# Composition Structure

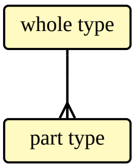
John Cartmell

December 2, 2014

## 1 Entities and Their Parts

An entity model is a system of entity types and relationships and for a reader of a model the most important relationships - those to understand first - are those that represent associations between entities and their parts. These relationships are called *composition relationships* and the term *composition structure* is used to refer to the subsystem consisting solely of entity types and compositions relationships.

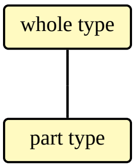
In the sections which follow we describe notations for describing the properties of entities and their composition relationships including cardinality and optionality (the crows foot and dashing of lines), exclusion (the exclusion arc), recursion (looping structures), abstraction (nested boxes), the absolute (diagram root). Almost all this notation applies equally to more general types of relationship but the approach is to introduce the notation in this more restricted setting. The notation is as used in the SSADM method as described, for example, in Richard Barker's book, but with further development including the special treatment of composition relationships, and thereby their foregrounding, and the notation for the absolute.

Composition relationships are shown top-down, which is to say that they are drawn leaving the lower edge of the box representing type of the whole and entering the upper edge of the type representing the part, as here: (a) 

. This fragment signifies that there are *one or more* entities of type *part type* within the whole.

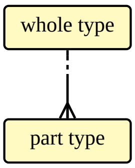
Looking at composition relationships the other way around - bottom up - then they are seen to relate entities with the contexts in which they exist and it is because of this that these are the most important relationships in an entity model - they provide context to entities.

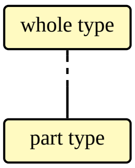
The presence of the crows foot is representative of multiplicity - if the crows foot is present the notation asserts that there may be many parts of type *part type* within each entity of type *whole type*.

If the crows foot is absent, as here: (b) 

then the assertion is that there is *exactly one* entity of type *part type* within the whole.

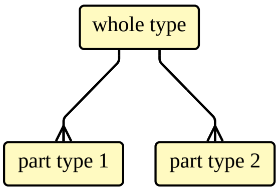
A further distinction is made by use of a half-dashed line to represent the possibility of zero; this gives us

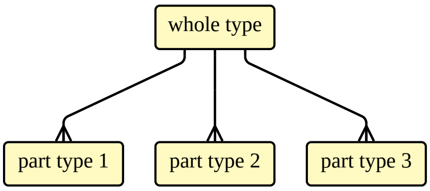
(c) there may be *zero, one or more* entities of *part type* within the whole: 

and (d) there may be *zero or one* entities of type *part type* : 

.

If there are parts of different types, then the structure is shown branching as for example here:

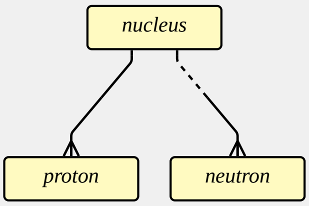
(e) 

or here: (f) 

.

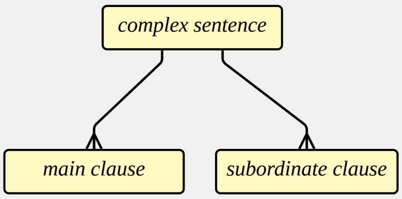
Figures 1, 2 and 3 gives basic examples using the notation. More examples follow as more features of the notation are introduced. If you find yourself disagreeing with these examples - thinking that what they express is contrary to your understanding then it is reasonable to suppose that I will have achieved my aim of showing how the notation works - how it can be used to express precise models and that it is then possible to consider these models, to disagree with them or, better, to refine them.

**Figure 1** – The atomic nucleus



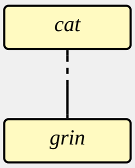
*a nucleus is composed of one or more protons and zero, one or more neutrons*

**Figure 2** – From grammar



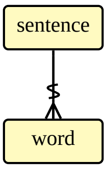
*a complex sentence is composed of exactly one main clause and one or more subordinate clauses*

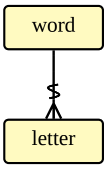
**Figure 3** – From Alice's dream:



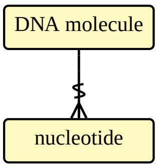
*'Well I've often seen a cat without a grin,' thought Alice; 'but a grin without a cat! It's the most curious thing I ever saw in my life!'*

## 2 Sequential Order

When the parts of a whole have a particular order amongst them then an 'S' shaped symbol (S for sequence) maybe added to the relationship. In this way we can represent a sentence as a sequence of words: (a) 

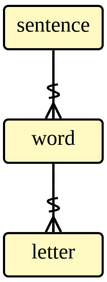
and a word as a sequence of letters: (b) 

Famously, we can represent a DNA molecule (a chromosome?) as a sequence of nucleotides.



## 3 Levels of Decomposition

When a whole is decomposed into parts then very often we find that the part types can be decomposed further. We have seen this already: ‘sentence’s can be broken down into ‘word’s and ‘word’s into ‘letter’s. We then put these together on the diagram the relationships are shown stacked up as follows:

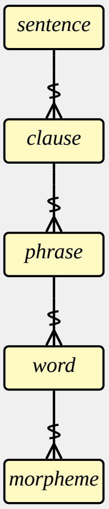


The decomposition of wholes into parts can continue over many levels. Thus some linguists, for example, speak of five levels of decomposition of a sentence. These are shown in figure 4. For another example in a dictionary you see each word with its different senses defined and so a dictionary is composed of words and these, within the context of a dictionary, are composed of a number of senses[1], see figure 5 for the three levels of a dictionary. Figure 6 shows the parts of an atom as understood circa 1930.

[1] For the word ‘sense’ itself we find no less than ten distinct senses given in the Concise Oxford Dictionary. The 7th sense and the one we intend for it here is this:

> *sense n... 7. meaning, way in which a word etc. is to be understood, intelligibility or coherence or possession of a meaning*

**Figure 4** – Some linguists describe five levels of decomposition of a sentence



*(i) a sentence is composed of one or more clauses*

*(ii) a clause is composed of one or more phrases*

*(iii) a phrase is composed of one or more words*

*(iv) a word is composed of one or more morphemes*

**Figure 5** – Dictionary Structure

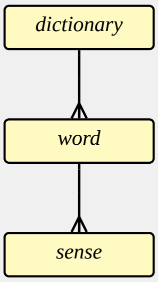
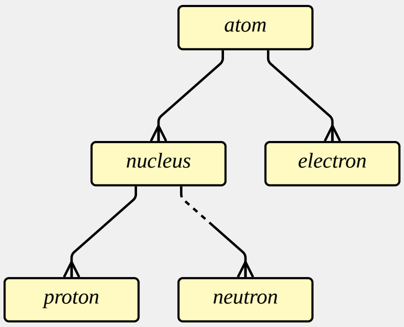


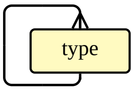
**Figure 6** – From physics



*(i) an atom is composed of exactly one nucleus and one or more electrons*

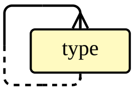
*(ii) a nucleus is composed of one or more protons and zero, one or more neutrons*

## 4 Recursion

When the type of a part is identical to the type of the whole then the relationship must be looped around to renter the top after leaving the bottom: (g) 

. This asserts that every *type* entity is composed of one or more *type* entities... each one of which, therefore, is itself composed of one more *type* entities ... and so on indefinitely. The implied infinity in this model is a concern which is elaborated by Jonathon Swift in rhyme wherein the type in question is the type 'flea':

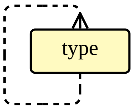
*Big fleas have little fleas,*
*Upon their backs to bite 'em,*
*And little fleas have lesser fleas,*
*and so, ad infinitum.*

In fact, if we are to avoid asserting an infinity of parts of parts of parts then we must specify *zero*, one or more parts within the whole as here: (h) 

. Further, if we are to avoid asserting an infinity of ever bigger conglomerates as suggested in Augustus De Morgan's rejoinder to Swift:

*And the great fleas themselves, in turn, have greater fleas to go on,*
*While these again have greater still, and greater still, and so on*

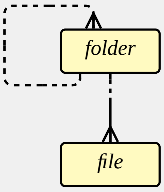
then we must specify that a part is *optionally* part of some other part as here:

(i) 

.

