Chapter 10
An Algorithm of Semantic Representation Assembly

Abstract An algorithm transforming a matrix semantic-syntactic representation $Matr$ of a natural language text into a formal expression $Semrepr \in Ls(B)$, where B is the conceptual basis being the first component of the used marked-up conceptual basis Cb, and $Ls(B)$ is the SK-language in the basis B, was called above *an algorithm of semantic assembly.* This chapter describes (a) an algorithm of semantic assembly *BuildSem*1, (b) an algorithm of semantic-syntactic analysis *SemSynt*1 being the composition of the algorithms *BuildMatr*1 and *BuildSem*1.

10.1 Initial Step of Building Semantic Representations of Input Texts

Let's consider the algorithm "$Preparation-to-constr-SemRepr$" – the first part of the algorithm of semantic assembly described in this chapter. The algorithm "$Preparation-to-constr-SemRepr$" delivers an initial value of the semantic representation (SR) of the input text; it is an initial value of the string *Semrepr* ("Semantic representation"). The form of this string depends on the form of the input text, i.e., on the value of the variable *kindtext* formed by the algorithm *BuildMatr*1.

The choice of the form of a semantic representation of the input text depends on the value of the string variable *kindtext*. To simplify the form of SR of the input texts, the existential quantifiers (when they are needed according to the approach described in Chap. 4) are not included in the formula but are implied.

It should be mentioned that the input language of the algorithm *BuildSem*1 is considerably broader than the output language of the algorithm *BuildMatr*1. The reason is that the algorithm *BuildSem*1 can deal with matrix semantic – syntactic representations corresponding to the NL-texts with participle constructions and attributive clauses. This possibility corresponds to the input language (a sublanguage of the Russian language) of the algorithm *SemSyn* developed by the author and published in [85].

V.A. Fomichov, *Semantics-Oriented Natural Language Processing*, IFSR International Series on Systems Science and Engineering 27, DOI 10.1007/978-0-387-72926-8_10,

10.1.1 Description of the Algorithm "Preparation-to-constr-SemRepr"

$$External specification$$

Input:

Rc – array – classifying representation of the input text;

Rm – array – morphological representation of the input text;

kindtext – string characterizing the form of the input text (the possible values of this string are $Stat$, Imp, $Genqs$, $Specqs.relat1$, $Specqs-relat2$, $Specqs.role$, $Specqs-quant1$, $Specqs-quant2$ (see Sect. 9.3);

mainpos – integer – the position of the interrogative word in the beginning of the text;

Matr – MSSR of the text.

Output:

Semrepr – string – an initial value of the semantic representation of the input text.

$$Algorithm\text{“}Preparation-to-constr-SemRepr\text{”}$$

Begin Case of kindtext from

$Stat:\ Semrepr := the\ empty\ string;$

Comment

Example 1. Let T1 = “Professor Igor Novikov teaches in Tomsk.”

Then initially

$$Semrepr := the\ empty\ string.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Situation\,(e1, teaching * (Time, \#now\#)$$

$$(Agent1, certn\ person * (Qualif, professor)$$

$$(Name, 'Igor')\,(Surname, 'Novikov') : x2)$$

$$(Place3, certn\ city * (Name1, 'Tomsk') : x3)).$$

End-of-comment

$Imp:\ Semrepr = (Command\,(\#Operator\#, \#Executor\#, \#now\#, e1)$

Comment

Example 2. Let T2 = “Deliver the container with the details to warehouse No. 3.”

Then initially

$$Semrepr := (Command\,(\#Operator\#, \#Executor\#, \#now\#, e1).$$

After the termination of algorithm *BuildSem*1

$$Semrepr = (Command\,(\#Operator\#, \#Executor\#, \#now\#, e1)$$

$$\wedge Target\,(e1, delivery1 * (Object1, certn\,container1 *$$

$$(Content1, certn\,set * (Qual-compos, detail)) : x1)$$

$$(Place2, certn\,warehouse * (Number, 3) : x2)))$$

End-of-comment

$$Genqs:\ Semrepr := Question\,(x1,\ (x1 \equiv Truth-value\,($$

Comment

Example 3. Let T3 = "Did the international scientific conference 'COLING' take place in Asia?"

