

PART II QUANTIFICATIONAL LOGIC

The language \mathcal{L} of part I was built from *sentence letters*, symbols that stand in for sentences. The logical truth of a sentence or the logical validity of an argument, however, may hinge on the internal structure of the sentences; and it is the study of this structure that we now take up. Under the label *quantificational logic*, we consider logical systems that allow predication, i.e., the application of predicates to terms, and that also allow quantification, i.e., the application of quantifier phrases, such as **some** and **all**, to predicate expressions. Chapter 1 describes the syntax of a simple language of this kind and chapter 2 describes its semantics. Chapter 3 discusses various logical notions in the context of the new language. Chapter 4 presents an axiom system, PL, and illustrates its use in proofs and derivations. Chapter 5 contains a proof of PL's completeness, and chapter 6, an argument for its adequacy. Chapters 7 and 8 introduce systems of quantificational logic that can be viewed as extensions of PL. The proper treatment of some of these extensions is controversial, and we outline some of the more attractive alternatives.

Chapter 1. Syntax

We shall now set up a new language $\mathcal{L}(\pi)$ of predicate logic. Recall that \mathcal{L} is the language of classical sentential logic and has an alphabet of sentence letters $\mathbf{p}_1, \mathbf{p}_2, \dots$, truth-functional connectives \vee and \neg , and brackets (and). The alphabet of $\mathcal{L}(\pi)$ then consists of the connectives and brackets of \mathcal{L} plus the universal quantifier \forall , the (object) variables $\mathbf{v}_1, \mathbf{v}_2, \dots$, and, for every non-negative integer n , the degree- n predicate letters $\mathbf{P}^n_1, \mathbf{P}^n_2, \dots$. An atomic formula of $\mathcal{L}(\pi)$ is a degree- n predicate letter followed by n occurrences of object variables. The formulas of $\mathcal{L}(\pi)$ are determined by the following formation rules:

- (i) Each atomic formula is a formula.
- (ii) If \mathbf{A} is a formula then so is $\neg\mathbf{A}$.
- (iii) If \mathbf{A} and \mathbf{B} are formulas then so is $(\mathbf{A}\vee\mathbf{B})$.
- (iv) If \mathbf{A} is a formula and \mathbf{x} is a variable then $\forall\mathbf{x}\mathbf{A}$ is a formula.

For example \mathbf{P}^0_6 is an atomic formula (because it is a degree-0 predicate letter followed by zero occurrences of variables) and $\mathbf{P}^3_1\mathbf{v}_1\mathbf{v}_2\mathbf{v}_1$ is an atomic formula. Therefore, by (iii), $(\mathbf{P}^3_1\mathbf{v}_1\mathbf{v}_2\mathbf{v}_1 \vee \mathbf{P}^0_6)$ is a formula and, by (iv), $\forall\mathbf{v}_2(\mathbf{P}^3_1\mathbf{v}_1\mathbf{v}_2\mathbf{v}_1 \vee \mathbf{P}^0_6)$ is a formula.

We have here posited a fixed and countable supply of sentence-letters and predicate-letters of arbitrary degree. We could generalize the approach as we did in chapter 8 and consider languages with different sets of sentence- and predicate-letters (and even of the variables).

The correspondence between the new language and a natural language like English is not

as simple and direct as was the correspondence between \mathcal{L} and English. Full discussion of this correspondence is postponed until the presentation of an informal semantics in the next chapter. But if \mathbf{F} is a predicate letter and \mathbf{x} and \mathbf{y} are variables we can read \mathbf{Fxy} as "F of \mathbf{x} and \mathbf{y} " and $\forall \mathbf{x}$ as "for every object \mathbf{x} "

