Sottolineo quelle che (credo) siano le differenze principali tra la mia notazione
la notazione delle dispense di Berarducci usate nel corso di Istituzioni.

Chapter 1

Preliminaries and notation

This chapter introduces the syntax and semantic of first order logic. We assume that the reader has at least some familiarity with first order logic.

The definitions of terms and formulas we give in Section 3 and 5 are more formal than is required in subsequent chapters. Our main objective is to convince the reader that a rigorous definition of language and truth is possible. However, the actual details of such a definition are not relevant for our purposes.

Notate che manca una categoria apposita per le costanti (convenzione molto usata)

1 Structures

A (first order) language L (also called signature) is a triple that consists of

1. a set L_{fun} whose elements are called function symbols;
2. a set L_{rel} whose elements are called relation symbols;
3. a function that assigns to every $f \in L_{\text{fun}}$, respectively $r \in L_{\text{rel}}$, non-negative integers n_f and n_r that we call arity of the function, respectively relation, symbol. We say that f is an n_f-ary function symbol, and similarly for r. A 0-ary function symbol is also called a constant.

Warning: it is customary to use the symbol L to denote both the language and the set of formulas (to be defined below) associated to it. We denote by $|L|$ the cardinality of $L_{\text{fun}} \cup L_{\text{rel}} \cup \omega$. Note that, by definition, $|L|$ is always infinite. (convenzione molto usata)

A (first order) structure M of signature L (for short L-structure) consists of

1. a set that we call the domain or support and denote by the same symbol M used for the structure as a whole;
2. a function that assigns to every $f \in L_{\text{fun}}$ a total map $f^M : M^{n_f} \to M$;
3. a function that assigns to every $r \in L_{\text{rel}}$ a relation $r^M \subseteq M^{n_r}$.

We call f^M, respectively r^M, the interpretation of f, respectively r, in M.

Così è come recuperiamo le costanti. Il formalismo verrà anche usato con formule/enunciati.

Recall that, by definition, $M^0 = \{\varnothing\}$. Therefore the interpretation of a constant c is a function that maps the unique element of M^0 to an element of M. We identify c^M with $c^M(\varnothing)$.

We may use the word model as a synonym for structure. But beware that, in some contexts, the word is used to denote a particular kind of structure.

If M is an L-structure and $A \subseteq M$ is any subset, we write $L(A)$ for the language obtained by adding to L_{fun} the elements of A as constants. In this context, the elements of A are called parameters. There is a canonical expansion of M to an $L(A)$-structure that is obtained by setting $a^M = a$ for every $a \in A$.

1.1 Example The language of additive groups consists of the following function symbols:

1. a constant (that is, a function symbol of arity 0) 0
2. a unary function symbol (that is, of arity 1) $-$
3. a binary function symbol (that is, of arity 2) $+$.

In the language of multiplicative groups the three symbols above are replaced by 1, $^{-1}$, and $\cdot$ respectively. Any group is a structure in either of these two signatures with the obvious interpretation. Needless to say, not all structures with these signatures are groups.

The language of (unitary) rings contains all the function symbols above except $^{-1}$. The language of ordered rings also contains the binary relation symbol $<$. □

The following example is less straightforward. The reason for the choice of the language of vector spaces will become clear in Example 1.9 below.

1.2 Example Let F be a field. The language of vector spaces over F, which we denote by L_F, extends that of additive groups by a unary function symbol k for every $k \in F$.

Recall that a vector space over F is an abelian group M together with a function $\mu : F \times M \to M$ satisfying some properties (that we assume well-known, see Example 2.4). To view a vector space over F as an L_F-structure, we interpret the group symbols in the obvious way and each $k \in F$ as the function $\mu(k, \text{-})$. □

The languages in Examples 1.1 and 1.2, with the exception of that of ordered rings, are functional languages, that is, $L_{\text{rel}} = \varnothing$. In what follows, we consider two important examples of relational languages, that is, languages where $L_{\text{fun}} = \varnothing$.

