13 Classical Logic

In constructive logic, a proposition is true exactly when it has a proof, a derivation of it from axioms and assumptions, and is false exactly when it has a refutation, a derivation of a contradiction from the assumption that it is true. Constructive logic is a logic of positive evidence. To affirm or deny a proposition requires a proof, either of the proposition itself, or of a contradiction, under the assumption that it has a proof. We are not always able to affirm or deny a proposition. An open problem is one for which we have neither a proof nor a refutation—constructively speaking, it is neither true nor false.

In contrast, classical logic (the one we learned in school) is a logic of perfect information where every proposition is either true or false. We may say that classical logic corresponds to "god's view" of the world—there are no open problems, rather all propositions are either true or false. Put another way, to assert that every proposition is either true or false is to weaken the notion of truth to encompass all that is not false, dually to the constructively (and classically) valid interpretation of falsity as all that is not true. The symmetry between truth and falsity is appealing, but there is a price to pay for this: the meanings of the logical connectives are weaker in the classical case than in the constructive.

The law of the excluded middle provides a prime example. Constructively, this principle is not universally valid, as we have seen in Exercise **12.1**. Classically, however, it is valid, because every proposition is either false or not false, and being not false is the same as being true. Nevertheless, classical logic is consistent with constructive logic in that constructive logic does not refute classical logic. As we have seen, constructive logic proves that the law of the excluded middle is positively not refuted (its double negation is constructively true). Consequently, constructive logic is stronger (more expressive) than classical logic, because it can express more distinctions (namely, between affirmation and irrefutability), and because it is consistent with classical logic.

Proofs in constructive logic have computational content: they can be executed as programs, and their behavior is described by their type. Proofs in classical logic also have computational content, but in a weaker sense than in constructive logic. Rather than positively affirm a proposition, a proof in classical logic is a computation that cannot be refuted. Computationally, a refutation consists of a continuation, or control stack, that takes a proof of a proposition and derives a contradiction from it. So a proof of a proposition in classical logic is a computation that, when given a refutation of that proposition derives a contradiction, witnessing the impossibility of refuting it. In this sense, the law of the excluded middle has a proof, precisely because it is irrefutable.

13.1 Classical Logic

In constructive logic, a connective is defined by giving its introduction and elimination rules. In classical logic, a connective is defined by giving its truth and falsity conditions. Its truth rules correspond to introduction, and its falsity rules to elimination. The symmetry between truth and falsity is expressed by the principle of indirect proof. To show that $\phi\ \mathsf{true}$ it is enough to show that $\phi\ \mathsf{false}$ entails a contradiction, and, conversely, to show that $\phi\ \mathsf{false}$ it is enough to show that $\phi\ \mathsf{true}$ leads to a contradiction. Although the second of these is constructively valid, the first is fundamentally classical, expressing the principle of indirect proof.

13.1.1 Provability and Refutability

There are three basic judgment forms in classical logic:

1. $\phi\ \mathsf{true}$, stating that the proposition ϕ is provable;
2. $\phi\ \mathsf{false}$, stating that the proposition ϕ is refutable;
3. #, stating that a contradiction has been derived.

These are extended to hypothetical judgments in which we admit both provability and refutability assumptions:

$$\phi_1\ \mathsf{false}, \ldots, \phi_m\ \mathsf{false}\ \psi_1\ \mathsf{true}, \ldots, \psi_n\ \mathsf{true} \vdash J.$$

The hypotheses are divided into two zones, one for falsity assumptions, Δ, and one for truth assumptions, Γ.

The rules of classical logic are organized around the symmetry between truth and falsity, which is mediated by the contradiction judgment.

The hypothetical judgment is reflexive:

$$\frac{}{\Delta, \phi\ \mathsf{false}\ \Gamma \vdash \phi\ \mathsf{false}} \tag{13.1a}$$

$$\frac{}{\Delta\ \Gamma, \phi\ \mathsf{true} \vdash \phi\ \mathsf{true}} \tag{13.1b}$$

The remaining rules are stated so that the structural properties of weakening, contraction, and transitivity are admissible.