A formula is quantificational if it contains some occurrences of \forall and quantifier-free if it does not. A set of formulas is quantifier-free if all its members are. The formula \mathbf{A} is a truth-functional compound of the formulas $\mathbf{B}_1, \dots, \mathbf{B}_n$ if \mathbf{A} can be obtained by applying rules (ii)-(iii) to $\mathbf{B}_1, \dots, \mathbf{B}_n$, i.e., if \mathbf{A} is either one of the $\mathbf{B}_1, \dots, \mathbf{B}_n$ or the result of successively forming disjunctions and negations of $\mathbf{B}_1, \dots, \mathbf{B}_n$. For example $\neg(\mathbf{P}^3_1 \mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_1 \vee \forall \mathbf{v}_1 \mathbf{P}^0_6)$ is a truth functional compound of $\mathbf{P}^3_1 \mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_1$ and $\forall \mathbf{v}_1 \mathbf{P}^0_6$ but $\forall \mathbf{v}_1 \neg(\mathbf{P}^3_1 \mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_1 \vee \forall \mathbf{v}_1 \mathbf{P}^0_6)$ is a truth-functional compound of no formula but itself. The formula \mathbf{A} is a universal formula if it is of the form $\forall \mathbf{x} \mathbf{B}$ for some variable \mathbf{x} and formula \mathbf{B} . Predicate letters of degree-one are sometimes called monadic.

Notice that $\mathcal{L}(\pi)$ is a truth-functional language in the sense of chapter I.8. Its constituents are the atomic formulas and the universal formulas. All of the previous conventions concerning parentheses, abbreviations, naming and metalinguistic variables remain in force. In addition we shall use the boldface letters ' \mathbf{x} ', ' \mathbf{y} ' and ' \mathbf{z} ', with or without subscripts and primes, as variables whose range is the object variables of $\mathcal{L}(\pi)$, ' \mathbf{F} ', ' \mathbf{G} ' and ' \mathbf{H} ', with or without subscripts, as variables whose range is the predicate letters (of any degree) of $\mathcal{L}(\pi)$. **Formula** will now mean formula of $\mathcal{L}(\pi)$ unless otherwise stated. We add the following clauses to the definition of direct abbreviation:

- PLi) $\mathbf{P}^n > \mathbf{P}^n_1$ for all non-negative n ,
- PLii) $\mathbf{Q}^n > \mathbf{P}^n_2$ for all non-negative integers n ,
- PLiii) $\mathbf{R}^n > \mathbf{P}^n_3$ for all non-negative integers n ,
- PLiv) $\mathbf{v} > \mathbf{v}_1$,
- PLv) $\mathbf{u} > \mathbf{v}_2$,
- PLvi) $\mathbf{w} > \mathbf{v}_3$,
- PLvii) $\exists \mathbf{x} > \neg \forall \mathbf{x} \neg$,
- PLviii) $\mathbf{Sx}_1 \dots \mathbf{x}_n > \mathbf{S}^n \mathbf{x}_1 \dots \mathbf{x}_n$
- PLix) $\mathbf{p}_j > \mathbf{P}^0_j$ for all positive integers j .

The first seven clauses, unlike previous ones, involve direct abbreviation of non-formula expressions. They allow us to write a variety of formulas without subscripts on predicate letters and object variables. The symbol \exists in clause vii is called the existential quantifier. $\exists \mathbf{x}$ is read as **for some \mathbf{x}** . Clause viii allows us to drop superscripts on the predicate letters in a formula. Doing so does not destroy uniqueness of unabbreviated form. The degree of the predicate letter can always be recovered by counting the variable occurrences that follow it. This abbreviatory device requires a modification on the definition of abbreviation. For under the previous definition, $\mathbf{Pv}_1 \mathbf{v}_2$ would abbreviate $\mathbf{P}^1 \mathbf{v}_1 \mathbf{v}_2$, which would violate the property that all disabbreviation chains from formulas terminate in formulas. To prevent this, we modify the definition of abbreviation so that clause viii may only be applied when \mathbf{x}_n is not followed by a variable. A way to formulate the definition of direct abbreviation without modifying the definition of abbreviation is suggested in the exercises). The reader should keep in mind that when a formula like $\forall \mathbf{v}(\mathbf{Pv} \vee \neg \mathbf{Pv})$ is disabbreviated the two occurrences of \mathbf{P} are replaced by