Then initially

$$Semrepr := Question\,(x1,\ (x1 \equiv Truth-value\,(.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,(x1,\ (x1 \equiv Truth-value$$

$$(Situation\,(e1, taking\_place *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Event, certn\,conference * (Type1, international)$$

$$(Type2, scientific)\,(Name, 'COLING') : x2)$$

$$(Place, certn\,continent * (Name, 'Asia') : x3))))).$$

End-of-comment

$$Specqs-relat1,\ Specqs-relat2:$$

$$if\ kindtext = Specqs-relat1\ then\ Semrepr := 'Question\,(x1, '$$

$$else\ Semrepr := 'Question\,(S1, (Compos\,(S1, '$$

$$end-if$$

$$end$$

Comment

Example 4. Let T4 = "What publishing house released the novel 'The Winds of Africa'?"

Then initially

$$Semrepr := Question\,(.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,(x1, Situation\,(e1, releasing1 *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Agent2, certn\,publish-house : x1)$$

$$(Object3, certn\,novel1 * (Name1, 'The\,Winds\,of\,Africa') : x2))).$$

End-of-comment

Comment

Example 5. Let T5 = "What foreign publishing houses the writer Igor Somov is collaborating with?"

Then initially

$$Semrepr := Question\,(S1,\,(Qual-compos\,(S1,\,.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,(S1,\,(Qual-compos\,(S1,\,publish-house*$$

$$(Type-geographic,\,foreign))\wedge Description$$

$$(arbitrary\,publish-house*(Element,\,S1):y1,$$

$$Situation\,(e1,\,collaboration*(Time,\,\#now\#)$$

$$(Agent1,\,certn\,person*(Occupation,\,writer)$$

$$(Name,\,'Igor')\,(Surname,\,'Somov'):x1)$$

$$(Organization1,\,y1))))).$$

End-of-comment

$$Specqs-role:\ Semrepr := 'Question\,('$$

Comment

Example 6. Let T6 = "Who produces the medicine 'Zinnat'?"

Then initially

$$Semrepr := Question\,(\,.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,(x1,\,Situation\,(e1,\,production1*$$

$$(Time,\,\#now\#)\,(Agent2,\,x1)$$

$$(Product2,\,certn\,medicine1*(Name1,\,'Zinnat'):x2)))$$

End-of-comment

Comment

Example 7. Let T7 = "For whom and where the three-ton aluminum container has been delivered from?"

Then initially

$$Semrepr := Question\,(\,.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,((x1\wedge x2),$$

$$Situation\,(e1,\,delivery2*$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Recipient, x1)\,(Place1, x2)$$

$$(Object1, certn\,container1 * (Weight, 3/ton)$$

$$(Material, aluminum) : x3))).$$

End-of-comment

$Specqs - quant1 :\ Semrepr := {}'Question\,(x1, ((x1 \equiv Numb\,({}'; $

Comment

Example 8. Let T8 = "How many people did participate in the creation of the text-book on statistics?"

Then initially

$$Semrepr := {}'Question\,(x1, ((x1 \equiv Numb\,({}'.$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,(x1, ((x1 \equiv Numb(S1))$$

$$\wedge Qual - compos\,(S1, person) \wedge$$

$$Description\,(arbitrary\,person * (Element, S1) : y1,$$

$$Situation\,(e1, participation1 *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Agent1, y1)\,(Type - of - activity, creation1 *$$

$$(Product1, certn\,text - book1 * (Field1, statistics) : x2))))).$$

End-of-comment

$Specqs - quant2 :$

$typesit :=$ selected type *sit* ("situation") of the used conceptual basis;

$Semrepr := Question\,(x1, ((x1 \equiv Numb\,(S1)) \wedge Qual - compos\,(S1,$

$+ sortsit +{}') \wedge$

$Description\,(arbitrary{}' + sortsit + {}' * (Element, S1) : e1, {}'$

Comment

Example 9. Let T9 = "How many times Mr. Stepan Semenov flew to Mexico?"

