Mathematical Logic

(Based on lecture slides by Stan Burris)

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 300

Subsection 1

Language of Algebras

The Language of Algebras

- A **language** $\mathcal{L}$ **of algebras** (or **algebraic structures**) consists of
  - a set $\mathcal{F}$ of **function symbols** $f, g, h, \ldots$;
  - a set $\mathcal{C}$ of **constant symbols** $c, d, e, \ldots$;
  - a set X of **variables** $x, y, z, \ldots$.
- Each function symbol has an **arity** to indicate how many arguments it takes.

  If the symbol takes n arguments we say it is n-**ary**.
- For small n's we have the terminology

Number of Arguments n	1	2	3	4
The Symbol is	**unary**	**binary**	**ternary**	**quaternary**

Example: The Language of Boolean Algebras

- The language $\mathcal{L}_{\rm BA}$ of Boolean algebras has

$$\mathcal{F} = \{\vee, \wedge, '\}, \quad \mathcal{C} = \{0, 1\},$$

  where
  - $\vee$ and $\wedge$ are binary function symbols;
  - $'$ is a unary function symbol.
- Names:

Symbol	Symbol Name
$\vee$	**join**
$\wedge$	**meet**
$'$	**complement**

- The constants are just called by the usual names **zero** and **one**.

Subsection 2

Interpretations and Algebras

The Meaning of the Symbols

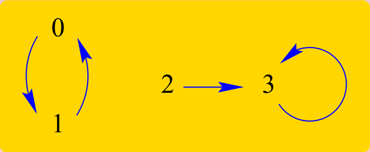
- To assign meaning to the symbols in a language of algebras, we start with a set A, called the **universe** of the algebra;
- Then the symbols of $\mathcal{L}$ are interpreted in A as follows:
  - Function symbols are interpreted as functions on the set.
    More specifically, an n-ary function symbol f is interpreted as a function $f^{\mathbf{A}} : A^n \to A$;
    These are called ***n*-ary functions** because they have n arguments (or inputs);
  - Constant symbols are interpreted as elements of the set.
    The interpretation of a constant symbol c is denoted by $c^{\mathbf{A}}$;
- Variables in X are left uninterpreted;
  They are intended to vary over arbitrary elements of A.

Example: A Simple Language $\mathcal{L}$

- Consider a language $\mathcal{L}$, such that $\mathcal{F} = \{f\}$ and $\mathcal{C} = \emptyset$, with f unary;
- If $A = \{0, 1, 2, 3\}$, we can describe an interpretation $f^{\mathbf{A}} : A \to A$ of f in A using
  - an element-wise description: $0 \mapsto 1, 1 \mapsto 0, 2 \mapsto 3$ and $3 \mapsto 3$;
  - a table, e.g.,

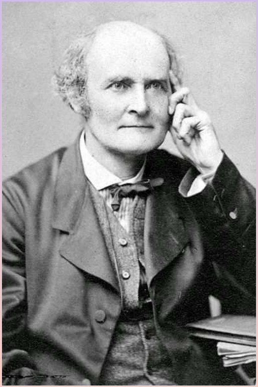
	f
0	1
1	0
2	3
3	3

  - or with a directed graph representation:



Arthur Cayley

- Arthur Cayley, born in Richmond, Surrey, United Kingdom (1821-1895)



Cayley Tables

- We can also describe small binary functions on a set A using a table, called a **Cayley table**;
- To describe the integers mod 4, with the binary operation of multiplication mod 4, we may use the following Cayley table:

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Function Tables

- We can also describe functions on a small set A using a table that is similar to the truth tables used to describe the connectives;
- To describe the ternary function

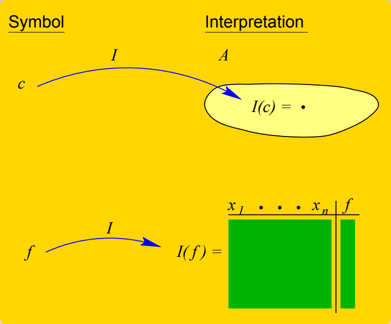
$$f(x, y, z) = 1 + xyz$$

on the integers mod 2, we could use the **function table**

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Interpretations of a Language $\mathcal{L}$: Formal Definition

