Action Logic and Pure Induction

Vaughan Pratt
Dept. of Computer Science, Stanford, CA 94305
pratt@cs.stanford.edu

November, 1990

Abstract

In Floyd-Hoare logic, programs are dynamic while assertions are static (hold at states). In action logic the two notions become one, with programs viewed as on-the-fly assertions whose truth is evaluated along intervals instead of at states. Action logic is an equational theory ACT conservatively extending the equational theory REG of regular expressions with operations *preimplication* $a{\rightarrow}b$ (*had a then b*) and *postimplication* $b{\leftarrow}a$ (*b if-ever a*). Unlike REG, ACT is finitely based, makes a^* reflexive transitive closure, and has an equivalent Hilbert system. The crucial axiom is that of *pure induction*, $(a{\rightarrow}a)^* = a{\rightarrow}a$.

This work was supported by the National Science Foundation under grant number CCR-8814921.

1 Introduction

Many logics of action have been proposed, most of them in the past two decades. Here we define action logic, ACT, a new yet simple juxtaposition of old ideas, and show off some of its attractive aspects. The language of action logic is that of equational regular expression logic REG together with two implications, one for each direction of time. Action logic is a conservative extension of REG, yet has three properties lacking in REG: it is finitely axiomatizable, it asserts that a^* is reflexive transitive closure, and it has an equivalent Hilbert system.

This tidiness of ACT depends on two principles, the long-established principle of residuation, and the virtually unknown principle of what we call pure induction used by Ng and Tarski to adjoin transitive closure to the language of relation algebras to obtain a finitely based variety RAT.[1]

Residuation was noticed early on by De Morgan in his study of the calculus of binary relations [DM60]. He called the phenomenon Theorem K "in remembrance of the office of that letter in [the syllogistic forms] Baroko and Bokardo." Ward and Dilworth christened and studied residuation [WD39], the concluding chapters of Birkhoff's *Lattice Theory* gave it considerable coverage [Bir67], and it has since become a cornerstone of lattice theory and algebraic logic. It is treated by Fuchs [Fuc63], Jónsson [Jón82] emphasizes its role in relation algebras, and it is the basis for Freyd and Scedrov's notion of division allegory [FS90, 2.3]. The

[1] Just as Jónsson and Tarski [JT51] do not mention modal logic, so do Ng and Tarski [NT77, Ng84] not mention regular expressions.

corresponding notion in category theory is the closed category, where for fixed a, $a \rightarrow x$ is written x^a, right adjoint to $a \otimes x$, our ax. And it is at the heart of the Curry-Howard "isomorphism".[2]

By contrast Tarski's induction principle, in either form $(a \rightarrow a)^* \rightarrow (a \rightarrow a)$ or $a(a \rightarrow a)^* \rightarrow a$, is virtually unknown. In a 15-line announcement Ng and Tarski [NT77] restrict Tarski's variety RA of relation algebras to the class RAT of relation algebras in which every element a has a transitive closure $a^+ = aa^*$ (a^ω in their notation). They give equations establishing that RAT is a variety, the crucial equation being in our notation

$$((a^{\smile} a^-)^-)^+ = (a^{\smile} a^-)^-$$

where $(a^{\smile} b^-)^-$ is the RA expression of $a \rightarrow b$. And they state that the chain CCRA $\subset$ RAT $\subset$ RA (CCRA = countably complete RA's, not a variety) of inclusions among classes is strict. The proofs of these claims along with other results about RAT appear in Ng's thesis [Ng84], where the principle, in its "curried" form

$$a(a^{\smile} a^-)^-)^+ = a,$$

is attributed to Tarski. Ng gives another principle,

$$(a \cdot (a^- a^{\smile})^-)^+ = a \cdot (a^- a^{\smile})^-,$$

where $a \cdot b$ denotes meet, and proves its equivalence to Tarski's principle.

A quite different but equipollent induction axiom for RAT is given elsewhere by the present author, namely

$$a^* b = b + a^*(b^- \cdot ab).$$

This adapts Segerberg's induction axiom for propositional dynamic logic to RA [Pra90]. This induction axiom exploits nonmonotonicity in the static (Boolean) structure of RA instead of in the residuation structure, and for that reason is more generally applicable to the larger class BM of Boolean monoids, i.e. RA without converse and hence without residuation.

ACT generalizes RAT by weakening the Boolean monoid structure of RAT to a residuated semilattice while retaining Tarski's induction principle. The philosophical conclusion one might then draw from the success of these two abstractions is that induction is quite versatile in being able to adapt itself successfully to these different sources of nonmonotonicity.

We thus have the chain $RAT \subset ACT \subset REG$ of classes of models of the respective equational theories of those names. On the one hand ACT expands the language of REG with conditionals, on the other it reduces that of RAT by deleting Boolean conjunction, Boolean negation, and converse while keeping residuation (and incidentally replacing a^+ by a^*).

The case for ACT in preference to RAT or REG is that RAT is too strong while REG is too weak. RAT omits many useful algebras for which ACT provides a useful logic, in particular all algebras not Boolean algebras. REG goes too far in the other direction, failing to rule out even finite nonstandard interpretations of a^* yet having no finite basis.

But although RAT is unnecessarily strong, the central properties of ACT that we prove below are simply inherited from RAT. In particular the key arguments in the proof of Theorem 7, that ACT is a finitely based variety, can all be obtained from Ng's thesis [Ng84] merely by ignoring the Boolean structure of RAT, which inspection shows need play no role once the translation of $(a^{\smile} b^-)^-$ to $a \rightarrow b$ has been made.

Our main objective is to study induction via residuation in the context of propositional reasoning about action. However we hope our paper, in conjunction with the companion paper of Mike Dunn in this volume, will also have some tutorial value in the area of philosophical and computer science applications of residuation.

[2]Over lunch at LICS-90 I had the pleasure of being able to point out this connection with Theorem K to Bill Howard after we discovered a shared enthusiasm for that theorem, the common cause being recent publicity for it by the relation algebraist Roger Maddux.

2 The Language of Action Logic

The language of action logic is unsorted in that all its terms denote actions, also interpretable as propositions. It has a minimum of symbols consistent with the requirements that its equational theory be finitely based yet conservatively extend the equational theory of regular expressions. The latter makes a nice logic of action because it contains the three basic control elements of imperative programming languages, namely choice, sequence, and iteration. These are as follows.

Given two actions a and b, the binary *choice* action $a+b$ performs one of a or b. The constant *block* action 0 is zeroary choice: it offers no choices, is never done, and is the immovable object. The binary *sequence* action $a;b$ or $a \otimes b$ or ab performs first a then b. The constant *skip* action 1 is zeroary sequence; it performs no actions, is done immediately, and is the irresistible force by dint of having the empty agenda. Finally the unary *iteration* or *star* action a^* performs a zero or more times sequentially.

We may think of regular expressions as a monotone logic with a disjunction $a+b$ and a "dynamic" conjunction ab that need be neither idempotent nor commutative. While iteration is not ordinarily associated with propositional logic, its presence is nevertheless in line with recent attractive and fruitful proposals to expand first order logic with a least fixpoint operator [Mos74, CH82, HK84]. Regular algebra is not a proper logic in the usual sense however since it lacks nonmonotone operations such as negation and implication.

Besides sequence, the conjunction ab may also be interpreted as *concurrence*, the performance of both a and b but in no specific order. If a and b have m and n subactions respectively then ab has $m+n$ subactions under either of these interpretations. Yet another interpretation is *interaction*, as when a consists of m trains passing through n stations constituting b; in this case ab has mn subactions.

The interpretation of iteration always matches that of conjunction. Thus if ab is concurrence then a^* performs a zero or more times in no specific order. If interaction then a^* is the interaction of arbitrarily many copies of a with itself.

A model of regular algebra is an algebra $(A, +, 0, ;, 1, *)$. There are two models of regular algebra either of which could be considered the standard. One interprets regular expressions as formal languages (sets of strings) over some set Σ, the *alphabet*, with $a+b$ denoting the union of languages a and b, ab their concatenation, and a^* as all concatenations of finitely many (including zero) strings of a. The other interprets them as binary relations (sets of pairs) on some set W, with $a+b$ as union, ab as composition, and a^* as ancestral or reflexive transitive closure. These two models turn out to have the same equational theory, REG.

The star-free fragment of REG can be completely axiomatized by equations asserting that $(A, +, 0)$ and $(A, ;, 1)$ are monoids, with $+$ commutative and idempotent, ; distributing over $+$ on both sides ($(a+b)c = ac+bc$, $a(b+c) = ab+ac$), and 0 an annihilator for ; on both sides ($0a = 0 = a0$). In addition REG contains the following equations involving star.

$$
\begin{aligned}
1 + a^*a^* + a &= a^* \\
1 + aa^* &= a^* \\
a^* + (a+b)^* &= (a+b)^* \\
a^{**} &= a^* \\
(a+b)^* &= (a^*b^*)^* \\
(aa)^* + a(aa)^* &= a^* \\
a + (aaa)^*(aa)^* &= a^* \\
a(ba)^* &= (ab)^*a
\end{aligned}
$$

However no such list can be complete, because REG is not finitely based [Red64, Con71]. That is, there is no finite list of equations of REG from which the rest of REG may be inferred.

