Automated Deduction in Matching Logic

FSL group

April 10, 2017

Recent success in building very fast automated theorem provers (especially for first-order theories) makes us look forward to an highly efficient automated deductive system for matching logic. This project aims at that.

1 Grammar

As in most logic, formulas of matching logic, called *patterns*, are written in a formal language, denoted as $\mathcal{L}$, who has a very similar grammar as first order logic.

The language $\mathcal{L}$ in general is a many-sorted language. A signature of $\mathcal{L}$ contains not only a finite set Σ of symbols, but also a finite nonempyty set S of sorts. Each symbol $\sigma \in \Sigma$ is, of course, sorted, with a fixed nonempty arity. We write $\sigma \in \Sigma_{s_1,\dots,s_n,s}$ when we want to emphasize that σ takes n arguments (with suggested sorts) and returns a pattern in sort s, but we hope in most cases sorting is clear from context.

The basic grammar for $\mathcal{L}$, as defined below, is almost identical to first-order logic, except that in $\mathcal{L}$ there is no difference between relational (predicate) and functional symbols, and we accept first-order terms as patterns in matching logic.

$$\begin{aligned} P ::= {} & x \\ & \mid P_1 \to P_2 \\ & \mid \neg P \\ & \mid \forall x.P \\ & \mid \sigma(P_1, \dots, P_n). \end{aligned}$$

For simplicity, we did not mention sorting in the grammar definition, and assume it should be clear to all readers. For example, in $P_1 \to P_2$, both patterns P_1 and P_2 should have the same sort, and that sort is the sort of $P_1 \to P_2$. The sort of $\forall x.P$ is the sort of P, where the sort of variable x does not matter. To see why it is the case, consider the pattern $\exists x.list(x, 1 \cdot 3 \cdot 5)$, which is the set of all memory configurations that has a list $(1, 3, 5)$ in it.

Propositional connectives are always assumed, including conjunction ($\wedge$), disjunction ($\vee$), and equivalence ($\leftrightarrow$). Existential quantifier ($\exists x$) is defined by universal quantifier ($\forall x$) in the normal way. Bottom ($\bot_s$) and top ($\top_s$) in sort s are given by $x \wedge \neg x$ and $\neg\bot_s$, respectively, where x is a variable in sort s. It does not matter which variable we pick.

1.1 Extension

The formal language is often extended with definedness symbols. A definedness symbol $\lceil \_ \rceil_{s_1}^{s_2} \in \Sigma_{s_1,s_2}$ is a *predicate* symbol, i.e., $\lceil P \rceil_{s_1}^{s_2}$ is either $\top$ or $\bot$ for any pattern P. We often write $\lceil P \rceil$ when sorting information can be derived from the context.

Definedness extends the formal language:

$$\begin{aligned}
&\lfloor P \rfloor \coloneqq \neg \lceil \neg P \rceil \\
&P_1 = P_2 \coloneqq \lfloor P_1 \leftrightarrow P_2 \rfloor \\
&P_1 \neq P_2 \coloneqq \neg (P_1 = P_2) \\
&P_1 \subseteq P_2 \coloneqq \lfloor P_1 \rightarrow P_2 \rfloor \\
&x \in P \coloneqq x \subseteq P
\end{aligned}$$

and provides convenient ways to write *predicate patterns*, patterns that is either $\top$ or $\bot$. The sort of a predicate patterns is often of no importance. It is its "truth value" that is important. For example, $\lceil P \rceil$ is "true" if P is not empty, and this fact can be used in any context. This leads us to polymorphic sorting in matching logic.

1.2 Polymorphism

Predicate symbols are often polymorphic sorting. As a result, many predicate connectives are of polymorphic types, too. Predicate pattern $P_1 = P_2$ only requires P_1 and P_2 have the same sort, and its result can be of any sort. If P_1 and P_2 have different sorts, $P_1 = P_2$ is *ill-sorted*.

Polymorphism means in mathematics a mapping τ from pattern set $\mathcal{P}$ to a sort algebra $\mathcal{S}$ built from S. The algebra $\mathcal{S}$ is obtained by adding two special elements, called the top-sort $s_\top$ and bottom-sort $s_\bot$, to the set S, and define a partial order relation by which $s_\top$ is the largest element in $\mathcal{S}$, $s_\bot$ is the smallest, and every other elements in S is in between, incomparable to each other. $\mathcal{S}$ is a lattice.

