Designing a parsing grammar

Atro Voutilainen

July 8, 1996

0.1 Finite-state Intersection Grammar

Koskenniemi (Coling'90)

Koskenniemi, Tapanainen and Voutilainen (Coling'92)

Voutilainen and Tapanainen (EACL'93)

Voutilainen (PhD'94)

Voutilainen (EACL'95)

Derives from

- Two-level model: rule formalism (Koskenniemi '83)

- Constraint Grammar: linguistic representation (Karlsson, Voutilainen, Heikkilä and Anttila 1995)

0.2 Related work

Roche (Coling'92)

Chanod and Tapanainen (EACL'95)

0.3 Basics

Parsing means eliminating illegitimate sentence readings (proposed by the morphological analyser and certain simple intermediary programs).

Monotonicity: The correct syntactic analysis is present in the input, so nothing is added during the parsing operation itself.

Technically, parsing is the intersection of two kinds of finite-state automata: the sentence automaton and the grammar automata.

The parse consists of those sentence readings that all of the grammar automata accept.

The grammarian does not directly operate with automata; rather, the sentence representation as well as the grammar rules are expressed as (extended) regular expressions that can be transformed into finite-state automata.

0.4 Representation of sentences

One of the two inputs of the finite-state parser is the morphologically analysed sentence, represented as a regular expression. Here is a simplified representation of the sentence *The program runs*:

```
[@@
the <*> <Def> DET CENTRAL ART SG/PL @DN>

<word-boundary>

[[program <SVO> V SUBJUNCTIVE VFIN @+FMAINV] |
 [program <SVO> V IMP VFIN @+FMAINV] |
 [program <SVO> V INF @-FMAINV] |
 [program <SVO> V PRES -SG3 VFIN @+FMAINV] |
 [program N NOM SG [@SUBJ | @OBJ]]]

<word-boundary>

[[run <SVC/A> <SVO> V PRES SG3 VFIN @+FMAINV] |
 [run N NOM PL [@SUBJ | @OBJ]]]

<word-boundary>

FULLSTOP @@]
```

'[' and ']' indicate grouping; '|' indicates disjunction. In all, this concise representation contains 12 sentence-level readings or traversals, for instance the following one:

```
@@ the <*> <Def> DET CENTRAL ART SG/PL @DN> <word-boundary>
program <SVO> V IMP VFIN @+FMAINV <word-boundary>
run <SVC/A> <SVO> V PRES SG3 VFIN @+FMAINV <word-boundary>
FULLSTOP @@
```

Here *the* is analysed as a determiner, *program* is analysed as a verb in the imperative, and *runs* is analysed as a verb in the present tense.

The task of the finite-state parser is to discard inappropriate readings like these. In the analysis of the above sentence, a successful grammar will discard all illegitimate sentence readings.

Only the following reading will be accepted by a realistic grammar here:

```
@@ the <*> <Def> DET CENTRAL ART SG/PL @DN> <word-boundary>
program N NOM SG <word-boundary>
run <SVC/A> <SVO> V PRES SG3 VFIN @+FMAINV <word-boundary>
FULLSTOP @@
```

Before parsing proper starts, the regular expression that represents the sentence is transformed into a finite-state automaton.

0.5 Rule formalism

The finite-state grammar is an unordered set of rules that constrain the combinations of tags and words.

Before parsing, the rules, represented as regular expressions, are compiled into finite-state automata.

The task of the finite-state parser is to find the logical intersection of the sentence automaton and all rule automata.

The grammar can employ **constants**, **predicates** and **implication rules**.

- A constant is simply a shorthand for a regular expression. For instance, it may be expedient to define a constant for nominals:

  ```
  Nominal = [N | ABBR | NUM | PRON] ;
  ```

- Predicates – parameterised constants – can be used in or as rules, and the grammarian can define new predicates as the need arises.

  ```
  nowhere(X) = \X ;
  ```

  X is a regular expression, and the backslash '\' denotes any sequence of tokens not containing occurrences of its argument. The *nowhere(X)* predicate can be used as a grammar rule, e.g.

  ```
  nowhere(DET Word-Boundary VFIN) ;
  ```

- The implication rule is of the form

```
X =>
LC1 _ RC1,
LC2 _ RC2,
...
LCn _ RCn;
```

  where *X*, *LC1*, *RC1*, *LC2* etc. are regular expressions.

  The rule reads: 'X *is legitimate only if it occurs in context* LC1 _ RC1 *or in context* LC2 _ RC2 .. *or in context* LCn _ RCn'. At least one of the alternative context conditions has to be satisfied for sentence readings containing the *X* part; otherwise that sentence reading is rejected.

Grammar rules are compiled into finite-state automata before parsing.

0.6 Parsing algorithms

Simple algorithm: intersect the rule automata with the sentence automaton successively.

Faster algorithms:

- before parsing, intersect some rules with each other, so the number of rule automata becomes smaller

- apply the most effective rule automata first

- apply some rule automata in parallel

- apply only those rule automata that are relevant for the sentence (activation alphabet)

0.7 Representation of sentences

- *Clause boundary representation.* In the finite-state framework, also word boundaries are described as ambiguous:
  - '@@' indicates a sentence boundary. Presently, sentence boundaries are determined during the preprocessing stage.
  - '@' indicates that the adjacent words (or punctuation marks) belong to the same finite clause.
  - '@/' signals the end of one finite clause and the beginning of another, e.g.

    `This is the man @/ who I saw yesterday.`

  - Centre-embeddings need a different treatment. The symbols '@<' and '@>' indicate the beginning and the end of a finite clause that interrupts its matrix clause:

    `The man @< who I saw yesterday @> is here.`

    The clause

    `The man .. is here`

    is interrupted by another clause:

    `.. who I saw yesterday ..`

    Both clauses have e.g. subjects and finite verbs of their own, and this kind of markup makes it easier for the grammarian to indicate these syntactic relations.

It is possible to introduce all types of ambiguity at once.

The rule formalism is not sensitive to the ‘type’ of description we are doing. Any rule can refer to any type of descriptor (morphological, syntactic, and clause-boundary tags).

