Two Proof Systems for Peirce Algebras*

Renate A. Schmidt
University of Manchester, UK, schmidt@cs.man.ac.uk

Introduction. Peirce algebras [1], formalise the properties of binary relations and sets, and their interactions. They provide a natural algebraic framework for modelling programming constructs, for natural language analysis, and for interpreting description logics which are used in the subbranch of AI concerned with knowledge representation [1]. Also the algebraic semantics of modal logics and extended modal logics, such as Boolean modal logic and dynamic modal logic can be studied in the framework of Peirce algebras.

A Peirce algebra is a two-sorted algebra $(\mathfrak{B}, \mathfrak{R}, :, {}^c)$ consisting of a Boolean algebra $\mathfrak{B}$ and a relation algebra $\mathfrak{R}$ plus two operations via which the Boolean algebra and the relation algebra are linked. On the set-theoretic level the Peirce product $(:)$ multiplies a binary relation R with a set A to give the set $R\!:\!A = \{x \mid \exists y\ (x, y) \in R \text{ and } y \in A\}$. The interpretation of the left cylindrification operation $({}^c)$ is given by: $A^c = \{(x, y) \mid x \in A\}$.

Peirce algebras are expressively equivalent to relation algebras and therefore inherit various properties of relation algebras [1], in particular, various negative results such as the non-representability and the impossibility to give a finite axiomatisation of the class of representable Peirce algebras. This means that it is not possible to express and derive every property of sets and binary relations in an algebraic (equational) setting. Thus we do not follow an equational approach, but restrict our attention to the reasoning problem in *Peirce logic*, the logic of the class of representable Peirce algebras. Peirce logic may also be viewed as a description logic.

We present two proof systems for Peirce logic. Both are tableaux-style proof systems, with the difference that one is a refutation calculus and the other is a calculus for proving validities of relations and sets. It turns out the calculi are dual to each other.

We expect that a result of Maddux for the sequent calculus of relational logic [2] carries over to Peirce algebra and proofs or refutations constructed by the systems presented in this abstract. This would provide a method to prove validities in Peirce algebra.

Peirce logic. We define two variants of Peirce logic, which are tailored either for refutation proofs using tableau or for proofs of validity in the style Rasiowa-Sikorski. The alphabet of both versions of Peirce logic consists of three syntactic types: (i) countably many Boolean symbols, denoted by A_i, for the Boolean elements, (ii) countably many relational symbols R_i, for the relational elements, and (iii) countably many individual constants, denoted by a_i, for the tableau version, or countably many individual variables, denoted by x_i, for the Rasiowa-Sikorski version of Peirce logic. The syntax is defined by the following.

Boolean expressions:
$A, B \ \longrightarrow\ A_i \mid 0 \mid -A \mid A \cap B \mid R\!:\!A$
Relational expressions:
$R, S \ \longrightarrow\ R_i \mid 0 \mid -R \mid R \cap S \mid R \,;\, S \mid R^{\smile} \mid \mathit{Id} \mid A^c$
Formulae:
$\psi \ \longrightarrow\ \bot \mid \top \mid sA \mid sRt$

where s, t are both constants in the language of Peirce logic for tableau, or they are both variables in the language of Peirce logic for Rasiowa-Sikorski. Expressions of the form sA and sRt are called constraints.

Tableau. A tableau is a finitely branching tree whose nodes are sets of constraints. Given an expression E of Peirce algebra to be tested for satisfiability the root node is the set $\{a\,E\}$, when E is a Boolean expression and $\{a\,E\,b\}$, when E is a relational expression. Successor nodes are constructed in accordance with a set of expansion rules. For a selected constraint ψ in a node, a rule $X/X_1|\ldots|X_n$ can be applied if ψ is an instance of the numerator X, or more generally, ψ together with other constraints in the node are instances of the constraints in X $(n \geq 0)$. n successor nodes are created which are extensions of the current node with the appropriate instances of X_i. It is assumed that no instance of a rule is applied twice to the same instance of the numerator. Sets of constraints are interpreted conjunctively and the vertical bar is interpreted disjunctively.

The tableau calculus for Peirce logic is defined by the rules of Figure 1. Intersection is assumed to be associative and commutative, and *Id* is assumed to be symmetric. When a node in a tableau contains $\bot$ then the branches on which the node resides are closed.