1.3 Example The language of strict orders only contains a binary relation symbol, usually denoted By $<$. The language of graphs, too, only contains a binary relation symbol (for which there is no standard notation). □

2 Tuples

La notazione per le tuple è molto compatta (a volte un po informale)
Al momento, è sufficiente chiarire la notazione per tuple finite o di lunghezza ω e le convenzioni sulla tupla vuota.

A sequence is a function $a : I \to A$ whose domain is a linear order $I, <_I$. We may use the notation $a = \langle a_i : i \in I \rangle$ for sequences. A tuple is a sequence whose domain is an ordinal, say α, then we write $a = \langle a_i : i < \alpha \rangle$. When α is finite, we may also write $a = a_0, \dots, a_{\alpha-1}$ The domain of the tuple a, the ordinal α, is denoted by $|a|$ and is called the length of a. If a is surjective, it is said to be an enumeration of A.

If $J \subseteq I$ is a subset of the domain of the sequence $a = \langle a_i : i \in I \rangle$, we write $a_{\restriction J}$ for the restriction of a to J. When J is well ordered by $<_I$, e.g. when a is a tuple or when J is finite, we identify $a_{\restriction J}$ with a tuple. This is the tuple $\langle a_{j_k} : k < \beta \rangle$ where $\langle j_k : k < \beta \rangle$ is the unique increasing enumeration on J.

⚠ Sometimes (i.e. not always) we may overline tuples or sequences as mnemonic. When a tuple $\bar{c}$ is introduced, we write c_i for the i-th element of $\bar{c}$. and $c_{\restriction J}$ for the restriction of $\bar{c}$ to $J \subseteq |\bar{c}|$. Note that the bar is dropped for ease of notation.

The set of tuples of elements of length α is denoted by A^{α}. The set of tuples of length $< \alpha$ is denoted by $A^{<\alpha}$. For instance, $A^{<\omega}$ is the set of all finite tuples of elements of A. When α is finite we do not distinguish between A^α and the α-th Cartesian power of A. In particular, we do not distinguish between A^1 and A.

If $a, b \in A^\alpha$ and h is a function defined on A, we write $h(a) = b$ for $h(a_i) = b_i$. We often do not distinguish between the pair $\langle a, b\rangle$ and the tuple of pairs $\langle a_i, b_i\rangle$. The context will resolve the ambiguity.

Note that there is a unique tuple of length 0, the empty set $\varnothing$, which in this context is called empty tuple. Recall that by definition $A^0 = \{\varnothing\}$ for every set A. Therefore, even when A is empty, A^0 contains the empty string.

We often concatenate tuples. If a and b are tuples, we write $a\,b$ or, equivalently, a, b.

3 Terms

Let V be an infinite set whose elements we call variables. We use the letters x, y, z, etc. to denote variables or tuples of variables. We rarely refer to V explicitly, and we always assume that V is large enough for our needs.

We fix a signature L for the whole section.

1.4 Definition *A term is a finite sequence of elements of $L_{\text{fun}} \cup V$ that are obtained inductively as follows:*

o. every variable, intended as a tuple of length 1, is a term;

i. if $f \in L_{\text{fun}}$ and t is a tuple obtained by concatenating n_f terms, then $f\,t$ is a term. We write $f\,t$ for the tuple obtained by prefixing t by f.

We say L-term when we need to specify the language L. □

Note that any constant f, intended as a tuple of length 1, is a term (by i, the term f is obtained concatenating $n_f = 0$ terms and prefixing by f). Terms that do not contain variables are called closed terms.

The intended meaning of, for instance, the term $+ + x\,y\,z$ is $(x + y) + z$. The first expression uses prefix notation; the second uses infix notation. When convenient, we informally use infix notation and add parentheses to improve legibility and avoid ambiguity.

The following lemma shows that prefix notation allows to write terms unambiguously without using parentheses.