Let P be a pattern. Intuitively, if $\tau(P) = s_\bot$ then P is ill-sorted. Most patterns have their normal sorts in S. Predicate patterns have sort $s_\top$, which makes them fit in any context. However, nothing prevents one having a non-predicate pattern of sort $s_\top$. All that matters is the *sorting signature* of a symbol $\sigma \in \Sigma$.

Normal symbols have regular sorting signatures. A symbol $\sigma \in \Sigma_{s_1,\ldots,s_n,s}$ expects n arguments of the sorts $s_1, \ldots, s_n$ respectively, and results in a pattern of sort s.

The same notation can be used to polymorphic symbols, most of which are predicates. For example, $\texttt{primeQ}\colon \texttt{Nat} \Rightarrow s_\top$ checks whether its argument is a prime number or not, and returns either $\top$ (if the argument is a prime number) or $\bot$ (otherwise). A two-arity example is $\texttt{ge}(x, y)$ that checks whether $x > y$, where $\texttt{ge}\colon \texttt{Nat} \times \texttt{Nat} \Rightarrow s_\top$. Predicates can appear in any context, for instance, $\forall x.\exists y.(\texttt{ge}(y, x) \wedge \texttt{primeQ}(y))$.

Symbols can take arguments of the top sort $s_\top$, too. One example is the definedness symbol $\lceil \_ \rceil\colon s_\top \Rightarrow s_\top$, a predicate over *all sorts* of patterns. It sounds more "polymorphic" than `ge` and `primeQ`, in the sense that not only it can be plugged in any context, but also any patterns can be plugged in it. The above discussion leads us to the following definition of polymorphism.

Definition 1 (Polymorphic sorting)**.** *Polymorphic sorts of patterns is a mapping* $\tau\colon \mathcal{P} \to \mathcal{S}$*, defined recursively by*

$$\begin{aligned}
&\tau(x) = s \quad \textit{if } x \textit{ is a variable of sort } s\\
&\tau(\neg P) = \tau(P)\\
&\tau(P_1 \to P_2) = \min(\tau(P_1), \tau(P_2))\\
&\tau(\exists x.P) = \tau(P).
\end{aligned}$$

Let $\sigma \in \Sigma_{s_1,\ldots,s_n,s}$ *be a polymorphic symbol, then*

$$\tau(\sigma(P_1, \ldots, P_n)) = \begin{cases} s & \textit{if } \min(\tau(P_i), s_i) \neq s_\perp \textit{ for all } i\\ s_\perp & \textit{otherwise}\end{cases}$$

The above definition, though, does not provide a connection between poly-sorted matching logic with many-sorted matching logic. On one hand, we often write patterns such as $(x \mapsto 1 * y \mapsto 2) \wedge x + 1 = y$, which are in fact poly-sorted (in the equality symbol "="). On the other hand, such polymorphism is regarded as nothing more than just syntactic sugar. The semantics and the inference system of matching logic are all for many-sorted matching logic. And we do not want to define a duplicate just for the poly-sorted case.

Therefore, we provide here a viewpoint to regard poly-sorted patterns *as* many-sorted patterns. By doing so, we treat poly-sorted patterns *as* nothing but just syntactic sugar for many-sorted patterns. This viewpoint is much useful in practice than Definition 1, and may eventually replace it.

Definition 2. *(Poly-sorted symbols and patterns) Poly-sorted symbols are symbols whose signatures have sort* $s_\top$*. A regular symbol is said to* instantiates *a poly-sorted symbol if it instantiates all* $s_\top$ *to some regular sorts. Poly-sorted patterns are built from many-sorted variables, logical connectives and quantifiers, and symbol applications, the same as many-sorted patterns, with the only exception that symbols can be polymorphic. A poly-sorted pattern P is a set of many-sorted patterns by instantiating all poly-sorted symbols in P in all possible ways such that the instances are well-sorted.*

1.3 ml2fol-Translation

Let us assume a matching logic signature (S, Σ) with some symbols in Σ having polymorphic types, such as the definedness symbol $\lceil \_ \rceil\colon s_\top \rightsquigarrow s_\top$ that we have seen. We define the first-order logic theory $T = \mathsf{ml2pl}(S, \Sigma)$ as follows.

2 Deductive system

Axioms in $\mathcal{L}$ are given by the following axiom schemata where P, Q, R are arbitrary patterns and x, y are variables.

