```
;;;======================== ========================================
;;; The tested version. The 'start' has bee crossed out.

;;;(<start> --> (<np>)
;;;       ((x1 = x0)))

(<np> --> (<adjp> <np>)
         (((x0 cat) =c *unit-range)
          ((x0 attribute) = *defined*)
         (x1 == (x0 attribute))
          (x2 = x0)))

(<np> --> (<pp> <np>)
         (((x0 cat) =c *unit-range)
          ((x0 above) = *defined*)
         (x1 == (x0 above))
          ((x1 prep root) <= "高于")
          (x2 = x0)))

(<np> --> (<np> <dao> <np> <zhijian>)
         (((x0 cat) =c *unit-range)
          (x1 = (x0 start))
          (x3 = (x0 end))))

;; sentence 24
(<np> --> (<pp-coordination> <n>)
         (((x0 coordination) = *defined*)
          (x1 == (x0 coordination))
          (x2 = x0)))
(<pp-coordination> --> (<pp-> <and-conj> <pp->)
         (((x0 cat) =c *pp-coordination)
         ((x0 and-coordination) =c +)
         (x1 <= (list (cadddr (x0))))
          (x3 <= (list (caddr (x0))))))
(<pp-coordination> --> (<pp-> <or-conj> <pp->)
         (((x0 cat) =c *pp-coordination)
         ((x0 or-coordination) =c +)
         (x1 <= (list (cadddr (x0))))
          (x3 <= (list (caddr (x0))))))
(<pp-> --> (<pp>)
         (((x0 source) = *defined*)
         (x1 = (x0 source))
          ((x1 prep root) <= "从")
          ((x1 surf root) <= "来的")))
(<pp-> --> (<pp>)
         (((x0 location) = *defined*)
         (*EOR* (((x0 location root) =c "角度"))
                (((x0 location root) =c "倾斜度")))
         (x1 = (x0 location))
          ((x1 prep root) <= "以")))
(<pp-> --> (<pp>)
         (((x0 location) = *defined*)
```

```
        (x1 = (x0 location))
         ((x1 prep root) <= "在")))
(<pp-> --> (<pp>)
        (((x0 outside) = *defined*)
        (x1 = (x0 outside))
         ((x1 prep root) <= "在")
        ((x1 surf root) <= "外面")))
(<and-conj> --> (%)
        (((x1 value) <= "和")))
(<or-conj> --> (%)
        (((x1 value) <= "或")))
(<and/or-conj> --> (%)
        (((x1 value) <= "与/或")))
(<duonghao> --> (%)
        (((x1 value) <= "、")))
;;;=================================================================
;;;
;;;
;;;np 1
(<np> --> (<units>)
        (((x0 cat) =c *units)
         (x1 = x0)))
;;;np 2
(<np> --> (<det> <np>)
        (((x0 possessive) =c +)
         ((x1 possessive) == (x0 possessive))
         (*EOR*(((x0 number) = *undefined*))(((x1 number) == (x0 number))))
         ((x1 person) == (x0 person))
         (x2 = x0)))

;;;np 3
(<np> --> (<det> <np>)
        ((*EOR*(((x0 near) = *defined*)((x1 near) == (x0 near)))
               (((x0 far) = *defined*)((x1 far) == (x0 far))))
         ((x1 number) == (x0 number))
         ((x1 reference) == (x0 reference))
         (x2 = x0)))

(<np> --> (<predicate3> <de> <np>)
     (((x0 proposition) = *DEFINED*)
      ((x0 proposition predicated-of-theme) = *DEFINED*)
      (x1 == (x0 proposition))
      ((x1 no-shi) = +)
      (x3 = x0)))

(<np> --> (<s> <de> <np>)
     (((x0 proposition) = *DEFINED*)
      (x1 == (x0 proposition))
      (x3 = x0)))

(<np> --> (<vp> <de> <np>)
     (((x0 proposition) = *DEFINED*)
      (x1 == (x0 proposition))
```

```
      (x3 = x0)))


;;;================================================================
;;; This four rules are for nouns with verb character. Probablly I
;;; can use another method to deal with it, such as to transfer this
;;; kind of noun to verb in the transfermation process.

