(i) In λ-calculus this is [16] Theorem 5.1. In CL, the subject-expansion theorem in [12] §9C4 applies, since reducing a **BCI**-combinator does not duplicate or cancel any terms.
(ii) By Theorem 2.3.6(ii).
(iii) By (i).

3.4 Condensed BCI-logic

Definition 3.4.1 *Condensed* BCI-*logic* is the set of all formulas provable from (B), (C), and (I) by Rule D (Definition 1.1.5). These formulas are called *condensed* BCI-*theorems.*

Theorem 3.4.2 *The condensed* BCI-*theorems are exactly the principal types of* **BCI**-*combinators.*

Proof: [17] Theorem 6.7.

Theorem 3.4.3 *If α is a condensed* BCI-*theorem then so is every formula obtained from α by a purely structural substitution.*

Proof: [30].

Theorem 3.4.4: The BCI-2 Theorem *Every condensed* BCI-*theorem has the* 2-*property.*

Proof: The axioms (B), (C), (I) clearly have it and Rule D preserves it by Theorem 1.1.8(ii).

Theorem 3.4.5: Decidability of condensed BCI *There is a decision procedure for provability in condensed* BCI-*logic which outputs a proof of each provable formula.*

Proof: Theorems 3.2.7 and 3.5.3.

Theorem 3.4.6: Characterization of condensed BCI *The theorems of condensed* BCI-*logic are exactly the* C′-*tautologies with the* 2-*property.*

Proof: Theorems 3.2.1 and 3.5.3.

Note 3.4.7 An alternative characterization of condensed BCI-theorems is in Hirokawa [20] Theorem 3.

3.5 Relation between BCI and condensed BCI

Note 3.5.1 Every BCI-theorem is a substitution instance of a condensed BCI-theorem (just as all the types of a combinator are instances of its p.t.s.). More strongly, we have the following.

Theorem 3.5.2 *Every* BCI-*theorem is obtainable from a condensed* BCI-*theorem by a variables-for-variables substitution.*

Proof: By Theorem 3.4.3 and Lemma 1.1.4. (Another proof is in [10].)

Theorem 3.5.3 *A* BCI-*theorem is a condensed* BCI-*theorem iff it has the* 2-*property.*

Proof: (Cf. Meyer and Bunder [30]) By [3] every condensed BCI theorem has the 2-property. For the converse, let α be a BCI-theorem with the 2-property. By Theorem 3.5.2, α is obtainable by a variables-for-variables substitution from a condensed BCI-theorem α^*, but by [3] α^* has the 2-property as well as α, so α^* must be just an alphabetic variant of α.

Theorem 3.5.4: Incompleteness of D *Rule* D *is not complete for* BCI-*logic, i.e., there are* BCI-*theorems that are not condensed* BCI-*theorems.*

Proof: The formula $(a \to a) \to (a \to a)$ is a BCI-theorem (because it is a type that can be assigned to **I**), but by Theorem 3.5.3 it is not a condensed BCI-theorem because it does not have the 2-property.

3.6 Particular BCI-theorems The following are some BCI-theorems which may be of interest. They are also condensed BCI-theorems. In the list each theorem is shown on the left and the combinator of which it is a principal type is on the right.

Parentheses omitted from formulas should be restored by association to the right.

3.6.1 The following are from [30] and are provable in condensed BB′I-logic as well as BCI. Let $\mathbf{B}' \equiv \mathbf{CB}$.

(1) $(a \to a') \to (b \to b') \to (a' \to b) \to (a \to b')$
$\lambda uvwx.v(w(ux))$ or $\mathbf{B(B'B)(BBB')}$;

(2) $(a \to a') \to (b \to c \to c') \to b \to (c' \to a) \to (c \to a')$
$\lambda uvwxy.u(x(vwy))$ or $\mathbf{BB(B(B'B')B)B}$;

(3) $(a \to a') \to (b \to c \to c') \to b \to (a' \to c) \to (a \to c')$
$\lambda uvwxy.vw(x(uy))$ or $\mathbf{BB(B(B'B)(BBB'))}$;

(4) $(a \to c \to c') \to (b \to c' \to c'') \to a \to b \to c \to c''$
$\lambda uvwxy.vx(uwy)$ or $\mathbf{B(B(B'B')B')(BB')}$;