Theorem 1. *A Peirce logic constraint is unsatisfiable iff the tableau system presented in Figure 1 can be used to construct a closed tableau.*

An implementation of the tableau system was developed by Nellas [3].

Rasiowa-Sikorski. Rasiowa-Sikorski proof systems aim to prove entailment [4]. Given a candidate formula they aim to prove its validity or, if it is not valid, the aim is to construct a counter-model. This is done by systematic case analysis until fundamental validities are found. Rasiowa-Sikorski deduction rules may be used top-down and bottom-up. They have one of two forms $(n \geq 0)$: $X//X_1|\ldots|X_n$ and $X/X_1|\ldots|X_n$, where X, X_i denote sets of constraints. Sets of constraints are interpreted as disjunctions of constraints, whereas branching is interpreted conjunctively.

The Rasiowa-Sikorski calculus for Peirce logic is defined by the rules of Figure 2. In the literature rules are normally presented in a more detailed form. Our

* Joint work with Ewa Orlowska (National Institute of Telecommunications, Warsaw) and Ullrich Hustadt (University of Liverpool, UK).

Decomposition rules:

$(\cap)\ \dfrac{a\,A\cap B}{a\,A, a\,B}$ $\quad(-\cap)\ \dfrac{a-(A\cap B)}{a-A\ \mid a-B}$ $\quad(--)\ \dfrac{x--A}{x\,A}$

$(:)\ \dfrac{a\,R:A}{a\,R\,c, c\,A}$ where c is a new constant

$(-:)\ \dfrac{a-(R:A)}{a-R\,c \mid c-A}$ where c is any constant

$(\cap)\ \dfrac{a\,R\cap S\,b}{a\,R\,b, a\,S\,b}$ $\quad(-\cap)\ \dfrac{a-(R\cap S)\,b}{a-R\,b \mid a-S\,b}$ $\quad(--)\ \dfrac{x--R\,y}{x\,R\,y}$

$(^\smile)\ \dfrac{a\,R^\smile\,b}{b\,R\,a}$ $\quad(-^\smile)\ \dfrac{a-(R^\smile)\,b}{b-R\,a}$ $\quad(^{\smile\smile})\ \dfrac{x\,R^{\smile\smile}\,y}{x\,R\,y}$

$(^c)\ \dfrac{a\,A^c\,b}{a\,A}$ $\quad(-^c)\ \dfrac{a-(A^c)\,b}{a-A}$

$(;)\ \dfrac{a\,R\,;S\,b}{a\,R\,c, c\,S\,b}$ where c is a new constant

$(-;)\ \dfrac{a-(R\,;S)\,b}{a-R\,c \mid c-S\,b}$ where c is any constant

Substitution rule: $(Id)\ \dfrac{\psi, a\,Id\,b}{\psi[b]_\lambda}$ if $\psi|_\lambda = a$

Closure rules:

$(\mathrm{cl}_1)\ \dfrac{a\,A_i, a-A_i}{\bot}$ $\quad(\mathrm{cl}_2)\ \dfrac{a\,0}{\bot}$ $\quad(\mathrm{refl})\ \dfrac{a-Id\,a}{\bot}$

$(\mathrm{cl}_3)\ \dfrac{a\,R_i\,b, a-R_i\,b}{\bot}$ $\quad(\mathrm{cl}_4)\ \dfrac{a\,0\,b}{\bot}$

Fig. 1. Tableau rules for Peirce logic.

Decomposition rules:

$(\cap)\ \dfrac{x\,A\cap B}{x\,A\ \mid x\,B}$ $\quad(-\cap)\ \dfrac{x-(A\cap B)}{x-A, x-B}$ $\quad(--)\ \dfrac{x--A}{x\,A}$

$(:)\ \dfrac{x\,R:A}{x\,R\,z \mid z\,A}$ where z is any variable

$(-:)\ \dfrac{x-(R:A)}{x-R\,z, z-A}$ where z is a new variable

$(\cap)\ \dfrac{x\,R\cap S\,y}{x\,R\,y \mid x\,S\,y}$ $\quad(-\cap)\ \dfrac{x-(R\cap S)\,y}{x-R\,y, x-S\,y}$ $\quad(--)\ \dfrac{x--R\,y}{x\,R\,y}$

$(^\smile)\ \dfrac{x\,R^\smile\,y}{y\,R\,x}$ $\quad(-^\smile)\ \dfrac{x-(R^\smile)\,y}{y-R\,x}$ $\quad(^{\smile\smile})\ \dfrac{x\,R^{\smile\smile}\,y}{x\,R\,y}$

