

Reference Sheet for “What I’m Currently Learning”

Hello, World!

Pretty PDF Enter `M-x compile` to produce a nice looking PDF of your reference sheet.
◊ I’ve bound this command to `C-c C-m` in my Emacs setup ;-)

Section Headers A usual Org header, say `* my section`, results in the boxed headers used in this cheat sheet.

Parallel Environments The sequence `<p TAB` produces a ‘parallel’ environment for producing text side-by-side. The column break is automatic, but as this is sugar for a `minipage` containing a `multicolumn` we can force a column separation with `\columnbreak`. This command, in Org, necessitates newlines between the items being separated.

To learn more, manipulating this source is the way to go!

Org-mode Basics

Read [Org-mode for beginners](#) for a refresher!
◊ For more see [The Compact Org-mode Guide](#).

Reloading To reload a file with updated org settings, press `C-c C-c` on a settings line –i.e., one beginning with a `#`, to reset the temporary file cache.

Inclusion During export, you can include the content of another file.

- ◊ Syntax: `#+INCLUDE: "<fileName>" [<markup> [<language>]]`
 - `markup ::= src | example`
 - `language ::= C | haskell | emacs-lisp | ...`
 - If the markup is not given, the text will be assumed to be in Org mode format and will be processed normally; c.f., [Setup files](#).
- ◊ To visit the file, `C-c '` while the cursor is on the line with the file name.

Grep

Find all files containing specific text

‘r’ecursively look for the ‘w’hole given pattern:
`grep -rw '/path/to/somewhere/' -e 'pattern'`

Better `ack 'text-to-find-here' locationToBeginLooking`

- ◊ `ack` is like `grep`, but for source code.
- ◊ It looks prettier and more informative.

The next page contains sample output from a logic-based class.

Propositional Calculus

Metatheorem Any two theorems are equivalent; ‘ true ’ is a theorem.

Equivalence is an equivalence relation that is associative — $((p \equiv q) \equiv r) \equiv (p \equiv (q \equiv r))$ — and has identity true.

Discrepancy ‘ \neq ’ is symmetric, associative, has identity ‘ false ’, mutually associates with equivalence — $((p \neq q) \equiv r) \equiv (p \neq (q \equiv r))$ — and mutually interchanges with it as well — $p \neq q \equiv r \equiv p \equiv q \neq r$ —.

Implication has the alternative definition $p \Rightarrow q \equiv \neg p \vee q$, has true as left identity and false as right zero, distributes over \equiv in the second argument, and is self-distributive; and has the properties

Shunting

$$p \wedge q \Rightarrow r \quad \equiv \quad p \Rightarrow (q \Rightarrow r)$$

Contrapositive

$$p \Rightarrow q \quad \equiv \quad \neg q \Rightarrow \neg p$$

Modus Ponens

$$\begin{aligned} p \wedge (p \Rightarrow q) &\equiv p \wedge q \\ p \wedge (q \Rightarrow p) &\equiv p \\ p \wedge (p \Rightarrow q) &\Rightarrow q \end{aligned}$$

Moreover it has the useful property “(3.62)”: $p \Rightarrow (q \equiv r) \equiv p \wedge q \equiv p \wedge r$.

Conjunction and disjunction distribute over one another, \vee distributes over \equiv , \wedge distributes over $\equiv - \equiv$ in that $p \wedge (q \equiv r \equiv s) \equiv p \wedge q \equiv p \wedge r \equiv p \wedge s$, and they satisfy,

Excluded Middle

$$p \vee \neg p$$

Contradiction

$$p \wedge \neg p \equiv \text{false}$$

Absorption

$$\begin{aligned} p \wedge (\neg p \vee q) &\equiv p \wedge q \\ p \vee (\neg p \wedge q) &\equiv p \vee q \end{aligned}$$

De Morgan

$$\begin{aligned} \neg(p \wedge q) &\equiv \neg p \vee \neg q \\ \neg(p \vee q) &\equiv \neg p \wedge \neg q \end{aligned}$$