3.6.2 The following are from [24].

(1) $a_1 \to \ldots \to a_n \to (a_1 \to \ldots \to a_n \to b) \to b$ $\qquad \lambda u_1 \ldots u_n v.vu_1 \ldots u_n$;

(2) $(a \to b_1 \to \ldots \to b_n \to a' \to c) \to (a' \to b_1 \to \ldots \to b_n \to a \to c)$
$\lambda uvw_1 \ldots w_n x.uxw_1 \ldots w_n v$;

(3) $(b_1 \to \ldots \to b_n \to a \to c) \to (a \to b_1 \to \ldots \to b_n \to c)$
$\lambda uvw_1 \ldots w_n \cdot uw_1 \ldots w_n v$;

(4) $(a \to b \to c) \to (d \to a) \to (b \to d \to c)$ $\qquad \lambda uvwx.u(vx)w$;

(5) $(a \to a') \to (b_1 \to \ldots \to b_n \to a' \to c) \to (b_1 \to \ldots b_n \to a \to c)$
$\lambda uvw_1 \ldots w_n x.vw_1 \ldots w_n(ux)$;

(6) $(a \to a') \to ((b_1 \to \ldots \to b_n \to a \to c) \to d) \to ((b_1 \to \ldots \to b_n \to a' \to c) \to d)$
$\lambda uvw.v(\lambda x_1 \ldots x_n y.wx_1 \ldots x_n(uy))$;

(7) $(a \to b \to a') \to ((c_1 \to \ldots \to c_n \to a \to d) \to e)$
$\to b \to ((c_1 \to \ldots \to c_n \to a' \to d) \to e)$
$\lambda uvwx.v(\lambda y_1 \ldots y_n z.xy_1 \ldots y_n(uzw))$;

(8) $(a \to b) \to (c \to d) \to c \to (d \to a) \to b$ $\qquad \lambda uvwx.u(x(vw))$.

Note 3.6.3 Formula (5) in 3.6.2 is also provable in condensed BB′I-logic, being a p.t.s. of $\mathbf{(BB)}^n\mathbf{B}'$, where X^nY is $X(X(\ldots(XY)\ldots))$ with n X's.

4 The 1-2- and 2-preservation theorems

4.1 Introduction In this section we shall prove that Rule D preserves the 1-2- and 2-properties (as stated in Theorem 1.1.8).

The first published proof of either of these results was the proof of 2-preservation in Belnap [3]. Belnap's method also shows that Rule D preserves the 1-2-property, although he did not claim this; in fact the 1-2-theorem needs slightly less proof. The core of [3] is an analysis of a series of repeated compositions of two one-to-one functions.

It may be of interest that the proof of a different result in Jaskowski [24] contains an argument rather like Belnap's, that would probably have led Jaskowski to the 1-2-preservation theorem had be been concerned with it. Jaskowski's argument is an analysis of two equivalence relations instead of two one-to-one functions, and does not go so far as Belnap's. It would not have given Belnap's 2-theorem without a little extra work, but would have given the slightly easier 1-2-theorem with very little modification.

In this section the 1-2-preservation theorem is proved by a modified Jaskowski argument. A proof of the 2-preservation theorem is given afterwards by adding an argument of Belnap's onto the end of Jaskowski's.

Note 4.2 Other proofs of the 1-2- and 2-preservation theorems are in Meyer and Bunder [30] and Bunder [8].

The 2-preservation theorem and its background are discussed in Kalman [25], though no proof is given. It is stated and used in Wos et al. [33] which refers to Belnap for the proof.

Lemma 4.3 *Purely structural substitutions preserve the* 1-2- *and* 2-*properties. That is, if α has the* 2- *or* 1-2-*property and σ is a purely structural substitution relative to α then $\sigma(\alpha)$ has the same property* (2- *or* 1-2-) *as α.*

Proof: Let $\sigma(\alpha) = [\delta_1/a_1, \ldots, \delta_k/a_k](\alpha)$. If a variable a_i occurs exactly n times in α and is replaced by n copies of δ_i in $\sigma(\alpha)$, then each variable in δ_i occurs exactly once in each copy and hence occurs exactly n times in $\sigma(\alpha)$.