;;;np 4

(<np> --> (<np> <ppadjunct> <np>)
      (((x0 object) = *defined*)
       ((x0 theme) = *defined*)
       (x1 == (x0 object))
       (x2 == (x0 theme))
       ((x2 prep root) <= "对")
       (x3 = x0)))

;;;np 5
(<np> --> (<np> <ppadjunct> <np>)
      (((x0 object) = *defined*)
       ((x0 path) = *defined*)
       (x1 == (x0 object))
       (x2 == (x0 path))
       ((x2 prep root) <= "从")
       (x4 = x0)))

;;;np 6
(<np> --> (<np> <np> <pp>)
      (((x0 object) = *defined*)
       ((x0 cofactor) = *defined*)
       (x2 == (x0 object))
       (x3 == (x0 cofactor))
       ((x3 prep root) <= "于")
       (x1 = x0)))

;;;np 7
(<np> --> (<pp> <np> <np>)
      (((x0 object) = *defined*)
       ((x0 location) = *defined*)
       (x3 == (x0 object))
       (x1 == (x0 location))
       ((x1 prep root) <= "在")
       (x2 = x0)))

;;;================================================================
;;; Following rules are for PP adjuct of noun phrase.

;;;np 8
(<np> --> (<ppadjunct> <np>)
      (((x0 under) = *defined*)
       (x1 == (x0 under))
```

```
       ((x1 prep root) <= "低于")
       (x2 = x0)))

;;;np 9
(<np> --> (<ppadjunct> <np>)
      (((x0 above) = *defined*)
       (x1 == (x0 above))
       ((x1 prep root) <= "高于")
       (x2 = x0)))

;;;np 10
(<np> --> (<ppadjunct> <vp>)
      (((x0 for-event) = *defined*)
       ((x0 for-event cat) =c n)
       (x1 == (x0 for-event))
       ((x1 prep root) <= "用于")
       (x2 = x0)))

(<np> --> (<vp> <de> <np>)
      (((x0 for-event) = *defined*)
       ((x0 for-event cat) =c v)
       (x1 == (x0 for-event))
       (x3 = x0)))

;;;np 11
(<np> --> (<pp> <np>)
      (((x0 root) =c "距离")
       ((x0 source) = *defined*)
       (x1 == (x0 source))
       ((x1 prep root) <= "离")
       (x2 = x0)))

(<np> --> (<np> <de> <np>)
      (((x0 source) = *defined*)
       (x1 == (x0 source))
       (x3 = x0)))

;;;np 12
(<np> --> (<ppadjunct> <np>)
      (((x0 goal) = *defined*)
       (x1 == (x0 goal))
       ((x1 prep root) <= "到")
       (x2 = x0)))

;;;np 13
(<np> --> (<ppadjunct> <np>)
      (((x0 destination) = *defined*)
       (x1 == (x0 destination))
       ((x1 prep root) <= "到")
       (x2 = x0)))

;;;np 14
```

```
(<np> --> (<ppadjunct> <np>)
      (((x0 has-part) = *defined*)
       (x1 == (x0 has-part))
       ((x1 prep root) <= "带")
       (x2 = x0)))

;;;np 15
(<np> --> (<ppadjunct> <np>)
      (((x0 path) = *defined*)
       (x1 == (x0 path))
       ((x1 prep root) <= "通过")
       (x2 = x0)))

 ;;;np 16
 (<np> --> (<np> <de> <np>)
       (((x0 location) = *defined*)
         (x1 == (x0 location))
         (x3 = x0)))

 ;;;np 17
 (<np> --> (<ppadjunct> <np>)
       (((x0 located-in) = *defined*)
         (x1 == (x0 located-in))
         ((x1 prep root) <= "在")
         ((x1 surf root) <= "内部")
         (x2 = x0)))

 ;;;np 18
 (<np> --> (<ppadjunct> <np>)
       (((x0 opposition) = *defined*)
         (x1 == (x0 opposition))
         ((x1 prep root) <= "反")
         (x2 = x0)))

 ;;;np 19
 (<np> --> (<ppadjunct> <np>)
       (((x0 between-items) = *defined*)
         (x1 == (x0 between-items))
         ((x1 prep root) <= "在")
         ((x1 surf root) <= "之间")
         (x2 = x0)))

 ;;;np 20
 (<np> --> (<np> <np>)
       (((x0 text-referent) = *defined*)
         (x1 == (x0 text-referent))
         (x2 = x0)))

 ;;;np 21
 (<np> --> (<np> <de> <np>)
       (((x0 condition) = *defined*)
         (x1 == (x0 condition))
```

```
        (x3 = x0)))

;;;np 22
(<np> --> (<np> <np>) ;;;contradictory (43,48):67. so canceled.