Example:

```
(@@
john <Proper> N NOM SG (:OR @SUBJ @OBJ @I-OBJ @<P)
(:OR @ @/ @< @>)
(:OR
  (likes N NOM PL (:OR @SUBJ @OBJ @I-OBJ @<P))
  (likes V PRES SG3 @+FMAINV))
(:OR @ @/ @< @>)
coffee N NOM SG (:OR @SUBJ @OBJ @I-OBJ @<P)
(:OR @ @/ @< @>)
FULLSTOP @@)
```

In all, this representation contains $4*4*5*4*4*4 = 5,120$ distinct sentence readings.

Depending on the distinctiveness of the adopted grammatical representation, an ambiguous representation of a real-text sentence may easily contain more than 10^{100} sentence readings.

The task of the grammar(ian) can perhaps be easiest understood by visualising the input as a (very great) number of distinct sentence readings: one or more of the readings are appropriate, and the rest are inappropriate.

Here, the first one is the intended analysis of the above sample sentence, and the second is one of the remaining suspect 5,119 analyses.

```
                                                @@
John      <Proper> N NOM SG   @SUBJ              @
like      V PRES SG3          @+FMAINV           @
coffee    N NOM SG            @OBJ               @
FULLSTOP                                        @@

                                                @@
John      <Proper> N NOM SG   @<P                @/
like      V PRES SG3          @+FMAINV           @
coffee    N NOM SG            @OBJ               @
FULLSTOP                                        @@
```

A comprehensive finite-state grammar checks the contextual licence of each distributional category in the sentence. If all distributional categories are contextually legitimate, the sentence reading is structurally well-formed. If some of the descriptors violate some of the rule automata, the sentence reading is discarded.

- *Reference to phrase-like units.* In the finite-state environment, regular expressions are available. Thus one can for instance define constants that can then be used in rules; cf. the following one:

```
. = \[@ | @@ | @/ | @< | @>];
```

The *dot* '.' accepts anything which does not contain word boundary symbols; i.e. anything within a morphological reading. It is quite useful; cf. the following expression which accepts any number of determiners or premodifiers followed by a nominal head:

```
>head = @AN> | @DN> | @NN> | @GN> | @QN> ;
NPhead = @SUBJ | @OBJ | @I-OBJ | @<P ;

[. >head @]* . NPhead
```

The first two are definitions, which are then used in the third expression. '|' indicates alternatives, '[' and ']' are for grouping, and the Kleene star '*' accepts any number of its arguments.

The most useful kind of rule appears to be the **implication rule**. Consider the following simplified rule about the distribution of the subject in a finite clause:

```
# Defs
>..< =  [. | @]*;
@<>  =  @< [. | @ | @/]* @>;
..   =  [@<> | >..<]*;

...  =  \@@;


# Rules
Subject =>
          _ .. FinVerbChain,
FinAux .. _ .. NonFinMainVerb ... QUESTION;
```

The *dotdot* '..' accepts anything within the same clause, centre-embedded clauses included. The *dotdotdot* '...' accepts anything but the sentence boundary symbol.

With the implication rule, statements about linear order and dependency can be expressed economically and in a positive fashion.

It is also possible to write expressly negative, prohibitive rules e.g. by using the *nowhere(X)* predicate, but this is not always necessary because the **consequence** of an implication rule is what negative rules seek explicitly to effect.

0.8 A surface-syntactic grammatical representation

An extension of the representation used in English Constraint Grammar.

Next, we sketch a more informative grammatical representation that has already been applied to over 200,000 words of running text from various genres and to some 2,000 test sentences from Quirk & al.

Together, these annotated corpora serve as a first approximation of the inventory of syntactic structures in written English, and they can be conveniently used in the validation of the new grammar currently under development.

SYNTACTIC TAGS IN OUTLINE

The following is a schematic representation of the syntactic tags:

`SUBJ`	`Subject`
`F-SUBJ`	`Formal subject`
`OBJ`	`Object`
`F-OBJ`	`Formal object`
`I-OBJ`	`Indirect object`
`SC`	`Subject complement`
`OC`	`Object complement`
`P<<`	`Preposition complement`
`>>P`	`Complement of deferred preposition`
`APP`	`Apposition`

`@>A`	`AD-A, head follows`
`@A<`	`AD-A, head precedes`
`@>N`	`Determiner or premodifier`
`@>P`	`Premodifier of a PP`
`N<`	`Postdeterminer or postmodifier`
`ADVL`	`Adverbial`
`ADVL/N<`	`Adverbial or postmodifier`
`@CC`	`Coordinator`
`@CS`	`Subordinator`
`AUX`	`Auxiliary`
`MV`	`Main verb`
`MAINC`	`Main clause`
`mainc`	`Independent nonfinite clause`
`n-head`	`Independent nominal construction`
`a-head`	`Independent adverbial construction`

This list represents the tags in a somewhat abstract fashion. The grammatical representation also employs a few notational conventions.

1. An explicit difference is made between two kinds of clause: finite and nonfinite.
2. A finite clause typically contains

3. a finite verb chain, one or more in length, one of which is a finite verb, and
4. a varying number of nominal and adverbial constructs.

  Verbs and nominal heads in a finite clause are indicated with a tag written in the upper case, e.g. *Sam/@SUBJ was/@MV a/@>N man/@SC.*

- A verb chain in a nonfinite clause, on the other hand, contains only nonfinite verbs. Verbs and nominal heads in a nonfinite clause are indicated with a tag written in the lower case, e.g. *To/@aux be/@mv or/@CC not/@ADVL to/@aux be/@mv.*
- While a distinction is made between the upper and the lower case in the description of verbs and nominal heads, no such distinction is made in the description of other categories, which are all furnished with tags in the upper case, cf. *or/@CC not/@ADVL* above.

The grammatical representation accounts both for the internal structure of clausal units and for their function in their matrix clause.

Usually, all tags start with the '@' sign, but those tags that indicate the function of a clausal unit rather than its internal structure *end* with the '@' sign.

The function tag of a clause is attached to the main verb of the clause, so main verbs always get *two* tags, one indicating the word's main verb status in the clause, the other indicating the clause's function is its context. An example:

```
                         @@
How     @ADVL             @
to      @aux              @
write   @mv     mainc@    @
books   @obj              @
.                        @@
```

Here *write* is a main verb in a nonfinite clause (*@mv*), and the nonfinite clause itself acts as an independent nonfinite clause (*mainc@*).