The example in figure 7 illustrates this looping in action; it is further developed in subsequent sections.

**Figure 7** – Describing the file system of a computer drive.

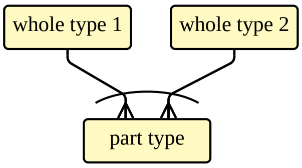


*(i) a folder is composed of zero, one or more folders and zero, one or more files*

## 5 Exclusion Arcs

When an entity of a particular type may be part of different types of whole, but not simultaneously so, then this is asserted by an arc drawn across the lines representing the composition relationships as here:

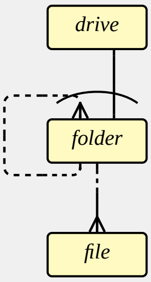
(j)



This arc is called an exclusion relationship. Note that the presence of the arc circumvents the necessity to dash the relationship lines to represent optionality - optionality of relationships from parts to whole is implied.

This now allows us to improve the model of a file system as shown in figure 8 in which we have introduced a drive entity ( i.e a disk drive) to provide context for those folders which do not themselves have folders for context.

**Figure 8** – From the file system of a computer



*(i) a drive has as a part exactly one folder*

*(ii) a folder has as parts zero, one or more folders and zero, one or more files*

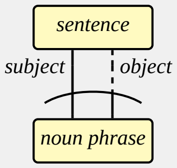
*(iii) every folder is exclusively either a folder that a drive has as a part or a folder which a folder has as a part*

## 6 Describing Roles

The front and rear wheels of a bicycle are parts which play different roles within the whole - arguably the front and back wheels should be modelled as different types of entity for the rear wheel has a sprocket, or at least a housing for one, whilst the front does not. In the case of a child's scooter though or in the case, prior to pedal power, of the velocipede, the whole thing has two identical parts playing different roles within the whole. It is for reasons like this that the role played by a part in a whole may be specified in a model. The role played is described as an annotation on the composition relationship at the 'whole' end of the relationship.

Sentence structure gives an example of this, it is shown in figure 10. There may be multiple noun phrases within a sentence - a simple sentence must have a subject noun phrase and it may have have an object noun phrase. Like the front wheel and the rear wheel of a child's scooter the subject and object noun phrases are in and by themselves indistinguishable.

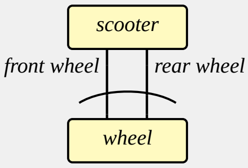
**Figure 9** – First example of naming of relationship roles



*(i) every sentence has a subject which is a noun phrase*

*(ii) a sentence may have an object which is a noun phrase*

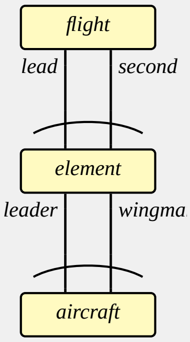
**Figure 10** – Second example of naming of relationship roles



*(i) every scooter has a front wheel which is a wheel*

*(ii) every scooter has a rear wheel which is a wheel*

**Figure 11** – The four-finger flight formation



*(i) every flight has exactly one lead which is an element*

*(ii) every flight has exactly one second which is an element*

*(iii) every element has exactly one leader which is an aircraft*

*(iv) every element has exactly one wingman which is an aircraft*

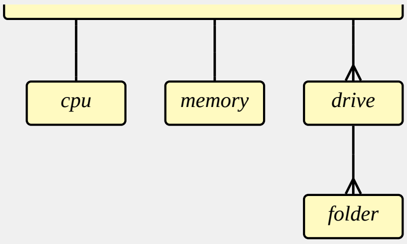
# 7 The Absolute Whole and Other Absolutes

The entirety of a modelling situation is represented by a half box at the upper edge of a diagram. Types whose entities may be free standing - not parts of or dependent on other entities are shown to be part of the entirety - which, relative to the situation, is the whole of everything.

The entirety is the ‘absolute’ whole. It is a principle of logic that when we model we model a specific situation and this establishes an absolute. Fiction operates the same way as logic in this regard.

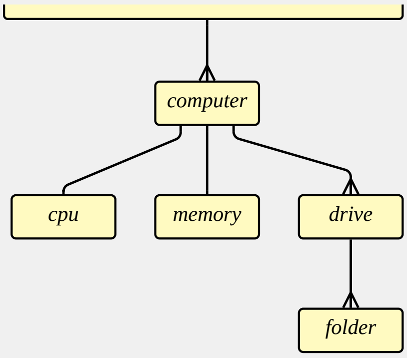
If the modelling situation is that of a single computer the entirety has cpu, drives and memory as parts as shown in figure 13.

**Figure 12** – Model of a single desktop computer



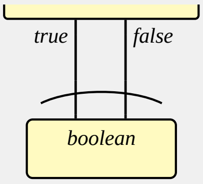
but if the entirety of the modelling situation is a network of computers then the whole has computers as parts which in turn each have drives and memory as shown in figure 14.

**Figure 13** – Model of a network of computers



Similarly, the terms ‘true’ and ‘false’ are not relative terms - they do not rely on context for their interpretation - therefore they are absolutes and can be represented in relationship to the absolute. We will follow the computer science convention of representing them as being of type ‘boolean’ which in common language might be translated as the made up word ‘truthliness’. This leads to the model shown in figure 15 which depicts true and false as absolutes and as entities of type ‘boolean’. Note that from this model we are able to infer the only boolean entities are either the absolute ‘true’ or the absolute ‘false’.

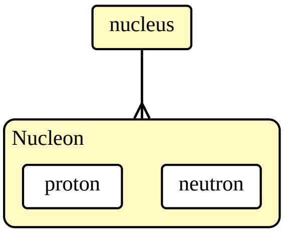
Figure 14 – Model of two absolutes



We might not have said this already: an entity model defines the rules by which entities of any of its types may exist; they do so by declaring the context for their existence i.e. other types of entities and the relationships in which they must be with these entities in order to exist. The uppermost model in figure 10 declares then that every noun phrase is either subject or object of a sentence [2]. Similarly, from the model in figure 15 we are able to infer that the only boolean entities are either the absolute 'true' or the absolute 'false'.

## 8 Use of Generalisations

We gave the example in figure 1 of an atomic nucleus being composed of many protons and neutrons. This can be expressed slightly differently using the generalisation 'nucleon' of the individual terms 'proton' and 'neutron' by stating: an atomic nucleus is composed of one or more nucleons; every nucleon is either a proton or a neutron; (it's pretty much expressed this way in a diagram from the physicist Harald Fritzsch's book - is it?). In an entity model diagram the generalisation type is shown as a box containing the specific types; so figure 1 may be expressed thus:

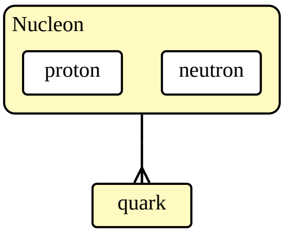


.