Lemma 4.4 (*i*) *Let $\{\alpha_1, \alpha_2\}$ have a most general unifier σ. Then there exist a purely structural substitution σ_{struc} relative to α_1 and α_2, and a variables-for-variables substitution σ_{var}, such that*

$$\sigma_{\text{var}}(\sigma_{\text{struc}}(\alpha_i)) \equiv \sigma(\alpha_i) \qquad (i = 1,2),$$

and if we define $\alpha_i^* \equiv \sigma_{\text{struc}}(\alpha_i)$, then σ_{var} is a most general unifier of $\{\alpha_1^*, \alpha_2^*\}$.

(ii) *If α_1 and α_2 have no common variables then σ_{struc} can be defined so that there are no variables common to α_1^* and α_2^*.*

Proof: For (i): apply Lemma 1.1.4 to σ. Clearly σ_{var} unifies $\{\alpha_1^*, \alpha_2^*\}$. To show σ_{var} is most general, suppose another substitution τ also unifies $\{\alpha_1^*, \alpha_2^*\}$. Then the composition $\tau \circ \sigma_{\text{struc}}$ unifies $\{\alpha_1, \alpha_2\}$. But the m.g.u. of $\{\alpha_1, \alpha_2\}$ is σ, so

$$\tau \circ \sigma_{\text{struc}} = \rho \circ \sigma$$

for some substitution ρ. Then for $i = 1,2$,

$$\tau(\alpha_i^*) \equiv \tau(\sigma_{\text{struc}}(\alpha_i)) \equiv \rho(\sigma(\alpha_i)) \equiv \rho(\sigma_{\text{var}}(\alpha_i^*)),$$

so σ_{var} is most general. For (ii): use the "also" clause in Lemma 1.1.4.

Theorem 4.5: The D-1-2 theorem *Rule* D (*the condensed detachment rule*) *preserves the* 1-2-*property. That is, if α_1 and $(\alpha_2 \to \beta_2)$ have the* 1-2-*property and* $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$ *exists, then it has the* 1-2-*property.*

Proof: (Based on the proofs of [24] Theorems 1.1 and 2.2.) Let $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$ exist and α_1 and $(\alpha_2 \to \beta_2)$ have the 1-2-property. Make an alphabetic variation if necessary to ensure that α_1 and $(\alpha_2 \to \beta_2)$ have no common variables. A fairly routine proof shows that this will leave $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$ unchanged modulo alphabetic variation. Let σ be an m.g.u. of $\{\alpha_1, \alpha_2\}$ satisfying the condition in Rule D, namely

$$\mathrm{Vars}(\sigma(\alpha_2)) \cap [\mathrm{Vars}(\beta_2) - \mathrm{Vars}(\alpha_2)] = \varnothing.$$

By Lemma 4.4 we can assume that σ is a variables-for-variables substitution. (Note that by Lemma 4.3, $\sigma_{\mathrm{struc}}(\alpha_1)$ and $\sigma_{\mathrm{struc}}(\alpha_2 \to \beta_2)$ have the 1-2-property, where σ_{struc} is determined by Lemma 4.4.) Let

(1) $$\alpha \equiv \sigma(\alpha_1) \equiv \sigma(\alpha_2), \qquad \beta \equiv \sigma(\beta_2).$$

Consider the formula $\alpha_2 \to \beta_2$: number all the variable-places in it from left to right; say $1, \ldots, n$ are in α_2 and $n+1, \ldots, m$ are in β_2. Now σ substitutes only variables for variables, so by (1) there are the same number of variable-places in α and in α_1 as in α_2, and in β as in β_2; number all these places from left to right as $1, \ldots, n$ in $\alpha, 1, \ldots, n$ in α_1, and $n+1, \ldots, m$ in β.

Define the following relations on the set $\{1, \ldots, m\}$ (cf. the proof of [24] Theorem 2.2).