A contradiction arises when a proposition is judged both true and false. A proposition is true if its falsity is absurd, and is false if its truth is absurd.

$$\frac{\Delta\ \Gamma \vdash \phi\ \mathsf{false} \qquad \Delta\ \Gamma \vdash \phi\ \mathsf{true}}{\Delta\ \Gamma \vdash \#} \tag{13.1c}$$

$$\frac{\Delta, \phi\ \mathsf{false}\ \Gamma \vdash \#}{\Delta\ \Gamma \vdash \phi\ \mathsf{true}} \tag{13.1d}$$

$$\frac{\Delta\ \Gamma, \phi\ \mathsf{true} \vdash \#}{\Delta\ \Gamma \vdash \phi\ \mathsf{false}} \tag{13.1e}$$

Truth is trivially true and cannot be refuted.

$$\frac{}{\Delta\ \Gamma \vdash \top \ \mathsf{true}} \tag{13.1f}$$

A conjunction is true if both conjuncts are true and is false if either conjunct is false.

$$\frac{\Delta\ \Gamma \vdash \phi_1 \ \mathsf{true} \quad \Delta\ \Gamma \vdash \phi_2 \ \mathsf{true}}{\Delta\ \Gamma \vdash \phi_1 \wedge \phi_2 \ \mathsf{true}} \tag{13.1g}$$

$$\frac{\Delta\ \Gamma \vdash \phi_1 \ \mathsf{false}}{\Delta\ \Gamma \vdash \phi_1 \wedge \phi_2 \ \mathsf{false}} \tag{13.1h}$$

$$\frac{\Delta\ \Gamma \vdash \phi_2 \ \mathsf{false}}{\Delta\ \Gamma \vdash \phi_1 \wedge \phi_2 \ \mathsf{false}} \tag{13.1i}$$

Falsity is trivially false and cannot be proved.

$$\frac{}{\Delta\ \Gamma \vdash \bot \ \mathsf{false}} \tag{13.1j}$$

A disjunction is true if either disjunct is true and is false if both disjuncts are false.

$$\frac{\Delta\ \Gamma \vdash \phi_1 \ \mathsf{true}}{\Delta\ \Gamma \vdash \phi_1 \vee \phi_2 \ \mathsf{true}} \tag{13.1k}$$

$$\frac{\Delta\ \Gamma \vdash \phi_2 \ \mathsf{true}}{\Delta\ \Gamma \vdash \phi_1 \vee \phi_2 \ \mathsf{true}} \tag{13.1l}$$

$$\frac{\Delta\ \Gamma \vdash \phi_1 \ \mathsf{false} \quad \Delta\ \Gamma \vdash \phi_2 \ \mathsf{false}}{\Delta\ \Gamma \vdash \phi_1 \vee \phi_2 \ \mathsf{false}} \tag{13.1m}$$

Negation inverts the sense of each judgment:

$$\frac{\Delta\ \Gamma \vdash \phi \ \mathsf{false}}{\Delta\ \Gamma \vdash \neg\phi \ \mathsf{true}} \tag{13.1n}$$

$$\frac{\Delta\ \Gamma \vdash \phi \ \mathsf{true}}{\Delta\ \Gamma \vdash \neg\phi \ \mathsf{false}} \tag{13.1o}$$

An implication is true if its conclusion is true when the assumption is true and is false if its conclusion is false yet its assumption is true.

$$\frac{\Delta\ \Gamma, \phi_1 \ \mathsf{true} \vdash \phi_2 \ \mathsf{true}}{\Delta\ \Gamma \vdash \phi_1 \supset \phi_2 \ \mathsf{true}} \tag{13.1p}$$

$$\frac{\Delta\ \Gamma \vdash \phi_1 \ \mathsf{true} \quad \Delta\ \Gamma \vdash \phi_2 \ \mathsf{false}}{\Delta\ \Gamma \vdash \phi_1 \supset \phi_2 \ \mathsf{false}} \tag{13.1q}$$

13.1.2 Proofs and Refutations

To explain the dynamics of classical proofs, we first introduce an explicit syntax for proofs and refutations. We will define three hypothetical judgments for classical logic with explicit derivations:

1. $\Delta\ \Gamma \vdash p : \phi$, stating that p is a proof of ϕ;

2. $\Delta\ \Gamma \vdash k \div \phi$, stating that k is a refutation of ϕ;
3. $\Delta\ \Gamma \vdash k\ \#\ p$, stating that k and p are contradictory.