$(^c)\ \dfrac{x\,A^c\,y}{x\,A}$ $\quad(-^c)\ \dfrac{x-(A^c)\,y}{x-A}$

$(;)\ \dfrac{x\,R\,;S\,y}{x\,R\,z \mid z\,S\,y}$ where z is any variable

$(-;)\ \dfrac{x-(R\,;S)\,y}{x-R\,z, z-S\,y}$ where z is new variable

Special rule for Id:

$(Id)\ \dfrac{\psi}{\psi[y]_\lambda \mid x\,Id\,y}$ if $\psi|_\lambda = x$ and y is any variable

Closure rules:

$(\mathrm{cl}_1)\ \dfrac{x\,A_i, x-A_i}{\top}$ $\quad(\mathrm{cl}_2)\ \dfrac{x-0}{\top}$ $\quad(\mathrm{refl})\ \dfrac{x\,Id\,x}{\top}$

$(\mathrm{cl}_3)\ \dfrac{x\,R_i\,y, x-R_i\,y}{\top}$ $\quad(\mathrm{cl}_4)\ \dfrac{x-0\,y}{\top}$

Cut rules: $(\mathrm{cut}_1)\ \dfrac{}{x\,A\ \mid x-A}$ $\quad(\mathrm{cut}_2)\ \dfrac{}{x\,R\,y \mid x-R\,y}$

Fig. 2. Rasiowa-Sikorski rules for Peirce logic.

presentation assumes the rules are extension rules similar as for tableau. By definition, in a Rasiowa-Sikorski decomposition tree a branch is closed iff $\top$ was derived.

Theorem 2. *A Peirce logic constraint is valid iff the Rasiowa-Sikorski system presented in Figure 2 can be used to construct a closed tree.*

Duality. We consider a tableau (Rasiowa-Sikorski) decomposition or closure rule R_1 as dual to a Rasiowa-Sikorski (tableau) decomposition or closure rule R_2 if the rule R_i is obtained from the rule R_j $(i, j \in \{1, 2\}$ and $i \neq j)$ by simultaneous replacement in accordance with the following table.

$\cap$	$-\cap$	0	-0	Id	$-Id$	$;$	$-;$	x,y,z	a,b,c	$\wedge$	$\vee$	$\exists$	$\forall$
$-\cap$	$\cap$	-0	0	$-Id$	Id	$-;$	$;$	a,b,c	x,y,z	$\vee$	$\wedge$	$\forall$	$\exists$

I.e. all occurrences of $E_1 \cap E_2$ are replaced with $-(E_1 \cap E_2)$, all occurrences of $-(E_1 \cap E_2)$ are replaced with $E_1 \cap E_2$, etc. $\wedge$ and $\vee$ refer to meta-level conjunction and disjunction (i.e. ',' and '|' for tableau and '|', and ',' for Rasiowa-Sikorski). The meaning of the last two columns is that universal and existential quantification in the side conditions is exchanged. Special rules are considered dual iff variables and constants are interchanged, the premise becomes the conclusion, and the conclusion becomes the premise.

There are some points which seem to spoil the duality between the rules of the two systems for Peirce logic. One difference concerns the decomposition rules. The tableau decomposition rules are only applied top-down whereas the Rasiowa-Sikorski decomposition rules may be applied top-down as well as bottom-up. It turns out that this difference is insignificant because we can show that the tableau decomposition rules are invertible. Moreover, we can show that the Rasiowa-Sikorski proof system given by the rules in Figure 2, but applied only top-down, is sound and complete.

Another difference is the presence of the cut rules in the Rasiowa-Sikorski system. However, the cut rules are admissible in the tableau system. Therefore, the duals of the tableau rules are admissible rules in the Rasiowa-Sikorski system, and vice versa.

References

1. C. Brink, K. Britz, and R. A. Schmidt. Peirce algebras. *Formal Aspects of Computing*, 6(3):339–358, 1994.
2. R. D. Maddux. A sequent calculus for relation algebras. *Ann. Pure Applied Logic*, 25:73–101, 1983.
3. K. Nellas. Reasoning about sets and relations: A tableaux-based automated theorem prover for Peirce logic. Master's thesis, Univ. Manchester, UK, 2001.
4. H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*. Polish Scientific Publ., Warsaw, 1963.