- (K1) $P \to (Q \to P)$

- (K2) $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$
- (K3) $(\neg P \to \neg Q) \to (Q \to P)$
- (K4) $\forall x.(P \to Q) \to (P \to \forall x.Q)$ if x does not occur free in P
- (K5) $\exists y.x = y$
- (K6) $\exists y.Q = y \to (\forall x.P(x) \to P[Q/x])$ if Q is free for x in P
- (K7) $P_1 = P_2 \to (Q[P_1/x] \to Q[P_2/x])$
- (M1) $x \in y = (x = y)$
- (M2) $x \in \neg P = \neg(x \in P)$
- (M3) $x \in P \wedge Q = (x \in P) \wedge (x \in Q)$
- (M4) $x \in \exists y.P = \exists y.x \in P$ where x is distinct from y
- (M5) $x \in \sigma(\dots, P_i, \dots) = \exists y.y \in P_i \wedge x \in \sigma(\dots, y, \dots)$

Inference rules include

- (Modus Ponens) From P and $P \to Q$, deduce Q.
- (Universal Generalization) From P, deduce $\forall x.P$.
- (Membership Introduction) From P, deduce $\forall x.(x \in P)$, where x does not occur free in P.
- (Membership Elimination) From $\forall x.(x \in P)$, deduce P, where x does not occur free in P.

Theorem 3 (Soundness of $K_{\mathcal{L}}$)**.** *Theorems of $K_{\mathcal{L}}$ are valid.*

Proof. Trivial. □

We provide some metatheorems of $K_{\mathcal{L}}$.

Proposition 4 (Tautology)**.** *For any propositional tautology $\mathcal{A}(p_1, \dots, p_n)$ where $p_1, \dots, p_n$ are propositional variables,*

$$\vdash \mathcal{A}(P_1, \dots, P_n).$$

Proof. Omit proof here. □

Remark Proposition 4 makes any metatheorem of propositional logic a metatheorem of $K_{\mathcal{L}}$.

Proposition 5 (Variable Substitution)**.** $\vdash \forall x.P \to P[y/x]$.

Proposition 6 (Functional Substitution)**.** $\vdash \exists y.(Q = y) \to (P[Q/x] \to \exists x.P(x))$.

Proposition 7 ($\vee$-Introduction)**.** $\vdash P$ *implies* $\vdash P \vee Q$.

Proof. Use Proposition 4 and Modus Ponens. Note that in general, $\vdash P \vee Q$ does not imply $\vdash P$ or $\vdash Q$. □

Proposition 8 ($\wedge$-Introduction and Elimination)**.** $\vdash P$ *and* $\vdash Q$ *iff* $\vdash P \wedge Q$.

Proof. Use Proposition 4 and Modus Ponens. □

Proposition 9 (Equality Introduction)**.** $\vdash P = P$.

Proof. Use Membership Introduction and Proposition 4. □

Proposition 10 (Equality Replacement)**.** $\vdash P_1 = P_2$ *and* $\vdash Q[P_1/x]$ *implies* $\vdash Q[P_2/x]$.

Proof. Use Axiom (K7) and Modus Ponens. □

Proposition 11 (Equality Establishment)**.** $\vdash P \leftrightarrow Q$ *implies* $\vdash P = Q$.

Proof. Use Membership Axoims and $\vee$-Introduction. □

Corollary 12. $\vdash P$ *implies* $\vdash P = \top$.

Proposition 13. $\vdash x \in \lceil y \rceil$.

Proof.

$$\begin{aligned} & \vdash x \in \lceil y \rceil \\ \text{if } & \vdash \forall x.(x \in \lceil y \rceil) && (K5,\ K6,\text{ and Modus Ponens}) \\ \text{iff } & \vdash \lceil y \rceil. \end{aligned}$$

□

Proposition 14. $\vdash P \to \lceil P \rceil$.