But in addition to this syntactic problem, REG has a semantic problem. It is not strong enough to constrain a^* to be the reflexive transitive closure of a. We shall call a *reflexive* when $1 \leq a$ and *transitive* when $aa \leq a$, and take the reflexive transitive closure of a to be the least reflexive transitive element b such that $a \leq b$. Thus the nature of this second problem is that REG has models in which a^* is not the least possible b for which $1 + bb + a \leq b$.

If all such nonstandard models were infinite this might be understandable. But Conway [Con71] has exhibited such a nonstandard model having only four elements, which we have elsewhere called *Conway's Leap* [Pra90]. Let $\{L_0, L_1, L_2, \ldots, L_\omega\}$ be the set of languages of the form $L_i = \{\mathbf{x}^j \mid j < i\}$ for $i = 0, 1, 2, \ldots, \omega$ over an alphabet whose single symbol is $\mathbf{x}$. This set is closed under the standard regular operations and has constants 0 and 1, namely L_0 and L_1 respectively, making it a subalgebra of the set of all languages on that alphabet and hence a model of REG. Now identify all its finite languages starting with L_2 to form a single element F for finite. This equivalence can be verified to be a congruence, whence the 4-element set $\{L_0, L_1, F, L_\omega\}$ is a quotient and hence also a model of REG. In this quotient $L_1 \leq F$ and $FF = F$ (the concatenation of two finite languages is still finite), so F is reflexive and transitive and so is its own reflexive transitive closure. But in this algebra $F^* = L_\omega$ since the star of any language containing a nonempty string is an infinite language. Hence iteration is interpreted nonstandardly in Conway's Leap.

So REG is simultaneously too big and too small. It is too big for the whole of it to be reached by equational reasoning from any finite part of it. Yet it is too small to prevent nonstandard models. In logic nonstandardness usually goes with the territory. What is so painful here is that REG is unable to rule out even *finite* nonstandard models!

Various solutions to these two problems can be found in the literature. Conway [Con71] gives five definitions of regular algebra: quantales (a term more recent than Conway's book), all regular subalgebras thereof, REG, and two finitely based fragments thereof. None both are finitely axiomatized and define a^*. Salomaa [Sal66] proposes a finite axiom schema that could prove all of REG. Kozen [Koz91, Koz90] modifies Salomaa's system to yield a finitely based Horn theory KA of *Kleene Algebras*. (By Horn theory we mean a universal Horn theory, one whose formulas are universally quantified implications $a_1 = b_1 \wedge \ldots \wedge a_n = b_n \;\Rightarrow\; a = b$ from a conjunction of equations to a single equation.) KA is a subclass of REG in which a^* is always reflexive transitive closure. This is a good solution, and we compare action logic with Kleene algebras in detail below. Pratt [Pra79, Pra80] takes the regular component of separable dynamic algebras as definitive of regular algebras. This system is equational save for the axiom of separability, but it uses two sorts, actions and propositions, unlike the other solutions. Here we shall only be interested in homogeneous (unsorted) solutions.

Action logic confers on regular expressions both a finite axiomatization and an induction principle, simply by expanding the language with two implications. *Preimplication* $a{\rightarrow}b$ denotes *had* a *then* b, as in "Had I bet on that horse I'd be rich now" or "Had this gun been pointed at you, you'd be dead now." *Postimplication* $b{\leftarrow}a$ denotes b *if-ever* a, as in "I'll be rich if that horse wins" or "You'll die if I pull this trigger."

Although each implication is monotone in its conclusion it is antimonotone in its premise: the stronger the premise the weaker the implication ("I'll be famous if I prove P≠NP"). This makes the two implications the only nonmonotonic operations of action logic, all the regular operations being monotone.

Action logic is an equational theory. This leads to a pleasant Hilbert-style formulation of action logic in which inequalities $a \leq b$ of the theory of action algebras are curried to either $1 \leq a{\rightarrow}b$ or $1 \leq b{\leftarrow}a$ and then the $1 \leq$ is read as the symbol $\vdash$ of theoremhood, i.e. the theorems of an action algebra are its reflexive elements.

We mention in passing certain useful abbreviations, analogous to $p{\rightarrow}q = \neg p \vee q$ in classical logic. These take care of themselves and we need not mention their properties beyond this paragraph. Analogously to intuitionistic negation, we write $\ulcorner a$ as abbreviating $a{\rightarrow}0$ and meaning *never-before* a, and $a\neg$ as abbreviating $0{\leftarrow}a$ and meaning a *never-again*. We write the constant $\top$ as abbreviating $\ulcorner 0$ and meaning the top element *any*, that is, no restriction whence any action may be performed. (We will see later that $0{\rightarrow}a = \top$ for any a; the implicit choice of $a = 0$ in the definition of $\top$ is arbitrary.) We could take 1 to be an abbreviation for 0^*, but since 1 features in the definition of a^* it is somewhat tidier to define it before defining a^*.

One final word on notation. Tradition writes 1 for what we have just notated above as $\top$. The roots of this tradition lie in the historical accident by which logic began with static conjunction $ab = a \cdot b$ and static truth $1 = \top$, and likewise in category theory where the first closed categories one encounters are cartesian closed. As the system begins to move, dynamic and static conjunction part company, and one must then decide whether the desirable notations ab and 1 are to remain with the stationary (traditionally $a \times b$ and 1) or moving ($a \otimes b$ and I or $1'$) constructs. At the risk of being judged nomadic, we express a strong preference for the latter. Our notation is consistent with that of Kozen [Koz91, Koz90] for Kleene algebras, Girard [Gir87] for linear logic (except that we write his $a \otimes b$ as ab since we only have one conjunction, and write his $a \oplus b$ more conventionally as $a + b$), and of course ordinary arithmetic $(\mathbf{R}, +, 0, \times, 1, \div, \div, \frac{1}{1-a})$.[3]

3 First Order Action Logic

3.1 Definition of Action Algebras

An *action algebra* is an algebra $\mathcal{A} = (A, +, 0, ;, 1, \rightarrow, \leftarrow, *)$ such that $(A, +, 0)$ and $(A, ;, 1)$ are monoids, with $+$ commutative and idempotent, and satisfying

$$a \leq c{\leftarrow}b \quad \overset{L}{\Leftrightarrow} \quad ab \leq c \quad \overset{R}{\Leftrightarrow} \quad b \leq a{\rightarrow}c \tag{1}$$

$$1 + a^*a^* + a \leq a^* \tag{2}$$

$$1 + b\,b + a \leq b \quad \Rightarrow \quad a^* \leq b \tag{3}$$

A *commutative* action algebra is an action algebra satisfying $ab = ba$. Whereas action logic in general is neutral as to whether ab combines a and b sequentially or concurrently, commutative action logic in effect commits to concurrency.

These axioms are substantially weaker than those of relation algebras [JT51], admitting as action algebras models that need not be Boolean algebras (models of classical propositional logic) or even Heyting algebras (models of intuitionistic propositional logic).

We refer to the class of all action algebras as ACT. We denote the Horn theory of ACT by Horn(ACT) and the equational theory of ACT by Eq(ACT) or simply ACT when context permits. As we will see later Eq(ACT) axiomatizes Horn(ACT), i.e. ACT is the class of models of both Horn(ACT) and Eq(ACT), making ACT a variety.

[3] Although arithmetic is not an action algebra as defined here, $+$ not being idempotent, it is conceivable that it could be constituted as an action category with idempotence being perhaps division by 2.

3.2 Examples of Action Algebras

We now list some examples of action algebras. The first two examples are Boolean algebras and hence relation algebras, but of the remainder the only Heyting algebras are a few of the "idempotent closed ordinals," namely the one of each cardinality for which $1 = \top$, only the two-element one of which is Boolean.

The algebra of all binary relations on a set W is an action algebra ordered by inclusion $R \subseteq S$, with RS composition, with the relation $R{\rightarrow}S$ defined as $\{(v,w) \mid \forall u[uRv{\rightarrow}uSw]\}$, $S{\leftarrow}R$ as $\{(u,v) \mid \forall w[vRw{\rightarrow}uSw]\}$, and the relation R^* as the ancestral or reflexive transitive closure of R. The implications have a quite natural meaning: for example if R and S are the relations *loves* and *pays* respectively then $R{\rightarrow}S$ is the relation which holds of (v,w) just when every u who loves v pays w. The sentence "king(loves→pays)taxes" is then pronounced "Who loves their king pays their taxes." In the area of relational databases the operation $L{\rightarrow}M$ has been termed *quotient* by E.F. Codd. This algebra can be expanded to an RAT by adjoining the Boolean operations and defining converse $R^{\smile}$ as either $R^-{\rightarrow}1^-$ or $1^-{\leftarrow}R^-$.

The algebra of all languages on a set Σ is an action algebra ordered by inclusion $L \subseteq M$, with LM concatenation, with the language $L{\rightarrow}M$ defined as $\{v \mid \forall u[u \in L \Rightarrow uv \in M]\}$, $M{\leftarrow}L$ defined as $\{u \mid \forall v[v \in L \Rightarrow uv \in M]\}$, and the language L^* being the Kleene closure of L, consisting of all strings each the concatenation of zero or more strings of L.