Be aware that there is a minor difference between this way of expressing the situation and the form in figure 1: this way, is consistent with a nucleus with one neutron and zero protons - in physics this cannot happen. Equally neither representation expresses that the number of protons must equal the number of electrons.

The same generalisation can be used to express that protons and neutrons are combinations of quarks:

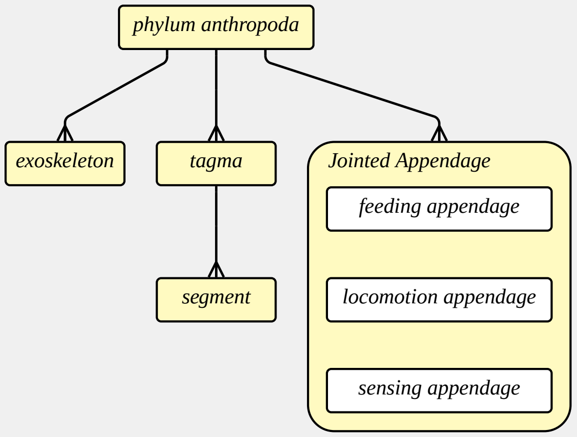
[2] though the model states this please be aware that the model might be misguided! Also, later, we shall we come back to many ambiguities with terms like 'word', and likewise is true for the terms 'noun' and 'noun phase'.



## 9 Two Examples from Biology

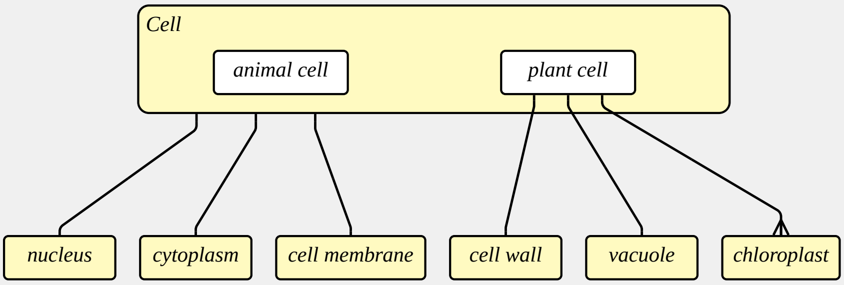
As a further illustration of the notations described so far figures 16 and 17 give examples representing descriptions given in Biology.

**Figure 15** – an elementary description of phylum anthropoda



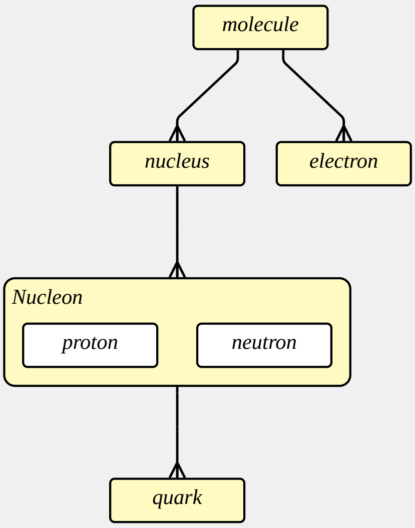
*the characteristics of phylum anthropoda are (i) exoskeleton, (ii) body segments grouped into specialized regions (= tagmata) (iii) jointed appendages variously specialized for feeding, locomotion, sensing*

**Figure 16** – You should know: animal cells have a nucleus, cytoplasm and a cell membrane. Plant cells also have a cell wall, a vacuole and chloroplasts.



## 10 Fundamental Levels of Matter

**Figure 17** – From physicist Harald Fritzsch



*(i) every molecule is composed of one or more nuclei and one or more electrons*

*(ii) every nucleus is composed of one or more nucleons*

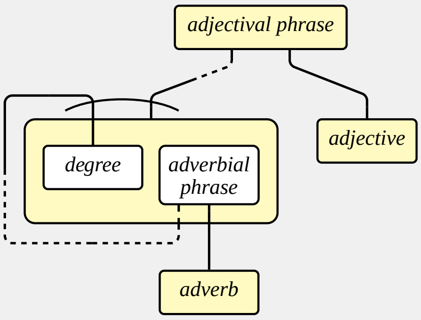
*(iii) every nucleon is either a neutron or a proton*

*(iv) every neutron is composed of one or more quarks*

## 11 An Example From Linguistics - Adjectival Phrases

A variation in the use of the exclusion arc is illustrated in the next example, figure 9, in which the model formally describes a certain part of the structure of English as given in many books of grammar by describing the possible appearances of degree words, adverbs and adjectives in adjectival phrases. In this example the ordering of composition relationships as they leave an entity from its lower edge corresponds to the relative positions of such consituents within an adjectival phrase.

**Figure 18** – Description of an adjectival phrase - an arrangement of degree words, adverbs and an adjective said to be the head of the phrase.



*Adjectives, wherever they appear in sentences as predicates or as qualifiers of nouns may be themselves be qualified by degree words or by adverbs; adverbs themselves may in turn be qualified by degree words.*

Adjectival phrases may appear as the predicates of sentences or as the qualifiers of nouns.

Examples of adjectival phrases and the types of the constitent words are given in table **??**. The table uses abbreviations for the different word types as were introduced in table 1.

| adjectival phrase | type sequence |
|---|---|
| fierce | A |
| very fierce | Deg A |
| fiercely barking | Adv A |
| very fiercely barking | Deg Adv A |

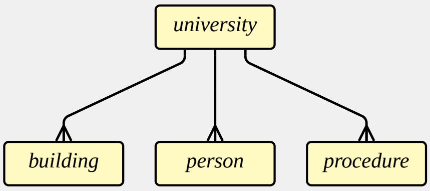
**Table 1** – Example adjectival phrases

The reason for the inclusion of a type 'adverbial phrase' is illustrated by the example *very fiercely barking* in which the degree word 'very' is understood as qualifying the word 'fiercely' which as a combination (an entity) 'very fiercely' then qualifies the word 'barking'. The phrase 'very fiercely' is an example of an 'adverbial phrase' as represented in the model in figure 9. There are futher extended examples of modelling of structure of English in section **??**.

## 12 Gilbert Ryle

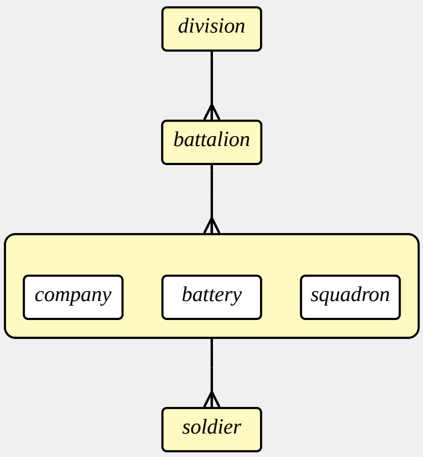
When Gilbert Ryle introduced the phrase 'category mistake' he gave a number of examples. The first example is of a visitor to Oxford. The visitor, upon viewing the colleges and library, reportedly inquired ”But where is the University?”. The visitor's mistake, explains Ryle, is presuming that a University is part of the category “units of physical infrastructure” or some such thing, rather than the category “institutions”, say, which are far more abstract and complex conglomerations of buildings, people, procedures, and so on.

**Figure 19** – A university as an abstract conglomeration of buildings, people and procedures.



Ryle's second example is of a child witnessing the march-past of a division of soldiers. After having had battalions, batteries, squadrons, etc. pointed out, the child asks when is the division going to appear. 'The march-past was not a parade of battalions, batteries, squadrons and a division; it was a parade of the battalions, batteries and squadrons of a division.'.