0.9 Sample analyses

Next, the new grammatical representation is illustrated with some concrete examples. Note, however, that most morphological tags are omitted here; only a part-of-speech tag is given. Likewise, the base forms are omitted; only running-text wordforms are given. Consider the following analysis:

```
                                             @@
smoking     PCP1       @mv     SUBJ@          @
cigarettes  N          @obj                   @
inspires    V          @MV     MAINC@         @
the         DET        @>N                    @
fat         A          @>N                    @
butcher's   N          @>N                    @
wife        N          @OBJ                   @
and         CC         @CC                    @
daughters   N          @OBJ                   @
.           FULLSTOP                         @@
```

The boundary markers ‘@@’, ‘@’, ‘@/’, ‘@<’ and ‘@>’ indicate a sentence boundary, a plain word boundary, an iterative clause boundary, the beginning, and the end, of a centre-embedding, respectively.

All words get a function tag. *Smoking* is a main verb in a nonfinite construction (hence the lower case tag *@mv*); *cigarette* is an object in a nonfinite construction; *inspires* is a main verb in a finite construction (hence the upper case tag *@MV*), and so on.

Main verbs also get a second tag that indicates the function of the verbal construction. The nonfinite verbal construction *Smoking cigarettes* is a subject in a finite clause, hence the tag *SUBJ@* for *Smoking*. The finite clause is a main clause, hence the tag *MAINC@* for *inspires*, the main verb of the finite clause.

The syntactic tags avoid telling what can be easily inferred from the context. For instance, the tag *@>N* indicates that the word is a determiner or a premodifier of a nominal.

The grammatical representation avoids introducing structurally unresolvable distinctions. Consider the analysis of *fat*. The syntactic tag *@>N* indicates that the word is a premodifier of a nominal, and the head is to the right – it is the nominal head of the noun phrase, or it is another intervening nominal premodifier. In other words, the tag *@>N* accounts for both of the following bracketings:

```
[[fat butcher's] wife]
[[fat [butcher's wife]
```

Coordination often introduces unresolvable ambiguities. On the basis of structural criteria, it is impossible to determine, for instance, whether *fat* modifies the coordinated *daughters* as well in *the fat butcher's wife and daughters*. In some cases, our grammatical representation keeps also this kind of ambiguity covert, which helps to keep the amount of ambiguity within reasonable limits.

In the present representation, the syntactic function is carried by the coordinates rather than by the coordinator – hence the object function tags on both *wife* and *daughters* rather than on *and*. An alternative convention would be the functional labelling of the conjunction. The difference appears to be merely notational.

A distinction is made between finite and nonfinite constructions. As shown above, nonfiniteness is expressed with lower case tags, and finite (as well as other) constructions are expressed with upper case tags.

This kind of split-up makes the grammarian's task easier. For instance, the grammarian might wish to state that a finite clause contains at most one potentially coordinated subject. Now if potential subjects in nonfinite clauses could not be treated separately, it would be more difficult to express the grammar statement as a rule because extra checks for the existence of subjects of nonfinite constructions would have to be incorporated in the rule as well, at a considerable cost to transparency and perhaps also to generality. Witness the following sample analysis:

```
                                             @@
Henry     N          @SUBJ                    @
dislikes  V          @MV     MAINC@           @
her       PRON       @subj                    @
leaving   PCP1       @mv     OBJ@             @
so        ADV        @>A                      @
early     ADV        @ADVL                    @
.         FULLSTOP                           @@
```

Apparently, there are two simplex subjects in the same clause; what makes them acceptable is that they have different verbal regents: *Henry* is a subject in a finite clause, with *dislikes* as the main verb, while *her* occurs in a nonfinite clausal construction, with *leaving* as the main verb.

With regard to the description of *so early* in the above sentence, the present representation makes no commitments as to whether the adverbial attaches to *dislikes* or *leaving*. In other words, there is no separate tag for adverbials in nonfinite constructions. Adverbial attachment is often structurally unresolvable; hence our description of these distinctions is rather modest.

Also finite clauses can have a nominal function. Consider the following sample.

```
                                       @@
What        PRON     @SUBJ              @
makes       V        @MV     SUBJ@      @
them        PRON     @OBJ               @
acceptable  A        @OC                @/
is          V        @MV     MAINC@     @/
that        CS       @CS                @
they        PRON     @SUBJ              @
have        V        @MV     SC@        @
different   A        @>N                @
verbal      A        @>N                @
regents     N        @OBJ               @
.           FULLSTOP                   @@
```

Here *What makes them acceptable* acts as a subject in a finite clause, and *that they have different verbal regents* acts as a subject complement. Clauses in a dependent role are always subordinate clauses that typically have a more fixed word order than main clauses. Thus clause-function tags like *SC@* can also be used for stating word order restrictions.

Another advantage of clause-function tags is that restricting the distribution of clauses becomes more straightforward. If, for instance, a clause is described as a postmodifying clause, then it has to follow something to postmodify; if a clause is described as a subject, then it should also have a predicate, and so on.

Also less frequent or 'basic' word orders can be conveniently accounted for with the present representation. For instance, in the following sentence there is a 'deferred' preposition; here the preposition complement is to the left of the preposition.

```
                                                @@
What              PRON      @>>P                 @
are               V         @AUX                 @
you               PRON      @SUBJ                @
talking           PCP1      @MV       MAINC@     @
about <Deferred>  PREP      @ADVL                @
?                 QUESTION                      @@
```

Here *@>>P* for *What* indicates that a deferred preposition is to be found in the right-hand context, and the feature *<Deferred>* indicates that *about* has no complement in the right-hand context: either the complement is to the left, as above, or it is missing altogether, as in

```
                                            @@
```

```
This            PRON     @SUBJ           @
is              V        @MV    MAINC@  @
the             DET      @>N             @
house           N        @SC             @/
she             PRON     @SUBJ           @
was             V        @AUX            @
looking         PCP1     @MV    N<@      @
for <Deferred>  PREP     @ADVL           @
.               FULLSTOP                @@
```

Ellipsis and coordination often co-occur. For instance, if finite clauses are coordinated, the verb is often left out from the non-first coordinates:

```
                                    @@
Pushkin    N        @SUBJ            @
was        V        @MV    MAINC@    @
Russia's   N        @>N              @
greatest   A        @>N              @
poet       N        @SC              @/
,          COMMA                     @
and        CC       @CC              @
Tolstoy    N        @SUBJ            @
her        PRON     @>N              @
greatest   A        @>N              @
novelist   N        @SC              @
.          FULLSTOP                 @@
```

Here, *and Tolstoy her greatest novelist* is granted a clause status, as indicated by the presence of the iterative clause boundary marker '@/'.