Proof.

$$
\begin{aligned}
&\vdash P \to \lceil P \rceil\\
\text{iff } &\vdash \forall x.(x \in P \to \lceil P \rceil)\\
\text{if } &\vdash x \in P \to \lceil P \rceil\\
\text{iff } &\vdash x \in P \to x \in \lceil P \rceil\\
\text{iff } &\vdash x \in P \to \exists y.(y \in P \wedge x \in \lceil y \rceil)\\
\text{iff } &\vdash x \in P \to \neg\forall y.(y \notin P \vee x \notin \lceil y \rceil)\\
\text{iff } &\vdash \forall y.(y \notin P \vee x \notin \lceil y \rceil) \to x \notin P\\
\text{if } &\vdash x \notin P \vee x \notin \lceil x \rceil \to x \notin P\\
\text{iff } &\vdash x \in P \to x \in P \wedge x \in \lceil x \rceil\\
\text{iff } &\vdash x \in P \to x \in \lceil x \rceil\\
\text{if } &\vdash x \in \lceil x \rceil
\end{aligned}
$$

Remark Similarly we can show $\vdash \lfloor P \rfloor \to P$. □

Proposition 15. $\vdash \forall x.(x \in P) = \lfloor P \rfloor$, *where x occurs free in P.*

Proof. By Proposition 11 and 8, it suffices to show

$$\vdash \forall x.(x \in P) \to \lfloor P \rfloor \tag{1}$$

and

$$\vdash \lfloor P \rfloor \to \forall x.(x \in P). \tag{2}$$

To show (1),

$$
\begin{aligned}
&\vdash \forall x.(x \in P) \to \lfloor P \rfloor\\
\text{iff } &\vdash \forall x.\lceil x \wedge P \rceil \to \neg\lceil \neg P \rceil\\
\text{iff } &\vdash \lceil \neg P \rceil \to \exists x.\neg\lceil x \wedge P \rceil\\
\text{iff } &\vdash \forall y.(y \in (\lceil \neg P \rceil \to \exists x.\neg\lceil x \wedge P \rceil))\\
\text{if } &\vdash y \in (\lceil \neg P \rceil \to \exists x.\neg\lceil x \wedge P \rceil\\
\text{iff } &\vdash \exists z_1.(z_1 \notin P \wedge y \in \lceil z_1 \rceil) \to\\
&\qquad \exists x.\neg(\exists z_2.(z_2 = x \wedge z_2 \in P \wedge y \in \lceil z_2 \rceil))\\
\text{iff } &\vdash \exists z_1.(z_1 \notin P \wedge \top) \to \qquad\qquad \text{(Proposition 13, 10, and Corollary 12)}\\
&\qquad \exists x.\neg(\exists z_2.(z_2 = x \wedge z_2 \in P \wedge \top))\\
\text{iff } &\vdash \exists z_1.(z_1 \notin P) \to \exists x.\neg(\exists z_2.(z_2 = x \wedge z_2 \in P))\\
\text{iff } &\vdash \forall x.(\exists z_2.(z_2 = x \wedge z_2 \in P)) \to \forall z_1.(z_1 \in P)\\
\text{if } &\vdash \forall z_1.(\forall x.(\exists z_2.(z_2 = x \wedge z_2 \in P)) \to (z_1 \in P))\\
\text{if } &\vdash \forall x.(\exists z_2.(z_2 = x \wedge z_2 \in P)) \to (z_1 \in P).
\end{aligned}
$$

Since $\vdash \forall x.(\exists z_2.(z_2 = x \wedge z_2 \in P)) \to \exists z_2.(z_2 = z_1 \wedge z_2 \in P)$, it suffices to show

$$\begin{aligned}
&\vdash \exists z_2.(z_2 = z_1 \wedge z_2 \in P) \to (z_1 \in P) \\
\text{iff } &\vdash z_1 \notin P \to \forall z_2.(z_2 \neq z_1 \vee z_2 \notin P) \\
\text{if } &\vdash \forall z_2.(z_1 \notin P \to z_2 \neq z_1 \vee z_2 \notin P) \\
\text{if } &\vdash z_1 \notin P \to z_2 \neq z_1 \vee z_2 \notin P \\
\text{if } &\vdash z_2 = z_1 \wedge z_2 \in P \to z_1 \in P.
\end{aligned}$$

And we proved (1).

Similarly, to show (2),

$$\begin{aligned}
&\vdash \lfloor P \rfloor \to \forall x.(x \in P) \\
\text{iff } &\vdash \exists x.\neg\lceil x \wedge P \rceil \to \lceil \neg P \rceil \\
\text{iff } &\vdash \forall y.(y \in \exists x.\neg\lceil x \wedge P \rceil \to \lceil \neg P \rceil) \\
\text{if } &\vdash y \in \exists x.\neg\lceil x \wedge P \rceil \to \lceil \neg P \rceil \\
\text{iff } &\vdash \exists x.\neg\exists z_2.(z_2 = x \wedge z_2 \in P) \to \exists z_1.(z_1 \notin P) \\
\text{iff } &\vdash \forall z_1.(z_1 \in P) \to \exists z_2.(z_2 = x \wedge z_2 \in P) \\
\text{iff } &\vdash x \in P \to \exists z_2.(z_2 = x \wedge z_2 \in P).
\end{aligned}$$

We proved (2).