distinct predicate letters, \mathbf{P}^1_1 and \mathbf{P}^2_1 . Predicate letters of degree zero are sentence letters. They constitute formulas themselves, without the addition of variables or other symbols. We use this terminology without prejudice to the question of whether there is a significant difference between sentences and predicates as ordinarily conceived. Notice that the sentence letters of $\mathcal{L}(\pi)$ are the capital letters $\mathbf{P}^0_1, \mathbf{P}^0_2, \dots$, whereas the sentence letters of \mathcal{L} were the lower case letters $\mathbf{p}_1, \mathbf{p}_2, \dots$. But the final clause in the definition of direct abbreviation permits us to use the latter to abbreviate the former. This makes it possible to view $\mathcal{L}(\pi)$ as an *extension* of \mathcal{L} as described in part I. Although the constituents of \mathcal{L} are not constituents of $\mathcal{L}(\pi)$, they are abbreviations in $\mathcal{L}(\pi)$ of such constituents, and consequently every formula of \mathcal{L} is a formula of $\mathcal{L}(\pi)$ -with-abbreviations.

Some terminology regarding quantifiers and variables will be useful later. It is sometimes helpful to regard the expressions $\forall \mathbf{x}$ and $\exists \mathbf{x}$ as unary connectives, like the negation sign. We refer to these expressions as (universal and existential) quantifier expressions (in \mathbf{x}). Recall that the scope of an occurrence of a connective is the smallest occurrence of a subformula to contain the connective occurrence as a part. Thus the scope of an occurrence of a quantifier expression in a formula is comprised of the occurrence of the quantifier expression itself and a subformula occurrence, which may be called the proper scope of the quantifier expression occurrence. Every occurrence of \mathbf{x} that is within the scope of an occurrence of a quantifier expression in \mathbf{x} is bound by that occurrence. Every occurrence of a variable that is not bound by any occurrence of a quantifier expression is free. A formula is said to be closed if it contains no free occurrences of any variable, otherwise it is open. \mathbf{y} is free at an occurrence of \mathbf{x} in \mathbf{A} if that occurrence does not lie within the scope of a quantifier expression in \mathbf{y} (so that when the occurrence of \mathbf{x} is replaced by a \mathbf{y} , that occurrence of \mathbf{y} is free). \mathbf{y} is free for \mathbf{x} in \mathbf{A} if \mathbf{y} is free at every free occurrence of \mathbf{x} in \mathbf{A} . For example, in the formula $\exists \mathbf{u}(\forall \mathbf{v} \mathbf{P} \mathbf{v} \vee \mathbf{R} \mathbf{v} \mathbf{w})$, the first two occurrences of \mathbf{v} are bound and the third is free. \mathbf{v} is free for \mathbf{w} , but \mathbf{u} is not free for \mathbf{v} .

Two occurrences of \mathbf{x} that are either both free or both bound by the same quantifier expression occurrence are linked. Two occurrences that of \mathbf{x} that are not linked are said to be independent. For example in $\forall \mathbf{v}(\mathbf{P} \mathbf{v} \vee \mathbf{Q} \mathbf{v})$ each of the three occurrences of \mathbf{v} is linked to each other occurrence of \mathbf{v} , whereas in $\forall \mathbf{v} \mathbf{P} \mathbf{v} \vee \forall \mathbf{v} \mathbf{Q} \mathbf{v}$ the first two occurrences of \mathbf{v} are each independent of the last two occurrences. The variable occurrences can be partitioned into families so that any two occurrences in a family are linked and any two occurrences in different families are independent. In the previous example, the first two variable occurrences comprise one family and the third and fourth occurrences comprise a second family. Since the members of a family will either be all free or all bound by a single quantifier, the families themselves can appropriately be labeled free or bound.