Then initially

$$Semrepr := Question\,(x1, ((x1 \equiv Numb\,(S1))$$

$$\wedge Qual - compos\,(S1, sit) \wedge$$

$$Description\,(arbitrary\,sit * (Element, S1) : e1, .$$

After the termination of algorithm *BuildSem*1

$$Semrepr = Question\,(x1, ((x1 \equiv Numb\,(S1))$$

$$\wedge\, Qual - compos\,(S1, sit) \wedge$$

$$Description\,(arbitrary\,sit * (Element, S1) : e1,$$

$$Situation\,(e1,\ flight * (Time, certn\,moment *$$

$$(Earlier, \#now\#) : t1)\,(Agent1, certn\,person *$$

$$(Name, 'Stepan')\,(Surname, 'Semenov') : x2)$$

$$(Place2, certn\,country * (Name1, 'Mexico') : x3)))))'.$$

End-of-comment

$end - case - of$

end

10.2 Semantic Representations of Short Fragments of the Input Texts

This section describes the algorithm "Begin-of-constr-SemRepr" aimed at forming the one-dimensional arrays *Sembase* ("Semantic Base"), *Semdes* ("Semantic Description"), *Performers* ("Role performers in the situations mentioned in the input text") and an initial configuration of the two-dimensional array *Sitdescr* ("Situation Description").

10.2.1 Description of the Algorithm "Calculation-of-the-kind-of-case"

Let's start to consider such auxiliary algorithms that their interaction allows for forming the arrays *Sembase*, *Semdes*, *Performers*, and an initial configuration of the array *Sitdescr.*

Externalspecification

Input:

Rc – array – classifying representation of the input text;

*k*1 – number of the row of the classifying representation of the input text, i.e. the ordered number of the text unit;

Arls – two-dimensional array – the projection of the lexico-semantic dictionary *Lsdic* on the input text *T*;

Matr – MSSR of the text;

*class*1 – string defining the class of the text unit;

*sem*1 – semantic unit corresponding to *k*1 text unit.

Output:

casemark – string - takes the values *case*1 – *case*7 depending on the form of the fragment of the classifying representation of the text.

Algorithm

*Begin if class*1 = *adject then casemark* := $'Case1'$
end − *if*
*if class*1 = *constr then casemark* := $'Case2'$
end − *if*
*if class*1 = *noun*
then if Rc[*k*1 + 1, *tclass*] = *name then casemark* := $'Case3'$
*else numb*1 := *Matr* [*k*1, *qt*]
Comment
The number corresponding to the noun in the position *k*1
End-of-comment
end − *if*
ref := *certn*
Comment
Referential quantifier
End-of-comment
*beg*1 := *sem*1 [1]
Comment
The first symbol of the string sem1, considering each element of the primary informational universe $X(B(Cb(Lingb)))$ and each variable from $V(B)$ as one symbol
End-of-comment
*setind*1 := 0
Comment
Mark of individual, not the set of individuals
End-of-comment
*len*1 := *Length* (*sem*1)
if (*len*1 ≥ 2) *and* (*sem*1 [2] = $'set'$)
*then setind*1 := 1
end − *if*
Comment
*sem*1 [2] – 2nd symbol of the structured semantic unit *sem*1, if we interpret the elements of primary informational universe $X(B(Cb))$ as symbols, where *Cb* – the used marked-up conceptual basis (m.c.b.)
End-of-comment
if ((*numb*1 = 0) *or* (*numb*1 = 1)) *and* (*beg*1 = *ref*) *and* (*setind*1 = 0)
Comment
i.e. *Rc* [*k*1, *unit*] – the designation of the individual, not the set
End-of-comment
then casemark := $'Case4'$
Comment
Example: “Belgium”
End-of-comment
end − *if*
if(*numb*1 = 0) *and* (*beg*1 ≠ *ref*) *and* (*sem*1 is not the designation of the function from $F(B(Cb))$, where *Cb* is the used marked-up conceptual basis)

$then\ begin\ loc1 := Rc[k1, mcoord],\ md1 := Rm[loc1, morph];$
$if\ (Number(md1) = 1)\ then\ casemark := 'Case5'$
Comment
Example: “conference”
End-of-comment
$else$
Comment
i.e., in the case when $Number(md1) = 2$
End-of-comment
$casemark := 'Case6'$
Comment
Examples: “5 articles”, “3 international conferences”
End-of-comment
$end - if$
end
$end - if$
$if\ (numb1 = 0)\ and\ (setind1 = 1)$
Comment
Example: “with Indian ceramics”
End-of-comment
$then\ casemark := 'Case7'$
$end - if$
$end - if$
end

10.2.2 External Specification of the Algorithms BuildSemdes1 – BuildSemdes7

Input:

Rc – array – classifying representation of the input text;

$k1$ – integer – number of a row from Rc;

$Arls$ – two-dimensional array – projection of the lexico-semantic dictionary $Lsdic$ on the input text T;

$Matr$ – MSSR of the text;

$sem1$ – string – semantic unit, corresponding to the text unit with the number $k1$;

$casemark$ – string taking the values $case1$ – $case7$ depending on the form of the fragment of the classifying representation of the input text;

arrays $Sembase$, $Semdes$, $Performers$.

