and is maximally consistent; then, by Henkin's Theorem (8), I_Γ is a model of $(\Gamma')^*$, hence a model of Γ too. $\square$

Exercise 32. **(*)** *Research in the suggested related references how a consistent set built over a countable set of symbols, but that uses infinite free variables can be extended to a maximal consistent set with witnesses. The problem, is that in this case there are no new variables that can be used as witnesses. Thus, one needs to extend the language with new constant symbols that will act as witnesses, but each time a new constant symbol is added to the language the set of existential formulas change.*

Exercise 33. **(*)** *Research the general case in which the language is not restricted, that is the case in which Γ is built over a non countable set of symbols.*

2.5.3 Compactness Theorem and Löwenheim-Skolem Theorem

The connections between $\models$ and $\vdash$ as well as between consistence and satisfiability provided in this section, give rise to other additional important consequences that relate semantic and syntactic elements of the predicate logic. Here we present two important theorems that are related with the scope and limits of the expressiveness of predicate logic.

Theorem 9 (Compactness)**.** *Given a set Γ of predicate formulas and a formula φ, the following holds*

i. $\Gamma \models \varphi$ *if and only if there is a finite set* $\Gamma_0 \subseteq \Gamma$ *such that* $\Gamma_0 \models \varphi$

ii. Γ *is satisfiable if and only if for all finite set* $\Gamma_0 \subseteq \Gamma$, Γ_0 *is satisfiable.*

Proof. i. For necessity, if $\Gamma \models \varphi$, by completeness one has that there exists a derivation ∇ for $\Gamma \vdash \varphi$. The derivation ∇ uses only a finite subset of assumptions, say $\Gamma_0 \subseteq \Gamma$. Thus, $\Gamma_0 \vdash \varphi$ and, by correctness, one concludes that $\Gamma_0 \models \varphi$. For sufficiency, suppose that $\Gamma_0 \models \varphi$, for a finite set $\Gamma_0 \subseteq \Gamma$. By completeness there exists a derivation ∇ for $\Gamma_0 \vdash \varphi$. But ∇ is also a derivation for $\Gamma \vdash \varphi$; hence, by correctness one concludes that $\Gamma \models \varphi$.

ii. Necessity is proved by contraposition: if Γ_0 were unsatisfiable for some finite set $\Gamma_0 \subseteq \Gamma$, then Γ_0 would be inconsistent, since consistency implies satisfiability (Corollary 2); thus, $\Gamma_0 \vdash \bot$, which implies also that $\Gamma \vdash \bot$ and by correctness that $\Gamma \models \bot$. Hence, Γ would be unsatisfiable. Sufficiency is proved also by contraposition: if we assume that Γ is unsatisfiable, then since there exists no model for Γ, $\Gamma \models \bot$ holds. By completeness also, $\Gamma \vdash \bot$ and hence, there exists a finite set $\Gamma_0 \subseteq \Gamma$, such that $\Gamma_0 \vdash \bot$, which by correctness implies that $\Gamma_0 \models \bot$. Thus, we conclude that Γ_0 is unsatisfiable.

□

The compactness theorem has several applications that are useful for restricting the analysis of consistency and satisfiability of arbitrary sets of predicate formulas to only finite subsets. This also has important implications in the possible cardinality of models of sets of predicate formulas such as the one given in the following theorem.

Theorem 10 (Löwenheim-Skolem). *Let Γ be a set of formulas such that for any natural $n \in \mathbb{N}$, there exists a model of Γ with a domain of cardinality at least n. Then Γ has also infinite models.*

Proof. Consider an additional binary predicate symbol E and the formulas φ_n for $n > 0$, defined as

$$\forall_x\ E(x,x) \land \exists_{x_1,\dots,x_n} \bigwedge_{i\neq j;i,j=1}^{n} \neg E(x_i,x_j)$$

For instance, the formulas φ_1 and φ_3 are given respectively as $\forall_x E(x,x)$ and $\forall_x\ E(x,x) \land \exists_{x_1}\exists_{x_2}\exists_{x_3}\ (\neg E(x_1,x_2) \land \neg E(x_1,x_3) \land \neg E(x_2,x_3))$.

