Bucanon introduction, continued: Motivating theory algebras

www.bucephalus.org

September 20, 2002

Throughout the rest of the text we assume that the output parameters of the Bucanon program are set to their default settings, i.e. *notation:stroke*, *formation:custom*, *ordered:no*, *simplified:yes*. Accordingly, all examples only use the stroke notation with the symbols `',;` for negation, conjunction and disjunction.

Formulas and worlds

Let us consider a formula φ and its table, say $\varphi =$ `[a , 'b]` for "a and not b", with the table

a	b	[a , 'b]
?	?	?
!	?	!
?	!	?
!	!	?

Each row of the table contains a *valuation*, that assigns a bit value to each of the atoms of φ. For example, the valuation of the last row is "`a:=!` and `b:=!`" for "a is true and b is true". The formula itself can be seen as a function that assigns bit values to each of these valuations. These values are written in the right column. For example, for the valuation of the last row, this value is `?` or "false". Each valuation is a function of the type $A \longrightarrow \mathcal{B}$, where A is the atom set containing `a` and `b` and $\mathcal{B}$ is the set of the bit values `?` and `!`. So, φ defines a function of the type $(A \longrightarrow \mathcal{B}) \longrightarrow \mathcal{B}$.

It makes sense to call such a function a **world**, so that each formula φ denotes its well defined world, based on its atom set or atom list A. Each valuation is a **state** of the world and the value assigned to this valuation determines if the state is **possible** or **impossible**. For example, the mentioned state "a is true and b is false" is impossible in the world of φ.

According to this interpretation, formulas are just descriptions of worlds. Every formula denotes a unique world, but on the other hand, every world can be represented by more than one formula. If we take the example φ again, the same world is represented by $\varphi' :=$ `'['a ; b]`.

Let us call two formulas φ_1 and φ_2

- **equiatomic** or **atomically identical** iff they have the same atom set
- **equivalent** or **boolean identical** iff the evaluation of `[`φ_1 `<=>` φ_2`]` is `!` (this is the usual equivalence in boolean algebras, as previously defined)
- **biequivalent** or **theoretically identical** iff they are equiatomic and equivalent

For every pair φ_1 and φ_2 of formulas holds:

> φ_1 and φ_2 have the same world iff they are biequivalent.

In terms of boolean algebras, two equivalent formulas like `['a , [b ; 'b]]` and `'a` cannot be distinguished. But in the algebra of worlds, formulas are not only determined by their boolean, but also by their atomic properties. Considered as worlds, they "mean" the same if and only if both, they have the same atom set and the same boolean semantic. For example, `['a , [b ; 'b]]` and `'a` are equivalent, but not equiatomic and thus not biequivalent.

Formulas and theories

In logic a *theory* is made of two ingredients:
- a *language*, i.e. a set of sets, generated from a given *vocabulary* or *signature*, and
- a subset of this language, where the elements are called *axioms* or *theorems*.

Usually, these theories are so–called theories of *first–order predicate logic*, where the signature comprises arbitrary function and predicate symbols. But the *propositional logic* theories we consider here are much more primitive and only made of nullary predicate symbols or *atoms* as we call them. So our theories are defined by
- their atom set, and
- their set of axioms.

Expressions in general and the Bucanon formulas in particular are always *finite*. The theories we consider here are always finite, too, in the sense that their atom set if always finite. Although the set of all formulas in general and the set of all theorems or consequences of a given finite theory is always infinite, there is always a finite set of axioms to define it properly. Thus, our theories are given by
- a finite set of atoms, and
- a finite set of axioms.

Saying that the axioms $\{\varphi_1, \ldots, \varphi_n\}$ must hold is the same as saying that their conjunction [φ_1,...,φ_n] holds, i.e. the set of axioms is essentially one axiom. Besides, when the atom set is implicitely present in this axiom, it is not really necessary to explicitely state the atom set. So finally, our specification of a theory reduces to the definition that a theory is given by just a formula. On the other hand, a given formula φ defines its unique theory on the atoms given in φ.

For every pair φ_1 and φ_2 of formulas holds:

> φ_1 and φ_2 denote the same theory iff they are biequivalent.

Propositional logic and boolean algebra

From the point of view we just scetched, formulas, worlds and theories are essentially the same, and this similarities between syntactical and semantical concepts is the basis for the science of (modern) logic. In fact, things become more complicated, when we generalize propositional logic to predicate logic. Our term *world* is usually not used, but *models* is the common title for concrete examples of abstract theories. However, the important and powerful interplay between syntax and semantics remains.