In the formal language literature the quotient $L \setminus M$ of languages has the dual meaning (to relation quotient) of $\{v \mid \exists u[u \in L \wedge uv \in M]\}$. The nature of this duality is brought out more clearly via the connections $L{\rightarrow}M = (L \setminus M^-)^-$ and $L \setminus M = (L{\rightarrow}M^-)^-$, where $M^- = \Sigma^* - M$. This dual relationship between relation quotient and language quotient parallels that of box and diamond modalities, or universal and existential quantifiers. This conflict between the relational and language uses of "quotient" appears not to have been noticed to date at the intersection of their respective communities. This algebra can be expanded to an RAT exactly as for the algebra of relations, in which case $L^{\smile}$ is either $\Sigma^+ = \Sigma\Sigma^*$ or Σ^* depending on whether or not $\varepsilon \in L$.

This example is the standard model of regular expressions for automata theory. Automata theorists will be familiar with the fact that regular sets are preserved by both complement and (language) quotient, and will correctly infer that they are therefore preserved by residuation. One is tempted to infer that if a and b are regular expressions then the regular set denoted by $a{\rightarrow}b$, when represented by a pure (→-free) regular expression c, satisfies $a{\rightarrow}b = c$ as an equation of ACT. This line of reasoning, which is perfectly reliable for pure regular expressions, falls to an easy counterexample, namely $a{\rightarrow}b$ itself, reading a and b as atoms (languages with one string containing one symbol). Here $a{\rightarrow}b = 0$, since no string when appended to the unit-length string a yields the (different) unit-length string b. But $a{\rightarrow}b = 0$ is certainly not a theorem of ACT.

A less standard example, but which will be recognized by those for whom the conjunction Floyd-Warshall [War62, Flo62] is meaningful, is provided by the algebra

$$(\mathbf{N}^+, max, \bot, +, \mathbf{0}, \dot{-}, \dot{-}, ^*)$$

where $\mathbf{N}^+ = \{\bot, \mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \top\}$ denotes the set of natural numbers together with $\bot$ and $\top$, and $\dot{-}$ is truncated subtraction ($7 \dot{-} 3 = 4$, $3 \dot{-} 7 = 0$). Addition is commutative (which is why the two residuations are the same) and satisfies $\bot i = \bot$ and otherwise $i\top = \top$ (the significance of the "otherwise" being that $\bot\top = \bot$), and * is the operation defined as $\bot^* = \mathbf{0}^* = \mathbf{0}$ and $i^* = \top$ for all $i \geq \mathbf{1}$ including $i = \top$.

Replacing max by min and taking $a^* = 0$ and $\bot\top = \top$ in this example (without negative numbers) yields a similar example.

A natural variation on these examples includes all the negative integers. In this case the common residuation becomes ordinary subtraction.

An action algebra for which $(A, +, 0)$ is a linear order $\{\mathbf{0} \leq \mathbf{1} \leq \mathbf{2} \leq \ldots \leq \mathbf{n}\}$ and $(A, ;, 1)$ is commutative and idempotent (so all elements transitive) is an *idempotent closed ordinal* (ico) [CCMP91]. Here $0 = \mathbf{0}$ necessarily, but 1 need not be $\mathbf{1}$ and may be anything but $\mathbf{0}$.

There is a pretty representation of these. It can be shown that $(A, ;, 1)$ is itself linearly ordered by $a \sqsubseteq b$ defined as $ab = a$, which also starts at 0 and agrees with $\leq$ up to 1 but then turns around and is thereafter opposite to it. On the way back down it has to interleave all a such that $1 < a$ with all b such that $b \leq 1$; there turn out to be 2^{n-1} such interleavings, over all choices for 1. The 2-element ico ($n = 1$) is the 2-element Boolean algebra with all logical operations (disjunction, conjunction, implication) standardly interpreted for Boolean algebras. The two three-element ico's find application in concurrency for modelling strict-nonstrict delay and causal-accidental precedence respectively. Exactly one of these is a Heyting algebra for each n, namely the one for which $1 = \top$, only the two element one of which is a Boolean algebra. Since all elements are transitive a^* is just reflexive closure $1 + a$.

3.3 Residuation as Implication

Axiom (1) defines *residuation* or *implication*, defining $b \leftarrow a$ and $a \rightarrow b$ as respectively the *left* and *right residuals of* b *over* a. Left residuation $c \leftarrow b$ is *postimplication*: "c *if-ever* b." Right residuation $a \rightarrow c$ is *preimplication*: "had a then c". When a is "bet on horse", b "horse wins," and c "get rich," the axiom asserts the equivalence of the three sentences "If you bet on a horse you get rich if the horse then wins," "If you bet on a horse and it then wins you get rich," and "If a horse wins then had you bet on it you would get rich." However these are not equivalent to any of $b \leq c \leftarrow a$, "If a horse wins you would get rich if you then bet on it," $ba \leq c$, "If a horse wins and then you bet on it you will get rich," or $a \leq b \rightarrow c$, "If you bet on a horse then had it previously won you would get rich."

Most of the properties of residuation flow from the form of these two equivalences, namely

$$f(x) \leq y \Leftrightarrow x \leq g(y).$$

This is the form of a *Galois connection*[4] (f, g) on a poset, the poset in our case being the structure $(A, \leq)$ where $a \leq b$ is defined by $a + b = b$. We call f and g the *left* and *right polarities* of the Galois connection, being just the left and right adjoints respectively of an adjunction of posets. We will not treat Galois connections explicitly here, but what we shall say about residuation can be read as applying to any Galois connection if in (1L) $x; b$ and $y \leftarrow b$ are viewed as the unary $f(x)$ and $g(y)$ respectively of a Galois connection, and similarly $a; x$ and $a \rightarrow y$ in (1R). That is, (1L) is really a family of Galois connections indexed by b, and similarly (1R) by a.

The existence of both residuals implies the following distributivity properties. This particularly attractive aspect of residuation is a special case of left (right) adjoints preserving colimits (limits).

$$a(b + c) = ab + ac \quad (4)$$
$$a0 = 0 \quad (5)$$
$$(a + b)c = ac + bc \quad (6)$$
$$0a = 0 \quad (7)$$
$$a \rightarrow (b \cdot c) = (a \rightarrow b) \cdot (a \rightarrow c) \quad (8)$$

[4] Galois connections have traditionally been defined contravariantly, having the advantage of making them symmetric. However there is some sentiment nowadays for taking them to be covariant and regarding what was previously taken to be a Galois connection between posets P and Q as being between P and Q^{op}, the order dual of Q. This then makes Galois connections simply adjunctions of posets, making one less concept in mathematics and giving a nice way to introduce adjunction.

$$
\begin{aligned}
a{\to}\top &= \top && (9)\\
(a+b){\to}c &= (a{\to}c)\cdot(b{\to}c) && (10)\\
0{\to}a &= \top && (11)
\end{aligned}
$$

Since binary meet, $a \cdot b$, is not in the language of action logic it need not be defined. The rule for our equations however is that if the left side of an equation is defined the right must be also. In particular in the last two equations $(a{\to}c)\cdot(b{\to}c)$ is always defined since $(a+b){\to}c$ is, and likewise $\top$.

These may be proved as follows, using the principle that $a \le c \Leftrightarrow b \le c$ implies $a = b$. The arguments are given only for the four binary cases, but generalize to arbitrary sups and infs because the conjunctions in the middle can also be arbitrary, and hence include the four zeroary cases. Only the last does not illuminate Galois connections along the lines we indicated earlier.

$$a(b+c) \le d \;\Leftrightarrow\; b+c \le a{\to}d \;\Leftrightarrow\; b \le a{\to}d \,\wedge\, c \le a{\to}d \;\Leftrightarrow\; ab \le d \,\wedge\, ac \le d \;\Leftrightarrow\; ab+ac \le d$$

$$(a+b)c \le d \;\Leftrightarrow\; a+b \le d{\leftarrow}c \;\Leftrightarrow\; a \le d{\leftarrow}c \,\wedge\, b \le d{\leftarrow}c \;\Leftrightarrow\; ac \le d \,\wedge\, bc \le d \;\Leftrightarrow\; ac+bc \le d$$

$$d \le a{\to}(b\cdot c) \;\Leftrightarrow\; ad \le b\cdot c \;\Leftrightarrow\; ad \le b \,\wedge\, ad \le c \;\Leftrightarrow\; d \le a{\to}b \,\wedge\, d \le a{\to}c \;\Leftrightarrow\; d \le (a{\to}b)\cdot(a{\to}c)$$

$$d \le (a+b){\to}c \;\Leftrightarrow\; a+b \le c{\leftarrow}d \;\Leftrightarrow\; a \le c{\leftarrow}d \wedge b \le c{\leftarrow}d \;\Leftrightarrow\; d \le a{\to}c \,\wedge\, d \le b{\to}c \;\Leftrightarrow\; d \le (a{\to}c)\cdot(b{\to}c)$$

Note how these arguments for distributivity all work by moving obstructing junk to the other side of the inequality and putting it back when done. In this respect residuation has some of the syntactic advantages of subtraction or division in arithmetic. Moreover, unlike $1/0$ in arithmetic, which quickly leads to inconsistencies in conjunction with otherwise reliable rules, $1{\leftarrow}0$ can be safely assumed to be $\top$ without risk of inconsistency.[5]

As corollaries we infer that ab is monotone in a and b, and $a{\to}b$ and $b{\leftarrow}a$ are monotone in b and antimonotone in a.