Notice that φ_n has models of cardinality at least n. It is enough to interpret E just as a the reflexive relation among the elements of the domain of the interpretation. Thus, pairs of different elements of the domain do not belong to the interpretation of E.

Let Φ be the set of formulas $\{\varphi_n \mid n \in \mathbb{N}\}$. We will prove that all finite subsets of the set of formulas $\Gamma \cup \Phi$ are satisfiable and then by the compactness theorem conclude that $\Gamma \cup \Phi$ is satisfiable too. An interpretation $I \models \Gamma \cup \Phi$ should have an infinite model, since also $I \models \Phi$

and all formulas in Φ are true in I only if there are infinitely many elements in the domain of I.

To prove that any finite set $\Gamma_0 \subset \Gamma \cup \Phi$ is satisfiable, let k be the maximum k such that $\varphi_k \in \Gamma_0$. Since Γ has models of arbitrary finite cardinality, let I' be a model of Γ with at least k elements in its domain $\mathtt{D}$. I' can be extended in such a manner that the binary predicate symbol E is interpreted just as the reflexive relation over $\mathtt{D}$. Let I be the extended interpretation. It is clear that $I \models \Gamma$ since E is a new symbol and also $I \models \Gamma_0 \cap \Phi$ since the domain has at least k different elements. Also, since $I \models \Gamma$, we have that $I \models \Gamma \cap \Gamma_0$. Hence, $I \models \Gamma_0$ and so we conclude that Γ_0 is satisfiable. $\square$

Exercise 34. *Prove that there is no predicate formula φ that holds exclusively for all finite interpretations.*

Exercise 35. *Let E be a binary predicate symbol, e a constant and $\cdot$ and -1 be binary and unary function symbols, respectively. The theory of groups is given by the models of the set of formulas Γ_G:*

$$\forall_x \; E(x, x)$$
$$\forall_{x,y} \; (E(x, y) \to E(y, x))$$
$$\forall_{x,y,z} \; (E(x, y) \wedge E(y, z) \to E(x, z))$$
$$\forall_x \; E(x \cdot e, x)$$
$$\forall_x \; E(x \cdot x^{-1}, e)$$
$$\forall_{x,y,z} \; E((x \cdot y) \cdot z, x \cdot (y \cdot z))$$

Notice that according to the three first axioms the symbol E should be interpreted as an equivalence relation such as the equality. Indeed, the three other axioms are those related with group theory itself: the fourth one states the existence of an identity element, the fifth one the inverse function and the sixth one the associativity of the binary operation.

Prove the existence of infinite models by proving that for any $n \in \mathbb{N}$, the structure of arithmetic modulo n is a group of cardinality n. The elements of this structure are all integers modulo n (i.e. the set $\{0, 1, \ldots, n\text{-}1\}$), with addition and identity element 0.

Exercise 36. *A graph is a structure of the form $G = \langle V, E\rangle$, where V is a finite set of vertices and $E \subset V \times V$ a set of edges between the vertices. The problem of reachability in graphs is the question whether there exists a finite path of consecutive edges, say $(u, u_1), (u_1, u_2), \ldots, (u_{n-1}, v)$, between two given nodes $u, v \in V$.*

Prove that there is no predicate formula that expresses reachability in graphs.

Hint: the key observation to conclude is that the problem of reachability between two nodes might be answered positively whenever there exists a path of arbitrary length.

2.6 Undecidability of the Predicate Logic

The gain of expressiveness obtained in predicate logic w.r.t. to the propositional logic comes at a price. Initially, remember that for a given propositional formula φ, one can always answer whether φ is valid or not by analyzing its truth table. This means that there is an algorithm that receives an arbitrary propositional formula as input and **always** answers after a finite amount of time `yes`, if the given formula is valid; or `no`, otherwise. The algorithm works as follows: build the truth table for φ and check whether it is true for all interpretations. Note that this algorithm is not efficient because the (finite) number of possible interpretations grows exponentially w.r.t. the number of propositional variables occurring in φ.