$i \sim_1 j$ *iff* *either* $i = j > n$, *or* $i \le n$ *and* $j \le n$ *and the* i*-th and* j*-th places in* α_1 *contain the same variable.*

$i \sim_2 j$ *iff* *the* i*-th and* j*-th places in* $\alpha_2 \to \beta_2$ *contain the same variable.*

$i \sim_0 j$ *iff* *there exist* $i_1, \ldots, i_{r+1}$ $(r \ge 0)$ *such that* $i_1 = i$, $i_{r+1} = j$, *and* $(\forall k \le r)$ $i_k \sim_1 i_{k+1}$ *or* $i_k \sim_2 i_{k+1}$.

Clearly $\sim_1$ and $\sim_2$ are equivalence relations, and hence so is $\sim_0$. Also

(2) $$i \sim_2 j \Rightarrow i \sim_0 j.$$

Let τ be a variables-for-variables substitution into $\alpha_2 \to \beta_2$ with the property that the variables in the i-th and j-th places are replaced by the same variable iff $i \sim_0 j$. Such a τ exists by (2) and the definition of $\sim_2$.

Note that $\tau(\alpha_2)$ is a substitution instance of α_1, since $i \sim_1 j \Rightarrow i \sim_0 j$. Say $\tau(\alpha_2) \equiv \rho(\alpha_1)$. Since α_1 and $\alpha_2 \to \beta_2$ have no common variables, we can define a substitution τ' which has the effect of τ on $\alpha_2 \to \beta_2$ and the effect of ρ on α_1. Then τ' unifies $\{\alpha_1, \alpha_2\}$.

Further, any unifier of $\{\alpha_1, \alpha_2\}$ must substitute the same formulas into places i and j such that $i \sim_0 j$. Therefore τ' is an m.g.u. of $\{\alpha_1, \alpha_2\}$.

Hence τ' is the same as σ, modulo alphabetic variation, so by (1) we can assume

$$\alpha \equiv \tau'(\alpha_1) \equiv \tau'(\alpha_2), \qquad \beta \equiv \tau'(\beta_2).$$

We must prove that β has the 1-2-property. By the definition of τ and τ' it is enough to show that each place-number i in β is related by $\sim_0$ to at most one other place-number in β.

Let i be a place-number in β, and consider any chain of form

(3) $$i = i_1 \sim_{q_1} i_2 \sim_{q_2} i_3 \sim_{q_3} \cdots \sim_{q_r} i_{r+1}$$

where $q_k = 1$ or 2 for each k. By cutting out repetitions we can assume that $i_1, \ldots, i_{r+1}$ are distinct. We must show that the chain contains at most one other number in β besides i_1.

Now i_1 is in β, so it is related by $\sim_1$ to nothing else (by the definition of $\sim_1$), and by $\sim_2$ to at most one other number (since $\alpha_2 \to \beta_2$ has the 1-2-property). So $q_1 = 2$ and i_2 is uniquely determined if it exists. (If it does not exist then $r = 0$.)

For $k = 2, \ldots, r+1$, the number i_k is related by $\sim_1$ and $\sim_2$ to at most one other number each, since α_1 and $\alpha_2 \to \beta_2$ both have the 1-2-property; hence $q_k = 2$ if $q_{k-1} = 1$, and $q_k = 1$ if $q_{k-1} = 2$; also i_{k+1} is uniquely determined if it exists.

But if i_k is in β, then it is related by $\sim_1$ to nothing else and by $\sim_2$ to only one other number. This number must be i_{k-1}, otherwise i_k would not be in the chain, so $q_{k-1} = 2$ and the chain cannot continue beyond i_k.

Thus the chain ends as soon as it meets an i_k in β with $k \geq 2$, so it contains at most two place-numbers in β, as required.

Theorem 4.6: The D-2 theorem *Rule* D (*the condensed detachment rule*) *preserves the* 2-*property. That is, if* α_1 *and* $(\alpha_2 \to \beta_2)$ *have the* 2-*property and* $\mathrm{D}((\alpha_2 \to \beta_2), \alpha_1)$ *exists, then it has the* 2-*property.*

Proof: Let α_1 and $\alpha_1 \to \beta_2$ both have the 2-property. Follow the proof of Theorem 4.5; the only extra thing is that now we must show that each place-number i in β is always related to another place-number j in β by $\sim_0$. This is done as follows (adapted from [3]).