Output:

arrays $Sembase$, $Semdes$, $Performers$ containing the blocks for forming the final value of the variable $Semrepr$ – semantic representation of the input text.

10.2.3 Description of the Algorithms BuildSemdes1 – BuildSemdes7

Description of the auxiliary algorithms

*Function Transform*1

Arguments:

s – string of either the form $r(z, b)$, where r – the designation of the binary relation and b – the second attribute of the relation, or of the form $(f(z) \equiv b)$, where f – name of function with one variable,

b – string designating the value of the function,

z – letter "z" interpreted as variable.

Value:

String t of the form (r, b) in the first case and of the form (f, b) in the second case.

Example 1. Let T1 = "How many two-ton aluminum containers came from Penza?" Then the linguistic basis can be defined in the following way:

$$for\ k1 = 2\ sem1 := (Weight\,(z) \equiv 2/ton),$$

$$Transform1\,(sem1) = (Weight, 2/ton),$$

$$for\ k1 = 3\ sem1 := Material\,(z, aluminum),$$

$$Transform1\,(sem1) = (Material, aluminum).$$

*Algorithm BuildSemdes*1

Begin
Comment
The reflection of semantics of adjectives in the array *Sembase*
End-of-comment
if $Matr\,[k1 - 1, nattr] = 0$
Comment
To the immediate left from the $k1$ position there are no adjectives, i.e., on the $k1$ position there is the first adjective from the group of consequent adjectives
End-of-comment
then $Sembase\,[k1] := Transform1\,(sem1)$
else
Comment
To the immediate left from the $k1$ position there is an adjective
End-of-comment
$Sembase\,[k1] := Sembase\,[k1 - 1] + Transform1\,(sem1)$
Comment

The sign "+" here designates the concatenation, i.e., the operation of addition of the string to the right

End-of-comment

$end - if$

end

Example 2. During processing the question T1 = "How many 2-ton aluminum containers came from Penza?" by the algorithm *BuildSemdes*1 the following operators will be used:

$$Sembase\,[2] := (Weight, 2/ton),$$

$$Sembase\,[3] := (Weight, 2/ton)\,(Material, aluminum).$$

*Algorithm BuildSemdes*2

Begin

Comment

Processing of a construct

End-of-comment

$Sembase\,[k1] := sem1;\ Performers\,[k1] := Rc\,[k1, unit]$

Comment

Example: $Performers\,[k1] := '720/km'$

End-of-comment

end

Description of the algorithm *BuildSemdes*3 **("Processing of the names")**

Purpose: building the semantic representation (SR) of the fragment of the text T, which is a combination of the form "noun + expression in quotation marks or apostrophes."

Application condition: in the $k1$ position there is a noun, in the $k1 + 1$ position there is an expression in quotation marks or apostrophes.

Example 3. Let T2 = "Who produces the medicine 'Zinnat'?" Then applying this algorithm will result in the following operator:

$$Performers\,[k1] := certn\,medicine1 * (Name, 'Zinnat').$$

Algorithm

Begin $name := Rc\,[k1 + 1, unit];$

if ($Performers\,[k1]$ does not include the symbol $*$)

$then\ Performers\,[k1] := Performers\,[k1] + '*(Name, ' + name + ')'$

$else\ Performers\,[k1] := Performers\,[k1] + '(Name, ' + name + ')'$

$end - if$

end

*Algorithm BuildSemdes*4

Begin

Comment

Processing of the proper names
Example of the context – published in Belgium
End-of-comment

$Sembase\,[k1] := sem1;\ Semdes\,[k1] := Sembase\,[k1]$
$Var1 := Matr\,[k1, mark];\ Performers\,[k1] := Semdes\,[k1] + ' : ' + var1$

Comment
Example:

$$Performers\,[k1] := 'certn\,country * (Name, 'Belgium') : x2'$$

End-of-comment
End

*Algorithm BuildSemdes*5

Begin
Comment
Processing of the common nouns
Example of the context: "published a monograph"
End-of-comment

