0.1. PC-PATR

PC-PATR-II is an implementation of PATR-II, a computer system for encoding linguistic information initially developed by Shieber in 1988 (McConnel, 1995). According to Schieber (1989), the PATR-II grammar formalism has been developed over the last few years at SRI International as a grammar formalism for codifying fragments of natural language. It is used to perform syntactic validation and analysis using a lexicon and grammar rules. Unification grammar and feature structures similar to PC-Kimmo are used to constrain the constituents of the rule.

Notation used in syntactic grammar rules are similar to the word grammar in PC-Kimmo version 2. PC-Kimmo is built into the PC-PATR and therefore it is possible to parse the inflected, derived and compound word forms in a sentence first using the word grammar and PC-Kimmo lexicons, and subsequently return the morphological features to the syntactic parser. This leads to the integration of the morphological and syntactic analyzers which is significant. The following is a simplified version of the syntactic grammar in Perslex.

1	Sentence → (Subject-P) (When-P) (How-P) (Where-P) Verb-P	10	Verb-P → V
2	Subject-P → PRON / N	11	Verb-P → CONT Verb-P
3	Object-P → Noun-P	12	Verb-P → Noun-P Verb-P
4	Noun-P → Noun-P RA	13	Verb-P → Verb-P VSFX
5	Noun-P → Noun-P AJ-P <AJ-P head ajform> = {REL ABS>	14	Verb-P → AUX Verb-P
6	Noun-P → Noun-P AJ-P TAR <AJ-P head ajform> = REL	15	How-P → BEH N <N case> = INFI
7	Noun-P → AJ Noun-P <AJ head ajform> = SUP	16	When-P → (PP) AD <AD head sem> = TIME
8	Noun-P → AJ-P TARIN Noun-P <AJ head ajform> = ABS	17	Where-P →PP Noun-P <Noun-P head sem> = PLACE
9	Noun-P → N PLSFX	18	How-P → AD <AD head sem> = MOOD

Figure 6.2

The following are some examples that can be successfully parsed using this grammar:

Surface form	Lexical form	Meaning
[avrdnd]	[~avár+d+ánd]	{they brought}
[mn miKvabm]	[mán mi+Kvab+ám]	{I sleep}
[mn mi Kvabm]	[mán mi Kvab+ám]	{I sleep}
[sib srK ra dvst darm]	[sib sröK ra dvst dar+ám]	{I like red apple}
[srKtrin sib ra Kvrdm]	[sörK+tárin sib ra Kövr+d+ám]	{I ate the most red apple}
[srK trin sib ra Kvrdm]	[sörK tárin sib ra Kövr+d+ám]	{I ate the most red apple}
[sib srK tr ra biavr]	[sib sörK ra bé+~avár}	{bring the more red apple}
[av amrvz bsKti bh mdrsh rft]	[av amrvz bé+sáKt+i béh mádrésh ráft]	{He went to school today with difficulty)

Syntactic sequence of SOV (Subject, Object, Verb) has been considered for the sake of simplicity otherwise phrases have no fixed position. In rule.1 all phrases except the verb phrase are optional. The subject can be a pronoun or a noun, and the object is a noun phrase that may be also followed by the object marker [ra]. Rules 5 to 8 describe the sequence in which an adjective and noun can appear. Rule 9 uses the plural free suffix [ha] to produce the plural form.
Rule 11 caters for the situation where the prefix [mi+], appears as a free morpheme. Rule 12 defines a compound verb and rule 13 handles the verb suffixes indicating that the person and the number need to appear as free morphemes. This is only necessary for verbs with perfect tenses which always end with suffix [+éh]. This is because the symbol [h] is of isolated variant and can not join to the next letter. Rules 15 to 18 describe adverbs which refer to either a place, some kind of mood or time, these are semantic features but can be also used to influence the sequence in which adverbs appear.

Successfully parsed word structures with their morpho-syntactic features can be saved in a lexicon which can be later directly accessed by PC-PATR. This provides an automated mechanism for creating a machine readable lexicon with the potential of reducing the processing time on subsequent attempts to parse the same word form.
For example PC-PATR generates the following lexical entries when it parses the sentence [mán mi+ráv+ám] meaning 'I am going'.

```
\w mn
\c PRON
\g mán
\f   <root_pos> = PRON
          <head pos> = PRON
          <head person> = 1
          <head number> = SG
```

```
\w mirvm
\c V
\g ráv
          \f   <root_pos> = V
          <head tense> = PRES
          <head v-mode> = CONT
          <head pos> = V
           <head number> = SG
           <head person> = 1
           <head case> = INTRANS
```

the first lexical entry comes directly from the Perslex lexicon, however the second entry is generated from within PC-PATR. The next diagram shows the syntactic analysis of the sentence [Hsn saKtman misazd] after every word is analyzed and morpho-syntactic features are returned to the syntactic parser:

```
        S_13
   _____|_______
Subject-P_2 Object-P_14 T-V_15
   |        |       |
 N_3+  Noun-P_5+ Verb-P_7+
  Hsn       |       |
          N_6+    V_8+
         saKtman misazd

S:
[ cat:   S
  obj:     [ genform:-
             number:SG
             object:-
             pos:   N
             tense: PAST ]
  subj:    [ case:  PERS
             number:SG
             person:3
             pos:   N
             reg:   +
             sem:   - ]
  verb:    [ number:SG
             person:3
             pos:   V
             tense: PRES
             v-mode:CONT
             vdir:  TRANS ] ]
```

```
saKtman

saz+t+man     saz+ PAST+INFI

1:
           Word
       ______|______
     Word          SFX
       |          +man
     Stem         +INFI
   ____|____
 Stem  TENSE-SF
   |      +t
 ROOT   + PAST
  saz
  saz

Word:
[ cat:   Word
  head:    [ number:SG
             pos:   N
             tense: PAST ]
  plform:HA
  root:  saz
  root_pos:V ]
```

```
Hsn

Hásán     Hásán

1:
 Word
   |
 Stem
   |
 ROOT
 Hásán
 Hásán

Word:
[ cat:   Word
  cosuff:  [ relajsfx:- ]
  head:    [ case:  PERS
             number:SG
             person:3
             pos:   N
             reg:   +
             sem:   - ]
  plform:-
  root:  Hásán
  root_pos:N ]
```

```
misazd

mi+saz+ád     CONT+saz+ 3SG

1:
      Word
   _____|_____
PREFIX      Word
  mi+      ___|____
 CONT+  Stem   VSUFF
          |     +ád
        ROOT   + 3SG
         saz
         saz

Word:
[ cat:   Word
  head:    [ number:SG
             person:3
             pos:   V
             tense: PRES
             v-mode:CONT
             vdir:  INTRANS ]
  root:  saz
  root_pos:V ]
```

Figure 6.3