As a corollary of $a0 \le 0$ we have $a \le 0{\to}0 = {⌐}0 = \top$. While we have $a{⌐}a = 0$ (it cannot be that a happens followed by a denial of a in the past), and dually that $a{\neg}a = 0$ (one cannot promise that a will never happen and then have it subsequently happen). On the other hand ${⌐}aa$ and $aa{\neg}$ need not be 0, though in the language model they so evaluate. They are however nonzero in the relation model, constituting a point of difference between these models that was lacking with just the regular language.

The following properties (12)-(16) of right residuation may each be obtained by making the substitution indicated on the right in (1R), simplifying, and renaming variables to suit. The corresponding properties of left residuation are obtained similarly; we shall refer to them by equation numbers suffixed with L, e.g. (16L) denotes $1 \le a{\leftarrow}a$.

$$
\begin{array}{lrcll}
 & b & \le & a{\to}ab & (c = ab) \qquad (12)\\
 & a(a{\to}b) & \le & b & (b = a{\to}c) \qquad (13)\\
\text{Hence} & a(a{\to}b)(b{\to}c) & \le & b(b{\to}c) & (\text{append } b{\to}c \text{ to both sides})\\
\text{which with} & b(b{\to}c) & \le & c & (\text{instance of } (13))\\
\text{yields} & a(a{\to}b)(b{\to}c) & \le & c &
\end{array}
$$

[5] Like all arguments about ordered sets, these arguments generalize quite directly to closed categories and show that tensor product $x \otimes y$ preserves colimits in each argument and x^y preserves limits in x and maps colimits in y to limits. The equivalences between truth values become natural isomorphisms between homsets, and the specific natural isomorphism from say the coproduct $ab+ac$ to $a(b+c)$ is that corresponding to the identity morphism on $a(b+c)$ after setting d to $a(b+c)$.

$$\begin{array}{rrcll} \text{which curries to} & (a\to b)(b\to c) & \leq & a\to c & \qquad (14) \\ \text{and specializes as} & (a\to a)(a\to a) & \leq & a\to a & \qquad (15) \\ \text{Reflexivity :} & 1 & \leq & a\to a \quad (b=1, c=a) & \qquad (16) \end{array}$$

The inequalities (12) and (13) are called the *unit* and *counit* respectively of the Galois connection. (14) constitutes an "internal" cut rule; note that order of premises matters. The last two inequalities assert that $a\to a$ is respectively transitive and reflexive. As exercises the reader may wish to show $a(a\to ab) = ab$ and $a\to a(a\to b) = a\to b$.

3.4 Reflexive Transitive Closure as Iteration

Axiom (2) asserts that a^* is reflexive ($1 \leq a^*$), transitive ($a^*a^* \leq a^*$), and includes a ($a \leq a^*$), while (3) asserts that a^* is the least element with these three properties.

A basic property of a^* is that it is monotone. For suppose $a \leq b$. Then by (2), $1 + b^*b^* + b \leq b^*$, whence $1 + b^*b^* + a \leq b^*$. But then by (3), $a^* \leq b^*$.

Setting a and b to $a\to a$ in (3) and using (15) and (16), we obtain *pure induction*, curried and uncurried, as follows.

$$\begin{array}{rrcl} & (a\to a)^* & \leq & a\to a \qquad (17) \\ \text{and} & a(a\to a)^* & \leq & a \qquad (18) \end{array}$$

We think of this induction rule as being pure in the sense that it involves not only just one sort of data, in contrast to numbers and predicates in Peano induction, or actions and propositions in Segerberg induction, but that it also involves only one variable, which however does appear four times. We will expound further on the meaning of pure induction in section 4.2.

The definition of star as reflexive transitive closure is symmetric. However the following equivalent asymmetric definition is often used.

$$\begin{array}{rcl} 1 + aa^* & \leq & a^* \qquad (19) \\ 1 + ab & \leq & b \;\Rightarrow\; a^* \leq b \qquad (20) \end{array}$$

This definition makes a^* the least b satisfying $1 + ab \leq b$.

Theorem 1 *In ACT, (2) is equipollent with (19), and independently (3) is equipollent with (20).*

Proof: The easy direction proves (19) from (2) and (20) from (3). The former is an easy exercise. For the latter, assume $1 + ab \leq b$, i.e. $1 \leq b$ and $ab \leq b$. Hence $a \leq b\leftarrow b$. So by (15)-(16),

$$\begin{array}{rrcl} & 1 + (b\leftarrow b)(b\leftarrow b) + a & \leq & b\leftarrow b \\ \text{so} & a^* & \leq & b\leftarrow b \qquad \text{(by (3))} \\ \text{whence} & a^*b & \leq & b \\ \text{and hence} & a^* & \leq & b \qquad (1 \leq b). \end{array}$$

In the other direction, to show (2) from (19) we must show each of:

$$
\begin{aligned}
1 &\leq a^* && (21)\\
a^*a^* &\leq a^* && (22)\\
a &\leq a^* && (23)
\end{aligned}
$$

(21) asserts that a^* is reflexive, and follows immediately from (19), as does (23). (22) says a^* is transitive, and is proved as follows.

$$
\begin{array}{llcll}
 & aa^* & \leq & a^* & \text{(by (19))}\\
\text{so} & a & \leq & a^*\leftarrow a^* & \text{(currying)}\\
\text{Now} & (a^*\leftarrow a^*)(a^*\leftarrow a^*) & \leq & a^*\leftarrow a^* & \text{(by (15))}\\
\text{whence} & a(a^*\leftarrow a^*) & \leq & a^*\leftarrow a^* & \\
\text{Also} & 1 & \leq & a^*\leftarrow a^* & \text{(by (16))}\\
\text{So} & a^* & \leq & a^*\leftarrow a^* & \text{(by (20))}\\
\text{giving} & a^*a^* & \leq & a^* & \text{(by uncurrying)}
\end{array}
$$

To show (3) from (20), assume $1 + bb + a \leq b$. Then $bb \leq b$, so $ab \leq b$. Hence $1 + ab \leq b$, giving $a^* \leq b$ by (20). To show independence of these equipollences it suffices to note that (3)$\Leftrightarrow$(20) did not appeal to either (2) or (19). ∎

Ng's Corollary 3.15 [Ng84] is the corresponding theorem for RAT, while Freyd and Scedrov's item 1.787 [FS90] is the corresponding theorem for endo-relations in a "logos," a category meeting certain conditions.

It follows by symmetry that we must have equipollence of (2)-(3) with the left-handed form of (19)-(20), namely:

$$
\begin{aligned}
1 + a^*a &\leq a^* && (24)\\
1 + b\,a &\leq b \;\Rightarrow\; a^* \leq b && (25)
\end{aligned}
$$

The last two properties in ACT that we shall prove here are motivated by Kozen's notion of a Kleene algebra, which we shall define shortly.

We say that a *left-preserves* b when $ab \leq b$, and dually *right-preserves* b when $ba \leq b$. The following axioms assert that a^* preserves everything a does, in either direction.

$$
\begin{aligned}
ab \leq b &\;\Rightarrow\; a^*b \leq b && (26)\\
ba \leq b &\;\Rightarrow\; ba^* \leq b && (27)
\end{aligned}
$$

We prove (26) thus.

$$
\begin{array}{rcll}
ab & \leq & b & \text{(premise)}\\
a & \leq & b\leftarrow b & \text{(curry)}\\
a^* & \leq & (b\leftarrow b)^* & \text{(mon. of *)}\\
 & \leq & b\leftarrow b & \text{(pure induction)}\\
\text{whence } a^*b & \leq & b & \text{(uncurry)}
\end{array}
$$

A similar argument proves (27).

This collection of theorems of action logic will suffice for our purposes, in particular to compare Kozen's Kleene algebras with action algebras.

One naturally wonders whether ACT has a preferred direction for particular applications, favoring one implication over the other and one of (26) or (27) over the other. There is no asymmetry in the definition. Nevertheless when a is the predicate transformer "incrementing x makes" and b is $x = 2$, then it is natural to read ab as "incrementing x makes $x = 2$," denoting the predicate $x = 1$. Note how the predicate transformer corresponding to an action transforms the predicate backwards in time.

On the other hand when we view a as some ongoing action and b as one more step, then ab is "more of a." Here we transform forwards in time.

3.5 Definition of ACT operations

A primitive set of operations of algebras of a class $\mathcal{C}$ is a set that implicitly defines the remaining operations. Formally, given any class $\mathcal{C}$ of algebras of a given signature S, a subset $T \subset S$ is called *primitive for* $\mathcal{C}$ when it determines the remaining operations of S, in the sense that any algebra with signature T has at most one expansion (same elements but additional operations) to an algebra in $\mathcal{C}$. For example either $\cdot$ or $+$ is primitive for Boolean algebras because they each give away the underlying poset of the algebra, via $a = a \cdot b$ or $a + b = b$, which then determines all the Boolean operations.

