

English Analysis Grammar (simplified)

This grammar is intended only to illustrate rules dealing with very simple structures (no attribute, one attribute, two attributes, three attributes and one relation). It presumes that:

- All the entries and corresponding features (D,N,SAV,VER,etc.) will be provided by the dictionary ;
- “att” is a special attribute of the dictionary entries associated to the empty UW and assumes UNL attributes as values

(e.g.: [the]{} “(D,ART,att=@def)<eng,255,0>;)

- The dictionary entries are unambiguous (i.e., the dictionary never selects the wrong word, and there is no need for disambiguation rules)

References

- The UNDL Foundation tagset (available at <http://www.unlweb.net/wiki/index.php/Tagset>)
- The Grammar Specs (available at [http://www.unlweb.net/wiki/index.php/Grammar Specs](http://www.unlweb.net/wiki/index.php/Grammar_Specs))
- English dictionaries (available at [http://www.unlweb.net/wiki/index.php/I UNL Grammar Workshop#English examples and templates](http://www.unlweb.net/wiki/index.php/I_UNL_Grammar_Workshop#English_examples_and_templates))

Rules

RULE NUMBER	RULE	EXAMPLE		DESCRIPTION
		BEFORE	AFTER	
01	(BLK):=;	[the][][book]	[the][book]	delete the blank space
02	(%a,D,att)(%b,N):=(%b,att=%a);	[the][book]	[book.@def]	if there is a node %a with the features D (determiner) and att (attribute) before a node %b with the feature N (noun), then delete the node %a and copy the value of the attribute "att" from %a to %b)
03	(%a,SAV,att)(%b,J):=(%b,att=%a);	[very][beautiful]	[beautiful.@plus]	if there is a node %a with the features SAV (specifier adverb) and att (attribute) before a node %b with the feature J (adjective), then delete the node %a and copy the value of the attribute "att" from %a to %b
04	(%a,"it")(%b,VER,VAL0,3PS):=(%b);	[it][rains]	[rains]	if there is a node %a that is the string "it" before a node %b with the features VER (verb), VAL0 (avalent) and 3PS (third person singular), then delete the node %a
05	(%a,N,PLR,^@pl):=(%a,+@pl);	[books]	[books.@pl]	if there is a node %a with the features N (noun) and PLR (plural) that does not have the attribute @pl yet, add the attribute @pl to it
06	(%a,POD)(%b,N):=(%c,+N,pos(%b;%a));	[my][book]	[pos(book;my)]	if there is node %a with the feature POD (possessive determiner) before a node %b with the feature N (noun), create a new node %c, with the feature N (noun); put both the node %a and the node %b inside this new node %c; and link them by a relation pos (possessor) going from %b to %a
07	(%a,N)(%b,"of")(%c,SPR):=(%d,+N,pos(%a;%c));	[book][of][mine]	[pos(book;my)]	if there is node %a with the feature N (noun) before a node %b that is the string "of" and that is before a node %c with the feature SPR (possessive pronoun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; and link them by a relation pos (possessor) going from %a to %c
08	(%a,NPR)(%b,"of")(%c,N):=(%d,+N,qua(%c;%a));	[all][of][books]	[qua(books;all)]	if there is node %a with the feature NPR (indefinite pronoun) before a node %b that is the string "of" and that is before a node %c with the feature N (noun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; and link them by a relation qua (quantifier) going from %c to %a)
09	(%a,N)(%b,"of")(%c,PPN):=(%d,+N,pos(%a;%c));	[car][of][John]	[pos(car;John)]	if there is node %a with the feature N (noun) before a node %b that is the string "of" and that is before a node %c with the feature PPN (proper noun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; and link them by a relation pos (possessor) going from %a to %c)
10	(%a,PPN)(%b,"'s")(%c,N):=(%d,+N,pos(%c;%a));	[John]['s][car]	[pos(car;John)]	if there is node %a with the feature PPN (proper noun) before a node %b that is the string "'s" and that is before a node %c with the feature N (noun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; and link them by a relation pos (possessor) going from %c to %a)

