

ColourPhrase-Unit

link: (((object ?t)))
m: ((equal-to-context ?s1)
(profile-colour-dimensions ?s2 ?s1)
(get-basic-colour-category-set ?cs)
(select-most-activated ?t ?s5))

syn-cat: ((constituent colourphrase))

ModifierColour-Unit

link: (((object-set ?s5) (object-set ?s2)))
c-link: (((col-cat ?c2) (col-cat-set ?cs)))
m: ((get-lightness-category-set ?cs2)
(scale-category-set-to-category ?cs3 ?cs2 ?c)
(profile-lightness-dimensions ?s4 ?s3))

syn-cat: ((constituent colour-category-set))
form: modifier-unit < colour-unit</pre>

Modifier-Unit

link: (((object-set ?s5) (object-set ?s4))) c-link: (((col-cat ?c2) (col-cat-set ?cs3))) m: ((filter-by-colour ?s5 ?s4 ?c2 ?cs3)

syn-cat: ((constituent colour-category))

Colour-Unit

link: (((object-set ?s3) (object-set ?s2)))
c-link: (((col-cat ?c) (col-cat-set ?cs)))
m: ((filter-by-colour ?s3 ?s2 ?c ?cs))

syn-cat: ((constituent colour-category))

Temno-Unit

link: (((colour-category ?c2)))
m: ((bind colour-category ?c2 [temno]))

form: "tëmno" syn-cat: ((lex-cat colour-category))

Rozovyj-Unit

link: (((colour-category ?c)))
m: ((bind colour-category ?c [rozovyj]))

form: "rozovyj" syn-cat: ((lex-cat colour-category))