$if\ Matr\,[k1, nattr] \geq 1$

Comment
There are adjectives to the left
End-of-comment

$then\ Sembase\,[k1] := sem1 + ' * ' + Sembase\,[k1 - 1]$
$else\ Sembase\,[k1] := sem1$
$end - if$
$Ref := 'certn';\ Semdes\,[k1] := ref\ sem1$
$Var1 := Matr\,[k1, mark];\ Performers\,[k1] := Semdes\,[k1] + ' : ' + var1$

Comment
Example 1:

$$Performers\,[k1] := 'certn\,monograph : x3'$$

End-of-comment
Comment
Example 2:

$$Performers\,[k1] := 'certn\,printer * (Form, ink - jet) : x4'$$

End-of-comment
end

*Algorithm BuildSemdes*6

Begin
Comment
Processing of a combination of nouns designating a set of objects.
Example of the context – "5 three-ton containers are delivered"

End-of-comment

$numb1 := Matr[k1, qt]$; $Sembase[k1] := sem1$

$if\ numb1 > 0\ then\ Semdes[k1] := 'certn\,set * (Quant, ' + numb1 + ')(Compos, ' + Sembase[k1] + ')'$

$else\ Semdes[k1] := 'certn\,set * (Compos, ' + Sembase[k1] + ')'$

$end - if$

$beg1 := sem1[1]$

Comment

First symbol of the string $sem1$ considering elements of the primary informational universe $X(B(Cb(Lingb)))$ and variables as symbols

End-of-comment

$Var1 := Matr[k1, mark]$; $Var2 := Varsetmember(var1)$;

Comment

The variable $var2$ designates an arbitrary element of the set with the mark $var1$.

Example: $If\ var1 = S2\ then\ var2 = y2$

End-of-comment

$Performers[k1] := 'arbitrary' + beg1 + ' * Elem('+Semdes[k1] + ' : ' + var1 + ') : ' + var2$

Comment

Example:

$$Performers[k1] := 'arbitrary\,container1 *$$

$$(Elem, certn\,set * (Quant, 5)$$

$$(Compos, container1 * (Weight, 3/ton)) : S1) : y1'$$

End-of-comment

end

Description of the algorithm *BuildSemdes*7 **(“Processing of the collective nouns”)**

Purpose: Building the fragment of the semantic representation (SR) of the text T which includes collective nouns (“Indian ceramics,” “Italian shoes,” etc.). Application condition: there is a collective noun on the $k1$ position.

Example 4. Let T3 = “Where three containers with Indian ceramics came from?” The word “ceramics” in the question T3 has the ordered number 6. The linguistic basis may be defined in a way that would lead to the execution of the following operators as a result of algorithm *BuildSemdes*7 application:

$$Semdes[6] := certn\,set * (Compos, article\_of\_ceramics *$$

$$(Geographic\_localization, certn\,country * (Name, 'India'))),$$

$$Performers[6] := certn\,set * (Compos, article\_of\_ceramics *$$

$$(Geographic\_localization, certn\,country * (Name, 'India'))) : S1.$$

$$The description of auxiliary algorithms$$

$$Function Transform2$$

Arguments:

s – string representing the semantics of an adjective or sequence of adjectives. For example, s may represent the semantics of an adjective "Indian" and be the string $(Geographic\_localization, certn\,country * (Name, 'India'))$.

String t – the structured semantic unit corresponding to the collective noun and including the substring $(Qual - compos$, (for example, t may correspond to the noun "ceramics" and be the string $certn\,set * (Qual - compos, article\_of\_ceramics)))$.

Value:

String u formed as follows.

Let $pos1$ – position of the first left bracket "("in the substring $(Qual - compos$ of the string s, and let $pos2$ – position of the right bracket")" closing the bracket opened in the position $pos1$.

Let h – substring of the string t placed between the substring $(Qual - compos$ and the right bracket in the position $pos2$.

Then u can be derived from the string t by replacing the substring h with the string $h * s$.