The key fact is that a chain (3) where each i_k is a distinct place-number cannot have more members than the number, m, of places in $\alpha_2 \to \beta_2$.

Let i be in β. We shall construct a chain (3) with $i_1 = i$ and extend it as far as possible, and show that it must eventually contain another number in β besides i.

Define $i_1 = i$. Then i_1, being in β, is related by $\sim_1$ only to itself and by $\sim_2$ to exactly one other place-number, call it i_2. If i_2 is in β, then the chain has the required two members in β. If not, then we extend the chain two steps at a time, as follows.

Suppose we have already constructed a chain with an even number of members, all distinct:

(4) $$i_1 \sim_2 i_2 \sim_1 \ldots \sim_1 i_{2h-1} \sim_2 i_{2h}.$$

(Note that $\sim_1$, $\sim_2$ must alternate and begin with $\sim_2$, by the proof of Theorem 4.5.)

Suppose i_{2h} is in α_1. Then by the definition of $\sim_1$ and since α_1 has the 2-property, i_{2h} is related by $\sim_1$ to exactly one other number and this number is in α_1; call it i_{2h+1}.

Now i_{2h+1} is distinct from $i_1, \ldots i_{2h}$. To see this, note first that $i_{2h+1} \neq i_{2h}$ by definition of i_{2h+1}. Next, if $i_{2h+1} = i_j$ for some $j \leq 2h-2$, then i_{2h}, being the unique number related to i_{2h+1} by $\sim_1$, would be the same as i_{j+1} or i_{j-1}, contrary to the distinctness of $i_1, \ldots, i_{2h}$. Finally, if $i_{2h+1} = i_{2h-1}$, then i_{2h+1} would be related by $\sim_1$ to i_{2h-2} and to i_{2h}, so these two would be the same, contrary to the distinctness of $i_1, \ldots, i_{2h}$. Thus we now have the following chain of distinct numbers:

(5) $$i_1 \sim_2 i_2 \sim_1 \ldots \sim_1 i_{2h-1} \sim_2 i_{2h} \sim_1 i_{2h+1}.$$

Now i_{2h+1} is in $\alpha_2 \to \beta_2$ which has the 2-property, so i_{2h+1} is related by $\sim_2$ to exactly one other number; call it i_{2h+2}. By the same argument as above, it is distinct from $i_1, \ldots, i_{2h+1}$. Thus the chain (4) has now been extended from $2h$ to $2h+2$ distinct members.

Such an extension is always possible when i_{2h} is in α_1. But there are only m place-numbers to choose from, so the chain must stop and thus some i_{2h} must be in β.

Acknowledgment I am very grateful to Martin Bunder for much useful information, help, and criticism during the preparation of this paper, and to Sachio Hirokawa, Yuichi Komori, David Meredith, Robert Meyer, and the referee for information about some of the results listed. I am also very grateful to the Australian Research Council for financial support and the Mathematics Department of the University of Wollongong for generous hospitality.

REFERENCES

[1] Anderson, A. and N. Belnap, *Entailment, Volume 1*, Princeton University Press, Princeton, 1975.

[2] Barendregt, H., *The Lambda Calculus*, Second Edition, North-Holland, Amsterdam, 1984.

[3] Belnap, N., "The two-property," *Relevance Logic Newsletter* (informally distributed), vol. 1 (1976), pp. 173–180.

[4] Ben-Yelles, C., *Type-Assignment in the Lambda Calculus; Syntax and Semantics*, thesis, University College, Swansea, 1979.

[5] Blok, W. and D. Pigozzi, *Algebraizable Logics*, Memoirs of the American Mathematical Society No. 396, American Mathematical Society, Providence, 1989.

[6] Bunder, M., "BCK-predicate logic as a foundation for multiset theory," unpublished manuscript, 1985.

[7] Bunder, M., "Some consistency proofs and a characterization of inconsistency proofs in illative combinatory logic," *Journal of Symbolic Logic*, vol. 52 (1987), pp. 89–110.

[8] Bunder, M., "The 2 and 1-2 properties in condensed BCI and BCK logics," informal notes, 1989.

[9] Bunder, M., "Corrections to some results for BCK logics and algebras," *Logique et Analyse*, vol. 31 (1991), pp. 115–122.