      (((x0 object) = *defined*)
        ((x0 root) =c "使用")
        (x2 == (x0 object))
        (x1 = x0)))

(<np> --> (<np> <de> <np>)
      (((x0 object) = *defined*)
        (x1 == (x0 object))
        (x3 = x0)))

;;;np 23
(<np> --> (<np> <np>)
      (((x0 theme) = *defined*)
        (x1 == (x0 theme))
        (x2 = x0)))

;;;==================================================================
;;; This unique rule is used with adjective phrase modifiers.

;;;np 24
(<np> --> (<np> <de> <np>)
      (((x0 belongs-to) = *defined*)
        (x1 == (x0 belongs-to))
        (x3 = x0)))

;;;np 25
(<np> --> (<adjp> <np>)
      (((x0 unit) = *defined*)
       ((x0 attribute) = *defined*)
        (x1 == (x0 attribute))
       ((x1 unit) = (x0 unit))
        (x2 = x0)))

(<np> --> (<adjp> <np>)
      (((x0 attribute) = *defined*)
        (x1 == (x0 attribute))
        (x2 = x0)))

;;;==================================================================
;;; Following rules are designed for simple NP with a noun modifier
;;; or a special modifier, such as serial number, etc.

;;;np 26
(<np> --> (<pp> <np>)
      (((x0 per-measurement-unit) = *defined*)
        (x1 == (x0 per-measurement-unit))
        ((x1 prep root) <= "每")
        (x2 = x0)))
```

```
;;;np 27
(<np> --> (<np> <np>)
      (((x0 serial-number-concept) = *defined*)
        (x1 == (x0 serial-number-concept))
        (x2 = x0)))

;;;np 28
(<np> --> (<serial-number> <np>)
      (((x0 serial-number-integer) = *defined*)
        ((x1 integer) == (x0 serial-number-integer))
        (x2 = x0)))

;;;np 29
(<np> --> (<np> <np>)
      (((x0 substance) = *defined*)
        (x1 == (x0 substance))
        (x2 = x0)))

;;;np 30
(<np> --> (<quoted-label> <np>)
      (((x0 quoted-label) = *defined*)
        (x1 == (x0 quoted-label))
        (x2 = x0)))

;;;np 31
(<np> --> (<number> <np>)
      (((x0 quantity) = *defined*)
        (x1 == (x0 quantity))
        (x2 = x0)))

;;;np 32
(<np> --> (<np> <np>)
      (((x0 type) = *defined*)
        (x1 == (x0 type))
        (x2 = x0)))

;;;np 33
(<np> --> (<np> <np>)
      (((x0 function) = *defined*)
        (x1 == (x0 function))
        (x2 = x0)))

;;;================================================================
;;; For parenthetical comments.