Theorem 2 *The operations + and ; form a primitive set for ACT.*

Proof: The unit of each monoid is determined by its operation. $a{\rightarrow}b$ is unique because it is the greatest element c satisfying $ac \leq b$, and similarly for $b{\leftarrow}a$, while a^* is unique because it is the least reflexive transitive element including a. ∎

A slight strengthening of this theorem is possible by starting not from $(A, +, ;)$ but from $(A, \leq, ;)$, the join $a + b$ then being definable as the (necessarily unique) least upper bound on a and b, e.g. via $a + b \leq c \Leftrightarrow a \leq c \wedge b \leq c$. This makes a slightly better connection with monoidal categories as the categorical generalization of such an ordered monoid [Mac71].

In the Horn axiomatization each operation $a{\rightarrow}b$, $b{\leftarrow}a$, and a^* is defined independently. (The use of $a + b$ in the definition of a^* is inessential; (2) and the premise of (3) can each be rephrased as a conjunction of three inequalities not mentioning $+$.) In the equational axiomatization however a^* is defined in terms of one or the other of $a{\rightarrow}b$ or $b{\leftarrow}a$, with $+$ apparently essential to express each of the four monotonicity axioms equationally.

When the class $\mathcal{C}$ is specified by a first-order formula Φ, as is ACT (take Φ to be the conjunction of the axioms of ACT) we say that the set $S - T$ of nonprimitive operations is *implicitly defined by* Φ from T. We then have Beth's theorem, that an n-ary relation R which is implicitly definable by a first order sentence is also explicitly definable, in the sense that there is a first order formula with free variables $x_1, \ldots, x_n$ in the language just of $+$ and ; that is equivalent to $R(x_1, \ldots, x_n)$. The reader may find it instructive to construct suitable such formulas for each n-ary operation of action logic, construed as an $(n{+}1)$-ary relation.

Primitive operations permit meaningful comparisons between classes having different signatures, e.g. REG and ACT. Any two classes with the same primitive operations can be compared merely by comparison between their respective reducts to those primitives.

On the other hand classes with different signatures but a common set of primitive operations cannot be shown to be unequal merely from the fact that one is a variety and the other not. For example ACT is variety, as we shall see, whereas the implication-free reduct of ACT is not a variety because Conway's Leap is then a quotient of $\{L_0, L_1, \ldots, L_\omega\}$. This is because the equivalence defining the Leap is compatible with the regular operations but not with implication.

3.6 Kleene Algebras

A *Kleene algebra* [Koz91, Koz90] is an algebra $\mathcal{A} = (A, +, 0, ;, 1, *)$ such that $(A, +, 0)$ and $(A, ;, 1)$ are monoids, with + commutative and idempotent, and satisfying

$$\begin{array}{rclcrcll}
a(b+c) & = & ab+ac & (4) & 1+aa^* & \leq & a^* & (19) \\
a0 & = & 0 & (5) & 1+a^*a & \leq & a^* & (24) \\
(a+b)c & = & ac+bc & (6) & ab \leq b & \Rightarrow & a^*b \leq b & (26) \\
0a & = & 0 & (7) & ba \leq b & \Rightarrow & ba^* \leq b & (27)
\end{array}$$

We denote by KA the class of models of these axioms, and write Horn(KA) and Eq(KA) for the Horn and equational theories of KA respectively.

Theorem 3 *ACT* $\subseteq$ *KA.*

Proof: It is straightforward to prove (2) and (3) in KA. Hence a^* is uniquely determined, i.e. Theorem 3 (all operations defined by + and ;) holds for KA as well as ACT. ∎

Theorem 4 (Kozen.) *Eq(KA) = Eq(REG).*

Kozen [Koz91] proves this by recasting familar automata constructions in a very appealing algebraic form. This makes REG the variety generated by KA. Since KA proscribes the Leap and REG does not it follows that KA is not a variety. It is however a quasivariety, being the class of models of a Horn theory, as evident from the axioms.

Theorem 5 *Eq(ACT) conservatively extends Eq(REG).*

Proof: Since Horn(ACT) extends Horn(KA), Eq(ACT) extends Eq(KA) = Eq(REG). Now Eq(REG) = Eq(A) where A is the algebra consisting of all languages of strings over the alphabet $\{0, 1\}$. But A was one of our examples of an action algebra, whence the regular equations of ACT are all equations of A and hence of REG, making the extension conservative. ∎

Is every Kleene algebra an action algebra, in the sense that it expands to one? Are (26) and (27) independent in KA? Are they still independent if Boolean negation is present? We conjectured [Pra90] that (26) were (27) independent in BMT, Boolean monoids with transitive closure, a BMT being a KA expanded with complement to make it a Boolean algebra. Kozen [Koz90] showed their independence for KA, leaving the question for BMT open, by showing that the strict finititely additive functions on an infinite power set satisfies all of KA save axiom (27) which it violates.

We first answer all three of these questions affirmatively for the finite case. An affirmative answer to the first question implies affirmative answers to the other two in that case.

Theorem 6 *Every finite Kleene algebra expands to an action algebra.*

Proof: Define $a \to c = \sum_{ab \leq c} b$, a finite (possibly empty) sum. It follows immediately that $ab \leq c \Rightarrow b \leq a \to c$. For the other direction, fixing a and c we have

$$\begin{aligned} a \sum_{ab \leq c} b &= \sum_{ab \leq c} ab \\ &\leq c \end{aligned}$$

whence if $b \leq a \to c = \sum_{ab \leq c} b$ then $ab \leq c$. The dual argument yields $b \leftarrow c$. ∎

We answer all three questions negatively in the infinite case by considering five small variations on a standard action algebra, and then showing how to extend the five examples to a Boolean monoid.

Recall the action algebra $(\mathbf{N}^+, max, \bot, +, \mathbf{0}, \dot{-}, \dot{-}, ^*)$ where $\mathbf{N}^+ = \{\bot, \mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \top\}$, encountered in section 3.2. To this algebra adjoin an element ∞ second from the top, $i \leq \infty \leq \top$ for all finite i; this determines the corresponding extension of $+$. Star at ∞ is determined by monotonicity, namely $\infty^* = \top$; thus star maps $\bot$ and $\mathbf{0}$ to $\mathbf{0}$ and everything else to $\top$.

If we now set $i\infty = j\infty$ and $\infty i = \infty j$ for all finite positive i and j, then inspection reveals that ab is determined by the choices for $\mathbf{1}\infty$, $\infty\infty$, and $\infty\mathbf{1}$, each of which can be ∞ or $\top$, monotonicity ruling out all else. But if $\infty\infty = \infty$ then by monotonicity the other two are also ∞, leaving only five possible choices. These correspond to five distinct algebras. It is readily verified that all axioms of KA save (26) and (27) are satisfied by all five algebras.

Inspection shows that (3) holds just when $\infty\infty = \top$, (26) just when $\mathbf{1}\infty = \top$, and (27) just when $\infty\mathbf{1} = \top$. This gives an alternative example to Kozen's showing that (26) and (27) are independent in KA, illustrated in each direction by taking one of the cases of $\infty\mathbf{1} \neq \mathbf{1}\infty$. The only Kleene algebra among the five is the one for which $\infty\mathbf{1} = \mathbf{1}\infty = \top$.

Unlike Kozen's example however these examples are easily converted to Boolean monoids with star. We take the Boolean algebra $2^{\mathbf{N} \cup \{\infty\}}$ and define $XY = \{x + y \mid x \in X, y \in Y\} \cup \{\infty \mid p(X,Y)\}$ for five different predicates $p(X,Y)$: false, both X and Y infinite, X infinite, Y infinite, and either X or Y infinite. (We take $\infty i = i\infty = \infty$ for all $i \neq 0$.) We take $0^* = 1^* = 1$, and $a^* = \mathbf{N} \cup \infty$ otherwise. The correspondence with the previous examples is that an infinite set X corresponds to $\top$ or ∞ respectively according to whether or not $\infty \in X$. If ∞ is in either X or Y, whether infinite or not, it is automatically in XY unless the other of X or Y is empty. This correspondence now makes it easy to verify that the same reasoning in the previous example applies here to yield algebras satisfying all of the axioms for a Boolean monoid with star save one or the other of (26) and (27).

We now answer the first question by giving a Kleene algebra that is not (i.e. does not expand to) an action algebra. When $\mathbf{1}\infty = \top$, $\mathbf{1} \to \infty$ is undefined, and dually when $\infty\mathbf{1} = \top$, $\infty \leftarrow \mathbf{1}$ is undefined. Hence the one Kleene algebra among our five algebras does not expand to an action algebra, and therefore the inclusion in Theorem 3 is strict. This argument also yields a Boolean monoid with star that is not an action algebra, via the same correspondence as before.

We showed [Pra90] that BMT is a variety, using the adaptation of Segerberg's induction axiom described in the introduction. We have just seen a BMT that is not an action algebra, and we have seen many examples of action algebras that are not Boolean algebras. It follows that BMT and ACT are incomparable varieties, yet both define a^* to be reflexive transitive closure. Hence these two ways of defining star equationally, Segerberg induction and Tarski induction, are independent.