- An **interpretation** I of the language $\mathcal{L}$ on a nonempty set A assigns to each symbol from $\mathcal{L}$ a function or constant as follows:
  - $I(c) = c^{\mathbf{A}}$ is an element of A for each constant symbol c in $\mathcal{C}$;
  - $I(f) = f^{\mathbf{A}} : A^n \to A$ is an n-ary function on A for each n-ary function symbol f in $\mathcal{F}$.
- Visualizing an interpretation I on a set A:



$\mathcal{L}$-Algebras

- An $\mathcal{L}$-**algebra** (or $\mathcal{L}$-**structure**) $\mathbf{A}$ is a pair $\mathbf{A} = (A, I)$ where
  - A is a set;
  - I is an interpretation of $\mathcal{L}$ on A;
- Given an algebra $\mathbf{A} = (A, I)$:
  - the interpretations of the constant symbols are called the **constants** of the algebra $\mathbf{A}$;
  - the interpretations of the function symbols are called the **fundamental operations** of the algebra $\mathbf{A}$.
- The following notations can all be used:
  - $I(c) = c^{\mathbf{A}} = c$;
  - $I(f) = f^{\mathbf{A}} = f$;
  - $(A, I) = (A, \mathcal{F}, \mathcal{C})$.
- For example, the integers with addition, multiplication, and the zero 0 and unit 1 as constant elements can be written $\mathbb{Z} = (\mathbb{Z}, +, \cdot, 0, 1)$.

Example: Boolean Algebra of Subsets of a Set U

- Let $\mathcal{L} = \mathcal{L}_{\rm BA} = \{\vee, \wedge, ', 0, 1\}$;
- Let $\mathcal{P}(U)$ be the collection of all subsets of a given set U (U is called the **universe** and $\mathcal{P}(U)$ the **powerset** of U);
- Interpret the function symbols and the constants of $\mathcal{L}$ as follows:
    - join as **union** $(\cup)$;
    - meet as **intersection** $(\cap)$;
    - complement as **complement** $(')$ in U;
    - 0 as the **empty set** $(\emptyset)$;
    - 1 as the **universe** (U).
- Then, $\boldsymbol{\mathcal{P}}(U) = (\mathcal{P}(U), \cup, \cap, ', \emptyset, U)$ is **the Boolean algebra of subsets of** U.

The 2-Element Boolean Algebra

- Let $\mathcal{L} = \mathcal{L}_{\mathrm{BA}} = \{\vee, \wedge, ', 0, 1\}$;
- Let $A = \{0, 1\}$ and let the function symbols be interpreted as follows:

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

	$'$
0	1
1	0

  and $0, 1$ are interpreted in the obvious manner (as $0, 1$).
- This is the best known of all the Boolean algebras.
  Sometimes, logicians denote
  - the set $A = \{0, 1\}$ by $2 = \{0, 1\}$;
  - and the algebra by $\mathbf{2} = (2, \vee, \wedge, ', 0, 1)$.

Subsection 3

Terms

Intuition Behind Use of Terms

- Terms are used to make the sides of equations;
- Examples of terms using familiar infix notation for the language $\{+, \cdot, -, 0, 1\}$:
  - $0 \quad 1 \quad x \quad y$
  - $-0 \quad -1 \quad -x \quad -y$
  - $1+0 \quad x \cdot y \quad -(-x) \quad x+1$
  - $x \cdot (y+z) \quad (-x) \cdot (-y) \quad 1+(0+1)$
- Terms constructing using familiar infix notation for the language of Boolean algebra $\mathcal{L}_{\mathrm{BA}} = \{\vee, \wedge, ', 0, 1\}$:
  - $0 \quad 1 \quad x \quad y$
  - $0' \quad 1' \quad x' \quad y'$
  - $1 \vee 0 \quad x \wedge y \quad x'' \quad x \vee 1$
  - $x \wedge (y \vee z) \quad (x') \wedge (y') \quad 1 \vee (0 \vee 1)$