;;;np 34
(<np> --> (<np> <paren-np>)
      (((x0 paren-material) = *defined*)
        (x2 == (x0 paren-material))
        (x1 = x0)))

;;;np 35
```

```
(<np> --> (<np> <duonghao> <np> <and-conj> <np>)
        (((x0 cat) =c *coordination)
         ((x0 and-conjunction) =c +)
         (*TEST* (cdddr (assoc 'conjuncts (x0))))
        (x1 <= (cadddr (assoc 'conjuncts (x0))))
        (x3 <= (caddr (assoc 'conjuncts (x0))))
        (x5 <= (cadr (assoc 'conjuncts (x0))))))
(<np> --> (<np> <duonghao> <np> <or-conj> <np>)
        (((x0 cat) =c *coordination)
         ((x0 or-conjunction) =c +)
         (*TEST* (cdddr (assoc 'conjuncts (x0))))
        (x1 <= (cadddr (assoc 'conjuncts (x0))))
        (x3 <= (caddr (assoc 'conjuncts (x0))))
        (x5 <= (cadr (assoc 'conjuncts (x0))))))
(<np> --> (<np> <and-conj> <np>)
        (((x0 cat) =c *coordination)
         ((x0 and-conjunction) =c +)
        (x1 <= (caddr (assoc 'conjuncts (x0))))
        (x3 <= (cadr (assoc 'conjuncts (x0))))))
(<np> --> (<np> <or-conj> <np>)
        (((x0 cat) =c *coordination)
         ((x0 or-conjunction) =c +)
        (x1 <= (caddr (assoc 'conjuncts (x0))))
        (x3 <= (cadr (assoc 'conjuncts (x0))))))
(<np> --> (<np> <and/or-conj> <np>)
        (((x0 cat) =c *coordination)
         ((x0 and/or-conjunction) =c +)
        (x1 <= (caddr (assoc 'conjuncts (x0))))
        (x3 <= (cadr (assoc 'conjuncts (x0))))))

 ;;;==================================================================
 ;;;
 ;;; NP with singal noun.

 ;;;np 36
 (<np> --> (<n>)
      ((*EOR* (((x0 cat) =c n)) (((x0 cat) =c *pron)))
       (x1 = x0)))

 (<np> --> (<adjp>)
        (((x0 cat) =c a)
        ((x0 no-de) = +)
        (x1 = x0)))

 (<np> --> (<s>)
        (((x0 cat) =c v)
        (x1 = x0)))

 (<np> --> (<vp>)
        (((x0 cat) =c v)
        (x1 = x0)))

 (<np> --> (<number>)
```

```
        (((x0 cat) =c *number)
        (x1 = x0)))

(<n> --> (<pron>)
        (((x0 cat) =c *pron)
        (x1 = x0)))

(<n> --> (%)
     (((x1 value) = (x0 root))))

(<pron> --> (%)
        (((x0 person) =c third)
        ((x0 number) =c pl)
         ((x0 gender) =c neutral)
         ((x1 value) <= "它们")))

(<pron> --> (%)
        (((x0 person) =c third)
        ((x0 number) =c sg)
         ((x0 gender) =c neutral)
         ((x1 value) <= "它")))

(<pron> --> (%)
        (((x0 person) =c third)
        ((x0 number) =c pl)
         ((x1 value) <= "他们")))

(<pron> --> (%)
        (((x0 person) =c third)
        ((x0 number) =c sg)
         ((x1 value) <= "他")))

(<pron> --> (%)
        (((x0 person) =c second)
         ((x1 value) <= "你")))

(<pron> --> (%)
        (((x0 pleonastic-pron) =c this)
         ((x1 value) <= "这")))

;;;=================================================================
;;; DETERMINER

(<det> --> (%)
        (((x0 possessive) =c +)
         ((x0 number) =c sg)
         ((x0 person) =c first)
         ((x1 value) <= "我的")))

(<det> --> (%)
        (((x0 possessive) =c +)
         ((x0 number) =c pl)
```

```
         ((x0 person) =c first)
         ((x1 value) <= "我们的")))

(<det> --> (%)
        (((x0 possessive) =c +)
         ((x0 number) =c pl)
         ((x0 person) =c second)
         ((x1 value) <= "你们的")))

(<det> --> (%)
        (((x0 possessive) =c +)
         (*EOR* (((x0 number) = *undefined*))(((x0 number) =c sg)))
         ((x0 person) =c second)
         ((x1 value) <= "你的")))

(<det> --> (%)
        (((x0 possessive) =c +)
         ((x0 number) =c sg)
         ((x0 person) =c third)
         ((x1 value) <= "他的")))

(<det> --> (%)
        (((x0 possessive) =c +)
         ((x0 number) =c pl)
         ((x0 person) =c third)
         ((x1 value) <= "他们的")))

(<det> --> (%)
        (((x0 near) =c +)
         ((x0 reference) =c definite)
         ((x0 number) =c sg)
         ((x1 value) <= "这")))

(<det> --> (%)
        (((x0 near) =c +)
         ((x0 reference) =c definite)
         ((x0 number) =c pl)
         ((x1 value) <= "这些")))

(<det> --> (%)
        (((x0 far) =c +)
         ((x0 reference) =c definite)
         ((x0 number) =c sg)
         ((x1 value) <= "那")))

(<det> --> (%)
        (((x0 far) =c +)
         ((x0 reference) =c definite)
         ((x0 number) =c pl)
         ((x1 value) <= "那些")))

(<ppadjunct> --> (<pp> <de>)
```

```
            ((x1 = x0)))

(<de> --> (%)
      (((x1 value) <= "的")))

;;;------------------------------------------------------------
;;;
;;; There are some ill defined structures in ILT with adj phrase,
;;; which sould be modify mail verb "BE" instead of adjp. I order
;;; to cope with these ill-defined structures I have to incluse
;;; following structures of adjp, which, I beleive sould be removed
;;; after the correction of the problems of the ILT.