1.5 Lemma (unique legibility of terms) *Let a be a sequence of terms. Suppose a can be obtained both by concatenating the terms $t_1, \ldots, t_n$ and by concatenating the terms $s_1, \ldots, s_m$. Then $n = m$ and $s_i = t_i$.*

Proof By induction on $|a|$. If $|a| = 0$ than $n = m = 0$ and there is nothing to prove. Suppose the claim holds for tuples of length k and let $a = a_1, \ldots, a_{k+1}$. Then a_1 is the first element of both t_1 and s_1. If a_1 is a variable, say x, then t_1 and s_1 are the term x and $n = m = 1$. Otherwise a_1 is a function symbol, say f. Then $t_1 = f\,\bar{t}$ and $s_1 = f\,\bar{s}$, where $\bar{t}$ and $\bar{s}$ are obtained by concatenating the terms $t'_1, \ldots, t'_p$ and $s'_1, \ldots, s'_p$. Now apply the induction hypothesis to $a_2, \ldots, a_{k+1}$ and to the terms $t'_1, \ldots, t'_p, t_2, \ldots, t_n$ and $s'_1, \ldots, s'_p, s_2, \ldots, s_m$. □

If $x = x_1, \dots, x_n$ is a tuple of distinct variables and $s = s_1, \dots, s_n$ is a tuple of terms, we write $t[x/s]$ for the sequence obtained by replacing x by s coordinatewise. Proving that $t[x/s]$ is indeed a term is a tedious task that can be safely skipped.

If t is a term and $x_1, \dots, x_n$ are (tuples of) variables, we write $t(x_1, \dots, x_n)$ to declare that the variables occurring in t are among those that occur in $x_1, \dots, x_n$. When a term has been presented as $t(x, y)$, we write $t(s, y)$ for $t[x/s]$.

Finally, we define the interpretation of a term in a structure M. We begin with closed terms. These are interpreted as 0-ary functions, i.e. as elements of the structure.

1.6 Definition *Let t be a closed $L(M)$ term. The interpretation of t, denoted by t^M, is defined by induction of the syntax of t as follows.*

i. *if $t = f\bar{t}$, where $f \in L_{\text{fun}}$ and $\bar{t}$ is a tuple obtained by concatenating the terms $t_1, \dots, t_{n_f}$, then $t^M = f^M(t_1^M, \dots, t_{n_f}^M)$.*

Note that in i we have used Lemma 1.5 in an essential way. In fact this ensures that the sequence $\bar{t}$ uniquely determines the terms $t_1, \dots, t_{n_f}$. □

The inductive definition above is based on the case $n_f = 0$, that is, the case where f a constant, or a parameter. When $t = c$, a constant, $\bar{t}$ is the empty tuple, and so $t^M = c^M(\varnothing)$, which we abbreviate as c^M. In particular, if $t = a$, a parameter, then $t^M = a^M = a$.

Now we generalize the interpretation to all (not necessarily closed) terms. If $t(x)$ is a term, we define $t^M(x) : M^{|x|} \to M$ to be the function that maps a to $t(a)^M$.

4 Substructures

In the working practice, a *substructure* is a subset of a structure that is closed under the interpretation of the functions in the language. But there are a few cases when we need the following formal definition.

1.7 Definition *Fix a signature L and let M and N be two L-structures. We say that M is a substructure of N, and write $M \subseteq N$, if*

1. *the domain of M is a subset of the domain of N*
2. *$f^M = f^N \upharpoonright M^{n_f}$ for every $f \in L_{\text{fun}}$*
3. *$r^M = r^N \cap M^{n_r}$ for every $f \in L_{\text{rel}}$.*

Note that when f is a constant 2 becomes $f^M = f^N$, in particular the substructures of N contains at least all the constants of N.

If a set $A \subseteq N$ is such that

1. $f^N[A^{n_f}] \subseteq A$ for every $f \in L_{\text{fun}}$

then there is a unique substructure $M \subseteq N$ with domain A, namely, the structure with the following interpretation

2. $f^M = f^N \upharpoonright A^{n_f}$ (which is a good definition by the assumption on A);
3. $r^M = r^N \cap A^{n_r}$.