Some More Abstract Examples of Terms

- In the following examples of terms prefix notation will be used:
- If f is a unary function symbol, the following are terms:

$$x \qquad fx \qquad ffx$$

- If c is a constant symbol, the following are terms:

$$c \qquad fc \qquad ffc$$

- If g is a binary function symbol, the following are terms:

$$gcx \qquad gyy \qquad ggfzcgcx$$

- If h is a ternary function symbol, the following are terms:

$$hxyz \qquad hccx \qquad hfgxcgxcggxyfc$$

Formal Definition of Terms

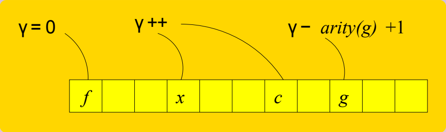
- The $\mathcal{L}$-**terms over** X are defined inductively by the following clauses:
  - A variable x in X is an $\mathcal{L}$-term;
  - A constant symbol c in $\mathcal{C}$ is an $\mathcal{L}$-term;
  - If $t_1, \ldots, t_n$ are $\mathcal{L}$-terms and f is an n-ary function symbol in $\mathcal{F}$, then

$$ft_1 \cdots t_n$$

    is an $\mathcal{L}$-term.

A Parsing Algorithm for Terms in Prefix Form

- Define an integer γ on the symbols of a string $s = fs_1 \cdots s_n$ by:
  - γ is 0 when at first symbol f;
  - Increase γ by 1 when scanning variables or constants;
  - Decrease γ by $\text{arity}(g) - 1$ when scanning a function symbol g;
- Schematically, we have



Decision of the Algorithm

1. s is a term iff the value of γ is always less than $\text{arity}(f)$ except at the last symbol, where γ has the value $\text{arity}(f)$.
2. If s is a term, say $s = ft_1 \cdots t_k$ where $k = \text{arity}(f)$, then, the end of t_i is the first symbol where γ is equal to i.

Illustration of the Algorithm

- Suppose $\mathcal{L} = \{f, g, c\}$, with
  - f unary;
  - g binary;
  - c a constant;
- We use the algorithm to determine if $s = ggcxfz$ is a term.
- Moreover, if it is, we find the subterms t_1 and t_2, such that $gt_1t_2 = ggcxfz$.
- Here is the computation of γ (according to the algorithm):

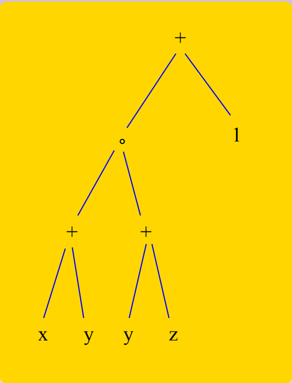
i	0	1	2	3	4	5
s_i	g	g	c	x	f	z
γ_i	0	-1	0	1	1	2

- Conclusions:
  - Since g is binary, $\gamma < 2$ except at last symbol and the algorithm terminates with $\gamma = 2$, the string is a valid term;
  - The first subterm t_1 ends at x; so it is gcx;
  - The second subterm t_2 ends at z; so it is fz.