;0
(<adjp> --> (<adjp> <adjp>)
        (((x0 cat) =c *multiple)
        ((x0 unit) = *defined*)
         (x1 <= (caddr (x0)))
        (x1 <= (cons '(no-de +) (x1)))
        (x2 <= (cadddr(x0)))
        ((x1 unit) == (x0 unit))))

(<adjp> --> (<adjp> <adjp>)
        (((x0 cat) =c *multiple)
         (x1 <= (cadr (x0)))
        (x1 <= (cons '(no-de +) (x1)))
        (x2 <= (caddr(x0)))))
;1
(<adjp> --> (<adv> <adjp>)
         (((x0 event-contrast) = *defined*)
          (x1 == (x0 event-contrast))
          (x2 = x0)))
;2
(<adjp> --> (<pp> <adjp>)
         (((x0 during-event) = *defined*)
          (x1 == (x0 during-event))
          ((x1 prep root) <= "在")
          ((x1 surf root) <= "期间")
          (x2 = x0)))
;3
(<adjp> --> (<pp> <adjp>)
         (((x0 location) = *defined*)
          (x1 == (x0 location))
          ((x1 prep root) <= "在")
          ((x1 surf root) <= "下")
          (x2 = x0)))
;4
(<adjp> --> (<adjp> <np>)
         (((x0 goal) = *defined*)
          (x2 == (x0 goal))
          (x1 = x0)
         ((x1 no-de) = +)))

;;;--------------------------------------------------------------
```

```
;5
(<adjp> --> (<as-clause> <adjp>)
         (((x0 as-theme) = *defined*)
          (x1 == (x0 as-theme))
          (x2 = x0)))
;6
(<adjp> --> (<adv> <adjp>)
         (((x0 manner) = *defined*)
          (x1 == (x0 manner))
          (x2 = x0)))
;7
(<adjp> --> (<adv> <adjp>)
         (((x0 intensifier) = *defined*)
          (x1 == (x0 intensifier))
          (x2 = x0)))
;8
(<adjp> --> (<as-possible> <adjp>)
         (((x0 as-possible) = *defined*)
          (x2 = x0)))
;9
(<adjp> --> (<adjp> <than-clause>)
         (((x0 than-object) = *defined*)
          (x2 == (x0 than-object))
         ((x0 comparative) = *remove*)
         ((x0 no-de) = +)
          (x1 = x0)))

(<adjp> --> (<adjp>)
         (((x0 comparative) = *defined*)
          (x1 = x0)
         ((x1 comparative) = *remove*)
         ((x1 intensifier root) = "较")
         ((x1 intensifier cat) = adv)))
;10
 (<adjp> --> (<pp> <adjp>)
        (((x0 purpose) = *defined*)
         (x1 == (x0 purpose))
          ((x1 prep root) <= "对")
         (x2 = x0)))
;11
(<adjp> --> (<adj> <unit>)
        ((*EOR* (((x0 subcat) =c number))
                (((x0 subcat) =c det)))
         ((x0 unit) = *defined*)
         ((x2 value) == (x0 unit))
         (x1 = x0)))

(<adjp> --> (<adj>)
        (((x0 no-de) =c +)
         (x1 = x0)))
;12
(<adjp> --> (<adj> <de>)
```

```
        ((x1 = x0)))
;;;-----------------------------------------------------------------
(<adj> --> (%)
        (((x0 cat) =c a)
        ((x1 value) = (x0 root))))

(<adv> --> (%)
        (((x0 cat) =c adv)
        ((x1 value) = (x0 root))))

(<than-clause> --> (<yu1> <np>)
                ((x2 = x0)))

(<as-clause> --> (<yu> <np> <yiyang>)