The falsity assumptions Δ are given by a context of the form

$$u_1 \div \phi_1, \ldots, u_m \div \phi_m,$$

where $m \geq 0$, in which the variables $u_1, \ldots, u_n$ stand for refutations. The truth assumptions Γ are given by a context of the form

$$x_1 : \psi_1, \ldots, x_n : \psi_n,$$

where $n \geq 0$, in which the variables $x_1, \ldots, x_n$ stand for proofs.

The syntax of proofs and refutations is given by the following grammar:

Prf	p	$::=$	$\texttt{true-T}$	$\langle\rangle$	truth
			$\texttt{and-T}(p_1; p_2)$	$\langle p_1, p_2\rangle$	conjunction
			$\texttt{or-T[l]}(p)$	$\texttt{l} \cdot p$	disjunction left
			$\texttt{or-T[r]}(p)$	$\texttt{r} \cdot p$	disjunction right
			$\texttt{not-T}(k)$	$\texttt{not}(k)$	negation
			$\texttt{imp-T}(x.p)$	$\lambda\,(x)\,p$	implication
			$\texttt{ccr}(u.(k\ \#\ p))$	$\texttt{ccr}(u.(k\ \#\ p))$	contradiction
Ref	k	$::=$	$\texttt{false-F}$	$\texttt{abort}$	falsehood
			$\texttt{and-F[l]}(k)$	$\texttt{fst}\,;k$	conjunction left
			$\texttt{and-F[r]}(k)$	$\texttt{snd}\,;k$	conjunction right
			$\texttt{or-F}(k_1; k_2)$	$\texttt{case}(k_1; k_2)$	disjunction
			$\texttt{not-F}(p)$	$\texttt{not}(p)$	negation
			$\texttt{imp-F}(p; k)$	$\texttt{ap}(p)\,;k$	implication
			$\texttt{ccp}(x.(k\ \#\ p))$	$\texttt{ccp}(x.(k\ \#\ p))$	contradiction

Proofs serve as evidence for truth judgments, and refutations serve as evidence for false judgments. Contradictions are witnessed by the juxtaposition of a proof and a refutation.

A contradiction arises when a proposition is both true and false:

$$\frac{\Delta\ \Gamma \vdash k \div \phi \quad \Delta\ \Gamma \vdash p : \phi}{\Delta\ \Gamma \vdash k\ \#\ p} \tag{13.2a}$$

Truth and falsity are defined symmetrically in terms of contradiction:

$$\frac{\Delta, u \div \phi\ \Gamma \vdash k\ \#\ p}{\Delta\ \Gamma \vdash \texttt{ccr}(u.(k\ \#\ p)) : \phi} \tag{13.2b}$$

$$\frac{\Delta\ \Gamma, x : \phi \vdash k\ \#\ p}{\Delta\ \Gamma \vdash \texttt{ccp}(x.(k\ \#\ p)) \div \phi} \tag{13.2c}$$

Reflexivity corresponds to the use of a variable hypothesis:

$$\overline{\Delta, u \div \phi\ \Gamma \vdash u \div \phi} \tag{13.2d}$$

$$\overline{\Delta\ \Gamma, x : \phi \vdash x : \phi} \tag{13.2e}$$

The other structure properties are admissible.

Truth is trivially true and cannot be refuted.

$$\frac{}{\Delta\ \Gamma \vdash \langle\rangle : \top} \tag{13.2f}$$

A conjunction is true if both conjuncts are true and is false if either conjunct is false.

$$\frac{\Delta\ \Gamma \vdash p_1 : \phi_1 \quad \Delta\ \Gamma \vdash p_2 : \phi_2}{\Delta\ \Gamma \vdash \langle p_1, p_2\rangle : \phi_1 \wedge \phi_2} \tag{13.2g}$$

$$\frac{\Delta\ \Gamma \vdash k_1 \div \phi_1}{\Delta\ \Gamma \vdash \texttt{fst}\,; k_1 \div \phi_1 \wedge \phi_2} \tag{13.2h}$$

$$\frac{\Delta\ \Gamma \vdash k_2 \div \phi_2}{\Delta\ \Gamma \vdash \texttt{snd}\,; k_2 \div \phi_1 \wedge \phi_2} \tag{13.2i}$$