11	$(\%a,N)(\%b,P,att)(\%c,N):=(\%d,+N,mod(\%a;\%c,att=\%b));$	[mug][of][glass]	(mod(mug;glass.@of))	if there is node %a with the feature N (noun) before a node %b with the features P (preposition) and att (attribute in the dictionary) and that is before a node %c with the feature N (noun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; copy the value of the attribute att from %b to %c; and link %a to %c by a relation mod (modification) going from %a to %c)
12	$(\%a,SAV,^att)(\%b,J):=(\%c,+J,qua(\%b;\%a));$	[extremely][beautiful]	[qua(beautiful;extremely)]	if there is node %a with the feature SAV (specifier adverb) before a node %b with the feature J (adjective): create a new node %c, with the feature J (adjective); put both the node %a and the node %b inside this new node %c; and link them by a relation qua (modification) going from %b to %a)
13	$(\%a,N)(\%b,"and")(\%c,N):=(\%d,+N,and(\%c;\%a));$	[John][and][Mary]	[and(Mary;John)]	if there is node %a with the feature N (noun) before a node %b that is the string "and" and that is before the node %c with the feature N (noun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; and link them by a relation and (conjunction) going from %c to %a)
14	$(\%a,N)(\%b,"or")(\%c,N):=(\%d,+N,or(\%c;\%a));$	[John][or][Mary]	[or(Mary;John)]	if there is node %a with the feature N (noun) before a node %b that is the string "or" and that is before the node %c with the feature N (noun): create a new node %d, with the feature N (noun); put both the node %a and the node %c inside this new node %d; and link them by a relation or (conjunction) going from %c to %a)
15	$(\%a,J)(\%b,N):=(\%c,+N,mod(\%b;\%a));$	[beautiful][car]	[mod(car;beautiful)]	if there is node %a with the feature J (adjective) before a node %b with the feature N (noun): create a new node %c, with the feature N (noun); put both the node %a and the node %b inside this new node %c; and link them by a relation mod (modification) going from %b to %a)
16	$(\%a,N)(\%b,N):=(\%c,+N,mod(\%b;\%a));$	[glass][mug]	[mod(mug;glass)]	if there is node %a with the feature N (noun) before a node %b with the feature N (noun): create a new node %c, with the feature N (noun); put both the node %a and the node %b inside this new node %c; and link them by a relation mod (modification) going from %b to %a)
17	$(\%a,N)(\%b,J):=(\%c,+N,mod(\%a;\%b));$	[books][available]	[mod(books;available)]	if there is node %a with the feature N (noun) before a node %b with the feature J (adjective): create a new node %c, with the feature N (noun); put both the node %a and the node %b inside this new node %c; and link them by a relation mod (modification) going from %a to %b)
18	$(\%a,V)(\%b,FPR)=(\%a,att=@reflexive);$	[love][herself]	[love.@reflexive]	if there is a node %a with the feature V (verb) before a node %b with the feature FPR(reflexive pronoun), then delete the node %b and assign the value @reflexive to the attribute "att" of %a)
19	$(\%a,N)(\%b,V,^AUX):=(\%c,+V,agt(\%b;\%a));$	[he][arrives]	[agt(arrives;he)]	if there is node %a with the feature N (noun) before a node %b with the features V (verb) and without the feature AUX (auxiliary): create a new node %c, with the feature V (verb); put both the node %a and the node %b inside this new node %c; and link them by a relation agt (agent) going from %b to %a)
20	$(\%a,AUX,att)(\%b,V):=(\%b,att=\%a);$	[will][arrive]	[arrive.@future]	if there is a node %a with the features AUX (auxiliary verb) and att (attribute) before a node %b with the feature V (verb), then delete the node %a and copy the value of the attribute "att" from %a to %b)