We remark that the algebra for $\infty\infty = \infty$ expands to satisfy all of the ACT axioms except (3), showing the independence of the latter from the other ACT axioms. We have now made use of all five algebras in distinguishing various theories from each other.

Another difference between ACT and KA is the absence of explicit distributivity axioms in ACT. As we saw earlier the presence of residuation ensures distributivity; the left residual gives distributivity on the left, and similarly for the right.

One might look the gift horse of residuation in the mouth by viewing its assumption as a large burden. However residuation is finitely and equationally definable, and moreover is found in nature, in particular in both standard models of regular algebra, not to mention the prevalence of closed categories in algebra. Moreover residuation is action logic's only source of nonmonotonicity, a *sine qua non* of a proper logic. It pays its own way by subsuming the distributivity and annihilation ($a0 = 0 = 0a$) axioms. And it simplifies logic by identifying the meanings of induction and reflexive transitive closure. In our opinion its benefits to regular algebra substantially outweigh its costs, and is a horse you can safely bet on.

4 Equational Action Logic

4.1 Equational Axiomatization

Theorem 7 *ACT is a finitely based variety.*

Proof: The following equations axiomatize ACT. We continue to write $a \leq b$ as an abbreviation for $a + b = b$.

$$a + (b + c) = (a + b) + c \qquad (28)$$
$$a + b = b + a \qquad (29)$$
$$a + a = a \qquad (30)$$
$$a + 0 = a \qquad (31)$$
$$a(bc) = (ab)c \qquad (32)$$
$$ab \leq (a + a')(b + b') \qquad (33)$$
$$1a = a = a1 \qquad (34)$$
$$a{\rightarrow}b \leq a{\rightarrow}(b + b') \qquad (35)$$
$$a(a{\rightarrow}b) \leq b \leq a{\rightarrow}ab \qquad (36)$$
$$b{\leftarrow}a \leq (b + b'){\leftarrow}a \qquad (37)$$
$$(b{\leftarrow}a)a \leq b \leq ba{\leftarrow}a \qquad (38)$$
$$1 + a^*a^* + a \leq a^* \qquad (39)$$
$$a^* \leq (a + b)^* \qquad (40)$$
$$(a{\rightarrow}a)^* \leq a{\rightarrow}a \qquad (41)$$

All these equations were proved in the section on Horn action logic. This verifies the soundness of this system with respect to our original definition of action algebra. For completeness, all the equations in the Horn axiomatization of ACT appear above, so it suffices to prove the nonequational axioms (1) and (3).

To prove (1R), assume $ab \leq c$. By (35) we have $a{\rightarrow}ab \leq a{\rightarrow}c$. But then by (36b), $b \leq a{\rightarrow}c$. Conversely assume $b \leq a{\rightarrow}c$. By (33) we have $ab \leq a(a{\rightarrow}c)$. But then by (36a), $ab \leq c$. A similar argument proves (1L).

To prove (3), assume $1 + bb + a \leq b$, that is, $1 \leq b$ and $bb \leq b$ and $a \leq b$. Currying the second yields $b \leq b{\leftarrow}b$, and by transitivity with the third we have $a \leq b{\leftarrow}b$. By (40), $a^* \leq (b{\leftarrow}b)^*$. But then by (41), $a^* \leq b{\leftarrow}b$. Uncurrying yields $a^*b \leq b$. $1 \leq b$ and (33) gives $a^* \leq a^*b$, whence by transitivity $a^* \leq b$. ∎

In contrast KA is not a variety since it does not contain Conway's Leap. Since the Leap is the regular quotient

(quotient by a regular congruence, one compatible with just the regular operations) of an action algebra, this difference between KA and ACT must be attributable to residuation being in the signature of ACT. In particular the quotient yielding the Leap is not compatible with residuation, since for $0 < m \leq n < \omega$, $L_m {\rightarrow} L_n = L_{n-m+1}$. Hence $L_2 {\rightarrow} L_n = L_{n-1}$. But then if $L_n = L_{n+1}$ for any n we have $L_2 {\rightarrow} L_n = L_2 {\rightarrow} L_{n+1}$ whence $L_{n-1} = L_n$. Thus residuation has a sort of iteration-like power to propagate any identification of consecutive elements downwards until 0 and 1 are identified. But then $a = a1 = a0 = 0 = b0 = b1 = b$. Hence identifying consecutive elements of this algebra must collapse it to the one-element or inconsistent algebra. So the equivalence creating the Leap, although a regular quotient, is not an action quotient.

4.2 The Meaning of Pure Induction

Let us now consider the meaning of the equation of pure induction, to which ACT owes its varietyhood. First let us look at the meaning of another of the several induction axioms we now have, (26), namely $ab \leq a \Rightarrow ab^* \leq a$. Interpreting action a as a proposition in this context (think for the moment of these particular propositions as actions whose common point of beginning is lost in the mists of time), this says that if action b preserves proposition a then so does b^*. That is, * preserves all preservers, of whatever proposition a.

Now $a {\rightarrow} a$ is the *weakest* (least constrained) preserver of a, being the greatest b for which $ab \leq a$. Hence the explicit content of pure induction, written as $(a {\rightarrow} a)^* = a {\rightarrow} a$, is that iteration preserves weakest preservers. That star also preserves the other preservers, which we have seen thus far only formally, can be seen a little more graphically as follows. For each a the weakest preserver $a {\rightarrow} a$ of a sits at the top of a pyramid or *principal order ideal* of preservers b all at least as strong as $a {\rightarrow} a$, i.e. $b \leq a {\rightarrow} a$ or $ab \leq a$. Iteration may move actions up in the pyramid, weakening them to vary the number of times they can happen from 1 to any, and thus thinning out the pyramid, but it does not move $a {\rightarrow} a$, the apex, by pure induction, and hence it does not move any of the other preservers of a out of the pyramid, by monotonicity of iteration. Thus even those preservers of a that are not fixpoints of iteration remain preservers of a when iterated, as demanded by intuition. For if kicking the customer's TV set is known not to fix it, repeated kicking is in vain, something every artificially intelligent TV repairbot should know instinctively.

Currying yields $a(a {\rightarrow} a)^* \leq a$. This is a pure albeit dynamic form of Peano induction: if a holds then no matter how long we run the weakest preserver of a, a still holds. Since every preserver b of a satisfies $b \leq a {\rightarrow} a$, it follows by monotonicity of iteration that $ab^* \leq a$, that is, if a holds then no matter how long we run b, a still holds.

Induction in action logic thus takes on a very natural, pleasing, and easy to work with form that the reader should find fully in agreement with intuition.

4.3 Hilbert-style Action Logic

The Boolean equation $p = q$ translates into the Boolean term $p \equiv q$, while the Boolean term p translates into the Boolean equation $p = 1$. This is the basis for Hilbert systems, whose theorems are terms instead of equations. Theoremhood for equations means membership in the equational theory BA of Boolean algebras, while theoremhood for terms means Boolean validity. These translations both preserve theoremhood. We then say that BA has an equivalent Hilbert system.

These translations are also both easily computed, a situation we express by saying that BA has an *efficiently* equivalent Hilbert system. If the translation is not easily computed then the systems are not practically interchangeable in the sense that it is not easy to read a theorem of one system as the corresponding theorem in the other.

The corresponding situation for ACT is that the ACT *inequality* $a \leq b$ translates into the ACT term $a \rightarrow b$. Conversely the ACT term a translates as the ACT inequality $1 \leq a$. If we take as the criterion of theoremhood of a term of ACT that $1 \leq a$ be a theorem in the inequational theory Ineq(ACT) of ACT, then these two translations constitute an efficient Hilbert system for inequational action logic. Formally these translations are a pair of maps $H : L^2_{ACT} \rightarrow L_{ACT}$ and $J : L_{ACT} \rightarrow L^2_{ACT}$, with $H(a, b) = a \rightarrow b$ and $J(a) = 1 \leq a$ (this would be made more precise by use of quotation marks in the obvious places). Thus we have:

Theorem 8 *Ineq(ACT) has an efficiently equivalent Hilbert system.*

For example the equation $(a \rightarrow a)^* \leq a \rightarrow a$ expressing pure induction becomes $(a \rightarrow a)^* \rightarrow (a \rightarrow a)$, while its Peano induction form $a(a \rightarrow a)^* \leq a$ now reads $a(a \rightarrow a)^* \rightarrow a$.

The canonical rule in a Hilbert system is Modus Ponens. The translation we have in mind carries over the rules of inequational logic to the Hilbert system. Such a direct translation of inequational reasoning will include such rules as, from $a \rightarrow b$ and $b \rightarrow c$ infer $a \rightarrow c$. Some optimization to reduce the number of rules should be possible, but we have not explored this very far.

In BA $1 = \top$ and so $p = 1$ and $1 \leq p$ are equivalent. Hence we may safely take $1 \leq p$ as the criterion for theoremhood in defining Hilbert systems for both BA and ACT. We therefore fix $1 \leq a$ as a general criterion for theoremhood in Hilbert systems in general.