Note that clausal constructions without a main verb do not get a function tag because at present the clause function tag is attached to the main verb. If the ellipsis co-occurs with coordination, then the presence of the coordinator in the beginning of the elliptical construction (i.e. to the right of the iterative clause boundary marker '@/') may be a sufficient clue to the function tag: it is to the left, in the first coordinate.

Verbless clauses also occur in simplex constructions. Consider the following real-text example:

```
                                         @@
Providing       PCP1   @mv    ADVL@   @<
the             DET    @>N            @
pin             N      @SUBJ          @
has             V      @AUX           @
been            V      @AUX           @
fully           ADV    @ADVL          @
inserted        V      @MV    obj@    @
into            PREP   @ADVL          @
the             DET    @>N            @
connect         PCP1   @>N            @
rod             N      @P<<           @>
,               COMMA                 @
final           A      @>N            @
centralization  N      @SUBJ          @
can             V      @AUX           @
,               COMMA                 @
if              CS     @CS            @
necessary       A      @sc            @
,               COMMA                 @
```

```
be                V        @AUX              @
done              PCP2     @MV     MAINC@    @
on                PREP     @ADVL             @
a                 DET      @>N               @
press             N        @P<<              @
using             PCP1     @mv     ADVL@     @
the               DET      @>N               @
support           N        @>N               @
stop              N        @>N               @
button            N        @obj              @
and               CC       @CC               @
driver            N        @obj              @
.                 FULLSTOP                  @@
```

In the analysis of *if necessary*, there is a subject complement tag for *necessary*. Subject complements typically occur in clauses; clauses in general are assigned a syntactic function in our description; here, however, no such analysis is given due to the lack of a main verb. Nevertheless, in this type of verbless clause there is a lexical marker in the beginning: a subordinating conjunction or a *WH* word, and from this we can imply that the verbless clause functions as an adverbial.

Next, consider the description of prepositional phrases. In general, the present grammar tries to distinguish here between the adverbial function (*@ADVL*) and the postmodifier function (*@N<*). In the following sentence, the distinction is straightforward to make in some cases, and problematic in one:

```
                                  @@
Somebody   PRON @SUBJ              @
with       PREP @N<                @
a          DET  @>N                @
telescope  N    @P<<               @
saw        V    @MV   MAINC@ @
with       PREP @ADVL              @
difficulty N    @P<<               @
the        DET  @>N                @
man        N    @OBJ               @
of         PREP @N<                @
honor      N    @P<<               @
with       PREP @ADVL/N<           @
the        DET  @>N                @
binoculars N    @P<<               @
.          FULLSTOP               @@
```

The phrase *with difficulty* is an unambiguous adverbial, because it is directly preceded by a verb, and verbs do not take postmodifiers.

Likewise, *with a telescope* and *of honor* are unambiguously postmodifiers: the former because postnominal prepositional phrases without a verb in the left-hand context are postmodifiers; the latter because a postnominal *of*-phrase is always a postmodifier unless the left-hand context contains a member of a limited class of verbs like 'consist' and 'accuse' which take an *of*-phrase as a complement.

Here *with the binoculars* is a problem case: generally postnominal prepositional phrases with a verb in the left-hand context are ambiguous due to the postmodifier and adverbial functions. Furthermore, several such ambiguous prepositional phrases can occur in a clause at once, so in combination they can produce quite many grammatically acceptable analyses for a sentence. To avoid this uncomfortable situation, an underspecific tag has been introduced: a prepositional phrase is described unambiguously as *@ADVL/N<* if it occurs in a context legitimate for adverbials and postmodifiers. That is, all other functions of prepositional phrases are disallowed in this context (with the exception of *of*-phrases). In all other contexts *@ADVL/N<* is disallowed. This solution may appear clumsy, e.g. a new tag is introduced for the purpose, but its advantage is that description can take full benefit of the unambiguous 'easy' cases without paying the penalty of unmanageable ambiguity as a price for the extra information.

Unsurprisingly, even the present grammatical representation leaves some ambiguities structurally unresolvable. For instance, some coordination phenomena are still somewhat problematic, cf. the following structurally unresolvable ambiguity due to the preposition complement and object analyses:

```
                                      @@
They          PRON  @SUBJ              @
established   V     @MV   MAINC@       @
networks      N     @OBJ               @
of            PREP  @N<                @
state         N     @P<<               @
and           CC    @CC                @
local         A     @>N                @
societies     N    [@OBJ | @P<<]       @
.             FULLSTOP                @@
```

0.10 Formalising the grammar of subjects

The implication rule is useful for expressing distributional generalisations. A construction is contextually legitimate if it occurs in a context listed in an implication rule about the construction.

The task of the grammarian, then, is to determine (i) the distributional categories and (ii) their necessary contexts.

As a starting point, we may use a comprehensive descriptive grammar. Let us sketch a description of subjects, as given in Quirk & al.

The present grammatical representation distinguishes between various kinds of subject. Their form can vary. Roughly speaking, a subject can be a simple noun phrase, a nonfinite clause, or even a finite clause. Also the constructions in which subjects can occur are of two main kinds: finite and nonfinite clauses.

Here we shall only be concerned with the most typical kind of subject: the noun phrase subject in finite clauses. This type of subject has the most variable distribution of all types of subject, so describing the other types of subject will be even more straightforward than what is the case with this type.