Falsity is trivially false and cannot be proved.

$$\frac{}{\Delta\ \Gamma \vdash \texttt{abort} \div \bot} \tag{13.2j}$$

A disjunction is true if either disjunct is true and is false if both disjuncts are false.

$$\frac{\Delta\ \Gamma \vdash p_1 : \phi_1}{\Delta\ \Gamma \vdash \texttt{l} \cdot p_1 : \phi_1 \vee \phi_2} \tag{13.2k}$$

$$\frac{\Delta\ \Gamma \vdash p_2 : \phi_2}{\Delta\ \Gamma \vdash \texttt{r} \cdot p_2 : \phi_1 \vee \phi_2} \tag{13.2l}$$

$$\frac{\Delta\ \Gamma \vdash k_1 \div \phi_1 \quad \Delta\ \Gamma \vdash k_2 \div \phi_2}{\Delta\ \Gamma \vdash \texttt{case}(k_1; k_2) \div \phi_1 \vee \phi_2} \tag{13.2m}$$

Negation inverts the sense of each judgment:

$$\frac{\Delta\ \Gamma \vdash k \div \phi}{\Delta\ \Gamma \vdash \texttt{not}(k) : \neg\phi} \tag{13.2n}$$

$$\frac{\Delta\ \Gamma \vdash p : \phi}{\Delta\ \Gamma \vdash \texttt{not}(p) \div \neg\phi} \tag{13.2o}$$

An implication is true if its conclusion is true when the assumption is true and is false if its conclusion is false, yet its assumption is true.

$$\frac{\Delta\ \Gamma, x : \phi_1 \vdash p_2 : \phi_2}{\Delta\ \Gamma \vdash \lambda\,(x)\, p_2 : \phi_1 \supset \phi_2} \tag{13.2p}$$

$$\frac{\Delta\ \Gamma \vdash p_1 : \phi_1 \quad \Delta\ \Gamma \vdash k_2 \div \phi_2}{\Delta\ \Gamma \vdash \mathsf{ap}(p_1)\,; k_2 \div \phi_1 \supset \phi_2} \tag{13.2q}$$

13.2 Deriving Elimination Forms

The price of achieving a symmetry between truth and falsity in classical logic is that we must very often rely on the principle of indirect proof: to show that a proposition is true, we often must derive a contradiction from the assumption of its falsity. For example, a proof of

$$(\phi \wedge (\psi \wedge \theta)) \supset (\theta \wedge \phi)$$

in classical logic has the form

$$\lambda\,(w)\,\texttt{ccr}(u.(k \mathbin{\#} w)),$$

where k is the refutation

$$\texttt{fst}\,;\texttt{ccp}(x.(\texttt{snd}\,;\texttt{ccp}(y.(\texttt{snd}\,;\texttt{ccp}(z.(u \mathbin{\#} \langle z, x\rangle)) \mathbin{\#} y)) \mathbin{\#} w)).$$

And yet in constructive logic this proposition has a direct proof that avoids the circumlocutions of proof by contradiction:

$$\lambda\,(w)\,\langle w \cdot \texttt{r} \cdot \texttt{r}, w \cdot \texttt{l}\rangle.$$

But this proof cannot be expressed (as is) in classical logic, because classical logic lacks the elimination forms of constructive logic.

However, we may package the use of indirect proof into a slightly more palatable form by deriving the elimination rules of constructive logic. For example, the rule

$$\frac{\Delta\ \Gamma \vdash \phi \wedge \psi \ \mathsf{true}}{\Delta\ \Gamma \vdash \phi \ \mathsf{true}}$$

is derivable in classical logic:

$$\dfrac{\dfrac{\dfrac{}{\Delta, \phi \ \mathsf{false}\ \Gamma \vdash \phi \ \mathsf{false}}}{\Delta, \phi \ \mathsf{false}\ \Gamma \vdash \phi \wedge \psi \ \mathsf{false}} \quad \dfrac{\Delta\ \Gamma \vdash \phi \wedge \psi \ \mathsf{true}}{\Delta, \phi \ \mathsf{false}\ \Gamma \vdash \phi \wedge \psi \ \mathsf{true}}}{\dfrac{\Delta, \phi \ \mathsf{false}\ \Gamma \vdash \#}{\Delta\ \Gamma \vdash \phi \ \mathsf{true}}}$$

The other elimination forms are derivable similarly, in each case relying on indirect proof to construct a proof of the truth of a proposition from a derivation of a contradiction from the assumption of its falsity.