It seems obvious that ACT is better suited to Hilbert systems than REG. But how can we be sure? After all REG is a logic too, with $a + b$ as disjunction, 0 as false, ab as conjunction, and 1 as true. Hence we can at least contemplate the notion of a Hilbert system for regular expressions even if we are convinced of its impossibility. We continue to take $1 \leq a$ as the criterion for theoremhood of terms. The translation of $a \leq b$ can no longer be $a \rightarrow b$, but perhaps it could be some other term of REG. If all we want is fidelity to theoremhood, it suffices to translate all theorems of Ineq(REG) to 1 and the rest of L^2_{REG} to 0. This particular translation is unreasonable if the systems are meant to be equivalent in any useful sense, since it requires deciding theoremhood, but perhaps there are other less onerous translations. In the other direction of course things are much easier, a simply translates as $1 \leq a$.

However reflexivity of regular expressions is decidable in DTIME(n), deterministic time linear in the number of operations in the expression by the obvious recursive procedure, whereas theoremhood in REG (and hence in Ineq(REG)) is PSPACE-complete [MS72]. Hence:

Theorem 9 *Any equivalent Hilbert system for Ineq(REG) is a reduction from a PSPACE-complete set to a DTIME(n) set.*

Such reductions are popularly suspected to be very hard to compute. On this assumption then we have that *there are no efficiently equivalent Hilbert systems for REG.*

This along with no finite axiomatization and failure to define a^* are the three defects of REG corrected by ACT.

The following example proves $a(ba)^* \rightarrow (ab)^* a$. We keep the proof short by liberal use of (reasonable) derived rules, the justification for each of which should be clear. The topmost $\rightarrow$ could as well have been $\leftarrow$ or $\leq$; we made it an implication for the sake of being able to call this a Hilbert-style proof.

$$
\begin{array}{rcl}
(ab)^*ab & \to & (ab)^* \\
(ab)^*aba & \to & (ab)^*a \\
(ab)^*a(ba)^* & \to & (ab)^*a \\
a(ba)^* & \to & (ab)^*a
\end{array}
$$

5 Side Issues

5.1 Reverse

Action logic with reverse admits an additional unary connective, *reverse* $a^\smile$, satisfying the following conditions.

$$a^\smile + b^\smile = (a+b)^\smile \tag{42}$$

$$(ab)^\smile = b^\smile a^\smile \tag{43}$$

$$a^{\smile\smile} = a \tag{44}$$

Reverse differs from the other operations in that it is not fully defined by its axioms, which is why we said "satisfying" rather than "defined by." This imprecision is in contrast to RA, where it is indeed defined, namely by $a^\smile = a^- \to 1^-$. The imprecision reveals itself in the standard language model viewed as an RA, which then defines RA converse $a^\smile$ to be either $\Sigma^+ = \Sigma\Sigma^*$ or Σ^* respectively according to whether or not $\varepsilon \in a$. Both this peculiar operation and ordinary language reversal satisfy our axioms for reverse. If reverse could be defined so that it was converse for binary relations but language reversal for languages this would then constitute a *very* intriguing branch off to one side of the present linear hierarchy REG $\subset$ ACT $\subset$ RAT. Until this is resolved we have chosen not to include reverse in basic action logic, but rather to treat it as an extension.

The imprecision notwithstanding, its presence allows us now to prove one of the two equations defining 1 from the other, to weaken associativity of ab to an inequality, and via the theorem $(a \to b)^\smile = (a^\smile \leftarrow b^\smile)$ to reduce the status of one of the implications to a mere abbreviation in terms of the other. We thus gain four equations and weaken another for the price of three, leave the number of operations unchanged, but muddy the dimensional interpretation described below with a third dimension of uncertain meaning. Also provable is $a^{*\smile} = a^{\smile *}$. We may also weaken the second equation for reverse to $(ab)^\smile \leq b^\smile a^\smile$ and the first to assert merely the monotonicity of reverse rather than its finite additivity. The involution equation however is sacrosanct; it prevents $a^\smile$ from being stuck at 0 or $\top$.

5.2 Comparisons with other logics.

In the realm of propositional logics, action logic is a close cousin to classical logic, intuitionistic logic, relevance logic, linear logic, relation algebras, regular expressions, and dynamic logic. Of these, perhaps its closest cousin is relation algebra, begun in 1860 by De Morgan [DM60] and further developed by Peirce [Pei33], Schröder [Sch95], Tarski [Tar41], and Jónsson [JT51], and many others since.

Classical and intuitionistic logics are pure *static* logics, while regular expressions constitute a pure *action* logic[6]. Relevance logic, linear logic, relation algebras, and dynamic logic are all hybrid static-dynamic logics, with the dynamics of the former two logics being concurrent (commutative) and of the latter two sequential.

The criterion we use for distinguishing these logics is the kind of conjunctions they have. We define static conjunction to be meet in the logic's lattice of propositions, as in both classical and intuitionistic logic. Dynamic conjunction on the other hand, while still being an associative binary operation, is not meet, and need not even be idempotent. Whether dynamic conjunction is commutative or not determines whether we regard it as *concurrent* or *sequential* respectively.

Action logic differs from its cousins in two notable ways. First, unlike all but RAT and PDL (propositional dynamic logic), it embeds the theory of regular expressions, in particular iteration. Second, whereas classical and intuitionistic logic achieve their simplicity relative to the other logics by not having dynamic conjunction, action logic achieves its simplicity by not having static conjunction. The essential premise here, whether insight or folly, is that one can have a perfectly usable action logic without a static conjunction.

Like classical and intuitionistic logic, and very much unlike relevant logic and linear logic, the semantics of action logic is very straightforward: an ordered semigroup has at most one expansion to an action algebra (Theorem 3). The corresponding statement for classical and intuitionistic logic is that any poset has at most one expansion to respectively a Boolean algebra and a Heyting algebra. (These logics are one-dimensional in that they are determined by just $\leq$, whereas action logic is determined by $\leq$ and ;, making it two-dimensional.) The equational logic of action algebra is also clearcut, like that of classical and intuitionistic logic. Various equational logics have been proposed for relevant logic, but there remains much room for debate as to why one might be preferable to another. And there is at present not even a candidate equational theory of linear logic to debate the suitability of. The equational theory ACT of action logic in contrast is as immutable as that of Heyting algebras[7]

5.3 Branching Time

The link between implication and distributivity can be exploited in the following intriguing way. If we had only one of the two implications then we would have only the corresponding distributivity principle, since $a \to b$ as right residuation yields distributivity of ; over + on the right: $a(b+c) = ab+ac$; and dually for $b \leftarrow a$ and distributivity on the left.

The "branching time" school of thought distinguishes $ab+ac$ from $a(b+c)$ in terms of information available before and after a, arguing that the early decision implied by the $ab+ac$ choice is more constraining than the later decision implicit in $a(b+c)$, justifying $ab+ac \leq a(b+c)$ but not conversely. But $(a+b)c = ac+bc$ is accepted because in both cases the decision is made at the outset. We can easily accommodate this viewpoint by dropping the preimplication $a \to b$ and keeping postimplication $b \leftarrow a$.

But this seems a case of the tail wagging the dog to make it happy. What is going on that would justify thus, beyond a purely means-end analysis? What if our explanation of the underlying cause predicted dropping the wrong implication?

Given two actions a and b starting together, $a \to b$ is the weakest or most general action which can continue a so as to satisfy b, $a(a \to b) \leq b$, where b is an upper bound not to be exceeded rather than a goal to be

[6]The term "dynamic logic" is too closely linked now with the present author's system of modal regular algebra to permit referring to regular expressions on their own as a "dynamic logic."

[7]But not quite as immutable as the equational theory of Boolean algebras, which so pack their vertical one-dimensional space that they would instantly become inconsistent if so much as one equation were added. RA renders BA consistently mutable by first adding a horizontal dimension $a;b$ that creates new space to grow in, see the section on 2D logic for more on these orientations.

exactly attained in full generality. The problem of determining $a \rightarrow b$ is that of locating the maximum in the poset of those actions, starting at the end of a and ending at the end of b, which on the assumption of a having been done would satisfy b. The dual problem of determining $b \leftarrow a$, where a and b end together, is to maximize in the poset of those actions, starting at the start of b and ending at the start of a, which when continued with a would satisfy b.

The asymmetry here is between the poset in a's past and that in its future. If for whatever reason we can find the maximal element of the past poset but not of the future, we lose $a \rightarrow b$ but not $b \leftarrow a$.

Having established the form the rest of the argument should take, we leave it to the reader to agonize over whether the past or future poset is more likely to yield up its maximal element on demand. The goal is clear: in order to make distributivity agree with the branching time point of view the future poset has to turn out to be the reluctant one. This is not as obvious as it might seem; if we have more information about the past than the future, the maximal element might well be more accessible in the future poset on cardinality grounds. A more detailed picture of the structure of these posets than we are willing to speculate on is surely needed here.

5.4 Two-dimensional Logic

We may think of action logic as a two-dimensional propositional logic. In both classical and intuitionistic propositional logic, the implication $p \rightarrow q$ is an arrow between points p and q (Fig. 1). We may view the arrow as the trajectory of a moving point travelling from p to q, a one-dimensional entity.