              ((x2 = x0)))

(<as-possible> --> (%)
                (((x1 value) <= "尽可能")))

(<paren-np> --> (<[> <np> <]>)
            ((x2 = x0)))
(<[> --> (%)
     (((x1 value) <= "[")))
(<]> --> (%)
     (((x1 value) <= "]")))

(<yu1> --> (%)
      (((x1 value) = "于")))

(<geng> --> (%)
         (((x1 value) = "更")))

(<yu> --> (%)
      (((x1 value) <= "与")))

(<yiyang> --> (%)
      (((x1 value) <= "一样")))

(<dao> --> (%)
        (((x1 value) <= "到")))

(<zhijian> --> (%)
        (((x1 value) <= "之间")))

(<pp> --> (<pp-coordination>)
       (((x0 cat) =c *pp-coordination)
        ((x0 prep) = *remove*)
        (x1 = x0)))

(<pp> --> (<prep> <np> <surf>)
      (((x0 prep) = *defined*)
        (x1 == (x0 prep))
```

```
        (*EOR* (((x0 surf) = *defined*) (x3 == (x0 surf)))
               ((x3 <= '((root "")))))
        (x2 = x0)))

(<pp> --> (<prep> <np>)
      (((x0 prep) = *defined*)
        (x1 == (x0 prep))
        (x2 = x0)))

(<prep> --> (%)
         (                          ;;; ((x0 cat) =c prep)
         ((x1 value) = (x0 root))))

(<surf> --> (%)
         (((x1 value) = (x0 root))))

(<de> --> (%)
      (((x1 value) <= "的")))


(<quoted-label> --> (<l-quote> <quoted-string> <r-quote>)
        (((x0 cat) =c *quoted-term)
         ((x2 root) = (x0 quoted-string))))
(<quoted-string> --> (%)
        (((x1 value) = (x0 root))))
(<l-quote> --> (%)
        (((x1 value) <= " “"))) ;;; “ 『
(<r-quote> --> (%)
        (((x1 value) <= "” "))) ;;;” 』

;;;-----------------------------------------------------------
;;;
;;;
;;; generation of numbers
(<serial-number> --> (<number> <#>)
        ((x1 = x0)))
(<number> --> (<number> <%>)
       (((x0 percent) = *defined*)
         ((x0 percent) = *remove*)
        (x1 = x0)))
(<number> --> (<integer>)
       (((x0 decimal) = *undefined*)
         ((x1 root) = (x0 integer))))
(<number> --> (<integer> <point> <integer>)
       (((x0 decimal) = *defined*)
         ((x1 root) = (x0 integer))
        ((x3 root) = (x0 decimal))))
(<integer> --> (%)
        (((x1 value) = (x0 root))))
(<point> --> (%)
        (((x1 value) <= ".")))
(<%> --> (%)
        (((x1 value) <= "%")))
```

```
(<#> --> (%)
        (((x1 value) <= "#")))
;;;-----------------------------------------------------------
;;;
;;; generation of units

(<units> --> (<units> <np>)
        (((x0 object) = *defined*)
         (x2 == (x0 object))
         (x1 = x0)))
(<units> --> (<units> <[> <np> <]>)
        (((x0 metric) = *defined*)
         (x3 == (x0 metric))
        (x1 = x0)))
(<units> --> (<adjp> <units>)
        (((x0 attribute) = *defined*)
         (x1 == (x0 attribute))
        (x2 = x0)))
(<units> --> (<np>)
        (((x0 english) = *defined*)
         (x1 == (x0 english))))

(<unit> --> (%)
       (((x1 value) = (x0 value))))
```