The derivations of the elimination forms of constructive logic are most easily exhibited using proof and refutation expressions, as follows:

$$\begin{aligned}
\texttt{abort}(p) &= \texttt{ccr}(u.(\texttt{abort} \mathbin{\#} p)) \\
p \cdot \texttt{l} &= \texttt{ccr}(u.(\texttt{fst}\,; u \mathbin{\#} p)) \\
p \cdot \texttt{r} &= \texttt{ccr}(u.(\texttt{snd}\,; u \mathbin{\#} p)) \\
p_1(p_2) &= \texttt{ccr}(u.(\texttt{ap}(p_2)\,; u \mathbin{\#} p_1)) \\
\texttt{case}\ p_1\,\{\texttt{l} \cdot x \hookrightarrow p_2 \mid \texttt{r} \cdot y \hookrightarrow p\} &= \texttt{ccr}(u.(\texttt{case}(\texttt{ccp}(x.(u \mathbin{\#} p_2)); \texttt{ccp}(y.(u \mathbin{\#} p))) \mathbin{\#} p_1))
\end{aligned}$$

The expected elimination rules are valid for these definitions. For example, the rule

$$\frac{\Delta\ \Gamma \vdash p_1 : \phi \supset \psi \quad \Delta\ \Gamma \vdash p_2 : \phi}{\Delta\ \Gamma \vdash p_1(p_2) : \psi} \tag{13.3}$$

is derivable using the definition of $p_1(p_2)$ given above. By suppressing proof terms, we may derive the corresponding provability rule

$$\frac{\Delta\ \Gamma \vdash \phi \supset \psi \ \mathsf{true} \quad \Delta\ \Gamma \vdash \phi \ \mathsf{true}}{\Delta\ \Gamma \vdash \psi \ \mathsf{true}}\ . \tag{13.4}$$

13.3 Proof Dynamics

The dynamics of classical logic arises from the simplification of the contradiction between a proof and a refutation of a proposition. To make this explicit, we will define a transition system whose states are contradictions $k \ \# \ p$ consisting of a proof p and a refutation k of the same proposition. The steps of the computation consist of simplifications of the contradictory state based on the form of p and k.

The truth and falsity rules for the connectives play off one another in a pleasing way:

$$\mathtt{fst}\,;k \ \# \ \langle p_1, p_2\rangle \longmapsto k \ \# \ p_1 \tag{13.5a}$$

$$\mathtt{snd}\,;k \ \# \ \langle p_1, p_2\rangle \longmapsto k \ \# \ p_2 \tag{13.5b}$$

$$\mathtt{case}(k_1;k_2) \ \# \ \mathtt{l}\cdot p_1 \longmapsto k_1 \ \# \ p_1 \tag{13.5c}$$

$$\mathtt{case}(k_1;k_2) \ \# \ \mathtt{r}\cdot p_2 \longmapsto k_2 \ \# \ p_2 \tag{13.5d}$$

$$\mathtt{not}(p) \ \# \ \mathtt{not}(k) \longmapsto k \ \# \ p \tag{13.5e}$$

$$\mathtt{ap}(p_1)\,;k \ \# \ \lambda\,(x)\,p_2 \longmapsto k \ \# \ [p_1/x]p_2 \tag{13.5f}$$

The rules of indirect proof give rise to the following transitions:

$$\mathtt{ccp}(x.(k_1 \ \# \ p_1)) \ \# \ p_2 \longmapsto [p_2/x]k_1 \ \# \ [p_2/x]p_1 \tag{13.5g}$$

$$k_1 \ \# \ \mathtt{ccr}(u.(k_2 \ \# \ p_2)) \longmapsto [k_1/u]k_2 \ \# \ [k_1/u]p_2 \tag{13.5h}$$

The first of these defines the behavior of the refutation of ϕ that proceeds by contradicting the assumption that ϕ is true. Such a refutation is activated by presenting it with a proof of ϕ, which is then substituted for the assumption in the new state. Thus, "ccp" stands for "call with current proof." The second transition defines the behavior of the proof of ϕ that proceeds by contradicting the assumption that ϕ is false. Such a proof is activated by presenting it with a refutation of ϕ, which is then substituted for the assumption in the new state. Thus, "ccr" stands for "call with current refutation."