In general, a computational question with a `yes` or `no` answer depending on the parameters is known as a *decision problem.* A decision problem is said to be *decidable* whenever there exists an algorithm that correctly answers `yes` or `no` for each instance of the problem, and when such algorithm does not exist the decision problem is said to be *undecidable* . Therefore, we conclude that that *validity* is decidable in propositional logic.

The natural question that arises at this point is whether validity is decidable or not in predicate logic. Note that the truth table approach is no longer possible because the number of different interpretations for a given predicate formula φ is not finite. In fact, as stated in the previous paragraph the gain of expressiveness of the predicate logic comes at a price: validity is undecidable in predicate logic. This fact is usually known as the *undecidability of predicate logic*, and has several important consequences. In fact, it is straightforward from the completeness of predicate logic that provability is also undecidable, i.e. there is no algorithm that receives a predicate formula φ as input and returns `yes` if $\vdash \varphi$, or `no` if not $\vdash \varphi$.

The standard technique for proving the undecidability of the predicate logic consists in reducing a known undecidable problem to the validity of the predicate logic in such a way that decidability of validity of the predicate logic entails the decidability of the other problem leading to a contradiction. In what follows, we consider the word problem for a specific monoid introduced by G. S. Tseitin, and that is well-known to be undecidable.

A semigroup is an algebraic structure with a binary associative operator $\cdot$ over a given set A. When in addition the structure has an identity element id it is called a monoid. By associativity, one understands that for all x, y, z in A, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, and for all $x \in A$ the identity satisfies the properties $id \cdot x = x$ and $x \cdot id = x$. In general, the word problem in a given semigroup with a given set of equations E (between pairs of elements of A), is the problem of answering whether two words are equal *applying* these equations.

By an application of an equation, say $u = v$ in E, one understand an equational transformation of the form below, where x y are any elements of A.

$$x \cdot (u \cdot y) = x \cdot (v \cdot y)$$

Hence, the word problem consists in answering for any pair of elements $x, y \in A$ if there exists a finite chain, possibly of length zero, of applications of equations that transform x in y:

$$x \equiv x_0 \overset{u_1=v_1}{=} x_1 \overset{u_2=v_2}{=} x_2 \overset{u_3=v_3}{=} \dots \overset{u_n=v_n}{=} x_n \equiv y \tag{2.2}$$

In the chain above, the notation $\equiv$ is used for syntactic equality and $\overset{u_i=v_i}{=}$ for highlighting that the equation applied in the application step is $u_i = v_i$.

Tseitin's monoid is given by the set Σ^* of words freely generated by the quinary alphabet $\Sigma = \{a, b, c, d, e\}$. In this structure the binary associative operator is the concatenation of words and the empty word plays the role of the identity. The set of equations is given below. For simplicity, we will omit parentheses and the concatenation operator.

$$\begin{aligned} ac &= ca \\ ad &= da \\ bc &= cb \\ bd &= db \\ ce &= eca \\ de &= edb \\ cdca &= cdcae \end{aligned} \tag{2.3}$$

As previously mentioned Tseitin introduced this specific monoid with the congruence generated by this set of equations and proved that the word problem in this structure is undecidable.

In order to reduce the above problem to validity of the predicate logic, we choose a logical language with a constant symbol $\Box$, five unary function symbols f_a, f_b, f_c, f_d and f_e, and a binary predicate P. The constant $\Box$ will be interpreted as the empty word, and each function symbol, say $f_\star$ for $\star \in \Sigma$, as the concatenation of the symbol $\star$ to the left of the term given as argument of $f_\star$. For example, the word *baaecde* will be encoded as $f_b(f_a(f_a(f_e(f_c(f_d(f_e(\Box)))))))$, which for brevity will be written simply as $f_{baaecde}(\Box)$. The binary predicate P will play the role of equality, i.e. $P(x, y)$ is interpreted as x is equal to y (modulo the congruence induced by the set of equations above, which would be assumed as axioms).