Example 5. In the context of the question T3 = "Where three containers with Indian ceramics came from?" let

$$s = (Geographic\_localization, certn\,country * (Name1, 'India')),$$

$$t = certn\,set * (Qual - compos, article\_of\_ceramics).$$

$$Then\ h = article\_of\_ceramics,$$

$$u = Transform2\,(s, t) = certn\,set *$$

$$(Qual - compos, article\_of\_ceramics *$$

$$(Geographic\_localization, certn\,country * (Name, 'India'))).$$

$$Algorithm\,BuildSemdes7$$

$Begin\ if\ Rc\,[k1 - 1, tclass] \neq adjective$
$\quad then\ Semdes\,[k1] := sem1$
$\quad else\ prop1 := Sembase\,[k1 - 1]$
$\quad\quad Semdes\,[k1] := Transform2\,(prop1, sem1)$

Comment

Example:

$$Semdes\,[k1] := certn\,set * (Qual - compos, article\_of\_ceramics *$$

$$(Place - of - production, certn\,country * (Name1, 'India')))$$

End-of-comment

end − *if*

end

10.2.4 Description of the Algorithm ProcessSit

The algorithm *ProcessSit* is designed for representation of the structured units of conceptual level (in other words, semantic units), corresponding to the situations mentioned in the input text by means verbs or participles, in the array *Sitdescr.*

External specification

Input:

Rc – array – classifying representation of the input text T;

$k1$ – number of the row of the classifying representation of the text T, i.e., the ordered number of the verbal text unit;

Rm – array – morphological representation of the text T;

kindtext – string designating the form of the text T;

Arls – projection of the lexico-semantic dictionary *Lsdic* on the input text T;

Matr – MSSR of the text;

Sitdescr – initial configuration of the array which includes descriptions of the situations mentioned in the text;

timevarnumb – maximal number of the variable designating the moment of time.

Output:

Sitdescr – transformed configuration of the array used for describing the situations mentioned in the text.

Description of the auxiliary algorithms

*Function Numb*1

Argument:

v – the string of the form RS, where R – letter of Latin alphabet and S – string representing an integer.

Value:

N – the integer associated with the string S.

Example 6. For the string $e3$ $Numb1(e3)$ is the number 3.

Function Stringvar

Arguments:

R – letter of Latin alphabet,

N – an integer

Value:

String of the form RS, where S – string representing the integer N.

Example 7. If R - letter "t", N – number 2, then $Stringvar\,(R, N)$ is the string $t2$.

Function Time

Argument:

M – set of the values of morphological properties related to the arbitrary verbal form $vbform$ (the verb or participle)

Value:

Figure "1" if $vbform$ corresponds to the past; figure "2" if $vbform$ corresponds to the present; figure "3" if $vbform$ corresponds to the future.

Algorithm ProcessSit

$Begin\ \ pos1 := Rc\,[k1, mcoord]$

$armorph := Rm\,[pos1, morph]$

Comment

The set of the values of morphological properties related to the verbal form in the position $k1$

End-of-comment

$timevarnumb := timevarnumb + 1$

$Vartime := Stringvar\,('t', timevarnumb)$

$time1 := Time\,(armorph)$

$Case\ of\ time1\ from$

$'1' : timesit := '(Time, certn moment * (Earlier, \#now\#) : ' + vartime + ')'$

$'2' : timesit := '(Time, \#now\#)'$

$'3' : timesit := '(Time, certn moment * (Later, \#now\#) : ' + vartime + ')'$

$End - case - of$

$linesit := Matr\,[k1, locunit];\ concsit := Arls\,[linesit, sem]$

$var1 := Matr\,[k1, mark];\ numbsit := Numb1\,(var1)$

$if\ (kindtext = Imp)\ and\ (numbsit = 1)$

$then\ Sitdescr\,[numbsit, expr] := 'Target\,(' + var1 + ', ' + concsit + ' * '$

$else\ Sitdescr\,[numbsit, expr] := 'Situation\,(' + var1 + ', ' + concsit + ' * ' + timesit$

$end - if$

Comment

Example 1:

$$Sitdescr\,[1, expr] := 'Situation\,(e1, releasing1 * '$$

$$(Time, certn moment * (Earlier, \#now\#) : t1)'$$

Example 2:

$$Sitdescr\,[1, expr] := 'Target\,(e1, delivery1 *$$

$$(Object1, certn\,container1 : x1)$$

$$(Place2, certn\,warehouse * (Number, 4) : x2))'$$

End-of-comment
end

10.2.5 Description of the Algorithm "Begin-of-constr-SemRepr"

External specification

Input:
Rc – array – classifying representation of the input text *T*;
Rm – array – morphological representation of the text *T*;
Arls – projection of the lexico-semantic dictionary *Lsdic* on the text *T*;
kindtext – string representing the form of the input text *T*;
mainpos – integer – position of the interrogative word in the beginning of the text;
Matr – MSSR of the text.
Output:
Semrepr – string – initial value of the semantic representation of the input text;
Performers – one-dimensional array containing semantic representations of the short fragments of the input text.

