Practical examples of writing programs and proving theorems in Idris.

Donovan Crichton

January 2020

Preliminaries

Propositional Logic

- Concerned with statements of verifiable facts.
- Used daily by programmers when reasoning about Boolean values.

Symbol	**Meaning**	**Example**
T, F	True, False	Boolean values.
$p, q, r, \ldots$	Propositions	Let $p =$ "It is raining."
$\neg$	Negation (Not)	$\neg p$
$\wedge$	Conjuction (And)	$p \wedge q$
$\vee$	Disjunction (Or)	$p \vee q$
$\rightarrow$	Implication (If)	$p \rightarrow q$
$\leftrightarrow$	Bi Implication (Iff)	$p \leftrightarrow q$
$\equiv$	Equivalence	$p \equiv q$
$\top$	Tautology	$p \vee \neg p \equiv \top$
$\bot$	Contradiction	$p \wedge \neg p \equiv \bot$

Definitions of Connectives

Conjunction (And)

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction (Or)

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Negation (Not)

p	$\neg p$
T	F
F	T

Implication (If)

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Bi Implication (Iff)

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Logical Equivalence

p	q	$p \equiv q$
T	T	T
T	F	F
F	T	F
F	F	T

Proof Techniques

By Exhaustion

Idea: Prove by enumerating all possible cases.

Prove: $(\neg p \vee q) \leftrightarrow (p \rightarrow q)$.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$	$(\neg p \vee q) \leftrightarrow (p \rightarrow q)$
T	T	F	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

Proof Techniques

By Appeal to Lemma

Idea: Introduce pre-proven smaller proofs (called a Lemma) to prove a larger proof.

- **Lemma 1.** $p \vee \neg p \equiv \top$.
- **Lemma 2.** $(p \equiv q) \equiv (p \leftrightarrow q)$.

Prove: $(p \leftrightarrow q) \vee \neg(p \equiv q) \leftrightarrow \top$.

$$
\begin{aligned}
(p \leftrightarrow q) \vee \neg(p \equiv q) \leftrightarrow \top &\qquad \text{Premise.} \\
(p \equiv q) \vee \neg(p \equiv q) \equiv \top &\qquad \text{Lemma 2.} \\
\top \equiv \top &\qquad \text{Lemma 1.}
\end{aligned}
$$

□

First Order (or Predicate) Logic

Extends propositional logic from reasoning about propositions to reasoning about sets.

Symbol	**Meaning**	**Example**
$X, Y, Z, \ldots$	Set Variables	Let $Y = \{2, 3, 4\}$.
$x, y, z, \ldots$	Member Variables	Let $z = 2$.
$P(x), Q(y), \ldots$	Predicate Variables	Let $Q(y) = y > 1$.
$\forall x \in X, P(x)$	Universal Quantifier	$\forall y \in Y, Q(y)$
$\exists x \in X, P(x)$	Existential Quantifier	$\exists z \in Y, z = 2$

Proof Techniques
Induction

- Allows us to prove that a property $P(x)$ holds $\forall x \in X$. Provided X is well-founded.
- Informally well-founded means "no infinite decreasing chains".

Prove: $\forall n \in \mathbb{N}(\exists y \in \mathbb{N}, y = n + 1)$

$$\begin{aligned} y &= 0 + 1 \\ &= 1 \end{aligned}$$

Base Case. $n = 0$ □

$$\begin{aligned} y &= (k + 1) + 1 \\ &= k + 2 \end{aligned}$$

Inductive Step. $n = k + 1$ □

Why should I care?

- Types are just sets with flavour!
- **Bool** $= \{\mathit{True}, \mathit{False}\}$
- **Int** $= \{-\infty, ..., -2, -1, 0, 1, 2, 3, ..., \infty\}$
- Mixing of flavours is not allowed!
- $\{\mathit{True}, -2, "\mathit{Hello}", 1\}$ Can really only be said to be a "thing" flavoured set.

Propositions as Types. Proofs as Programs

Curry-Howard in Idris

What is truth?

Vacuous Truths

Statements that are true because the premise cannot be satisfied.
Often arise as a base case in proofs by induction.

- $F \to q \equiv T$
- $\forall x \in \{\} P(x)$.

Natural Numbers

Some small examples in Idris