It is usual to confuse subsets of N that satisfy 1 with the unuque substructure they support.

It is immediate to verify that the intersection of an arbitrary family of substructures of N is a substructure of N. Therefore, for any given $A \subseteq N$ we may define the substructure of N generated A as the intersection of all substructures of N that contain A. We write $\langle A\rangle_N$. The following easy proposition gives more concrete representation of $\langle A\rangle_N$

La notazione mi sembra la stessa di quella delle dispense di Berarducci. È comunque bene ricordare questi semplici fatti.

1.8 Lemma *The following hold for every* $A \subseteq N$

1. $\langle A\rangle_N = \Big\{t^N \ : \ t \text{ a closed } L(A)\text{-term}\Big\}$

2. $\langle A\rangle_N = \Big\{t^N(a) \ : \ t(x) \text{ an } L\text{-term and } a \in A^{|x|}\Big\}$

3. $\langle A\rangle_N = \bigcup_{n\in\omega} A_n$, *where*

$$A_0 = A$$
$$A_{n+1} = A_n \cup \Big\{f^N(a) \ : \ f \in L_{\rm fun},\ a \in A_n^{n_f}\Big\}. \qquad \square$$

1.9 Example Let L be the language of groups. Let N be a group, which we consider as an L-structure in the natural way. Then the substructures of N are exactly the subgroups of N and $\langle A\rangle_N$ is the group generated by $A \subseteq N$. A similar claim is true when L_F is the signature of vector spaces over some fixed field F. The choice of the language is more or less fixed if we want that the algebraic and the model theoretic notion of substructure coincide. □

5 Formulas

Fix a language L and a set of variables V as in Section 3. A formula is a finite sequence of symbols in $L_{\rm fun} \cup L_{\rm rel} \cup V \cup \{\doteq, \bot, \neg, \vee, \exists\}$. The last set contains the logical symbols that are called respectively

$\doteq$	equality	$\bot$	contradiction	$\neg$	negation
$\vee$	disjunction	$\exists$	existential quantifier.		

Syntactically, $\doteq$ behaves like a binary relation symbol. So, for convenience set $n_{\doteq} = 2$. However $\doteq$ is considered as a logic symbol because its semantic is fixed (it is always interpreted in the diagonal).

The definition below uses the prefix notation which simplifies the proof of the unique legibility lemma. However, in practice we always we use the infix notation: $t \doteq s$, $\varphi \vee \psi$, etc.

1.10 Definition *A formula is any finite sequence is obtained with the following inductive procedure*

0. *if* $r \in L_{\rm rel} \cup \{\doteq\}$ *and* t *is a tuple obtained concatenating* n_r *terms then* $r\,t$ *is a formula. Formulas of this form are called atomic;*

i. *if* φ *e* ψ *are formulas then the following are formulas:* $\bot$, $\neg\,\varphi$, $\vee\,\varphi\,\psi$, *and* $\exists x\,\varphi$, *for any* $x \in V$. □

We use L to denote both the language and the set of formulas. We write $L_{\rm at}$ for the set of atomic formulas and $L_{\rm qf}$ for the set of quantifier-free formulas i.e. formulas where $\exists$ does not occur.

The proof of the following is similar to the analogous lemma for terms.

1.11 Lemma (unique legibility of formulas) *Let a be a sequence of formulas. Suppose a can be obtained both by the concatenation of the formulas $\varphi_1, \ldots, \varphi_n$ or by the concatenation of the formulas $\psi_1, \ldots, \psi_m$. Then $n = m$ and $\varphi_i = \psi_i$.* □

A formula is closed if all its variables occur under the scope of a quantifier. Closed formulas are also called sentences. We will do without a formal definition of *occurs under the scope of a quantifier* which is too lengthy. An example suffices: all occurrences of x are under the scope a quantifiers in the formula $\exists x\, \varphi$. These occurrences are called bonded. The formula $x \doteq y \wedge \exists x\, \varphi$ has free (i.e., not bond) occurrences of x and y.