Bound variables serve primarily to indicate linkage or independence among predicate letter arguments and quantifiers. Beyond that, their identity is unimportant. Accordingly, we can say that formulas are alphabetic variants if they are identical except for their bound variables and these variables exhibit the same pattern of links. Some additional notation will help to make this notion more precise. If $\mathbf{x}_1, \dots, \mathbf{x}_n$ is a list, in order of occurrence and including repetitions, of all the variables that occur bound in \mathbf{A} , we may write \mathbf{A} as $(\mathbf{A})\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$ or as $\mathbf{A}\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$. A subsequent use of $(\mathbf{A})\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$ or $\mathbf{A}\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$ in the same context then refers to the result of replacing the i 'th bound variable occurrence in \mathbf{A} for $1 \leq i \leq n$ by \mathbf{y}_i . \mathbf{A} is an alphabetic variant of \mathbf{B}

(in symbols $\mathbf{A} \approx \mathbf{B}$) if $\mathbf{A} = \mathbf{A}\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$, $\mathbf{B} = \mathbf{A}\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$ and, for $1 \leq i, j \leq n$ x_i is linked to x_j in \mathbf{A} iff y_i is linked to y_j in \mathbf{B} . For example, $(\forall \mathbf{x} \mathbf{F} \mathbf{x} \mathbf{x} \wedge \exists \mathbf{x} \mathbf{G} \mathbf{x} \mathbf{y})$ is an alphabetic variant of $(\forall \mathbf{y} \mathbf{F} \mathbf{y} \mathbf{y} \wedge \exists \mathbf{z} \mathbf{G} \mathbf{z} \mathbf{y})$. It follows from the definition that the relation of being an alphabetic variant is an equivalence relation. The fact that a particular sequence of bound variables occurs in a formula can be regarded as an artifact of the notation. The linguistic objects of primary concern are then the equivalence classes of formulas under alphabetic variance. We refer to such classes as statements of $\mathcal{L}(\pi)$; and let $|\mathbf{A}|$ be the statement corresponding to \mathbf{A} , i.e. the set of all formulas that are alphabetic variants of \mathbf{A} .

The alphabetic variants of \mathbf{A} can be obtained by successively replacing families of occurrences of one variable by another. In replacing a family of occurrences of \mathbf{x} by \mathbf{y} 's, however, care must be taken that the resulting occurrences of \mathbf{y} remain linked only to each other. This requires that \mathbf{y} be free at each occurrence of \mathbf{x} in the family and that \mathbf{y} not occur free in the scope of a quantifier expression that binds an occurrence of \mathbf{x} in the family. Let us call a replacement that meets this condition a permissible one-family replacement of \mathbf{x} by \mathbf{y} in \mathbf{A} . A permissible one-family replacement in \mathbf{A} , then, produces an alphabetic variant of \mathbf{A} . Indeed, every alphabetic variant of \mathbf{A} can be obtained by series of such replacements. For suppose $\mathbf{A}\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$ and $\mathbf{A}\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$ are alphabetic variants, each with k families of variables. Let $\mathbf{z}_1, \dots, \mathbf{z}_k$ be distinct variables that do not occur in $\mathbf{A}\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$ or $\mathbf{A}\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$. Then successively replacing the k families of distinct variables in \mathbf{A} by $\mathbf{z}_1, \dots, \mathbf{z}_k$ is a series of permissible one-family replacements, resulting in an alphabetic variant \mathbf{A}' of $\mathbf{A}\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$ that has no variables in common with $\mathbf{A}\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$. $\mathbf{A}\langle \mathbf{y}_1, \dots, \mathbf{y}_n \rangle$ can then be obtained by a similar series of k permissible one-family replacements on \mathbf{A}' .

Some additional notational devices are useful in describing substitutions and replacements in formulas. If $\mathbf{x}_1, \dots, \mathbf{x}_n$ are distinct variables, we sometimes introduce $(\mathbf{A})(\mathbf{x}_1, \dots, \mathbf{x}_n)$ as a notation for \mathbf{A} to indicate that $\mathbf{x}_1, \dots, \mathbf{x}_n$ may have free occurrences in \mathbf{A} . This notation does not presume that any of $\mathbf{x}_1, \dots, \mathbf{x}_n$ actually do occur free in \mathbf{A} , nor that variables other than $\mathbf{x}_1, \dots, \mathbf{x}_n$ do not occur free in \mathbf{A} . If \mathbf{y}_i is free for \mathbf{x}_i , for $1 \leq i \leq n$, then any subsequent use of $(\mathbf{A})(\mathbf{y}_1, \dots, \mathbf{y}_n)$ in the same context refers to the result of (simultaneously) substituting \mathbf{y}_i for each free occurrence of \mathbf{x}_i , for $1 \leq i \leq n$, in \mathbf{A} . Alternatively, we may use the notation $(\mathbf{A})(\mathbf{y}_1, \dots, \mathbf{y}_n / \mathbf{x}_1, \dots, \mathbf{x}_n)$ to indicate the formula that results from this substitution. The latter notation makes clear that the resulting formula is a function of both $\mathbf{y}_1, \dots, \mathbf{y}_n$ and $\mathbf{x}_1, \dots, \mathbf{x}_n$. If, for some i , $1 \leq i \leq n$, \mathbf{y}_i is not free for \mathbf{x}_i in \mathbf{A} , $(\mathbf{A})(\mathbf{y}_1, \dots, \mathbf{y}_n)$ and $(\mathbf{A})(\mathbf{y}_1, \dots, \mathbf{y}_n / \mathbf{x}_1, \dots, \mathbf{x}_n)$ are undefined. Later the definition of these notations will be extended to include this case.