Noun phrase subjects in finite clauses are marked with the tag *@SUBJ* that is attached to the head of the noun phrase (and, in the case of coordination, to the head of each coordinate noun phrase). For instance, consider *boys* in the following analysis:

```
                                @@
all   DET   @>N                  @
the   DET   @>N                  @
boys  N     @SUBJ                @
will  V     @AUX                 @
be    V     @MV       MAINC@     @
there ADV   @ADVL                @
.     FULLSTOP                  @@
```

The use of upper case in the tag indicates that the construction serves a function in a finite clause, and the existence of '@' at the beginning of the tag indicates that the functional category is a phrase rather than a clause. If we need more specific information e.g. about the part of speech of the subject, we can of course consult the morphological features. For instance, a noun acting as a subject in a finite clause can be expressed as the following regular expression:

```
N . @SUBJ
```

Note in particular that we need not be concerned about the internal structure of a category whose distribution we are trying to express, because a comprehensive grammar will in any case account for the distribution of e.g. determiners and modifiers with other rules. What has been said once in a grammar need not be repeated elsewhere.

The grammarian's other task is to determine the distribution of the construction. Again, a natural starting point is a comprehensive grammar, and perhaps extensive corpora as well. The type of subjects we are concerned with occur e.g. in declarative clauses with 'normal' word order, for instance:

```
                                 @@
sam    N      @SUBJ               @
was    V      @MV    MAINC@       @
a      DET    @>N                 @
man    N      @SC                 @
.      FULLSTOP                  @@
```

In this kind of clause, the subject is followed by the main verb of the finite clause. Main verbs of this kind are labelled as *@MV* in our syntax, so the generalisation can be expressed as the following expression:

```
@SUBJ .. @MV
```

As we saw above, '..' is a constant that accepts anything within the same clause, centre-embeddings included. So this expression accepts not only the sentence *Sam was a man* but also sentences like

```
                         @@
the    DET   @>N          @
man    N     @SUBJ        @<
who    PRON  @SUBJ        @
came   V     @MV   N<@    @
here   ADV   @ADVL        @/
when   ADV   @ADVL        @
he     PRON  @SUBJ        @
was    V     @MV   ADVL@  @
on     PREP  @ADVL        @
duty   N     @P<<         @
last   DET   @>N          @
time   N     @ADVL        @>
is     V     @MV   MAINC@ @
her    PRON  @>N          @
father N     @SC          @
.     FULLSTOP           @@
```

Here *man*, which is the subject, is followed by *is*, the main verb of the clause, and the regular expression recognises them as parts of the same clause although there is an intervening centre-embedded clause.

A declarative clause may also have other word orders. For instance, the clause may start with an adverbial or a 'formal subject' like *there*, continue with the finite verb chain, and end with the subject, e.g.

```
                          @@
here ADV   @ADVL           @
is   V     @MV   MAINC@    @
my   PRON  @>N             @
bus  N     @SUBJ           @
.    FULLSTOP             @@
```

```
                                @@
there    ADV  @F-SUBJ            @
used=to  V    @AUX               @
be       V    @MV      MAINC@    @
a        DET  @>N                @
school   N    @SUBJ              @
on       PREP @ADVL/N<           @
the      DET  @>N                @
island   N    @P<<               @
.        FULLSTOP               @@
```

This construction can be expressed with the following regular expression:

```
[@ADVL | @F-SUBJ] .. @MV .. @SUBJ
```

A subject can occur in other constructions as well. We will not attempt an exhaustive listing, let us only illustrate with interrogative clauses. Here are some examples and the corresponding regular expressions ('...' accepts anything save a sentence boundary):

```
Can Sally be so cruel?
@AUX .. @SUBJ .. @MV ... QUESTIONMARK

Why is Sally so cruel?
WH .. @MV .. @SUBJ ... QUESTIONMARK

Is Sally so cruel?
be . MV .. @SUBJ ... QUESTIONMARK
```

Now we have seen how to express the constructions where a given distributional category can occur. The main point is to keep the formalisations sufficiently general. It suffices to explicitly represent the necessary contextual conditions; the rest of the (intervening) context can be represented in less detail, e.g. by using the *dotdot* constant. A little underspecificity in one rule is not harmful as long as the missing information is given somewhere else.

Our next move is to combine the distributional category and its contexts into an implication rule. Consider the skeleton of an implication rule:

```
X =>

LC1 _ RC1,
LC2 _ RC2,
LC3 _ RC3,
.
.
.
LCn _ RCn;
```

We replace X with the regular expression that corresponds to the distributional category (*@SUBJ* in this case) and in the place of the context conditions *LC1* _ *RC1* to *LCn* _ *RCn*, we list the regular expressions that correspond to the constructions in which the distributional category is legitimate. There is one change: in the place of the X part, an underscore is inserted. Here is an implication rule based on the above generalisations:

```
                    @SUBJ =>
                            _ .. @MV,
[@ADVL | @F-SUBJ] .. @MV .. _,
                    @AUX .. _ .. @MV ... QUESTIONMARK,
               WH .. @MV .. _ ... QUESTIONMARK,
                be . @MV .. _ ... QUESTIONMARK;
```

Describing other distributions for the category, e.g. the subject in elliptical clauses, would merely mean adding new contexts to the rule, or making some of the existing contexts more permissive.

0.11 Excerpts from the English grammar

This section presents some substantial examples from various grammatical constructions, starting from the description of the simple noun phrase.

0.11.1 Determiners

Quirk & al. (1985) provide a positional classification of determiners: while determiners in general can be followed by a coordination, premodifier or a nominal head[1], there are specific constraints on the co-occurrence of determiners. Predeterminers such as *all* are mutually exclusive and can be followed by other types of determiner; central determiners, e.g. *the*, are also mutually exclusive and can be followed by postdeterminers; a postdeterminer like *other* and *additional*, on the other hand, can be followed by another postdeterminer. For instance,

```
all the other additional boxes
```

is acceptable, while

```
* the those all boxes
```

is not: the constraint of mutual exclusivity as well as the right-hand context of central determiners is violated.