**Figure 20** – Ryle's March Past



## 13 Linguistics: Modelling Sentence Structure

In this section we give an extended example of entity modelling by demonstrating the modelling of English sentence structure. Our starting point has been the syntactic rules presented in chapters x-z of the book of Brinton: *The Structure of Modern English*[3].

In the study of syntax, linguists seek to understand and express the rules by which some sequences of words can be comprehended as having a structure which is well-formed - that is, we find them grammatical - and some do not. In earlier sections we have spoken about an entity model as representing a theory of what is; we illustrate this now for, as Noam Chomsky wrote in 1957, *a grammar of the language L is a theory of L*[4]. Chomsky says that each grammar is simply a description of a certain set of utterances, namely those that it generates and he illustrates with reference to a part of chemical theory concerned with structurally possible compounds. Such a theory, says Chomsky, might be said to generate all physically possible compounds just as a grammar generates all grammatically 'possible' utterances[5][6].

Different authors have described English by various grammatical systems and with varying and over lapping degrees of success and agreement but each grammar do described is a rule system and what all such rule systems do is to identify and classify relationships between different parts of grammatical sentences; linguists bring to our attention types of entities and accompanying composition relationships which otherwise remain hidden.

Rule systems invariably focus on different word classes i.e. different types of words such as

[3] Brinton, 2000

[4] Syntactic Structures, N. Chomsky, 1957 (page 49)

[5] Syntactic Structures(page 48)

[6] Since Chomsky's 1957 descriptions of phrase structure rules, transformation rules and morphophonemic rules there many different approaches to expressing sentence structure. The models developed in this section cover the same ground as the phrase structure rules of Chomsky which are expressed as a class of grammar know as a context free. All context free grammars lead naturally and mechanically to entity models of parse trees. This section is an illustration of this.

were introduced in section **??**.

The type of each word of an example sentence can be shown by annotating the word entities as follows:

$\boxed{\text{The}}_{\text{Det}}$ $\boxed{\text{young}}_{\text{A}}$ $\boxed{\text{athelete}}_{\text{N}}$ $\boxed{\text{set}}_{\text{V}}$ $\boxed{\text{a}}_{\text{Det}}$ $\boxed{\text{new}}_{\text{A}}$ $\boxed{\text{record}}_{\text{NP}}$ .

With regard to the types of words - many words can be used say as nouns and as verbs and so it is the *use* of the word that is the entity to which a type is assigned.

Figure 21 shows the hierarchically structure of a simple sentence. It shows the composition relationships between its constituents parts. Each of the boxes in figure 21 is an example of a sentence constituent - a single word or a phrase; this is a key entity type for the linguist.

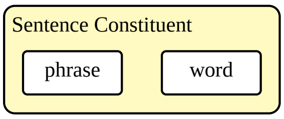
## 13.1 Sentence Constituents

According to Brinton:

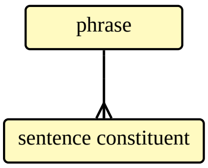
> *The study of syntax is the analysis of the constituent parts of a sentence: their form, positioning, and function. Constituents are the proper subparts of sentences. There are different types of constituents classified by the categories which constitute them; these have different functions and internal structures with elements arranged in a specific way. And they may themselves be complex, containing other constituents. The structure of a sentence is hence hierarchical.*

Brinton notes that not all sequences of words function as constituents. It is the context, he says, which determines whether a particular sequence forms a constituent or not. For example, the sequence of words *beautiful flowers* is a constituent in *I received beautiful flowers for my birthday* but not in *Though they are beautiful, flowers cause me to sneeze*. The sequence *the house on the hill* is a constituent in one reading of the ambiguous sentence *I bought the house on the hill*, but not in the other; it is a constituent in the sense 'I bought the house which is on the hill', but not in the sense 'I bought the house while standing on the hill'.

We can represent sentence constituent as an entity type generalising 'phrase' and 'word':

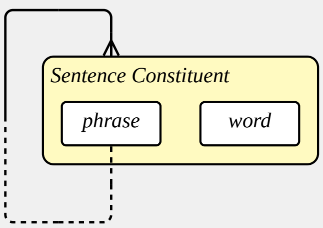


and show phrases as sequences of words:



this leads us to a single diagram as shown in figure 22.

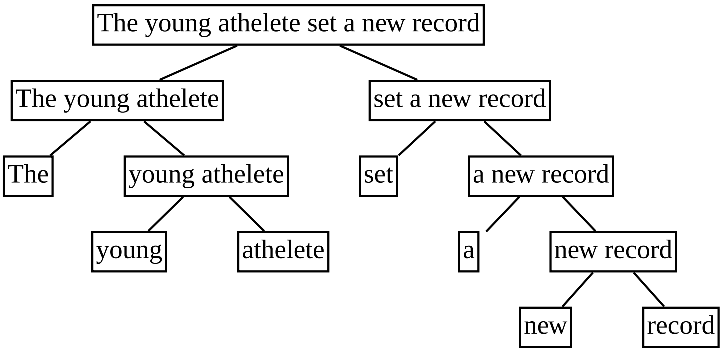
**Figure 21** – From Understanding Language



*(i) a sentence constituent is either a word or a phrase*

*(ii) a phrase is composed of one or more sentence constituents*

**Figure 22** – Constituent Parts of an Example Sentence[8]- phrases are shown in relationship to their constituent parts - referred to as 'sentence constituents'.



## 13.2 Types of Phrase

Our example sentence is of a type that is often termed a 'simple' sentence and consists of a subject followed by a predicate as shown in figure 23. Other examples of the same type, subject followed by predicate, are shown in table 2; each subject in the table can equally combine with any of the given predicates to form a well-formed sentence. Though ten sentences are presented by combining different subjects with different predicates we can generate 100 different sentences[7]. What we cannot do is pair any two subjects to produce a well-formed sentence nor any two predicates. This illustrates that subject phrases and predicate phrases are of distinct types. These are the types 'noun phrase' and 'verb phrase' respectively. These types of phrase can appear in other contexts within sentences but for now they appear as subject and predicate respectively.

This simple rule:

$$\mathrm{S} \longrightarrow \mathrm{NP\ VP} \tag{1}$$

using some self evident abbreviations is of a type sometimes referred to as a production,

[8] Reproduced from Fowler, "Understanding Language", page 100.

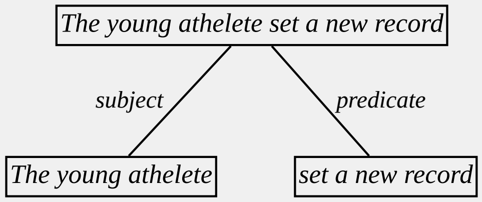
[7] But only the proviso that some verbs need be inflected to match their subjects - for example 'We is for sale' needs become 'We are for sale'.

|  | Subject | Predicate |
|---|---|---|
| 1. | We | had arrived at the wrong house. |
| 2. | The man | could open the door. |
| 3. | Truffles | are a kind of fungus. |
| 4. | The house on the corner | is for sale. |
| 5. | She | went to see who was at the door. |
| 6. | The rain and fog | dissipated. |
| 7. | There | are five cats in the hall. |
| 8. | The film that I saw last night | was depressing. |
| 9. | It | is raining. |
| 10. | It | is possible that Alfred will know the answer. |

**Table 2** – Ten simple sentences used by Brinton to illustrate the subject predicate distinction.

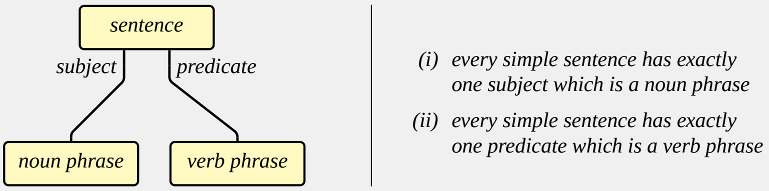
aggregates of which are used to express grammars technically referred to as context free grammars. This is a formal expression of the fact that a sentence(S) may be constructed as a noun phrase(NP) followed by a verb phrase(VP).