If $\mathbf{x}_1, \dots, \mathbf{x}_n$ is a list, in order and including repetitions, of all the variables with free occurrences in \mathbf{A} , we sometimes write \mathbf{A} as $(\mathbf{A})[\mathbf{x}_1, \dots, \mathbf{x}_n]$. If \mathbf{y}_i is free for \mathbf{x}_i , for $1 \leq i \leq n$, then a subsequent use of $(\mathbf{A})[\mathbf{y}_1, \dots, \mathbf{y}_n]$ is the result of replacing the i 'th free variable occurrence in \mathbf{A} by \mathbf{y}_i for $1 \leq i \leq n$. Alternatively, we may use the notation $(\mathbf{A})[\mathbf{y}_1, \dots, \mathbf{y}_n / \mathbf{x}_1, \dots, \mathbf{x}_n]$ to indicate the formula that results from this replacement. Again, these notations will later be extended to cover the case in which, for some i , $1 \leq i \leq n$, \mathbf{y}_i is not free for \mathbf{x}_i .

For example, if \mathbf{A} is $\forall \mathbf{v}(\mathbf{R} \mathbf{u} \mathbf{v} \vee \mathbf{P} \mathbf{u} \mathbf{w} \mathbf{u})$ we might write \mathbf{A} either as $(\mathbf{A})(\mathbf{u})$ or as $(\mathbf{A})[\mathbf{u}, \mathbf{u}, \mathbf{w}, \mathbf{u}]$. In the first case, $(\mathbf{A})(\mathbf{w})$ is $\forall \mathbf{v}(\mathbf{R} \mathbf{w} \mathbf{v} \vee \mathbf{P} \mathbf{w} \mathbf{w} \mathbf{w})$. In the second case $(\mathbf{A})[\mathbf{w}, \mathbf{w}, \mathbf{u}, \mathbf{w}]$ is $\forall \mathbf{v}(\mathbf{R} \mathbf{w} \mathbf{v} \vee \mathbf{P} \mathbf{w} \mathbf{u} \mathbf{w})$. The parentheses around \mathbf{A} in these notations may be dropped when no ambiguity results from so doing.

Transparency, the result that any occurrence of a formula within a formula is a syntactic occurrence, carries over to $\mathcal{L}(\pi)$, and, as before, a unique readability result for the language follows. When the abbreviatory devices are added to $\mathcal{L}(\pi)$, however, transparency is lost. \mathbf{Pvw} , for example, contains a non-syntactic occurrence of \mathbf{Pv} . Unique readability of the enlarged version of $\mathcal{L}(\pi)$ can still be proved by the method outlined in problem 1.21.

Drill, Exercises and Problems

1[d]. Give a counterexample to each of the following claims:

- a. If $\mathbf{A}=\mathbf{B}^{(x/y)}$ then $\mathbf{B}=\mathbf{A}^{(y/x)}$;
- b. $\mathbf{A}^{(y/x)}^{(z/y)}=\mathbf{A}^{(z/x)}$;
- c. $\mathbf{A}^{(x/y)}^{(y/x)}=\mathbf{A}$
- d. $\mathbf{A}^{(y/x)}^{(z/w)}=\mathbf{A}^{(z/w)}^{(x/y)}$;

2[e] For each of a) - d) above, state conditions that are necessary and sufficient for the identities to hold.