For formalising these statements as finite-state rules, there are several alternatives. One solution would be to formulate one

[1] In the case of the determiner *no*, also an *ing*–form is legitimate as a main verb in a nonfinite construction

general, somewhat underspecific rule, and worry about the details elsewhere. Consider the following grammar fragment:

```
# Defs

Ad-A = . @>A ;
NominalHead = . [@SUBJ | @F-SUBJ |
                 @OBJ | @F-OBJ | @I-OBJ |
                 @SC | @OC | @subj |
                 @f-subj | @obj | @f-obj |
                 @i-obj | @sc | @oc |
                 @APP | @ADVL | @>>P |
                 @P<< | @n-head]
              . ;
Participle = . [PCP1 | PCP2] . ;
Nominal = . [A | N | NUM |
             PRON | Participle | ABBR]
          . ;

..NominalHead = [[@ Ad-A]* @ NominalHead] ;
Premodifier = Nominal @>N ;

..Premodifier = [[@ Ad-A]* @ Premodifier] ;
```

```
# Rules

DET . @>N =>
            _ @ . @CC,
            _ @ . DET,
            _ ..Premodifier,
            _ ..NominalHead,
       no . _ @ . PCP1 . @mv;

nowhere([CENTRAL | POST] . @ . [PRE | CENTRAL]) ;
```

Here, the constants used in the rules are listed to illustrate how an increasing degree of detail can be hidden behind a suggestive name. Readability is not the only motivation for using constants and predicates: shorthands should also be defined for commonly used expressions. If the expression needs correction later, the modification has to be made once only. In the rest of this chapter, we will leave most of the definitions implicit.

The above implication is a general rule that allows any kind of determiner after a determiner, thus also violations of the positional restrictions. These positional restrictions are expressed as a separate rule, a *nowhere()* predicate that explicitly forbids expressions given as the argument.

Of course the positional restrictions can also be expressed entirely with positive implication rules, for instance:

```
Predeterminer =>
                _ @ Coordinator,
                _ @ CentralDeterminer,
                _ @ Postdeterminer,
                _ ..Premodifier,
                _ ..NominalHead;
```

```
Central/Postdeterminer =>
                         _ @ Coordinator,
                         _ @ Postdeterminer,
                         _ ..Premodifier,
                         _ ..NominalHead,
                    no . _ @ . PCP1 . @mv;
```

There are also determiners that follow their head: *enough* does so optionally (it can also be a postdeterminer), but for *galore*, this position is compulsory

```
    DET . @N< =>
NominalHead @ . _ ;
```

Some determiners are sensitive to number. Singular determiners such as *a* take only a singular head, and plural determiners like *those* take only a plural head. Thus,

```
a man
those men
```

```
* a men
* those man
```

There is a special case, namely measure noun phrases (Quirk & al. 1985: 758), where a singular determiner can take as head a plural noun phrase head, e.g.

```
We spent a happy three months there.
```

The measure phrase is notionally regarded as a single unit, so this is a special case of notional concord, typically involving a quantifier as part of the phrase.

Here is the formulation:

```
SingularDeterminer =>
                        _ ..SingularNominalHead,
                        _ ..MeasurePhrase ;

PluralDeterminer =>
                      _ ..PluralNominalHead ;
```

Note how these rules constrain not only the distribution of determiners but of several other constructs as well. For instance, a typical determiner rule is also an implicit rule about clause boundaries: because a determiner of a nominal in the right-hand context is always followed by a plain word boundary, the clause and sentence boundaries are constrained to occur elsewhere.

0.11.2 Premodifiers

Our description of noun phrases is extremely shallow; the functional syntactic codes distinguish only between determiners, premodifiers, heads and postmodifiers. No explicit difference is made between definite descriptions, compounds, names, titles etc; witness for instance the following examples:

```
white        A ABS                           @>N       @
stone        N NOM SG                        @>N       @
wall         N NOM SG                        @n-head   @

Mr.          <Title> ABBR NOM SG             @>N       @
Peter        <Proper> N NOM SG               @>N       @
Smith        N NOM SG                        @n-head   @

the          <Def> DET CENTRAL ART SG/PL @>N           @
word         N NOM SG                        @>N       @
"            DQUOTE                                    @
data         N NOM SG/PL                     @n-head   @
"            DQUOTE                                    @

interesting  PCP1                            @>N       @
story        N NOM PL                        @n-head   @

red          A ABS                           @>N       @
herring      N NOM SG                        @n-head   @
```

Note, however, that some of the 'angle-bracket features' like $<$*Title*$>$ are useful clues if the more exact relation of the word to its head is to be determined.

Regarding the mapping from (morphological) form to (syntactic) function, generally only non-plurals can serve the premodifier function. Sometimes, however, an occasional compound can contain a plural premodifier, for instance:

```
systems analyst
consumer goods people
financial-services industry
```

Here the issue of optimising both recall and precision becomes relevant: if the mapping from a plural to premodifier function is severely restricted, we are in danger of missing some actual uses because of the occasionality of compounds of this type. On the other hand, if no difference is made between typical non-plural premodifiers and the less typical plural ones, we are likely to get uncomfortably many parses, i.e. lower precision.

A solution is to allow a free mapping to the premodifier function for all plural (as well as non-plural) nominals in the lexicon. If multiple syntactic analyses result, a ranking mechanism can be used: a penalty could be assigned to all sentence readings with plural premodifiers, unless the compound in question is a previously attested, more or less conventionalised one like *systems analyst*. In this way, also occasional uses get a chance to become recognised as such – which maximises recall – while in the case of remaining ambiguity, the more conventional analyses are preferred, which will maximise precision.

There are some constraints on the linear order of premodifiers. Generally, a premodifier can only be followed by another, potentially coordinated, premodifier or a nominal head. There is an exception to this general rule, however: a premodifying adjective can precede an indefinite article if the adjective is immediately preceded by any of the following qualifiers: *too*, *so*, *as* or *how*, e.g.

```
It was too boring a book to read.
```

Although in general premodifiers do not have postmodifiers of their own, a premodifying adjective can be immediately followed by the postmodifying qualifier *enough*, e.g.

```
It was an interesting enough book.
```

There are also some part-of-speech specific restrictions on the co-occurrence of premodifiers: closest to the noun phrase head are noun premodifiers; numeral and adjectival premodifiers typically have a more distant position. For instance,

```
three new car manuals
* three car new manuals
```

There are some exceptions, however. Firstly, the rule does not apply to genitival premodifiers; for instance:

```
the gardener's beautiful daughter
the beautiful gardener's daughter
```

Another, more problematic exception concerns the possible form of a premodifier. In the present description, each premodifier is described as a single-word unit. Sometimes, however, it seems necessary to assume a more complex structure; witness the following example taken from Quirk & al.:

```
speech and language impaired children
```

Superficially, the restriction on the part of speech of premodifiers is violated: the premodifying noun *language* is followed by the premodifying adjective *impaired*. However, if we view the noun phrase as a more structured unit, e.g.

```
[A [N speech and language] impaired] [N children]
```

the restriction is not violated: here *speech and language* modifies *impaired*, i.e. *impaired* is not a premodifier at this level of representation, and, secondly, *speech and language impaired* as a whole forms an adjectival phrase that appropriately acts as a premodifier of a following noun, *children* in this case.