Let x is a tuple of variables and t is a tuple of terms such that $|x| = |t|$. We write $\varphi[x/t]$ for the formula obtained substituting t for all free occurrences of x, coordinatewise.

We write $\varphi(x)$ to declare that the free variables in the formula φ are all among those of the tuple x. In this case we write $\varphi(t)$ for $\varphi[x/t]$.

We will often use without explicit mention the following useful syntactic decomposition of formulas with parameters.

1.12 Lemma *For every formula $\varphi(x) \in L(A)$ there is a formula $\psi(x\,;z) \in L$ and a tuple of parameters $a \in A^{|z|}$ such that $\varphi(x) = \psi(x\,;a)$.* □

Just as a term $t(x)$ is a name for a function $t(x)^M : M^{|x|} \to M$, a formula $\varphi(x)$ is a name for a subset $\varphi(x)^M \subseteq M^{|x|}$ which we call the subset of M defined by $\varphi(x)$. It is also very common to write $\varphi(M)$ for the set defined by $\varphi(x)$. In general sets of the form $\varphi(M)$ for some $\varphi(x) \in M$ are called definable.

1.13 Definitionof truth *For every formula φ with variables among those of the tuple x we define $\varphi(x)^M$ by induction as follows*

o1. $$(\doteq t\, s)(x)^M = \Big\{a \in M^{|x|} \ :\ t^M(a) = s^M(a)\Big\}$$

o2. $$(r\, t_1 \ldots t_n)(x)^M = \Big\{a \in M^{|x|} \ :\ \langle t_1^M(a), \ldots, t_n^M(a)\rangle \in r^M\Big\}$$

i0. $$\bot(x)^M = \varnothing$$

i1. $$(\neg\xi)(x)^M = M^{|x|} \smallsetminus \xi(x)^M$$

i2. $$(\vee\, \xi\, \psi)(x)^M = \xi(x)^M \cup \psi(x)^M$$

i3. $$(\exists y\, \varphi)(x)^M = \bigcup_{a \in M} (\varphi[y/a])(x)^M$$

Condition i2 assumes that ξ and ψ are uniquely determined by $\vee\, \xi\, \psi$. This is a guaranteed by the unique legibility o formulas, Lemma 1.11. Analogously, o1 e o2 assume Lemma 1.5.

The case when x is the empty tuple is far from trivial. Note that $\varphi(\varnothing)^M$ is a subset of $M^0 = \{\varnothing\}$. Then there are two possibilities either $\{\varnothing\}$ or $\varnothing$. We wil read them as two truth values: True and False, respectively. If $\varphi^M = \{\varnothing\}$ we say that φ is true in M, if $\varphi^M = \varnothing$, we say that φ is false M. We write $M \vDash \varphi$, respectively $M \nvDash \varphi$. Or we may say that M models φ, respectively M does not model φ. It is immediate to verify at

Questa è una notazione usatissima.

$$\varphi(M) = \{a \in M^{|x|} : M \vDash \varphi(a)\}.$$

Note that usually, we say *formula* when, strictly speaking, we mean *pair* that consists of a formula and a tuple of variables. Such pairs are interpreted in definable sets (cfr. Definition 1.13). In fact, if the tuple of variables were not given, the arity of the corresponding set is not determined.

In some contexts we also want to distinguish between two sorts of variables that play different roles. Some are placeholder for parameters, some are used to define a set. In the the first chapters this distinction is only a clue for the reader, in the last chapters it is an essential part of the definitions.

1.14 Definition *A partitioned formula (strictly speaking, we should say a 2-partitioned formula) is a triple $\varphi(x\,;z)$ consisting of a formula and two tuples of variables such that the variables occurring in φ are all among x, z.* □

We use a semicolon to separate the two tuples of variables. Typically, z is the placeholder for parameters and x runs over the elements of the set defined by the formula.