**Figure 23** – Example sentence showing the decomposition into subject phrase and a predicate phrase.



What such a rule (1) does not do is to give a name to the role played by the noun phrase (NP) in the sentence – that of subject – and nor does it give a name – predicate – to the role played by the verb phrase (VP). In Entity Modelling we have a notation for just such matters and accordingly we can represent the subject-predicate rule inclusive of role names by the entity model fragment shown in figure 24.

**Figure 24** – Sentence as subject and predicate

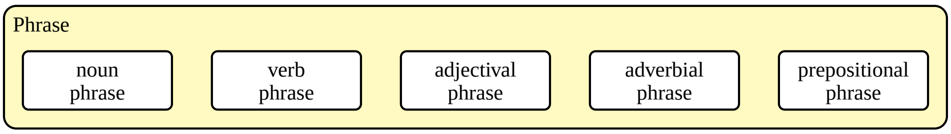

sentence
subject
predicate
noun phrase
verb phrase
(i) every simple sentence has exactly one subject which is a noun phrase
(ii) every simple sentence has exactly one predicate which is a verb phrase

The entity type ‘phrase’ of figure 22 is very general - it covers entities such as whole sentences, and the noun phrases and verb phrases shown in the subject and predicate columns of table 2. Linguists recognise these different phrase types and, just as for the different types of words, phrases may be substituted in sentences by phrases of like type without effecting the acceptability of the sentence. Table 3 shows some of the major distinctions within phrase types.

| phrase type | abbreviation | examples |
|---|---|---|
| whole sentence | S | *the young athelete set a new record* |
| noun phase | NP | *the young athelete, a new record* |
| verb phrase | VP | *set a new record* |
| adjectival phrase | AP | |
| prepositional phase | PP | |

**Table 3** – Types of phrase and abbreviations used.

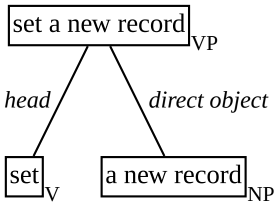
This leads us to be able to refine the entity type ‘phrase’ in figure 22 by a generalisation as follows:



In the next two sections we continue the process of refinement - we describe the composition relationships that are possible between the different phrase types of a sentence. In this way we use the entity modelling notation as a meta-language in which we describe how different types of sentence constituent can be combined into grammatical sentences and in so doing present a description of a set of rules for sentence construction. There is not, in linguistic theory, an unequivocal set of rules to follow - the rules we choose to model follow close to those presented by Brinton.

## 13.3 The grammar of the Verb Phrase

We have examples of verb phrases in the right hand column of table 2. They are quite varied in their structure. Here is a typical analysis of a verb phrase, in this case for the predicate of figure 23:

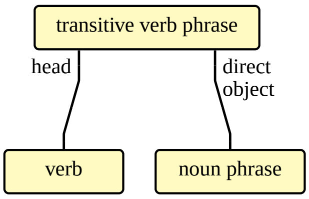


Generally, a verb phrase contains a verb which is said to be the head of the phrase. In this example the verb is complemented by a direct object which is of type 'noun phrase'. Note that we can be sure that the phrase 'a new record' is of type 'noun phrase' because we can construct a sentence with it as the subject. In fact it can replace any of the subjects in table 2 and new sentences result and, providing that verbs are conjugated appropriately, the results are well-formed though admittedly they are in some cases very unusual.

According to this analysis the phrase 'set a new record' is a verb phrase constructed as a 'verb' followed by a 'noun phrase' and this is one general rule of a number of such for the construction of verb phrases. Formally expressed the rule is:

$$\mathrm{VP} \longrightarrow \mathrm{V\ NP} \tag{2}$$

In an entity model we can represent this like this:

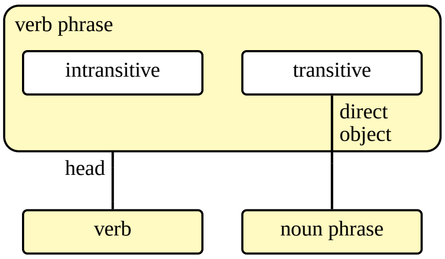


(3)

In other cases a verb phase consists just of a verb alone and has no constituent noun phrase. Only certain verbs that can be used in this way and these are said to be intransitive. Thus 'he cries' is an example in which 'he' is the subject and 'cries' is a predicate verb phrase consisting solely of the verb. Formally:

$$\mathrm{VP} \longrightarrow \mathrm{V} \tag{4}$$

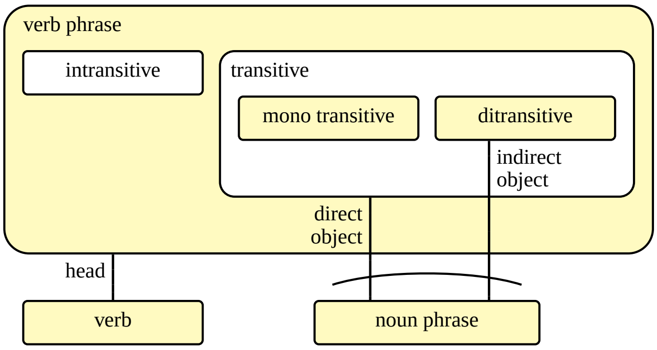
The entity model equivalent is simply this:
These fragments can be combined like this:



Still other verb phrases have two obhjects a direct object of the action and an indirect object. Formally:

$$\mathrm{VP} \longrightarrow \mathrm{V\ NP\ NP} \tag{5}$$

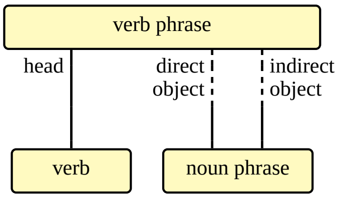
This leads to a model combining representations of rules (2), (4) and (5)like this:



It is worth saying at this point as an illustration of the choices that are open to an entity modeller in any modelling situation that sometimes the two rules above will be represented together as a single combined rule in which parentheses are used to indicate optionality of the noun phrase:

$$\mathrm{VP} \longrightarrow \mathrm{V\ (NP)} \tag{6}$$

Likewise the entity modeller may dispense with the types representing the transitive and intransitive cases and represent the combined situation like this:



In total, Brinton describes seven different constructions for verb phrase. These constructions are both summarised and illustrated in table 4. Alternatively the rules summarised in table 4 can be expressed by combining all the different rules into the entity model of a verb phrase shown in figure 25 in which types ‘noun phrase’, and ‘prepositional phrase’ are yet to be described but ‘adjectival phrase’ was described earlier in section xxx.