Our goal is, given an instance of the word problem $x, y \in \Sigma^*$ specified above, to build a formula $\varphi_{x,y}$ such that x equals y in this structure if and only if $\models \varphi_{x,y}$. The formula $\varphi_{x,y}$ is of the form

$$\varphi' \to P(f_x(\Box), f_y(\Box)) \tag{2.4}$$

where φ' is the following formula:

$$
\begin{aligned}
\forall_x(P(x,x)) \land \\
\forall_x \forall_y(P(x,y) \to P(y,x)) \land \\
\forall_x \forall_y, \forall_z(P(x,y) \land P(y,z) \to P(x,z)) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{ac}(x), f_{ca}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{ad}(x), f_{da}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{bc}(x), f_{cb}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{bd}(x), f_{db}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{ce}(x), f_{eca}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{de}(x), f_{edb}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_{cdca}(x), f_{cdcae}(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_a(x), f_a(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_b(x), f_b(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_c(x), f_c(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_d(x), f_d(y))) \land \\
\forall_x \forall_y(P(x,y) \to P(f_e(x), f_e(y)))
\end{aligned}
\tag{2.5}
$$

Suppose $\models \varphi_{x,y}$. Our goal is to find a model for $\varphi_{x,y}$ which tells us if there is a solution to the instance $x, y \in \Sigma^*$. Consider the interpretation I with domain Σ^* and such that:

- the constant $\square$ is interpreted as the empty word;
- each unary function symbol $f_\star$, for $\star \in \Sigma$, is interpreted as the function $f_\star^I : \Sigma^* \to \Sigma^*$ that appends the symbol $\star$ to the word $x \in \Sigma^*$ given as argument, i.e. $f_\star^I(x) = \star x$;
- and the binary predicate P is interpreted as follows:

  $P(x,y)^I$ if and only if there exists a chain, possibly of length zero, of applications of the equations (2.3) that transform x into the word y.

We claim that $I \models \varphi'$. Let us consider each case:

- $I \models \forall x(P(x,x))$: take the empty chain.
- $I \models \forall_x \forall_y(P(x,y) \to P(y,x))$: for any x, y such that $I \models P(x,y)$, take the chain given for $P(x,y)$ in reverse order.
- $I \models \forall_x \forall_y \forall_z(P(x,y) \wedge P(y,z) \to P(x,z))$: for any x, y, z such that $I \models P(x,y)$ and $I \models P(y,z)$, append the chains given for $P(x,y)$ and $P(y,z)$.
- $I \models \forall_x \forall_y(P(x,y) \to P(f_{ac}(x), f_{ca}(y)))$: for any x, y such that $I \models P(x,y)$, take the chain given for $P(x,y)$ and use this for the chain of equations for $acx = acy$; then add an application of the equation $ac = ca$ to obtain cay. A similar justification is given for all other cases related with equations (2.3), but the last.
- $I \models \forall_x \forall_y(P(x,y) \to P(f_\star(x), f_\star(y)))$ where $\star \in \Sigma$: for any x, y such that $I \models P(x,y)$, take the chain given for $P(x,y)$ and use it for the chain for the equation $\star x = \star y$.

Since $I \models \varphi_{x,y}$ and $I \models \varphi'$, we conclude that $I \models P(f_x(\Box), f_y(\Box))$. Therefore, the instance x, y of the word problem has a solution.

Conversely, suppose the instance x, y of the word problem has a solution in Tseitin's monoid; i.e., there is a chain of applications of the equations (2.3) from x resulting in the word y as given in the chain (2.2). We will suppose that this chain is of length n.