6 Yet more notation

Now we abandon the prefix notation in favor of the infix notation. We also use the following logical connectives as abbreviations

$\top$	stands for	$\neg\bot$	tautology
$\varphi \wedge \psi$	stands for	$\neg[\neg\varphi \vee \neg\psi]$	conjunction
$\varphi \rightarrow \psi$	stands for	$\neg\varphi \vee \psi$	implication
$\varphi \leftrightarrow \psi$	stands for	$[\varphi \rightarrow \psi] \wedge [\psi \rightarrow \varphi]$	bi-implication
$\varphi \nleftrightarrow \psi$	stands for	$\neg[\varphi \leftrightarrow \psi]$	exclusive disjunction
$\forall x\, \varphi$	stands for	$\neg\exists x \neg\, \varphi$	universal quantifier

We agree that $\rightarrow$ e $\leftrightarrow$ bind less than $\wedge$ e $\vee$. Unary connnectives (quantifiers and negation) bind stronger then binarary connectives. For example

$$\exists x\, \varphi \wedge \psi \;\rightarrow\; \neg\xi \vee \vartheta \quad \text{reads as} \quad \Big[[\exists x\, \varphi] \wedge \psi\Big] \;\rightarrow\; \Big[[\neg\xi] \vee \vartheta\Big]$$

We say that $\forall x\, \varphi(x)$ and $\exists x\, \varphi(x)$ are the universal, respectively, existential closure of $\varphi(x)$. We say that $\varphi(x)$ holds in M when its universal closure is true in M. We say that $\varphi(x)$ is consistent in M when its existential closure is true in M.

The semantic of conjunction and disjunction is associative. Then for any finite set of formulas $\{\varphi_i : i \in I\}$ we can write without ambiguities

$$\bigwedge_{i\in I}\varphi_i \qquad\qquad \bigvee_{i\in I}\varphi_i$$

When $x = x_1,\dots,x_n$ is a tuple of variables we write $\exists x\,\varphi$ or $\exists x_1,\dots,x_n\,\varphi$ for $\exists x_1\dots\exists x_n\,\varphi$. With first order sentences we are able to say that $\varphi(M)$ has at least n elements (also, no more than, or exactly n). It is convenient to use the following abbreviations.

$$\exists^{\geq n}x\,\varphi(x) \quad \text{stands for} \quad \exists x_1,\dots,x_n\Big[\bigwedge_{1\leq i\leq n}\varphi(x_i)\ \wedge \bigwedge_{1\leq i<j\leq n} x_i \dot{\neq} x_j\Big].$$

$$\exists^{\leq n}x\,\varphi(x) \quad \text{stands for} \quad \neg\exists^{\geq n+1}x\,\varphi(x)$$

Abbreviazioni molto importanti

$$\exists^{= n}x\,\varphi(x) \quad \text{stands for} \quad \exists^{\geq n}x\,\varphi(x)\ \wedge\ \exists^{\leq n}x\,\varphi(x)$$

1.15 Exercise Let M be an L-structure and let $\psi(x), \varphi(x,y) \in L$. For each of the following conditions, write a sentence true in M exactly when

Raccomando questo esercizio

a. $\psi(M) \in \{\varphi(a,M) : a \in M\}$;

b. $\{\varphi(a,M) : a \in M\}$ contains at least two sets;

c. $\{\varphi(a,M) : a \in M\}$ contains only sets that are pairwise disjoint. □

1.16 Exercise Let M be a structure in a signature that contains a symbol r for a binary relation. Write a sentence φ such that

e magari anche questo.

a. $M \vDash \varphi$ if and only if there is an $A \subseteq M$ such that $r^M \subseteq A \times \neg A$.

Remark: φ assert an asymmetric version of the property below

b. $M \vDash \psi$ if and only if there is an $A \subseteq M$ such that $r^M \subseteq (A \times \neg A) \cup (\neg A \times A)$.

Assume M is a graph, what required in b is equivalent to saying that M is a *bipartite graph*, or equivalently that it has *chromatic number* 2 i.e., we can color the vertices with 2 colors so that no two adjacent vertices share the same color. □