Rules (13.5g) to (13.5h) overlap in that there are two transitions for a state of the form

$$\mathtt{ccp}(x.(k_1 \ \# \ p_1)) \ \# \ \mathtt{ccr}(u.(k_2 \ \# \ p_2)),$$

one to the state $[p/x]k_1 \ \# \ [p/x]p_1$, where p is $\mathtt{ccr}(u.(k_2 \ \# \ p_2))$, and one to the state $[k/u]k_2 \ \# \ [k/u]p_2$, where k is $\mathtt{ccp}(x.(k_1 \ \# \ p_1))$. The dynamics of classical logic is non-deterministic. To avoid this one may impose a priority ordering among the two cases, preferring one transition over the other when there is a choice. Preferring the first corresponds to a "lazy" dynamics for proofs, because we pass the unevaluated proof p to the refutation on the left, which is thereby activated. Preferring the second corresponds to an "eager" dynamics for proofs, in which we pass the unevaluated refutation k to the proof, which is thereby activated.

All proofs in classical logic proceed by contradicting the assumption that it is false. In terms of the classical logic machine the initial and final states of a computation are as follows:

$$\frac{}{\mathtt{halt}_{\phi} \;\#\; p \ \mathsf{initial}} \tag{13.6a}$$

$$\frac{p \ \mathsf{canonical}}{\mathtt{halt}_{\phi} \;\#\; p \ \mathsf{final}} \tag{13.6b}$$

where p is a proof of ϕ, and $\mathtt{halt}_{\phi}$ is the assumed refutation of ϕ. The judgment p $\mathsf{canonical}$ states that p is a canonical proof, which holds of any proof other than an indirect proof. Execution consists of driving a general proof to a canonical proof, under the assumption that the theorem is false.

Theorem 13.1 (Preservation)**.** *If $k \div \phi$, $p : \phi$, and $k \;\#\; p \longmapsto k' \;\#\; p'$, then there exists ϕ' such that $k' \div \phi'$ and $p' : \phi'$.*

Proof By rule induction on the dynamics of classical logic. □

Theorem 13.2 (Progress)**.** *If $k \div \phi$ and $p : \phi$, then either $k \;\#\; p$ final or $k \;\#\; p \longmapsto k' \;\#\; p'$.*

Proof By rule induction on the statics of classical logic. □

13.4 Law of the Excluded Middle

The law of the excluded middle is derivable in classical logic:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\phi \vee \neg\phi \ \mathsf{false}, \phi \ \mathsf{true} \vdash \phi \ \mathsf{true}}}{\phi \vee \neg\phi \ \mathsf{false}, \phi \ \mathsf{true} \vdash \phi \vee \neg\phi \ \mathsf{true}} \quad \dfrac{}{\phi \vee \neg\phi \ \mathsf{false}, \phi \ \mathsf{true} \vdash \phi \vee \neg\phi \ \mathsf{false}}}{\phi \vee \neg\phi \ \mathsf{false}, \phi \ \mathsf{true} \vdash \#}}{\phi \vee \neg\phi \ \mathsf{false} \vdash \phi \ \mathsf{false}}}{\phi \vee \neg\phi \ \mathsf{false} \vdash \neg\phi \ \mathsf{true}}}{\phi \vee \neg\phi \ \mathsf{false} \vdash \phi \vee \neg\phi \ \mathsf{true}} \quad \dfrac{}{\phi \vee \neg\phi \ \mathsf{false} \vdash \phi \vee \neg\phi \ \mathsf{false}}}{\phi \vee \neg\phi \ \mathsf{false} \vdash \#}}{\phi \vee \neg\phi \ \mathsf{true}}$$