Remark If x occurs free in P, the result does not hold. For example, let P be $upto(x)$ where $upto(\cdot)$ is interpreted to $upto(n) = \{0, 1, \ldots, n\}$ on $\mathbb{N}$. □

Remark From Membership Introduction and Elimination inference rules and Proposition 15, $\vdash P$ iff $\vdash \lfloor P \rfloor$.

Proposition 16 (Classification Reasoning)**.** *For any P and Q, from $\vdash P \to Q$ and $\vdash \neg P \to Q$ deduce $\vdash Q$.*

Proof. From $\vdash \neg P \to Q$ deduce $\vdash \neg Q \to P$. Notice that $\vdash P \to Q$, so we have $\vdash \neg Q \to Q$, i.e., $\vdash \neg\neg Q \vee Q$ which concludes the proof. □

Corollary 17. *For any P_1, P_2, and Q are patterns with $\vdash P_1 \vee P_2$, from $\vdash P_1 \to Q$ and $\vdash P_2 \to Q$, deduce $\vdash Q$.*

Definition 18 (Predicate Pattern)**.** *A pattern P is called a predicate pattern or a predicate if $\vdash (P = \top) \vee (P = \bot)$.*

Remark Predicate patterns are closed under all logic connectives.

Remark For any P, $\lceil P \rceil$ is a predicate pattern.

Proposition 19. $\vdash (\lceil P \rceil = \bot) = (P = \bot)$ *and* $\vdash (\lfloor P \rfloor = \top) = (P = \top)$.

Proof. It is easy to prove one derivation from the other, so we only prove the first one. By Proposition 11, it suffices to prove

$$\vdash (\lceil P \rceil = \bot) \to (P = \bot) \tag{3}$$

and

$$\vdash (P = \bot) \to (\lceil P \rceil = \bot) \tag{4}$$

The proof of (4) is trivial and we left it as an exercise. We now prove (3) through the following backward reasoning.

$$\begin{array}{rl} & \vdash (\lceil P \rceil = \bot) \to (P = \bot) \\ \text{iff} & \vdash \forall y.(y \in ((\lceil P \rceil = \bot) \to (P = \bot))) \\ \text{if} & \vdash y \in ((\lceil P \rceil = \bot) \to (P = \bot)) \\ \text{iff} & \vdash (y \in (\lceil P \rceil = \bot) \to (y \in (P = \bot)). \end{array} \tag{5}$$

While for any pattern Q,

$$\begin{array}{rl} & \vdash y \in (Q = \bot) \\ \text{iff} & \vdash y \in \neg\lceil\neg(Q \leftrightarrow \bot)\rceil \\ \text{iff} & \vdash y \in \neg\lceil Q \rceil \\ \text{iff} & \vdash \neg\exists z.(z \in Q \wedge y \in \lceil z \rceil) \\ \text{iff} & \vdash \neg\exists z.(z \in Q) \end{array}$$

So we continue to prove (5) by showing

$$\begin{array}{rl} & \vdash (y \in (\lceil P \rceil = \bot)) \to (y \in (P = \bot)) \\ \text{iff} & \vdash \neg\exists z.(z \in \lceil P \rceil) \to \neg\exists z.(z \in P) \\ \text{iff} & \vdash \exists z.(z \in P) \to \exists z.(z \in \lceil P \rceil) \\ \text{iff} & \vdash \exists z.(z \in P) \to \exists z.(\exists z_1.(z_1 \in P \wedge z \in \lceil z_1 \rceil)) \\ \text{iff} & \vdash \exists z.(z \in P) \to \exists z.\exists z_1.(z_1 \in P) \\ \text{iff} & \vdash \exists z_1.(z_1 \in P) \to \exists z.\exists z_1.(z_1 \in P). \end{array}$$

And we finish the proof by noticing the fact that for any pattern Q and variable x,

$$\vdash Q \to \exists x.Q.$$

□

Proposition 20. *For any predicate P, $\vdash (P \neq \top) = (P = \bot)$ and $\vdash (P \neq \bot) = (P = \top)$.*

