Semantics

Greg Kobele

May 7, 2018

Review of Results

The big problems

1. The meaning(s) of constructions
2. Same words, different constructions

The (proposed) solutions

1. The meaning(s) of constructions
   - for any construction *C*, its meaning contribution is simply function application
2. Same words, different constructions
   - words like *and*, *or*, and *not* do the same thing in different domains

Plan

Today
Explore the consequences of these proposals

1. Types
2. NPs

Boolean Algebra Review

What does it mean to
do the same thing in different domains?

Structure in domains

Boolean Algebra

- greatest lower bound
- least upper bound
- complement

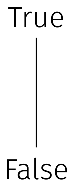
Based on *ordering*

$x \wedge y$ the biggest thing in: $\{z : z \leq x \ \& \ z \leq y\}$

$x \vee y$ the smallest thing in: $\{z : x \leq z \ \& \ y \leq z\}$

$\neg x$ the unique thing s.t. $\neg x \wedge x = 0$ and $\neg x \vee x = 1$

Hasse Diagrams



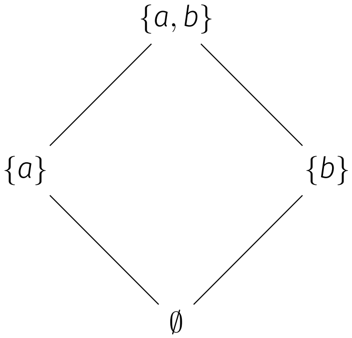
$a \leq b$ **if**

1. a is connected to b
2. but not above it

1
the top element

0
the bottom element

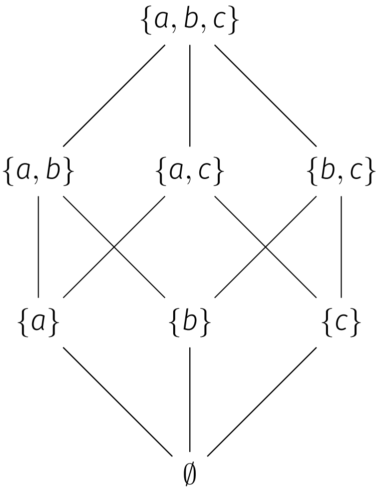
The Hasse Diagram for $\wp(\{a, b\})$



$x \wedge y$
the smallest thing above (or equal to) both x and y

$\neg x$
the element ’opposite’ x

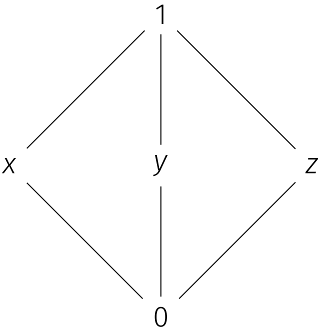
The Hasse Diagram for $\wp(\{a, b, c\})$



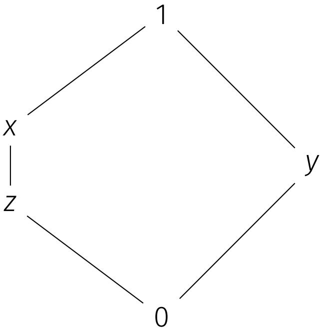
Non-Boolean Algebras

Distributivity
$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

Diamond

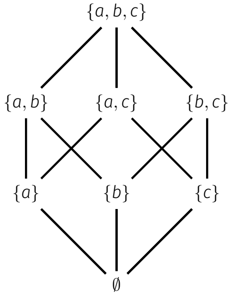


Pentagon



Atoms

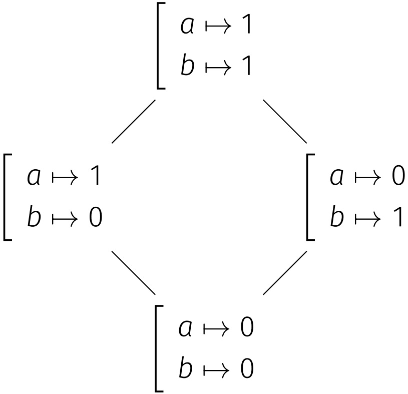
An atom is
a minimal non-zero element



Functions and Types

Sets and Functions

$\wp(A) \cong [A \to 2]$



Boolean operations

complement

$$\neg \begin{bmatrix} a \mapsto 1 \\ b \mapsto 0 \end{bmatrix} = \begin{bmatrix} a \mapsto \neg 1 \\ b \mapsto \neg 0 \end{bmatrix} = \begin{bmatrix} a \mapsto 0 \\ b \mapsto 1 \end{bmatrix}$$

glb

$$\begin{bmatrix} a \mapsto 1 \\ b \mapsto 0 \end{bmatrix} \wedge \begin{bmatrix} a \mapsto 0 \\ b \mapsto 1 \end{bmatrix} = \begin{bmatrix} a \mapsto 1 \wedge 0 \\ b \mapsto 0 \wedge 1 \end{bmatrix} = \begin{bmatrix} a \mapsto 0 \\ b \mapsto 0 \end{bmatrix}$$