When written out using explicit proofs and refutations, we obtain the proof term $p_0 : \phi \vee \neg\phi$:

$$\texttt{ccr}(u.(u \# \texttt{r} \cdot \texttt{not}(\texttt{ccp}(x.(u \# \texttt{l} \cdot x))))).$$

To understand the computational meaning of this proof, let us juxtapose it with a refutation $k \div \phi \vee \neg\phi$ and simplify it using the dynamics given in Section 13.3. The first step is the transition

$$k \# \texttt{ccr}(u.(u \# \texttt{r} \cdot \texttt{not}(\texttt{ccp}(x.(u \# \texttt{l} \cdot x)))))$$
$$\longmapsto$$
$$k \# \texttt{r} \cdot \texttt{not}(\texttt{ccp}(x.(k \# \texttt{l} \cdot x))),$$

wherein we have replicated k so that it occurs in two places in the result state. By virtue of its type, the refutation k must have the form $\texttt{case}(k_1; k_2)$, where $k_1 \div \phi$ and $k_2 \div \neg\phi$. Continuing the reduction, we obtain:

$$\texttt{case}(k_1; k_2) \# \texttt{r} \cdot \texttt{not}(\texttt{ccp}(x.(\texttt{case}(k_1; k_2) \# \texttt{l} \cdot x)))$$
$$\longmapsto$$
$$k_2 \# \texttt{not}(\texttt{ccp}(x.(\texttt{case}(k_1; k_2) \# \texttt{l} \cdot x))).$$

By virtue of its type k_2 must have the form $\texttt{not}(p_2)$, where $p_2 : \phi$, and hence the transition proceeds as follows:

$$\texttt{not}(p_2) \# \texttt{not}(\texttt{ccp}(x.(\texttt{case}(k_1; k_2) \# \texttt{l} \cdot x)))$$
$$\longmapsto$$
$$\texttt{ccp}(x.(\texttt{case}(k_1; k_2) \# \texttt{l} \cdot x)) \# p_2.$$

Observe that p_2 is a valid proof of ϕ. Proceeding, we obtain

$$\texttt{ccp}(x.(\texttt{case}(k_1; k_2) \# \texttt{l} \cdot x)) \# p_2$$
$$\longmapsto$$
$$\texttt{case}(k_1; k_2) \# \texttt{l} \cdot p_2$$
$$\longmapsto$$
$$k_1 \# p_2$$

The first of these two steps is the crux of the matter: the refutation, $k = \texttt{case}(k_1; k_2)$, which was replicated at the outset of the derivation, is re-used, but with a different argument. At the first use, the refutation k which is provided by the context of use of the law of the excluded middle, is presented with a proof $\texttt{r} \cdot p_1$ of $\phi \vee \neg\phi$. That is, the proof behaves as though the right disjunct of the law is true, which is to say that ϕ is false. If the context is such that it inspects this proof, it can only be by providing the proof p_2 of ϕ that refutes the claim that ϕ is false. Should this occur, the proof of the law of the excluded middle "backtracks" the context, providing instead the proof $\texttt{l} \cdot p_2$ to k, which then passes p_2 to k_1 without further incident. The proof of the law of the excluded middle boldly asserts

$\neg\phi$ true, regardless of the form of ϕ. Then, if caught in its lie by the context providing a proof of ϕ, it "changes its mind" and asserts ϕ to the original context k after all. No further reversion is possible, because the context has itself provided a proof p_2 of ϕ.

The law of the excluded middle illustrates that classical proofs are interactions between proofs and refutations, which is to say interactions between a proof and the context in which it is used. In programming terms, this corresponds to an abstract machine with an explicit control stack, or continuation, representing the context of evaluation of an expression. That expression may access the context (stack, continuation) to backtrack so as to maintain the perfect symmetry between truth and falsity. The penalty is that a closed proof of a disjunction no longer need show which disjunct it proves, for as we have just seen, it may, on further inspection, "change its mind."

13.5 The Double-Negation Translation

One consequence of the greater expressiveness of constructive logic is that classical proofs may be translated systematically into constructive proofs of a classically equivalent proposition. Therefore, by systematically reorganizing the classical proof, we may, without changing its meaning from a classical perspective, turn it into a constructive proof of a constructively weaker proposition. Consequently, there is no loss in adhering to constructive proofs, because every classical proof is a constructive proof of a constructively weaker, but classically equivalent, proposition. Moreover, it proves that classical logic is weaker (less expressive) than constructive logic, contrary to a naïve interpretation which would say that the added reasoning principles, such as the law of the excluded middle, afforded by classical logic makes it stronger. In programming language terms adding a "feature" does not necessarily strengthen (improve the expressive power) of your language; on the contrary, it may weaken it.