3[e] Show that the context-sensitive rule of abbreviation for \mathbf{P}^n can be replaced with a context-free rule. (Hint. Choose rules that will permit the following chain of disabbreviations: $\mathbf{Pvw} > \mathbf{P}^1\mathbf{uvw} > \mathbf{P}^2\mathbf{uvw} > \mathbf{P}^3\mathbf{uvw}$.)

4[e]. State conditions necessary and sufficient for the following identity to hold. (\mathbf{A} , \mathbf{B} , and \mathbf{C} do not occur previously in this context.): If $\mathbf{B}(y)=\mathbf{A}(y)$ and $\mathbf{C}(x)=\mathbf{A}(x)$ then $\mathbf{C}(y)=\mathbf{A}(y)$.

5[e]. Prove that the relation that holds between two occurrences of x in \mathbf{A} iff they are linked is an equivalence relation.

6[e]. Suppose that there are two occurrences of x in a subformula \mathbf{A}' of \mathbf{A} . Prove that the two occurrences are linked in \mathbf{A} iff they are linked in \mathbf{A}' .

7[e]. Prove that alphabetic variance is an equivalence relation.

8[e]. a. Show that $\forall \mathbf{x}\mathbf{A} \approx \forall \mathbf{y}(\mathbf{B})^{(x/y)}$ implies $\mathbf{A} \approx \mathbf{B}$ (when \mathbf{y} is free for \mathbf{x} in \mathbf{B}).

b. Show that $\mathbf{A} \approx \mathbf{B}$ does not imply $\forall \mathbf{x}\mathbf{A} \approx \forall \mathbf{y}(\mathbf{B})^{(x/y)}$.

c. Show that $(\mathbf{A} \vee \mathbf{B}) \approx (\mathbf{A}' \vee \mathbf{B}')$ iff $\mathbf{A} \approx \mathbf{A}'$ and $\mathbf{B} \approx \mathbf{B}'$, and that $\neg \mathbf{A} \approx \neg \mathbf{A}'$ iff $\mathbf{A} \approx \mathbf{A}'$.

9[p]. Let the link-map of \mathbf{A} be the triple $(\mathbf{F}, \mathbf{B}, \equiv)$, where \mathbf{F} is the set of all occurrences of free variables in \mathbf{A} , \mathbf{B} is the set of all occurrences of bound variables in \mathbf{A} , and \equiv is the linkage relation (as defined on $\mathbf{F} \cup \mathbf{B}$). a. Give an inductive definition of the link-map of \mathbf{A} (making use of the notion of alignment as defined in problem **). b. Redo exercises 5-8 using the new definition of linkage.

10[p]. Let n be the largest integer such that \mathbf{v}_n occurs free in \mathbf{A} . Say that \mathbf{A} is in standard form if no two occurrences of the same variable are independent and the bound variables of \mathbf{A} , in order of their first occurrence, are $\mathbf{v}_{n+1}, \dots, \mathbf{v}_{n+m}$ for some number m . Prove that every formula has a unique alphabetic variant in standard form.

11[p]. Suppose that \mathbf{B} is obtained from \mathbf{A} by replacing variable occurrences (be they free or

bound) by other variables. What are the necessary and sufficient conditions for **B** to be an alphabetic variant of **A**?

12[p]. (replacement in a general setting) Suppose we have a language with expressions of three types: terms, predicates, and variable-binders. Consider expressions **X**, **Y** and **Z** of this language, **Y** and **Z** of the same type. Give a general definition of a free replacement of **Z** for **Y** in **X** (written $\mathbf{X}^{\mathbf{Y}/\mathbf{Z}}$). [Hint: first explain an appropriate notion of an occurrence of **Z**'s being free for **Y** in **X** and then define free replacement in terms of replacement in alphabetic variants in which the desired freedom obtains.] We shall consider one instance of this notion--free replacement of predicates for predicates in $\mathcal{L}(\pi)$ --in chapter 6 below.