Proof. We only prove the first derivation, by showing both

$$\vdash (P \neq \top) \to (P = \bot) \tag{6}$$

and

$$\vdash (P = \bot) \to (P \neq \top). \tag{7}$$

Proving (7) is trivial. We now prove (6), which is also trivial by transforming disjunction to implication. □

Proposition 21. *For any pattern Q and any predicate pattern P, $\vdash P \vee Q$ iff $\vdash P \vee \lfloor Q \rfloor$.*

Proof. ($\Leftarrow$) is obtained immediately by the remark of Proposition 14. We now prove ($\Rightarrow$).

Because $\vdash Q = \top \vee Q \neq \top$, it suffices to show

$$\vdash Q = \top \rightarrow (P \vee \lfloor Q \rfloor = \top) \tag{8}$$

and

$$\vdash Q \neq \top \rightarrow (P \vee \lfloor Q \rfloor = \top) \tag{9}$$

by Corollary 17, and the fact that $\vdash P \vee \lfloor Q \rfloor = \top$ and $\vdash \top$ imply $\vdash P \vee \lfloor Q \rfloor$.

The proof of (8) is straightforward as follows.

$$\begin{array}{rl} & \vdash Q = \top \rightarrow (P \vee \lfloor Q \rfloor = \top) \\ \text{if} & \vdash Q = \top \rightarrow (P \vee \lfloor \top \rfloor = \top) \\ \text{if} & \vdash Q = \top \rightarrow (\top = \top) \\ \text{if} & \vdash \top. \end{array}$$

The proof of (9) needs more effort:

$$\begin{array}{rl} & \vdash Q \neq \top \rightarrow (P \vee \lfloor Q \rfloor = \top) \\ \text{iff} & \vdash (Q = \top) \vee (P \vee \lfloor Q \rfloor = \top) \\ \text{iff} & \vdash (\lfloor Q \rfloor = \top) \vee (P \vee \lfloor Q \rfloor = \top) \\ \text{iff} & \vdash \lfloor Q \rfloor \neq \top \rightarrow (P \vee \lfloor Q \rfloor = \top) \\ \text{iff} & \vdash \lfloor Q \rfloor = \bot \rightarrow (P \vee \lfloor Q \rfloor = \top) \\ \text{if} & \vdash \lfloor Q \rfloor = \bot \rightarrow (P \vee \bot = \top) \\ \text{iff} & \vdash \lfloor Q \rfloor = \bot \rightarrow (P = \top) \\ \text{if} & \vdash Q = \top \vee P = \top. \end{array}$$

Notice that P is a predicate pattern, so it suffices to show

$$\vdash P = \top \rightarrow (Q = \top \vee P = \top),$$

whose validity is obvious, and

$$\vdash P = \bot \rightarrow (Q = \top \vee P = \top),$$

which is proved by showing

$$\vdash P = \bot \rightarrow Q = \top. \tag{10}$$

Because $\vdash P \vee Q$, it suffices to show

$$\begin{array}{rl} & \vdash P = \bot \rightarrow (P \vee Q) \rightarrow (Q = \top) \\ \text{if} & \vdash P = \bot \rightarrow (\bot \vee Q) \rightarrow (Q = \top) \\ \text{iff} & \vdash P = \bot \rightarrow Q \rightarrow (Q = \top) \\ \text{if} & \vdash Q \rightarrow (Q = \top) \\ \text{iff} & \vdash (Q \neq \top) \rightarrow \neg Q \\ \text{iff} & \vdash (\lfloor Q \rfloor = \bot) \rightarrow \neg Q. \end{array}$$

Notice we have $\vdash Q \to \lfloor Q \rfloor$, which means $\vdash \neg\lfloor Q \rfloor \to \neg Q$, so it suffices to show

$$\begin{array}{ll} & \vdash (\lfloor Q \rfloor = \bot) \to \neg\lfloor Q \rfloor \\ \text{iff} & \vdash (\lfloor Q \rfloor = \bot) \to \neg\bot \\ \text{iff} & \vdash (\lfloor Q \rfloor = \bot) \to \top \\ \text{iff} & \vdash \top. \end{array}$$

And this concludes the proof. □

Proposition 22 (Deduction Theorem)**.** *If $\Gamma \cup \{P\} \vdash Q$ and the derivation does not use $\forall x$-Generalization where x is free in P, then $\Gamma \vdash \lfloor P \rfloor \to Q$.*

Proof. The proof is by induction on n, the length of the derivation of Q from $\Gamma \cup \{P\}$.