Fig. 1

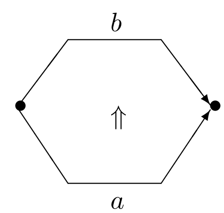

Fig. 2

In action logic p and q become periods a and b with a common start and end, and $p \rightarrow q$ becomes the trajectory $a \rightarrow b$ of a moving period with fixed endpoints sweeping through the surface enclosed by a and b, starting at a and ending at b, as in Fig. 2.

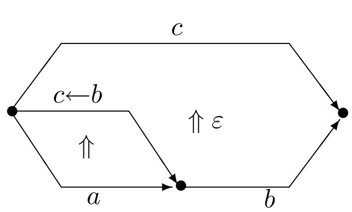

Fig. 3

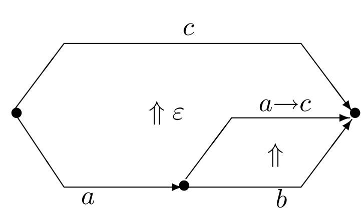

Fig. 4

These surfaces compose rather like arrows, but abutting in lines instead of points. Fig. 3 factors the nonlogical or empirical implication $ab \leq c$ into the composition of a tautological implication $(c \leftarrow b)b \leq c$

and a nonlogical implication $a \leq c \leftarrow b$. Fig. 4 dually factors the same implication into the composition of the action tautology $a(a \rightarrow c) \leq c$ and the nonlogical implication $b \leq a \rightarrow c$. In 2-category terminology the tautologies are counits, labeled ε by tradition for εvaluation map.

In our standard example in which a is "bet on horse", b "horse wins," and c "get rich," Fig. 3 factors the empirical "If you bet on a horse and it then wins you get rich" into the tautology "If you will get rich if the horse ever wins, and the horse then wins, you will get rich" and the empirical "If you bet on a horse you get rich if the horse then wins." Dually Fig. 4 factors the same empirical statement into the tautology "If you bet on a horse, and then had you bet on the horse you would get rich, then you will get rich" and the empirical "If a horse wins then had you bet on it you would get rich." The example "If I point the gun at you and pull the trigger you will die" may be factored similarly.

These *pasting diagrams* as they are called in the 2-category literature [KS74] thus provide a geometrical interpretation of Axiom (1) in which one direction of the equivalence $\stackrel{L}{\Leftrightarrow}$ is given constructively as the composition of the counit ε with $a \leq c \leftarrow b$. To see this as transitivity of $\leq$ we must first compose $a \leq c \leftarrow b$ with b to yield $ab \leq (c \leftarrow b)b$, by monotonicity of xy in x. The other direction of $\stackrel{L}{\Leftrightarrow}$ is given by the composition of $ab \leftarrow b \leq c \leftarrow b$ (obtained from $ab \leq c$ by monotonicity of $x \leftarrow y$ in x) with the unit $a \leq ab \leftarrow b$, another tautology. (1R) is the dual case.

By counting primitives as dimensions we may call classical and intuitionistic logic one-dimensional and action logic (along with relation algebras, relevant logic, and linear logic) two-dimensional.

Readers familiar with 2-categories will not only recognize the significance of this geometric analogy but will also be aware of the respective orientations of the dimensions: $\leq$ is vertical and ; horizontal, laying out logical strength plotted vertically against time passing horizontally. The 2-categories arising here consist of a monoid for their horizontal category and a poset for their vertical, thereby constituting an ordered monoid $(A, \leq, ;)$, the stuff of which action algebras are made.

Monads enter this 2-categorical picture via star. A monad in a 2-category is a triple (T, η, μ) where T is a 1-cell (a morphism of the horizontal category and an object of the vertical) and η, μ are 2-cells (morphisms of the vertical category). Every element a^* determines a monad (T, η, μ) where $T = a^*$ and η, μ are the laws of reflexivity and transitivity for a^*.

There is no reason to limit action algebras to a single monoid $(A, ;, 1)$. A logic of concurrency would quite naturally have two such structures, a noncommutative one for sequence and a commutative one for concurrence. Each may then have its own residuations and iteration as required. This would then constitute a 3D logic, $\leq$ still being a dimension, namely the static one.

Elsewhere [Pra86, CCMP91] we have described a language for concurrency that has three such operations, two of which are for concurrent combination, making this four-dimensional. One, called concurrence, combines objects "travelling in the same direction," and the other, orthocurrence, combines objects travelling in opposite directions. The latter describes all flow situations, whether of a river along a bed, signals through a circuit, or trains through stations. The elements of this model are a certain kind of generalized labeled metric space, equipped with the structure not of a partial order but rather of a category defined by the contracting maps of those spaces. Taking distances to be truth values yields event structures, but they may also be taken to be reals for modelling real time; other sorts of distances are also of use. Action logic, with its order structure suitably generalized to categories and with ab taken as a tensor product $a \otimes b$, is then applicable to this model, no matter what notion of distance is used. This is in contrast say to either Boolean monoids or relation algebras, both of which support induction but neither of which we have as yet been able to apply to this model of concurrency.

References

[Bir67] G. Birkhoff. *Lattice Theory*, volume 25. A.M.S. Colloq. Publications, 1967.

[CCMP91] R.T Casley, R.F. Crew, J. Meseguer, and V.R. Pratt. Temporal structures. *Math. Structures in Comp. Sci.*, 1(2):179–213, July 1991.

[CH82] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[Con71] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, London, 1971.

[DM60] A. De Morgan. On the syllogism, no. IV, and on the logic of relations. *Trans. Cambridge Phil. Soc.*, 10:331–358, 1860.

[Flo62] R.W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[FS90] P.J. Freyd and A. Scedrov. *Categories, Allegories*. North Holland, 1990.

[Fuc63] L. Fuchs. *Partially Ordered Algebraic Systems*. Pergamon Press, 1963.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[HK84] D. Harel and D. Kozen. A programming language for the inductive sets, and applications. *IC*, 63(1-2):118–139, 1984. Also in *Proc. 9th Int. Colloq. on Automata, Languages, and Programming*, Aarhus, Denmark, July 1982.

[Jón82] B. Jónsson. Varieties of relation algebras. *Algebra Universalis*, 15:273–298, 1982.

[JT51] B. Jónsson and A. Tarski. Boolean algebras with operators. Part I. *Amer. J. Math.*, 73:891–939, 1951.

[Koz90] D. Kozen. On Kleene algebras and closed semirings. In B. Rovan, editor, *Mathematical Foundations of Computer Science 1990*, volume 452 of *Lecture Notes in Computer Science*, pages 26–47, Banská Bystrica, 1990. Springer-Verlag.

[Koz91] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Proc. 6th Annual IEEE Symp. on Logic in Computer Science*, pages 214–225, Amsterdam, July 1991.

[KS74] G.M. Kelly and R. Street. Review of the elements of 2-categories. volume 420 of *Lecture Notes in Mathematics*. Springer-Verlag, 1974.

[Mac71] S. Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.

[Mos74] Y. N. Moschovakis. *Elementary Induction on Abstract Structures*. North-Holland, 1974.

[MS72] A.R. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential time. In *Proc. 13th IEEE Symp. on Switching and Automata Theory*, pages 125–129, Long Beach, CA, 1972. IEEE Computer Society.

[Ng84] K.C. Ng. *Relation Algebras with Transitive Closure*. PhD thesis, University of California, Berkeley, 1984. 157+iv pp.

[NT77] K.C. Ng and A. Tarski. Relation algebras with transitive closure, Abstract 742-02-09. *Notices Amer. Math. Soc.*, 24:A29–A30, 1977.

[Pei33] C.S. Peirce. Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic. In *Collected Papers of Charles Sanders Peirce. III. Exact Logic*. Harvard University Press, 1933.

[Pra79] V.R. Pratt. Models of program logics. In *20th Symposium on foundations of Computer Science*, San Juan, October 1979.

[Pra80] V.R. Pratt. Dynamic algebras and the nature of induction. In *12th ACM Symposium on Theory of Computation*, Los Angeles, April 1980.

[Pra86] V.R. Pratt. Modeling concurrency with partial orders. *Int. J. of Parallel Programming*, 15(1):33–71, February 1986.

[Pra90] V.R. Pratt. Dynamic algebras as a well-behaved fragment of relation algebras. In *Algebraic Logic and Universal Algebra in Computer Science*, volume 425 of *Lecture Notes in Computer Science*, pages 77–110, Ames, Iowa, June 1988, 1990. Springer-Verlag.

[Red64] V.N. Redko. On defining relations for the algebra of regular events (Russian). *Ukrain. Mat. Z.*, 16:120–126, 1964.

[Sal66] A. Salomaa. Two complete axiom systems for the algebra of regular events. *Journal of the ACM*, 13:158–169, 1966.

[Sch95] E. Schröder. *Vorlesungen über die Algebra der Logik (Exakte Logik). Dritter Band: Algebra und Logik der Relative.* B.G. Teubner, Leipzig, 1895.

[Tar41] A. Tarski. On the calculus of relations. *J. Symbolic Logic*, 6:73–89, 1941.

[War62] S. Warshall. A theorem on Boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.

[WD39] M. Ward and R.P. Dilworth. Residuated lattices. *Trans. AMS*, 45:335–354, 1939.