Our description of premodifiers is very shallow: no dependencies between successive premodifiers are postulated. A more ambitious description would assign a more complex structure, e.g.

```
speech    N  @>A   @>MOD @
and       CC @CC         @
language  N  @>A   @>MOD @
impaired  A  @>N   @HEAD @
children  N  @SUBJ       @
```

Here *speech* and *language* would be described as nouns (*N*) that premodify an adjective (*@>A*) that itself is a premodifier (*@>MOD*), and *impaired* would be described as an adjective (*A*) that premodifies a noun (*@>N*) that is a noun phrase head (*@HEAD*).

Our system, on the other hand, is content with the following flat description (note that word boundaries have been ignored in these examples):

```
speech    N  @>N   @
and       CC @CC   @
language  N  @>N   @
impaired  A  @>N   @
children  N  @SUBJ @
```

In brief: tags certainly could also be used for expressing more complex relationships, but we have chosen a more shallow representation to avoid introducing structurally unresolvable ambiguities. Here the shallowness results in some inconvenience for the grammarian because it tends to compromise the usability of the above mentioned statement about the linear order of parts of speech in premodifying constructions.

Next, let us attempt a formalisation:

```
# Defs

Premodifier = @>N ;
TooEtc>A = . [too | how | as | so] . @>A ;
IndefiniteArticle = <Indef> . @>N ;
A<ENOUGH = . enough . @A< ;
..NomH = [[@ Ad-A]* @ NominalHead] ;
..Pre> = [[@ Ad-A]* @ Premodifier] ;

# Rules

     Premodifier =>
TooEtc>A @ . Adj . _ @ IndefiniteArticle,
         Nominal . _ (@ A<ENOUGH) @ [Coord | ..Pre> | ..NomH];
```

0.11.3 On the distribution of prepositions

A preposition can consist of one word, e.g. *in*, or of several words, e.g. *in spite of*. In the ENGTWOL description, multi-word prepositions are represented as one-word tokens, e.g. *in=spite=of*.

Typically, a preposition has a complement – a noun phrase or a nominal clause – in the right-hand context, e.g:

```
on    PREP      @
the   DET  @>N  @
table N    @P<< @
```

```
from  PREP             @/
what  PRON @OBJ        @
he    PRON @SUBJ       @
said  V    @MV   P<<@  @
```

```
without    PREP              @
your       PRON  @subj       @
signing    PCP1  @mv   P<<@  @
the        DET   @>N         @
agreement  N     obj         @
```

Occasionally, also an adverb, an adjective or a prepositional phrase can serve as preposition complement:

```
in      PREP        @
brief   A     @P<<  @
```

```
at     PREP         @
once   ADV    @P<<  @
```

```
from  PREP       @
under PREP @P<< @
the   DET  @>N  @
table N    @P<< @
```

A preposition may also be deferred: either the complement is in the left-hand context or it is missing altogether. There are five basic constructions:

1. Passive constructions with a prepositional verb where the subject corresponds to the prepositional complement in the active:

```
                                        @@
the             DET  @>N                 @
room            N    @SUBJ               @
has             V    @AUX                @
been            PCP2 @AUX                @
paid <P/for>    PCP2 @MV    MAINC@      @
for <Deferred>  PREP @ADVL               @
.               FULLSTOP                @@
```

2. Infinitive clauses with thematisation of the prepositional complement:

```
                                        @@
he               PRON    @SUBJ           @
is               V       @MV   MAINC@    @
impossible       A       @SC             @
to               INFMARK @aux            @
work             V       @mv   N<@       @
with <Deferred>  PREP    @ADVL           @
.                FULLSTOP               @@
```

3. *-ing*-clauses with thematisation:

```
                                    @@
he             PRON   @SUBJ          @
is             V      @MV   MAINC@   @
worth          A      @SC            @
listening      PCP1   @mv   N<@      @
to <Deferred>  PREP   @ADVL          @
.              FULLSTOP             @@
```

4. *WH*–questions:

```
                                    @@
what             PRON  @>>P          @
did              V     @AUX          @
she              PRON  @SUBJ         @
look             V     @MV  MAINC@ @
like <Deferred>  PREP  @ADVL         @
?                QUESTION           @@
```

5. Relative clauses:

```
                                               @@
the                DET    @>N                   @
house              N      @SUBJ                 @<
she                PRON   @SUBJ                 @
was                V      @AUX                  @
talking            PCP1   @MV     N<@           @
about <Deferred>   PREP   @ADVL                 @>
was                V      @AUX                  @
sold               PCP2   @MV     MAINC@        @
yesterday          N      @ADVL                 @
.                  FULLSTOP                    @@
```

As we can see, a preposition can occur in several different constructions. Nevertheless, expressing the distributions as a readable implication rule is quite straightforward:

```
                          PREP =>
                                 _ . @ Coordinator,
                                 _ . ..PrepComp,
         PassVChain.. <Deferred> . _,
PostmodifyingClause.. <Deferred> . _,
        WH-Question.. <Deferred> . _;
```

Using constants makes for readability and compactness; for instance, *..PrepComp* lists the possible preposition complements in an underspecific fashion, by referring to the crucial tags, e.g. *@P<<*.