We will define a translation ϕ^* of propositions that interprets classical into constructive logic according to the following correspondences:

Classical	*Constructive*	
$\Delta\ \Gamma \vdash \phi\ \mathsf{true}$	$\neg\Delta^*\ \Gamma^* \vdash \neg\neg\phi^*\ \mathsf{true}$	truth
$\Delta\ \Gamma \vdash \phi\ \mathsf{false}$	$\neg\Delta^*\ \Gamma^* \vdash \neg\phi^*\ \mathsf{true}$	falsity
$\Delta\ \Gamma \vdash \#$	$\neg\Delta^*\ \Gamma^* \vdash \bot\ \mathsf{true}$	contradiction

Classical truth is weakened to constructive irrefutability, but classical falsehood is constructive refutability, and classical contradiction is constructive falsehood. Falsity assumptions are negated after translation to express their falsehood; truth assumptions are merely translated as is. Because the double negations are classically cancelable, the translation will be easily seen to yield a classically equivalent proposition. But because $\neg\neg\phi$ is constructively weaker than ϕ, we also see that a proof in classical logic is translated to a constructive proof of a weaker statement.

There are many choices for the translation; here is one that makes the proof of the correspondence between classical and constructive logic especially simple:

$$\begin{aligned} \top^* &= \top \\ (\phi_1 \wedge \phi_2)^* &= \phi_1^* \wedge \phi_2^* \\ \bot^* &= \bot \\ (\phi_1 \vee \phi_2)^* &= \phi_1^* \vee \phi_2^* \\ (\phi_1 \supset \phi_2)^* &= \phi_1^* \supset \neg\neg\phi_2^* \\ (\neg\phi)^* &= \neg\phi^* \end{aligned}$$

One may show by induction on the rules of classical logic that the correspondences summarized above hold, using constructively valid entailments such as

$$\neg\neg\phi \ \mathsf{true} \ \neg\neg\psi \ \mathsf{true} \vdash \neg\neg(\phi \wedge \psi) \ \mathsf{true}.$$

13.6 Notes

The computational interpretation of classical logic was first explored by Griffin (1990) and Murthy (1991). The present account is influenced by Wadler (2003), transposed by Nanevski from sequent calculus to natural deduction using multiple forms of judgment. The terminology is inspired by Lakatos (1976), an insightful and inspiring analysis of the discovery of proofs and refutations of conjectures in mathematics. Versions of the double-negation translation were originally given by Gödel and Gentzen. The computational content of the double-negation translation was first elucidated by Murthy (1991), who established the important relationship with continuation passing.

Exercises

13.1. If the continuation type expresses negation, the types shown to be inhabited in Exercise **30.2**, when interpreted under the proposition-as-types interpretation, look suspiciously like the following propositions:
(a) $\phi \vee \neg\phi$.
(b) $(\neg\neg\phi) \supset \phi$.
(c) $(\neg\phi_2 \supset \neg\phi_1) \supset (\phi_1 \supset \phi_2)$.
(d) $\neg(\phi_1 \vee \phi_2) \supset (\neg\phi_1 \wedge \neg\phi_2)$.
None of these propositions is true, in general, in constructive logic. Show that each of these propositions is true in classical logic by exhibiting a proof term for each. (The first one is done for you in Section 13.4; you need only do the other three.) Compare the proof term you get for each with the inhabitant of the corresponding type that you gave in your solution to Exercise **30.2**.

13.2. Complete the proof of the double-negation interpretation sketched in Section 13.5, providing explicit proof terms for clarity. Because $(\phi \vee \neg\phi)^* = \phi^* \vee \neg\phi^*$, the double-negation translation applied to the proof of LEM (for ϕ) given in Section 13.4 yields a proof of the double negation of LEM (for ϕ^*) in constructive logic. How does the translated proof compare to the one you derived by hand in Exercise **12.1**?