Algorithm

Begin
Preparation − to − constr − SemRepr
(*Rc*, *Rm*, *Matr*, *kindtext*, *mainpos*, *Semrepr*)
Comment
Example:
if kindtext = genqs
(closed question, i.e., the question with the answer "Yes/No")
then Semrepr := $'Question\,(x1, (x1 \equiv Truth - value\,('$
End-of-comment
*loop for k*1 *from* 1 *to nt*
Comment
Formation of the arrays *Sembase*, *Semdes*, *Performers*, and the initial configuration of the array *Sitdescr*
End-of-comment
$class1 := Rc\,[k1, tclass]$
$if\ (class1 \neq construct)\ and\ (class1 \neq name)\ and\ (class1 \neq marker)$
$then\ loc1 := Matr\,[k1, locunit];\ sem1 := Arls\,[loc1, sem]$
$end - if$

$if\ (class1 = verb)\ or\ (class1 = participle)$
$then\ ProcessSit\ (Rc, Rm, Arls, Matr, Sitdescr, timevarnumb)$
$end - if$
$if\ (class1 - of\ the\ set\ \{adject, constz, noun\})\ element$
$then\ Calculation - of\ the\ form - of\ case$
$(Rc, k1, Arls, Matr, class1, sem1, casemark1);$
$end - if$
$Case\ of\ casemark1\ from$
$'Case1' :\ BuildSemdes1\ (List1),$
where $List1$ - the list of parameters
$Rc, k1, Arls, Matr, Sembase, Semdes, Performers, casemark1;$
$'Case2' :\ BuildSemdes2\ (List1);$
$'Case3' :\ BuildSemdes3\ (List1);$
$'Case4' :\ BuildSemdes4\ (List1);$
$'Case5' :\ BuildSemdes5\ (List1);$
$'Case6' :\ BuildSemdes6\ (List1);$
$'Case7' :\ BuildSemdes7\ (List1)$
$End - case - of$
$End - of - loop$
End

10.3 Development of the Algorithm "Representation-of-situations"

Let's develop an algorithm constructing separate semantic representations (being K-representations) of the situations mentioned by the verbs and participles in the input NL-text. The principal sourse of data for this algorithm is a matrix semantic – syntactic representation (MSSR) of the input text.

10.3.1 The Key Ideas of the Algorithm

By the time this algorithm is applied, the array *Performers* and the initial configuration of the situation description array *Sitdescr* are already formed. The array *Performers* includes semantic units (primary and compound) corresponding to the constructs (digital values of the parameters), nouns and combinations of the forms "Group of adjectives + Noun," "Number + Noun," "Number + Group of adjectives + Noun," "Cardinal number + Noun," "Cardinal number + Group of adjectives + Noun."

The number of filled-in rows of the array *Sitdescr* equals the number of verbs and participles in the text. The mark of the situation is placed in the column mark (the link to the MSSR *Matr* is realized through the elements of this column). The

column *expr* (abbreviation of "the expression") is designed for keeping the semantic descriptions of the situations (events) mentioned in the text.

In the considered algorithm $Representation-of-situations$, the transformation of the information is carried out in two consequent stages. The first stage represents a loop for m from 1 to nt, where m is number of a row of the classifying representation Rc, nt – quantity of the text elements. In this loop the information about the semantic-syntactic relations in the combinations "Verbal form (verb or participle two or gerund) + Dependent fragment of the sentence" is represented in the elements of the column *expr* of the array $Sitdescr$ (each of these elements is a description of the certain situation mentioned in the input text).

Here the dependent fragment of the sentence means a construct or a noun or a combination of one of the following forms: "Group of adjectives + Noun," "Number + Noun," "Number + Group of adjectives + Noun," "Cardinal number + Noun," "Cardinal number + Group of adjectives + Noun."

The examples of the dependent fragments of the sentence could be the expressions "in 2002nd year," "European scientific publishers," "two-ton containers," "5 containers," "12 personal computers."

Example 1. Let Qs1 = "How many two-ton containers with Indian ceramics delivered from Novorossiysk were shipped to the firm 'Sail'?"

Then during the first stage of applying the algorithm, the expressions that are the elements of the column *expr* of the array $Sitdescr$ are supplemented with the information about the semantic-syntactic relations in the combinations "delivered + two-ton containers," "delivered + Novorossiysk," "shipped + two-ton containers," "shipped + firm "Sail."