21	(%a,SAV,att)(%b,V):=(%b,att=%a);	[not][arrive]	[arrive.@not]	if there is a node %a with the features SAV (specifier adverb) and att (attribute) before a node %b with the feature V (verb), then delete the node %a and copy the value of the attribute "att" from %a to %b)
22	(%a,"is")(%b,GER):= (%b,att=@present,att=@progressive);	[is][arriving]	[arrive.@present.@progressive]	if there is a node %a that is the string "is" before a node %b with the feature GER (gerund), then delete the node %a and assign the values @present and @progressive to the attribute att of %b)
23	(%a,N)(%b,COP)(%c,J):=(%d,+V,aoj(%c;%a));	[this][is][beautiful]	[aoj(beautiful;this)]	if there is node %a with the feature N (noun) before a node %b with the feature COP (copula) and that is before a node %c with the feature J (adjective): create a new node %d, with the feature V (verb); put both the node %a and the node %c inside this new node %d; and link them by a relation aoj (predicative) going from %c to %a)

Examples

CASE	EXAMPLE	STRUCTURE	RULES	TRACE
zero attribute	car	N	No rule is necessary, provided that the corresponding words are in the dictionary	0 [car] (initial state = final state)
one attribute	books	N (inflected)	5	0 [books] (initial state) 1 [books.@pl] (after applying rule 05)
	a book	D (with att) + N	1 + 2	0 [a][][book] (initial state) 1 [a][book] (after applying rule 01) 2 [book.@indef] (after applying rule 02)
	very beautiful	SAV (with att) + J	1 + 3	0 [very][][beautiful] (initial state) 1 [very][beautiful] (after applying rule 01) 2 [beautiful.@plus] (after applying rule 03)
	it rains	“it” + V	1 + 4	0 [it][][rains] (initial state) 1 [it][rains] (after applying rule 01) 2 [rains] (after applying rule 04)
two attributes	the books	D + N (inflected)	1 + 2 + 5	0 [the][][books] (initial state) 1 [the][books] (after applying rule 01) 2 [books.@def] (after applying rule 02) 3 [book.@def.@pl] (after applying rule 05)
	the only book	D + D + N	1 + 2	0 [all][][the][][books] (initial state) 1 [the][only][book] (after applying rule 01 twice) 2 [the][book.@only] (after applying rule 02 for the first time) 3 [book.@only.@def] (after applying rule 02 for the second time)
three attributes	any more books	D + D + N (inflected)	1 + 2 + 5	0 [any][][more][][books] (initial state) 1 [any][more][books] (after applying rule 01 twice) 2 [any][books.@more] (after applying rule 02 for the first time) 3 [books.@more.@any] (after applying rule 02 for the second time) 4 [books.@more.@any.@pl] (after applying rule 05)
one relation	my book	POD + N	1 + 6	0 [my][][book] (initial state) 1 [my][book] (after applying rule 01) 2 [pos(book;my)] (after applying rule 06)
	my own book	POD + D + N	1 + 2 + 6	0 [my][][own][][book] (initial state) 1 [my][own][book] (after applying rule 01 twice) 2 [my][book.@own] (after applying rule 02) 3 [pos(book.@own;my)] (after applying rule 06)
	a book of mine	D + N + “of” + SPR	1 + 2 + 7	0 [a][][book][][of][][mine] (initial state) 1 [a][book][of][mine] (after applying rule 01 three times) 2 [book.@indef][of][mine] (after applying rule 02) 3 [pos(book.@own;my)] (after applying rule 07)
	all of the books	NPR + “of” + D + N (inflected)	1 + 2 + 5 + 8	0 [all][][of][][the][][books] (initial state) 1 [all][of][the][books] (after applying rule 01 three times) 2 [all][of][books.@def] (after applying rule 02) 3 [all][of][book.@def.@pl] (after applying rule 05) 4 [qua(book.@def.@pl;all)] (after applying rule 08)
	car of John	N + “of” + PPN	1 + 9	0 [car][][of][][John] (initial state) 1 [car][of][John] (after applying rule 01 twice) 2 [pos(car;John)] (after applying rule 09)