**Table 4** – Examples of different types of verb phrase.

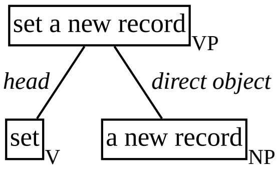
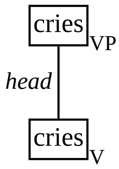
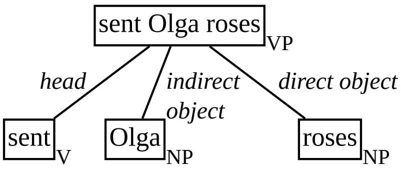
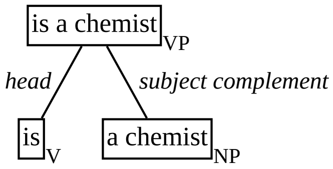
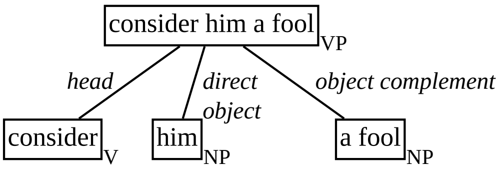
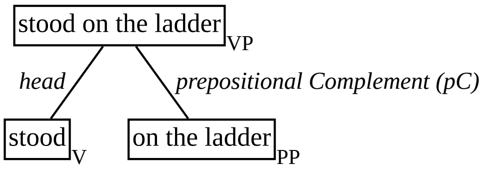
| Type of verb phrase/Rule | Example |
|---|---|
| mono transitive<br>VP ⟶ V NP |  set a new record VP<br>head<br>direct object<br>set V<br>a new record NP |
| intransitive<br>VP ⟶ V |  |
| ditransitive<br>VP ⟶ V NP NP |  sent Olga roses VP<br>head<br>indirect object<br>direct object<br>sent V<br>Olga NP<br>roses NP |
| copulative<br>VP ⟶ V {NP,AP,PP} |  is a chemist VP<br>head<br>subject complement<br>is V<br>a chemist NP |
| complex transitive<br>VP ⟶ V NP {NP,AP,PP} |  consider him a fool VP<br>head<br>direct object<br>object complement<br>consider V<br>him NP<br>a fool NP |
| mono prepositional<br>VP ⟶ V PP |  stood on the ladder VP<br>head<br>prepositional Complement (pC)<br>stood V<br>on the ladder PP |

**Table 4** – (continued)

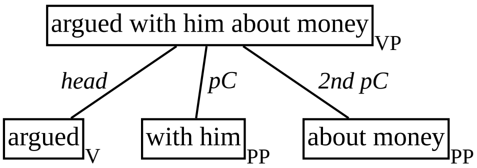
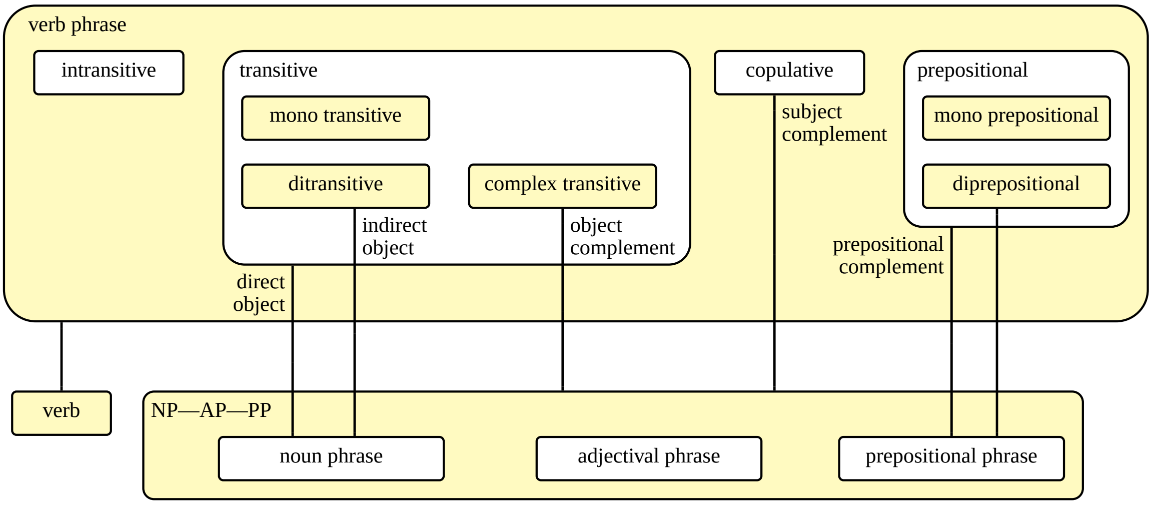
| | |
|---|---|
| dipreprositional<br>VP $\longrightarrow$ V PP PP |   argued with him about money VP<br>head<br>pC<br>2nd pC<br>argued V<br>with him PP<br>about money PP |

**Figure 25** – The grammar of the Verb Phrase

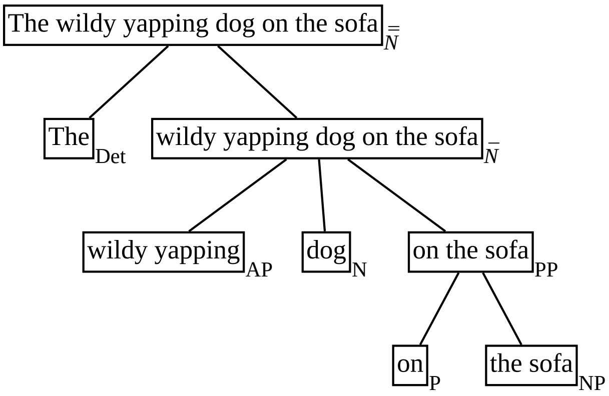


## 13.4 The Grammar of the Noun Phrase

There are three main types of Noun Phrase and paradoxically two of these types consist of phrases which are degenerate in the sense that they consist of single words - Pronouns and Proper Names because words of these types are grammatically acceptable as the subjects of sentences. For simplicity of the modelling we will henceforth consider pronouns and proper names as types of noun phrase. The other type of noun phrase takes a number of forms but does actually contain a noun and for technical reasons is known as type ‘N-bar-bar’[9].

By way of illustration for the model of N-bar-bar consider the phrase *The wildly yapping dog on the sofa*. We can annotate the types of word and constituent phrase are as follows:

[9] It may be abbreviated $\overline{\overline{N}}$



In our earlier example the noun phrase which was the subject of our sentence is analysed like this:

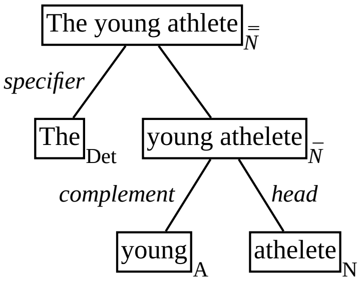
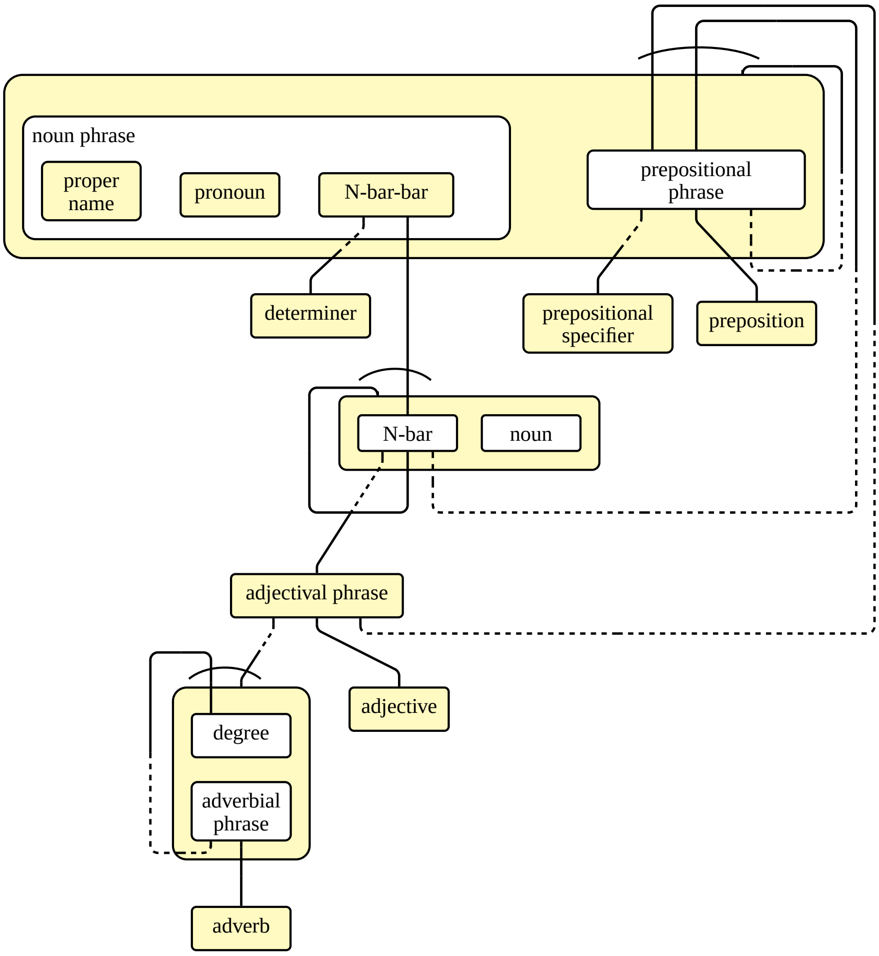


**Figure 26** – The grammar of the Noun Phrase



## 13.5 Model of a Simple Sentence

The final step in constructing a model of a simple sentence is achieved by putting together the models represented in figures 24,25 and 26 and rearranging the diagram to suit. The resulting model is shown in figure 27.

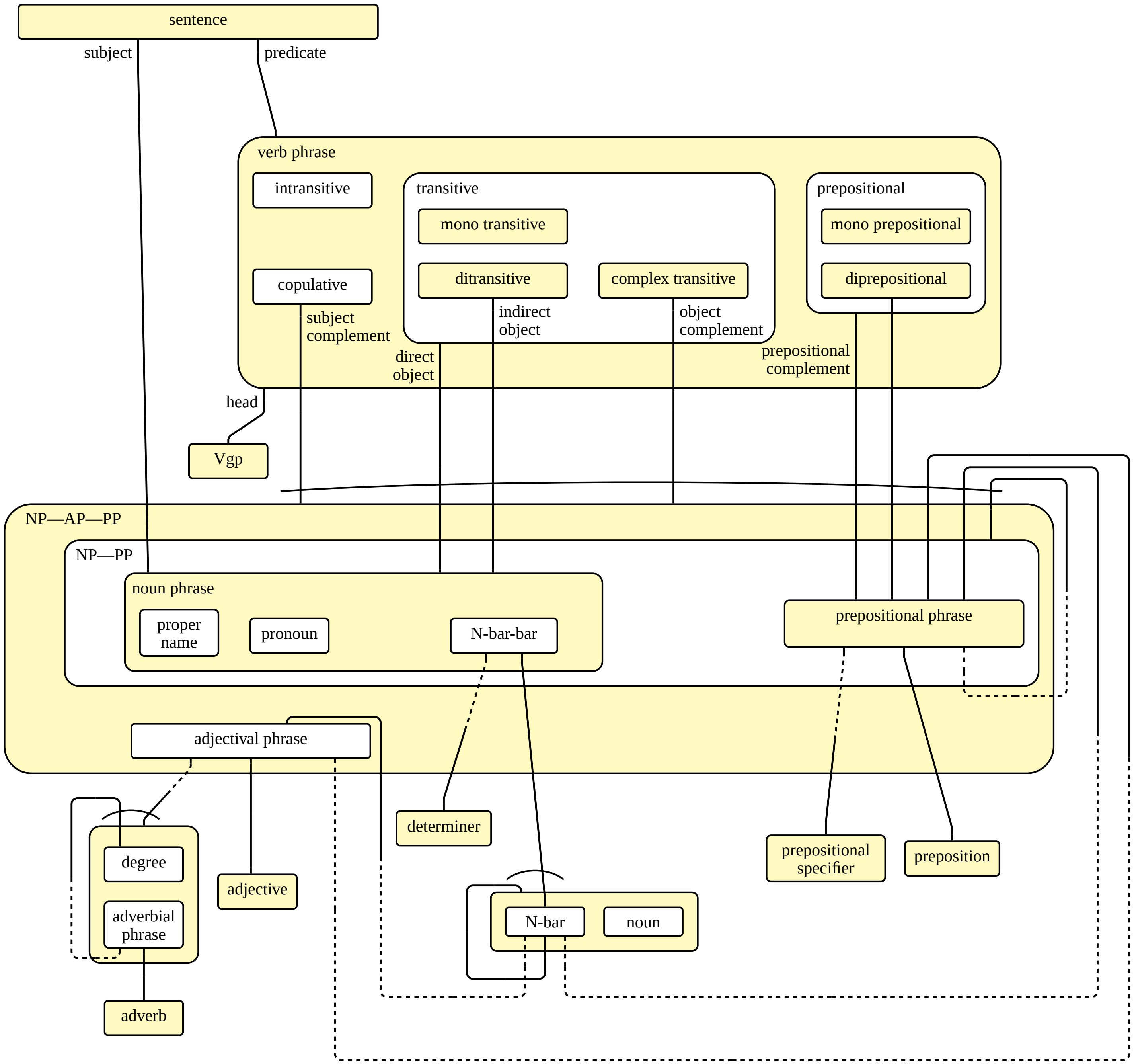
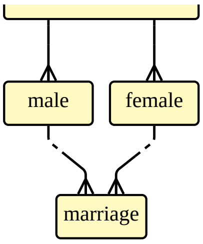


**Figure 27** – Structure of Simple Sentences - Based on Phrase Structure Rules given in Brinton.

# 14 Network and Matrix Structure

## 14.1 Marriages and Directed Graphs

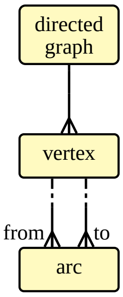
If we model the marriage entity then we make it double dependent on person entity and since at the time of writing same sex marriage is not legal in my part of the world I can model it as a dependent on a male entity on the one hand, and a female on the other:



In this model, unlike in those of figures 10 and 11 to which it is superficially similar, the subordinate entity, ‘marriage’, has no exclusion arc between its dependencies – which is to say that the model shows each subordinate entity ‘marriage’ to be dependent on two superordinate entities - those shown as ‘male’ and ‘female’. For the first time we see that the modelling notation is not constrained to situations where every entity has at most one other that it is dependent on. It is common practice to describe situations that do have this characteristic as *hierarchical* - it is the defining principle of hierarchy that in a hierarchical system every entity within the system is subordinate to at most one other entity.

In other instances of matrix structure the branch types (by which I mean for example the types *row* and *column* above) are not distinct types. We’ll see this in the next example - the type in question being *arc*.