0.12 On heuristics

The present grammatical representation avoids introducing certain structurally unresolvable distinctions connected with modifier scope, attachment and coordination. Still, some kinds of structurally unresolvable distinction remain to bother the parser. For instance, consider the following three alternative parses:

```
                                        @@
remove    V       @MV     MAINC@   @
the       DET     @>N               @
lower     A       @>N               @
clutch    N       @>N               @
housing   PCP1    @>N               @
bolts     N       @OBJ              @
and       CC      @COORD            @
the       DET     @>N               @
cover     N       @OBJ              @
.         FULLSTOP                 @@

                                        @@
remove    V       @MV     MAINC@   @
the       DET     @>N               @
lower     A       @>N               @
clutch    N       @OBJ              @
housing   PCP1    @mv     @N<       @
bolts     N       @obj              @
and       CC      @COORD            @
the       DET     @>N               @
```

```
cover     N     @OBJ          @
.         FULLSTOP           @@

                             @@
remove    V     @MV   MAINC@  @
the       DET   @>N           @
lower     A     @>N           @
clutch    N     @OBJ          @
housing   PCP1  @mv   @N<     @
bolts     N     @obj          @
and       CC    @COORD        @
the       DET   @>N           @
cover     N     @obj          @
.         FULLSTOP           @@
```

On the basis of structural information, all three readings are possible: *clutch housing bolts* can be analysed as a simplex noun phrase, or *housing* can be the main verb of a nonfinite construction that takes as object either *bolts* or *bolts and the cover*.

A grammar should minimally make a distinction between what is possible in a language and what is not. In a realistic application, however, this may not always suffice: a grammar that accounts for (virtually) every construction in the language is likely to assign uncomfortably many analyses to some of the input sentences, even when a considerably resolvable grammatical representation is available. Therefore a robust parsing system will also benefit from properly used less categorical generalisations.

The present finite-state formalism contains a mechanism for ranking analyses accepted by the strict grammar rules. A heuristic rule can formally look like any other grammar rule; yet so far we have been satisfied with imposing credit points or penalties to regular expressions. For instance, penalties can be imposed like this:

```
heuristic{

1:      @APP;
        PL . @>N;

2:      AdjectiveAsNPHead;

         }
```

Given the above heuristics, a sentence reading containing one apposition reading and two adjectival noun phrase heads would be assigned $1 + 2 + 2 = 5$ penalty points. If heuristics are applied, the parser will return the 'best' analyses first.

Next we sketch some potential applications for the heuristic component.

0.12.1 Function assignment

Although conventional grammars like Quirk & al. are primarily concerned with what is possible in the language, occasionally one will also find mention of what is typical and what is less so. Consider for instance the form–function mapping. In their discussion of the possible form of grammatical subjects, Quirk & al. note that occasionally also a prepositional phrase can be used as a subject; witness the following sentence:

```
On the lake will be fine.
```

Here *On the lake* is described as a subject. If the grammar accounts for this untypical use of the prepositional phrase, sentences like the above will be analysed correctly. The practically inconvenient consequence is that sentences like

```
In that case the meeting may end.
```

will receive at least two analyses, one with *In that case* as the subject and *the meeting* as an apposition:

```
                                @@
in       PREP   @SUBJ            @
that     DET    @>N              @
case     N      @P<<             @
the      DET    @>N              @
meeting  PCP1   @APP             @
may      AUXMOD @AUX             @
end      V      @MV   MAINC@     @
.        FULLSTOP               @@
```

and the other with *In that case* as an adverbial and *the meeting* as a subject:

```
                                  @@
in       PREP    @ADVL             @
that     DET     @>N               @
case     N       @P<<              @
the      DET     @>N               @
meeting  PCP1    @SUBJ             @
may      AUXMOD  @AUX              @
end      V       @MV    MAINC@     @
.        FULLSTOP                 @@
```

The heuristic component can be used to 'foreground' the more typical analysis. For instance, penalties can be imposed on analyses with untypical form–function mapping, e.g.

```
PREP . @SUBJ ;
```

In this way, the parser can maximise both recall and precision: also untypical uses can be recognised as such without flooding the practical application with a host of alternatives. *On the lake will be fine* will be analysed correctly, although the parser may return the analysis with some penalties which indicate that something may be a little untypical in the sentence; on the other hand, the comprehensiveness of the grammar will not result in massive ambiguity in the parser's output.

0.12.2 Modifier–Head constructions

Another potential application of the heuristics component is the detection of modifier–head constructions, e.g. compounds. Consider the following analyses of the sentence *They are computing officers*:

```
                                  @@
they      PRON @SUBJ               @
be        V    @MV   MAINC@ @
computing PCP1 @>N                 @
officer   N    @SC                 @
.         FULLSTOP                @@
```

```
                                  @@
they      PRON @SUBJ               @
be        V    @AUX                @
compute   PCP1 @MV  MAINC@        @
officer   N    @OBJ                @
.         FULLSTOP                @@
```

In the first analysis, *computing officers* is regarded as a simplex noun phrase while in the latter *computing* is analysed as a main verb, with *officers* as an object. This particular type of structural ambiguity can be resolved if the system employs knowledge about lexical co-occurrence. For this purpose, the heuristic mechanism is suitable: for instance analyses that satisfy the following expression:

```
computing . @>N @ officer ;
```

could be preferred over alternative ones.

How can these constructions be collected?

- NP parsers

- annotated corpora and lexica

0.12.3 Corpus probabilities

The heuristic section could also be used for expressing corpus frequencies. For instance, part-of-speech frequencies could be expressed with the credit mechanism. Consider the classical ambiguous sentence *Time flies like an arrow.* For instance, the following part-of-speech assignments would be structurally possible.

```
[N time] [N flies] [V like] [DET an] [N arrow]
[N time] [V flies] [PREP like] [DET an] [N arrow]
[V time] [N flies] [V like] [DET an] [N arrow]
[V time] [N flies] [PREP like] [DET an] [N arrow]
```

From annotated corpora it is possible to derive statistics about part-of-speech frequencies. The strongest tendencies could then be used as part of the heuristics. For example, a consultation of a the Lancaster–Oslo/Bergen Corpus indicates that *time* is almost always used as a noun (100%: 1654 instances out of 1658), so the heuristics section could assign 0 penalty points for noun readings of *time* and 100 penalty points to the other readings:

```
penalties{



.
.
.
100:
        time . Vbase ;
.
.
.
         }
```