Building Boolean lattices

if B is a boolean lattice, then for any set A

$[A \to B]$ is a boolean lattice

- $f \leq g$ iff for every a, $f(a) \leq g(a)$
- $(f \wedge g)(a) := f(a) \wedge g(a)$
- $1_{A \to B}(a) := 1_B$
- $(\neg f)(a) := \neg(f(a))$

(Recall:) Functions

A special kind of binary relation - $f \subseteq A \times B$

- each left-hand-side is paired with exactly one right-hand-side
- A is the domain
- B is the codomain

$[A \to B]$
is the set of all *functions* with domain A and codomain B

`It doesn't make sense`
to apply a function to an argument of the 'wrong' type

Types

A type
is a description of what *kind* of object something is

The type of entities
is conventionally given as e

The type of truth values
as t

The type of a function with domain A and codomain B
can be written as $a \to b$, or even (ab)

- here a is the *type* of things in domain A
- and b the *type* of things in codomain B

Solving for Unknowns

Function application

when can f apply to a?

- when a is the type of thing f can apply to

$$\underbrace{f}_{(xy)}(\underbrace{a}_{x}) = \underbrace{b}_{y}$$

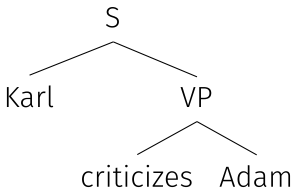
If you know two types, you can find the third

$$\underbrace{f}_{(xy)}(\underbrace{a}_{x}) = \underbrace{b}_{\square}$$

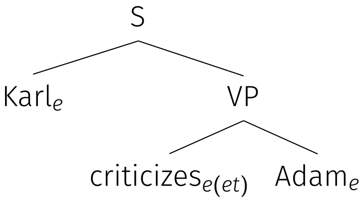
$$\underbrace{f}_{(xy)}(\underbrace{a}_{\square}) = \underbrace{b}_{y}$$

Examples

Karl criticizes Adam

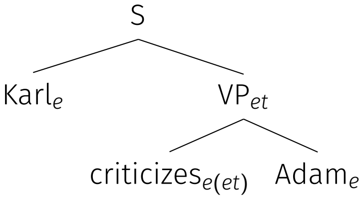


Karl criticizes Adam



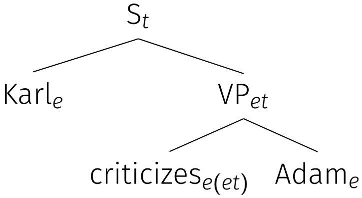
- Types of words

Karl criticizes Adam



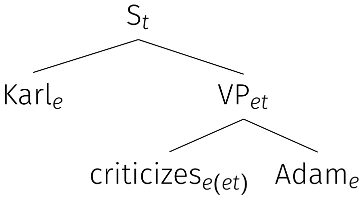
- Types of words
- Solve for type at each node

Karl criticizes Adam



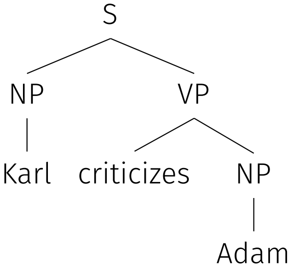
- Types of words
- Solve for type at each node

Karl criticizes Adam

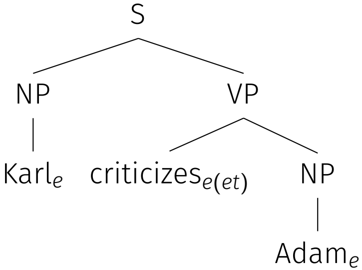


- Types of words
- Solve for type at each node
- Compute meaning

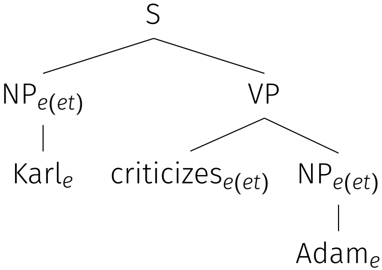
Non-branching nodes?



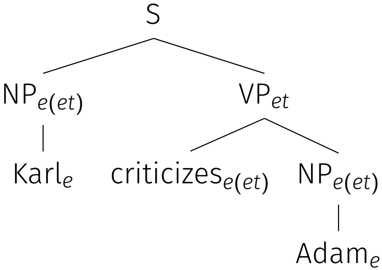
Non-branching nodes?



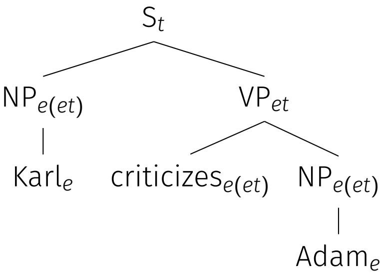
Non-branching nodes?



Non-branching nodes?



Non-branching nodes?



Bill drew a pretty picture of Mark