When we look at introductions of propositional logic, the boolean equivalence relation is always defined, but the atomic equivalence relation isn't. For a great deal that has to do with the fact that mathematicians nowadays tend to see propositonal logic as a special branch of (boolean) lattice theory. Lattices are partially ordered sets and the equivalence relation $\Leftrightarrow$ of this partial order $\Rightarrow$ (defined by $x \Leftrightarrow y$ iff $x \Rightarrow y$ and $y \Rightarrow x$) is always the identity, because a partial order is *antisymmetric* by definition, i.e. $x \Rightarrow y$ and $y \Rightarrow x$ implies $x = y$. For formulas, this is not the case, of course, as there are equivalent, but non–identical formulas like `[a , b]` and `[b , a]`. So in order to subsume propositional logic under boolean lattices, the set of formulas is replaced by the set of their *equivalence classes*. This construction of equivalence classes or *quotient structures* is a common trick in mathematics. For example, rational numbers are not defined as fractions $\frac{n}{d}$ of two integers, but as the equivalence classes of those fractions. Otherwise, $\frac{3}{2}$ and $\frac{6}{4}$ would only be equivalent, but not equal and $\frac{3}{2} = \frac{6}{4}$ would be wrong. From a boolean point of view this quotient structure (sometimes called *Lindenbaum–algebra*) is fine, but the atomic properties of formulas are not preserved during the transformation. The two formulas `[a , 'a]` and `[b , 'b]` are equivalent, thus their classes are equal in the boolean algebra. But their atom sets are different. The function `@`, that assigns the set or ordered list `@`φ of its atoms to each given formula φ cannot be redefined in the quotient structure. Accordingly, there is no redefinition of the atomic and theoretical equivalence relations possible.

In order to overcome this shortcoming, we generalize boolean algebras to *quasi–boolean algebras*, where elements can be equivalent without necessarily being equal. While lattices are partially ordered sets (transitive, reflexive and antisymmetric), *quasi–lattices* are special *quasi–ordered* sets (just transitive and reflexive). In this way we can generalize order and lattice theory, define the concept of *quasi–boolean algebras* and *(quasi–) theory algebras* as special quasi–boolean algebras.[1] The standard model of such a (quasi–) theory algebra is the *algebra of worlds*, which are functions of the type $(A \longrightarrow \mathcal{B}) \longrightarrow \mathcal{B}$. Another model is the *algebra of formulas*, which is implemented in Bucanon.

The usual boolean relations (sub– and equivalence), constants (bit values) and operations (negation, conjunction, etc) are defined in a theory algebra. Furthermore there are:[2]

- The atomic and theoretical relations as mentioned above.
- `@`, `-@`, `+@`, three operations that take an element of the algebra and return an atom set or ordered atom list. For example, the Bucanon program evaluates `@[a , 'b , a]` to `[a b]`.
- One operation $\|$ to add atoms and two operations $\Uparrow$ and $\Downarrow$ to reduce the atom set of a given element. In the Bucanon program these three symbols are written as `||`, `<|` and `|>`, respectively.

XPDNF and XPCNF

Let φ be a boolean formula, Δ the PDNF and Γ the PCNF of φ. Let `[`α_1 α_2 ... α_n`]` be the list of *negative atoms* of φ, i.e. these are the atoms of φ that don't occur in Δ and Γ anymore, then

`[`Δ `||` `[`α_1 α_2 ... α_n`]]` is the **extended PDNF** or **XPDNF** of φ
`[`Γ `||` `[`α_1 α_2 ... α_n`]]` is the **extended PCNF** or **XPCNF** of φ

For example, for φ = `[a , 'b , [d ; 'd ; c]]` we have

- Δ = `[; [a , 'b]]` the PDNF
- Γ = `[[; a] , [; 'b]]` the PCNF
- `[c d]` the list of negative atoms of φ and so we get
- `[ [; [a , 'b]] || [c d] ]` the XPDNF and
- `[ [[; a] , [; 'b]] || [c d] ]` the XPCNF of φ

These forms often look much better if they are **simplified**: Δ and Γ are replaced by their simplified forms and an expansion `[`Φ `|| [] ]` with an empty atom list is replaced by Φ only.

In particular, for the example φ = `[a , 'b , [d ; 'd ; c]]` this is

$\Delta = \Gamma$ = `[a , 'b]` the simplified PDNF and PCNF , and so
`[ [a , 'b] || [c d] ]` the simplified XPDNF and XPCNF of φ

The idea of the expansion `||` is, to have an operation available that increases the atom set of φ by $\alpha_1, \ldots, \alpha_n$, but preserves the boolean properties. We can actually define `[...||...]` as a boolean junctor by putting

`[`φ `||` α_1 ... α_n`]` := `[`φ `||` `[`α_1 ... α_n`]]` := `[`φ `,` `[!` `;` α_1 `;` ... `;` α_n`]]`

or alternatively

`[`φ `||` α_1 ... α_n`]` := `[`φ `||` `[`α_1 ... α_n`]]` := `[`φ `;` `[?` `,` α_1 `,` ... `,` α_n`]]`

because in both cases we exactly get, what we expected from `||`:

- `[`φ `||` α_1 ... α_n`]` is equivalent to φ and
- `@[`φ `||` α_1 ... α_n`]` = `@`φ $\cup$ `[`α_1 ... α_n`]`

[1] This whole procedure of generalizing order and lattice theory accordingly is subject of a forthcoming paper.

[2] For an intuitive introduction of the operations and relations of theory algebras, see *The algebra of worlds*, available from *www.bucephalus.org*.