one relation	John's car	PPN + "'s" + N	1 + 10	0 [John]['s][][car] (initial state) 1 [John]['s][car] (after applying rule 01) 2 [pos(car;John)] (after applying rule 10)
	a book about cars	D + N + P (with att) + N (inflected)	1 + 2 + 5 + 11	0 [a][][book][][about][][cars] (initial state) 1 [a][book][about][cars] (after applying rule 01 three times) 2 [book.@def][about][cars] (after applying rule 02) 3 [book.@def][about][car.@pl] (after applying rule 05) 4 [mod(book.@def;car.@pl.@about)] (after applying rule 11)
	extremely beautiful	SAV (without att) + J	1 + 12	0 [extremely][][beautiful] (initial state) 1 [extremely][beautiful] (after applying rule 01) 2 [qua(beautiful;extremely)] (after applying rule 12)
	John and Mary	N + "and" + N	1 + 13	0 [John][][and][][Mary] (initial state) 1 [John][and][Mary] (after applying rule 01 twice) 2 [and(Mary;John)] (after applying rule 13)
	John or Mary	N + "or" + N	1 + 14	0 [John][][or][][Mary] (initial state) 1 [John][or][Mary] (after applying rule 01 twice) 2 [or(Mary;John)] (after applying rule 14)
	the beautiful car	D + J + N	1 + 15 + 2	0 [the][][beautiful][][car] (initial state) 1 [the][beautiful][car] (after applying rule 01 twice) 2 [the][mod(car;beautiful)] (after applying rule 15) 3 [mod(car;beautiful).@def] (after applying rule 02)
	the glass mug	D + N + N	1 + 16 + 2	0 [the][][glass][][mug] (initial state) 1 [the][glass][mug] (after applying rule 01 twice) 2 [the][mod(mug;glass)] (after applying rule 16) 3 [mod(mug;glass).@def] (after applying rule 02)
	the most expensive car	D + SAV + J + N	1 + 3 + 15 + 2	0 [the][][most][][expensive][][car] (initial state) 1 [the][most][expensive][car] (after applying rule 01 three times) 2 [the][expensive.@most][car] (after applying rule 03) 3 [the][mod(car;expensive.@most)] (after applying rule 15) 4 [mod(car;expensive.@most).@def] (after applying rule 02)
	all the books available	D + D + N (inflected) + J	1 + 2 + 5 + 17	0 [all][][the][][books][][available] (initial state) 1 [all][the][books][available] (after applying rule 01 three times) 2 [all][books.@def][available] (after applying rule 02 for the first time) 3 [books.@def.@all][available] (after applying rule 02 for the second time) 4 [books.@def.@all.@pl][available] (after applying rule 05) 5 [mod(book.@pl.@def.@all;available)] (after applying rule 17)
	he arrives	N + V	1 + 19	0 [he][][arrives] (initial state) 1 [he][arrives] (after applying rule 1) 2 [agt(arrives;he)] (after applying rule 19)
	he will arrive	N + AUX + V	1 + 20 + 19	0 [he][][will][][arrive] (initial state) 1 [he][will][arrive] (after applying rule 1 twice) 2 [he][arrive.@future] (after applying rule 20) 3 [agt(arrive.@future;he)] (after applying rule 19)
	he will not arrive	N + AUX + SAV + V	1 + 21 + 20 + 19	0 [he][][will][][not][][arrive] (initial state) 1 [he][will][not][arrive] (after applying rule 1 three times) 2 [he][will][arrive.@not] (after applying rule 21) 3 [he][arrive.@not.@future] (after applying rule 20) 4 [agt(arrive.@not.@future;he)] (after applying rule 19)
	he is arriving	N + "is" + GER	1 + 22	0 [he][][is][][arriving] (initial state) 1 [he][is][arriving] (after applying rule 1 twice) 2 [he][arrive.@present.@progressive] (after applying rule 22) 3 [agt(arrive.@present.@progressive;he)] (after applying rule 19)

one relation	this is beautiful	N + “is” + J	1 + 23	0 [this][][is][][beautiful] (initial state) 1 [this][is][beautiful] (after applying rule 1 twice) 2 [aoj(beautiful;this)] (after applying rule 23)
	I love myself	N + V + FPR	1 + 18 + 19	0 [I][][love][][myself] (initial state) 1 [I][love][myself] (after applying rule 1 twice) 2 [I][love.@reflexive] (after applying rule 18) 3 [agt(love.@reflexive;I)] (after applying rule 19)