Base step: $n = 1$, and Q is an axiom, or P, or a member of Γ. If Q is an axiom or a member of Γ, then $\Gamma \vdash Q$ and as a result, $\Gamma \vdash \lfloor P \rfloor \to Q$. If Q is P, then $\Gamma \vdash \lfloor P \rfloor \to Q$ by Proposition 14.

Induction step: Let $n > 1$. Suppose that if P' can be deduced from $\Gamma \cup \{P\}$ without using $\forall x$-Generalization where x is free in P, in a derivation containing fewer than n steps, then $\Gamma \vdash \lfloor P \rfloor \to P'$.

Case 1: Q is an axiom, or P, or a member of Γ. Precisely as in the Base step, we show that $\vdash \lfloor P \rfloor \to Q$.

Case 2: Q follows from two previous patterns in the derivation by an application of Modus Ponens. These two patterns must have the forms Q_1 and $Q_1 \to Q$, and each one can certainly be deduced from $\Gamma \cup \{P\}$ by a derivation with fewer than n steps, by just omitting the subsequent members from the original derivation from $\Gamma \cup \{P\} \vdash Q$. So we have $\Gamma \cup \{P\} \vdash Q_1$ and $\Gamma \cup \{P\} \vdash Q_1 \to Q$, and, applying the hypothesis of induction, $\Gamma \vdash \lfloor P \rfloor \to Q_1$ and $\Gamma \vdash \lfloor P \rfloor \to (Q_1 \to Q)$. It follows immediately that $\Gamma \vdash \lfloor P \rfloor \to Q$.

Case 3: Q follows from a previous pattern in the derivation by an application of $\forall x_i$-Generalization where x_i does not occur free in P. So Q is $\forall x_i.Q_1$, say, and Q_1 appears previously in the derivation. Thus $\Gamma \cup \{P\} \vdash Q_1$, and the derivation has fewer than n steps, so $\Gamma \vdash \lfloor P \rfloor \to Q_1$, since there is no application of Universal Generalization involving a free variable of P. Also x_i cannot occur free in P, as it is involved in an application of Universal Generalization in the deduction of Q from $\Gamma \cup \{P\}$. So we have a derivation of $\Gamma \vdash \lfloor P \rfloor \to Q$ as follows.

$$\begin{array}{rl} & \Gamma \vdash \lfloor P \rfloor \to Q \\ \text{iff} & \Gamma \vdash \lfloor P \rfloor \to \forall x_i.Q_1 \\ \text{if} & \Gamma \vdash \forall x_i.(\lfloor P \rfloor \to Q_1) \\ \text{if} & \Gamma \vdash \lfloor P \rfloor \to Q_1. \end{array}$$

So $\Gamma \vdash \lfloor P \rfloor \to Q$ as required.

Case 4: Q follows from a previous pattern in the derivation by an application of Membership Introduction. So Q is $\forall x_i.(x_i \in Q_1)$ with x_i is free in Q_1, say, and Q_1 appears previously in the derivation. Thus $\Gamma \cup \{P\} \vdash Q_1$, and the derivation has fewer than n steps, so $\Gamma \vdash \lfloor P \rfloor \to Q_1$, since there is no application of Universal Generalization involving a free variable of P. So we have a derivation of $\Gamma \vdash \lfloor P \rfloor \to Q$ as follows.

$$\begin{array}{ll} & \Gamma \vdash \lfloor P \rfloor \to Q \\ \text{iff} & \Gamma \vdash \lfloor P \rfloor \to \forall x_i.(x_i \in Q_1) \\ \text{iff} & \Gamma \vdash \lfloor P \rfloor \to \lfloor Q_1 \rfloor, \end{array}$$

which follows by the hypothesis of induction $\Gamma \vdash \lfloor P \rfloor \to Q_1$ and the fact that $\Gamma \vdash Q_1 \to \lfloor Q_1 \rfloor$ (by the Remark in Proposition 14).