[10] Bunder, M., *Standardization of Proofs in Propositional Logic*, unpublished manuscript, 1990.

[11] Bunder, M. and N. da Costa, "On BCK logic and set theory," Preprint 4/86, Mathematics Dept., University of Wollongong, 1986.

[12] Curry, H. and R. Feys, *Combinatory Logic, Volume 1*, North-Holland, Amsterdam, 1958.

[13] Fitch, F., "A system of formal logic without an analogue to the Curry W operator," *Journal of Symbolic Logic*, vol. 1 (1936), pp. 92–100.

[14] Girard, J., "Linear logic," *Theoretical Computer Science*, vol. 50 (1987), pp. 1–101.

[15] Hindley, R., "The principal type-scheme of an object in combinatory logic," *Transactions of the American Mathematical Society*, vol. 146 (1969), pp. 29–60.

[16] Hindley, R., "BCK-combinators and linear lambda-terms have types," *Theoretical Computer Science*, vol. 64 (1989), pp. 97–105.

[17] Hindley, R. and D. Meredith, "Principal type-schemes and condensed detachment," *Journal of Symbolic Logic*, vol. 55 (1990), pp. 90–105.

[18] Hindley, R. and J. Seldin, *Introduction to Combinators and λ-calculus*, Cambridge University Press, Cambridge, 1986.

[19] Hirokawa, S., "Principal types of BCK-lambda-terms," *Theoretical Computer Science*, vol. 107 (1993), pp. 253–276.

[20] Hirokawa, S., "Principal type-schemes of BCI-lambda-terms," pp. 633–650 in *Theoretical Aspects of Computer Software*, edited by T. Ito and A. Meyer, Lecture Notes in Computer Science No. 530, Springer-Verlag, Berlin, 1991.

[21] Hirokawa, S., "BCK-formulas having unique proofs," pp. 106–120 in *Category Theory and Computer Science*, edited by D. H. Pitt et al., Lecture Notes in Computer Science No. 530, Springer-Verlag, Berlin, 1991.

[22] Hirokawa, S., "The relevance graph of a BCK-formula," *Journal of Logic and Computation*, forthcoming.

[23] Hirokawa, S. and Y. Komori, "Number of proofs for BCK-formulas," *Journal of Symbolic Logic*, vol. 58 (1993).

[24] Jaskowski, S., "Über Tautologien, in welchen keine Variable mehr als zweimal vorkommt," *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, vol. 9 (1963), pp. 219–228.

[25] Kalman, J., "The two-property and condensed detachment," *Studia Logica*, vol. 41 (1982), pp. 173–179.

[26] Komori, Y., "BCK algebras and lambda calculus," pp. 5–11 in *Proceedings of the 10th Symposium on Semigroups, Sakado 1986*, Josai University, Sakado, Japan, 1987.

[27] Komori, Y., "Illative combinatory logic based on BCK-logic," *Mathematica Japonica*, vol. 34 (1989), pp. 585–596.

[28] Meredith, C. and A. Prior, "Notes on the axiomatics of the propositional calculus," *Notre Dame Journal of Formal Logic*, vol. 4 (1963), pp. 171–187.

[29] Meredith, D., "Positive logic and λ-constants," *Studia Logica*, vol. 37 (1978), pp. 269–285.

[30] Meyer, R. and M. Bunder, "Condensed detachment and combinators," Report TR-ARP-8/88, Research School of the Social Sciences, Australian National University, Canberra, Australia, 1988.

[31] Meyer, R. and Z. Parks, "Independent axioms for the implicational fragment of Sobocinski's three-valued logic," *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, vol. 18 (1972), pp. 291–295.

[32] Prior, A., *Formal Logic*, Second Edition, Oxford University Press, Oxford, 1962 [Note: There are no references to BCI- and BCK-logics in the first edition, 1955.].

[33] Wos, L., S. Winker, R. Veroff, B. Smith, and L. Henschen, "Questions concerning possible shortest single axioms for the equivalential calculus," *Notre Dame Journal of Formal Logic*, vol. 24 (1983), pp. 205–223.

Mathematics Department
University College
Swansea SA2 8PP
Wales, U.K.