Mathematicians define and use various notions each which abstract the idea of a network of points and connecting lines independently of how or whether physically realised. They define such abstract notions using the language of sets and relations; they use the term ‘graph’ for the most abstract concept of such a network and they variously use the terms ‘point’, ‘node’ or ‘vertex’ for the things connected; generally they use the term ‘edge’, ‘directed edge’, ‘arc’ or ‘arrow’ for the connections. There is not a single terminology and so we have to plump for one of several available; our definitions will clarify the choices made. For an entity modeller, and therefore for a database designer, the most straight forward of the graph notions is that of a ‘directed graph’.

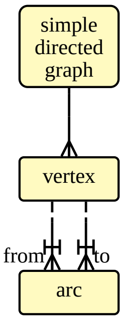


## 14.2 Simple Graphs and Identifying Relationships

One of the restrictions often places on graph structures is that there be at most one edge between any two vertices. The term ‘simple graph’ is used for graphs having this property and having the further property that vertices are not linked to themselves. Now the fact that there is at most one edge between any two verticies means that an edge can be uniquely identified by identifying the vertices that it connects.

The ‘arc’ type as used in models of simple directed graphs is an example of one each whose instances may be uniquely identified by relationships with other entities. Such relationships in entity modelling terminology are said to be *identifying relationships* though this terminology lacks rigour for it is the set of them which are identifying not the individual relationships.

Identifying relationships are represented with a mark across the relationship having the form of the letter ‘I’. Adding these marks to the model of a directed graph we get the model of a simple directed graph:



The terms *network* and *matrix* are commonly used in contrast to the term hierarchical to refer to arrangements of entities not constrained to be hierarchical; for example in organisational structure the term *matrix management* is used in situations were different dimensions are managed by different management hierarchies and in which individuals therefore have multiple reporting lines. The term hierarchical is etymologically derived from Greek *sacred ruler* and emerged in its modern sense via its use in medieval times in relation to the church organisation.

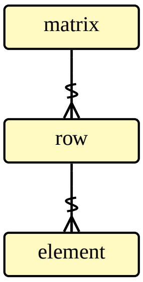
## 14.3 Matrices and Tabular Displays

In mathematics, a matrix is a rectangular array of numeric elements, for example

$$\begin{pmatrix} 23 & 15 & 29 \\ 31 & 6 & 9 \\ -1 & 8 & 17 \end{pmatrix}$$

is a matrix having 3 rows and 3 columns. Though, as with all matrix structure, this is essentially 2 dimensional, the content can be communicated linearly either row by row ([23,15,29], [31,6,9] and [-1,8,17]) or column by column ([23,31,-1],[15,6,8] and [29,9,17]).

As you see with this example, each element of any matrix is both part of a row and part of a column and so, as with the ‘marriage’ entity, the ‘element’ entity is modelled as a subordinate to two others of different types as shown in figure 28. The two ways of communicating the matrix correspond to the two branches of this entity model. The row by row communication is described by this model:



and the column by column communication by this one:

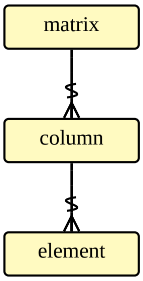
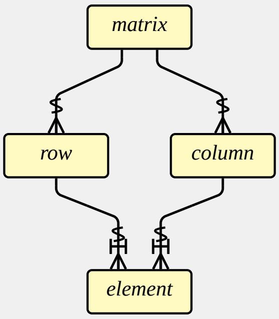


**Figure 28** – A mathematical matrix



*(i) every matrix has one or more rows*

*(ii) every matrix has one or more columns*

*(iii) every matrix has one or more elements*

*(iv) every element is part of row and part of a column*

There is a similar shape to the models representing the structure[10] of a rectangular table of data. An example is given in figure 29. In the HTML language, and in other computer markup languages, data tables are communicated row by row rather than column by column.

In other tabular displays the rows or columns of a table, or both, may be grouped together to represent some grouping of the subjects. The structure then has different branches that are hierarchical and joined at the detail level into the recognisable shape of the 2 dimensional matrix structure. One such is illustrated in figure 30.

[10] I am distinguishing here between the structure of the tabular display from the structure of the subject entitites

**Figure 29** – Model of the structure of a tabular display. Note that what is modelled is the structure of the display rather than the structure of the subject entities though, having said which, there is an important meta-relationship between the two - for of necessity there is a meta-relationship between the structure of a system of subject entities and the structure of a medium through which details of such a system may be communicated or visualised.

| Team sheet | | |
|---|---|---|
| Goalkeeper | GK | Paul Robinson |
| Defenders | LB | Lucus Radebe |
| | DC | Michael Duberry |
| | DC | Dominic Matteo |
| | RB | Didier Domi |
| Midfielders | MC | David Batty |
| | MC | Eirik Bakke |
| | MC | Jody Morris |
| Forward | FW | Jamie McMaster |
| Strikers | ST | Alan Smith |
| | ST | Mark Viduka |

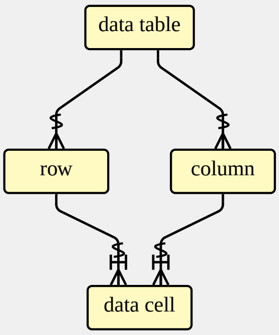
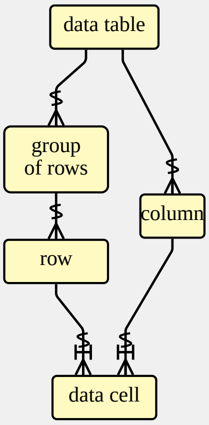


**Figure 30** – Tabular structure in which rows are grouped - part of dataset from U.S. Bureau of the Census.[11]

| Rank by Population of the 100 Largest Urban Places | | | | | |
|---|---|---|---|---|---|
| State | Urban Place | 1960 | 1970 | 1980 | 1990 |
| ARIZONA | Phoenix | - | - | - | 85 |
| | Tucson | 61 | 23 | 22 | 15 |
| CALIFORNIA | Fresno | 24 | 27 | 29 | 36 |
| | Long Beech | - | - | - | - |
| | Los Angeles | - | 90 | 88 | 93 |
| | Oakland | 83 | - | - | - |
| ⋮ | | | | | |
| TEXAS | Dallas | 43 | 44 | 36 | 44 |
| | Los Angeles | - | 90 | 88 | 93 |
| | Oakland | 83 | - | - | - |
| VIRGINIA | Virginia Beech | 2 | 2 | 2 | 3 |
| WASHINGTON | Chicago | 2 | 2 | 2 | 3 |
| WISCIONSIN | Milwauke | 2 | 2 | 2 | 3 |



## 15 Communicating Hierarchy - XML

Whereas, matrix structure, as considered in the previous section, is essentially 2 dimensional, hierarchically structured information may be flattened into a linear i.e. 1 dimensional structure in which nesting of detail represents the hierarchy. XML is a language designed for this

[11] http:/www.census.gov/population/www/documentation/twps0027/tab01.txt, Internet Release date: June 15, 1998

purpose.

In the XML language communication of a hierarchically structured entity, each type $X$ of entity is enclosed by its own parenthesis in the form of character sequences `<X>` and `</X>` giving us the conventions `<matrix>` and `</matrix>` for start and end matrix, `<row>` and `</row>` for start and end row, and `<element>` and `</element>` for start and end element; this message that follows these conventions[12]:

```
<matrix>
   <row>
      <element>23</element>
      <element>15</element>
      <element>29</element>
   </row>
   <row>
      <element>31</element>
      <element>6</element>
      <element>9</element>
   </row>
   <row>
      <element>-1</element>
      <element>8</element>
      <element>17</element>
   </row>
</matrix>
```

[12] I have written the message on many lines and indented to make more readable