Case 5: Q follows from a previous pattern in the derivation by an application of Membership Elimination. The previous pattern must have the form $\forall x_i.(x_i \in Q)$, and can be deduced from $\Gamma \cup \{P\}$ by a derivation with fewer than n steps, by just omitting the subsequent members from the original derivation from $\Gamma \cup \{P\} \vdash Q$. So we have $\Gamma \cup \{P\} \vdash \forall x_i.(x_i \in Q)$, and, applying the hypothesis of induction, $\Gamma \vdash \lfloor P \rfloor \to \forall x_i.(x_i \in Q)$. So we have a derivation of $\Gamma \vdash \lfloor P \rfloor \to Q$ as follows.

$$\begin{array}{lll} & \Gamma \vdash \lfloor P \rfloor \to Q & \\ \text{iff} & \Gamma \vdash \neg\lfloor P \rfloor \vee Q & \\ \text{iff} & \Gamma \vdash \neg\lfloor P \rfloor \vee \lfloor Q \rfloor & \text{(Proposition 21)} \\ \text{iff} & \Gamma \vdash \neg\lfloor P \rfloor \vee \forall x_i.(x_i \in Q) & \\ \text{iff} & \Gamma \vdash \lfloor P \rfloor \to \forall x_i.(x_i \in Q), & \end{array}$$

which is the hypothesis of induction. And this concludes our inductive proof. □

Corollary 23 (Closed-form Deduction Theorem)**.** *If P is closed, $\Gamma \cup \{P\} \vdash Q$ implies $\Gamma \vdash \lfloor P \rfloor \to Q$.*

Theorem 24 (Frame Rule)**.** *Let $\sigma \in \Sigma$ be a symbol in the signature. From $P_1 \to P_2$, deduce $\sigma(P_1) \to \sigma(P_2)$. In its most general form, $P_1 \to P_2$ deduces $\sigma(Q_1, \ldots, P_1, \ldots, Q_n) \to \sigma(Q_1, \ldots, P_2, \ldots, Q_n)$.*

Proof. we write $\sigma(Q_1, \ldots, P_i, \ldots, Q_n)$ as $\sigma(P_i, \vec{Q})$ for short, for any $i \in \{1, 2\}$.

$$\begin{array}{ll} & \vdash \sigma(P_1, \vec{Q}) \to \sigma(P_2, \vec{Q}) \\ \text{iff} & \vdash y \in (\sigma(P_1, \vec{Q}) \to \sigma(P_2, \vec{Q})) \\ \text{iff} & \vdash (y \in \sigma(P_1, \vec{Q})) \to (y \in \sigma(P_2, \vec{Q})) \\ \text{iff} & \vdash \exists z_1.\exists \vec{z}.(z_1 \in P_1 \wedge \vec{z} \in \vec{Q} \wedge y \in \sigma(z_1, \vec{z})) \\ & \to \exists z_2.\exists \vec{z}.(z_2 \in P_2 \wedge \vec{z} \in \vec{Q} \wedge y \in \sigma(z_2, \vec{z})) \\ \text{iff} & \vdash \exists z_1.\exists \vec{z}.(z_1 \in P_1 \wedge \vec{z} \in \vec{Q} \wedge y \in \sigma(z_1, \vec{z}) \\ & \qquad \to z_1 \in P_2 \wedge \vec{z} \in \vec{Q} \wedge y \in \sigma(z_1, \vec{z})) \\ \text{iff} & \vdash \exists z_1.\exists \vec{z}.(z_1 \in P_1 \to z_1 \in P_2) \\ \text{if} & \vdash \exists z_1.(z_1 \in P_1 \to z_1 \in P_2) \\ \text{if} & \vdash P_1 \to P_2. \end{array}$$

□

Corollary 25 (Frame Rule as Implication)**.** $\vdash \lfloor P \to Q \rfloor \to (\sigma(P) \to \sigma(Q))$

3 Inference rules

Axioms

$$\frac{.}{\Gamma \vdash A}$$

where A is an axiom.

Inclusion

$$\frac{.}{\Gamma \vdash P}$$

where $P \in \Gamma$.

Modus Ponens

$$\frac{\Gamma \vdash Q \to P \quad \Gamma \vdash Q}{\Gamma \vdash P}$$

Closed-Form Deduction Theorem

$$\frac{\Gamma \cup \{P\} \vdash Q}{\Gamma \vdash P \to Q}$$

where P is closed.

Universal Generalization

$$\frac{\Gamma \vdash P}{\Gamma \vdash \forall x.P}\ (\forall x)$$

Conjunction Splitting

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}$$