We need to show that $\varphi_{x,y}$ is valid; i.e., that $\models \varphi_{x,y}$. Let us consider an arbitrary interpretation I' over a domain $\mathtt{D}$ with an element $\Box^{I'}$, five unary functions $f_a^{I'}$, $f_b^{I'}$, $f_c^{I'}$, $f_d^{I'}$, $f_e^{I'}$ and a binary relation $P^{I'}$. Since $\varphi_{x,y}$ is equal to $\varphi' \to P(f_u(\Box), f_v(\Box))$, we have to show that if $I' \models \varphi'$ then $I' \models P(f_u(\Box), f_v(\Box))$.

We proceed by induction in n, the length of the chain of applications of equations (2.3) for transforming x in y.

IB: case $n = 0$, we have that $x \equiv y$ and if $I' \models \varphi'$, $I' \models \forall_x P(x,x)$ which also implies that $I' \models P(x,x)$.

IS: case $n > 0$, the chain of applications of equations to transform x in y is of the form

$$x \equiv x_0 \overset{u_1=v_1}{=} x_1 \overset{u_2=v_2}{=} x_2 \overset{u_3=v_3}{=} \dots x_{n-1} \overset{u_n=v_n}{=} x_n \equiv y$$

By induction hypothesis we have that $I' \models P(x, x_{n-1})$. If we prove that $I' \models P(x_{n-1}, y)$, we can conclude that $I' \models P(x, y)$, since $I' \models \forall_x \forall_y \forall_z P(x, y) \wedge P(y, z) \to P(x, z)$ because we are assuming that $I' \models \varphi'$.

Thus, the proof resumes to prove that equalities obtained by one step of application of equations in (2.3) hold in I': in particular if we suppose that $u_n = v_n$ is the equation $u = v$ in (2.3), $x_{n-1} \equiv wuz$ and $y = wvz$, we need to prove that $I' \models P(f_{wuz}(\square), f_{wvz}(\square))$, which is done by the following three steps:

1. First, one has that $I' \models P(f_z(\square), f_z(\square))$, since $I' \models \forall_x P(x, x)$.

2. Second, since $u = v$ in (2.3), $I' \models \forall_x \forall_y (P(x, y) \to P(f_u(x), f_v(y)))$. Thus, by the previous item one has that $I' \models P(f_{uz}(\square), f_{vz}(\square))$;

3. Third, $I' \models P(f_{wuz}(\square), f_{wvz}(\square))$ is obtained from the last item, inductively on the length of w, since $I' \models \forall_x \forall_y (P(x, y) \to P(f_\star(x), f_\star(y)))$, for all $\star \in \Sigma$.

To conclude the undecidability of validity of the predicate logic, if we suppose the contrary, we will be able to answer for any $x, y \in \Sigma^*$ if $\models P(f_x(\square), f_y(\square))$ answering consequently if x equals y in Tseitin's monoid, which is impossible since the word problem in this structure is undecidable.

Theorem 11 (Undecidability of the Predicate Logic)**.** *Validity in the predicate logic, that is answering whether for a given formula φ, $\models \varphi$ is undecidable.*

Notice, that by Gödel completeness theorem undecidability of validity immediately implies undecidability of derivability in the predicate logic. Indeed, in the above reasoning one can use the completeness theorem to alternate between validity and derivability.

Exercise 37. *Accordingly to the three steps above to prove $I' \models P(f_{wuz}(\square), f_{wvz}(\square))$, build a derivation for the sequent $\vdash P(f_{wuz}(\square), f_{wvz}(\square))$. Concretely, prove that:*

a. $\varphi' \vdash P(f_z(\square), f_z(\square))$, for $z \in \Sigma^*$;

b. $\varphi', P(f_z(\square), f_z(\square)) \vdash P(f_{uz}(\square), f_{vz}(\square))$, for $u = v$ in the set of equations (2.3);

c. $\varphi', P(f_{uz}(\Box), f_{vz}(\Box)) \vdash P(f_{wuz}(\Box), f_{wvz}(\Box))$*, for* $w \in \Sigma^*$*;*

d. $\varphi' \vdash P(f_{wuz}(\Box), f_{wvz}(\Box))$.