Loops implement finite quantifications

A finite quantification can be defined axiomatically by the empty-range rule and split-off term rules. Together these form a recursive definition which can be phrased as a loop.

```
// For  $\oplus$  :  $T \rightarrow T \rightarrow T$ ,
// fold(A,a,b)  $\approx (\oplus x : a..b-1 \bullet A[x])$ 
/*@ axiomatic Fold {
  @
  @ logic T
  @ fold{L}(T *A, integer a, integer b)
  @ reads a,b,A, A[...] ;
  @
  @ axiom foldEmptyRange{L} :
  @   \forall a,b,A, integer a, b; a >= b
  @   ==> fold(A,a,b) == identity( $\oplus$ );
  @
  @ axiom foldSplitOffTerm{L} :
  @   \forall a,b,A, integer a, b; a <= b
  @   ==> fold(A, a, b+1)
  @       == fold(A, a, b )  $\oplus$  A[b];
  @ }
  @*/
```

```
/*@ requires \valid(A+(0..N-1));
   @ assigns \nothing;
   @ ensures \result == fold(A,0,N);
   @*/
T fold(int N, T* A) {
  T total = identity( $\oplus$ );

  /*@ loop invariant 0 <= n <= N;
   @ loop invariant total == fold(A,0,n);
   @ loop assigns n, total;
   @ loop variant N-n;
   */
  for(int n = 0; n != N; n++)
    total = total  $\oplus$  A[n];
  return total;
}
```

This pseudo-code is reified by giving concrete values for (T, \oplus , identity) such as (int, +, 0) or (bool, ||, false). Any *monoid* will do.

Lattices

The distributive lattice interface $(L, \sqsubseteq, \sqcap, \sqcup, \perp, \top)$ has the following implementations:

- ◊ Booleans: $(\mathbb{B}, \Rightarrow, \wedge, \vee, \text{false}, \text{true})$ —Our ambient logic!
- ◊ Extended Number Line: $(\mathbb{R}, \leq, \min, \max, -\infty, +\infty)$
- ◊ Naturals under division: $(\mathbb{N}, |, \gcd, \text{lcm}, 1, 0)$
- ◊ Substructures of a given *datatype* with the substructure ordering.

E.g., sets, lists, and graphs with subset, subsequence, and subgraph ordering.

An *order* is a relation $\sqsubseteq : L \rightarrow L \rightarrow \mathbb{B}$ satisfying the following three properties:

Reflexivity

$$a \sqsubseteq a$$

Transitivity

$$a \sqsubseteq b \wedge b \sqsubseteq c \Rightarrow a \sqsubseteq c$$

Antisymmetry

$$a \sqsubseteq b \wedge b \sqsubseteq a \Rightarrow a = b$$

An order is *bounded* if there are elements $\top, \perp : L$ being the lower and upper bounds of all other elements:

Top Element

$$a \sqsubseteq \top$$

Bottom Element

$$\perp \sqsubseteq a$$

A *lattice* is a pair of operations $\sqcap, \sqcup : L \rightarrow L \rightarrow L$ specified by the properties:

\sqcup -Characterisation

$$a \sqsubseteq c \wedge b \sqsubseteq c \equiv a \sqcup b \sqsubseteq c$$

\sqcap -Characterisation

$$c \sqsubseteq a \wedge c \sqsubseteq b \equiv c \sqsubseteq a \sqcap b$$

Let \square be one of \sqcap or \sqcup , then:

Symmetry of \square

$$a \square b = b \square a$$

Associativity of \square

$$(a \square b) \square c = a \square (b \square c)$$

Idempotency of \square

$$a \square a = a$$

Zero of \square

$$a \sqcup \top = \top$$

$$a \sqcap \perp = \perp$$

Identity of \square

$$a \sqcup \perp = a$$

$$a \sqcap \top = a$$

Absorption

$$a \sqcap (a \sqcup b) = a$$

$$a \sqcup (a \sqcap b) = a$$

Self-Distributivity of \square

$$a \square (b \square c) = (a \square b) \square (a \square c)$$

Weakening / Strengthening

$$a \sqsubseteq a \sqcup b$$

$$a \sqcap b \sqsubseteq a$$

$$a \sqcap b \sqsubseteq a \sqcup b$$

$$a \sqcup (b \sqcap c) \sqsubseteq a \sqcup b$$

$$a \sqcap b \sqsubseteq a \sqcap (b \sqcup c)$$

Induced Defs. of Inclusion

$$a \sqsubseteq b \equiv a \sqcup b = b$$

$$a \sqsubseteq b \equiv a \sqcap b = a$$

Monotonicity of \square

$$a \sqsubseteq b \wedge c \sqsubseteq d \Rightarrow a \square c \sqsubseteq b \square d$$

Golden Rule

$$a \sqcap b = a \equiv b = a \sqcup b$$

$$a \sqcap b = a \sqcup b \equiv a = b$$

$$a \sqcup b \sqsubseteq a \sqcap b \equiv a = b$$

Duality Principle:

If a statement S is a theorem, then so is $S[(\sqsubseteq, \sqcap, \sqcup, \top, \perp) := (\sqsupseteq, \sqcup, \sqcap, \perp, \top)]$.