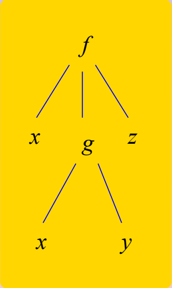
The Syntax Tree of a Term

- The way a term is built can be depicted using a **syntax tree**;
- The following are two examples:

The term $((x + y) \cdot (y + z)) + 1$:



The term $fxgxyz$ (f ternary, g binary):



Syntax Trees and Subterms

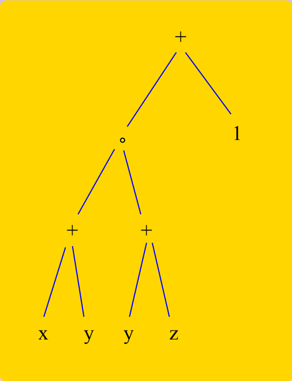
- Looking at the tree of a term we see that it is built up in stages called **subterms**.
- Using infix notation, the subterms of $((x+y)\cdot(y+z))+1$ are

$x \qquad y \qquad z \qquad 1$

$x+y \qquad y+z$

$(x+y)\cdot(y+z)$

$((x+y)\cdot(y+z))+1$



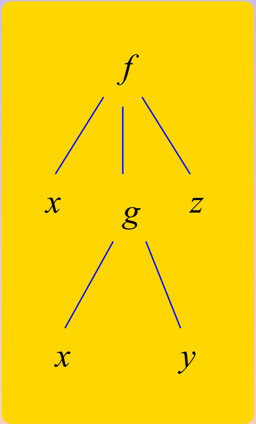
Syntax Trees and Subterms: Another Example

- Using prefix notation, the subterms of $fxgxyz$, with f ternary and g binary, are

$x \quad y \quad z$

gxy

$fxgxyz$



Subterms: Formal Definition

- The **subterms** of a term t are defined inductively:
  - The only subterm of a variable x is the variable x itself;
  - The only subterm of a constant symbol c is the symbol c itself;
  - The subterms of the term $ft_1 \cdots t_n$ are $ft_1 \cdots t_n$ itself and all the subterms of the t_i, for $1 \leq i \leq n$.
- Can we find all subterms of $(x \wedge y) \vee (x' \wedge z)$ carefully using the inductive definition?

$$(x \wedge y) \vee (x' \wedge z)$$

$$x \wedge y \qquad x' \wedge z$$

$$x \qquad y \qquad x' \qquad z$$

x (but we had it already)

Subsection 4

Term Functions

Term Functions Intuitively

- We interpret terms in an algebra as functions;
- Terms $t(x_1, \ldots, x_n)$ define functions $t^{\mathbf{A}} : A^n \to A$;
- Example: Using the usual language for the natural numbers, consider the term

$$t(x, y, z) = (x \cdot (y + 1)) + z$$

The corresponding term function $t^{\mathbb{N}} : \mathbb{N}^3 \to \mathbb{N}$ maps the triple $(1, 0, 2)$ to 3 since $t^{\mathbb{N}}(1, 0, 2) = (1 \cdot (0 + 1)) + 2 = 3$.

Term Functions: Formal Definition

- **Term functions** $t^{\mathbf{A}}$ for terms $t(x_1, \ldots, x_n)$ are the functions on the algebra $\mathbf{A}$ defined inductively by the following:
  - If t is the variable x_i then
  $$t^{\mathbf{A}}(a_1, \ldots, a_n) = a_i;$$
  - If t is the constant $c \in \mathcal{C}$ then
  $$t^{\mathbf{A}}(a_1, \ldots, a_n) = c^{\mathbf{A}};$$
  - If t is the term $ft_1 \cdots t_k$ then
  $$t^{\mathbf{A}} = f^{\mathbf{A}}(t_1^{\mathbf{A}}, \ldots, t_k^{\mathbf{A}}).$$

Evaluation Tables: An Example

- Let $\mathbf{2} = (\{0,1\}, \vee, \wedge, ', 0, 1)$ be our familiar 2-element Boolean algebra;
- Let
$$t(x,y,z) = x \vee (y \wedge z')$$
- The function $t^{\mathbf{2}} : \{0,1\}^3 \to \{0,1\}$ may be described by the following **evaluation table**:

x	y	z	z'	$y \wedge z'$	t
1	1	1	0	0	1
1	1	0	1	1	1
1	0	1	0	0	1
1	0	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
0	0	1	0	0	0
0	0	0	1	0	0

or

x	y	z	t
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	0