As a result of this stage of processing the matrix semantic-syntactic representation of the question Qs1 by the algorithm, the array $Sitdescr$ acquires the following configuration:

Table 10.1 The structure of the array $Sitdescr$ at the intermediate stage of constructing a semantic representation of the input text

mrk	expr
$e1$	$Description1$
$e2$	$Description2$

where

$$Description1 = Situation\,(e1, delivery2 * (Time,$$
$$certn\,moment * (Earlier, \#now\#) : t1)$$
$$(Object1, arbitrary\,container1 * (Element, certn\,set *$$
$$(Qual-compos, container1 * (Weight, 2/ton) : S1) : y1)$$
$$(Place1, certn\,city * (Name1, 'Novorossiysk') : x2)),$$

$$Description2 = Situation\,(e2, shipment1 * (Time,$$

$$certn\,moment * (Earlier, \#now\#) : t2)\,(Object1, y1)$$
$$(Recipient, certn\,firm * (Name1, 'Sail') : x3))\,.$$

The auxiliary array $Used$ of the length nt helps to avoid repeating the semantic representations of the same expression (designating an object or a set of objects) in the column $expr$ of the array $Sitdescr$. Initially for each m from 1 to nt, $Used\,[m] = 0$. If for a certain k, the string $Performers\,[k]$ is a fragment of a certain row of the array $Sitdescr$, then $Used\,[k] := 1$.

That is why not the string $Performers\,[k]$ but the variable being the ending of the string $Performers\,[k]$ is being added to other rows of the array $Sitdescr$, if needed.

For example, the first row of the array $Sitdescr$ (Table 10.1) includes the expression

$$arbitrary\,container1 * (Element, certn\,set *$$

$$(Qual - compos, container1 * (Weight, 2/ton) : S1) : y1,$$

being an element of the one-dimensional array $Performers\,[3]$. That is why the second row of the array $Sitdescr$ uses the variable $y1$ instead of this expression.

Let's call the fragments of the sentence directly governed by the verbal forms (verbs or participles) the dependent elements of the first level.

The second stage of applying the algorithm "$Representation - of - situations$" is searching in the MSSR $Matr$ for such constructs or combinations with noun that are directly governed by the dependent elements of the first level, i.e., searching for the dependent elements of the second level.

For example, in the question Qs1, the fragment "two-ton containers" is governed by the verbal form "were shipped," therefore, this fragment is a dependent element of the first level. At the same time, the combination "two-ton containers" governs the noun group "with Indian ceramics."

The formal representation $descr1$ of the information conveyed by the combination of the form "Dependent element of the first level X + Dependent element of the second level Y" is being added from right to the element $Sitdescr\,[k, expr]$ with the help of conjunction. Here k here is the ordered number of the situation, its participant is designated by the dependent element X of the first level.

The auxiliary array $Conj$ is initially filled in with zeros. If, as a result of the second stage of applying the algorithm, a certain expression is being added from the right to the element $Sitdescr\,[k, expr]$ with the help of conjunction, then $Conj\,[k] := 1$. The value 1 serves as the signal to put the element $Sitdescr\,[k, expr]$ in the brackets before including this element into the semantic representation of the input text.

Example 2. As a result of the second stage of applying the algorithm to the question Qs1, the following expression will be added from the right to the element $Sitdescr\,[1, expr]$ via conjunction:

$$Content1\,(y1, certn\,set * (Qual - compos, ceramic\_article *$$

$$(Geographic\_localization, certn\,country * (Name1, 'India'))))).$$

As the conjunction was used, the element $Conj\,[1]$ will get the value 1. Finally the array *Sitdescr* will acquire the following configuration:

Table 10.2 The structure of the array *Sitdescr* at the final stage of constructing a semantic representation of the input text

mrk	expr
$e1$	$SemRepr1$
$e2$	$SemRepr2$

where

$$SemRepr1 = Situation\,(e1, delivery2 * (Time,$$

$$certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Object1, arbitrary\,container1 * (Element, certn\,set *$$

$$(Qual - compos, container1 * (Weight, 2/ton) : S1) : y1)$$

$$(Place1, certn\,city * (Name1, 'Novorossiysk') : x2)),$$

$$\vee Content1, (y1, certn\,set *$$

$$(Qual - compos, ceramic\_article *$$

$$(Geographic\_localization, certn\,country * (Name, 'India'))),$$

$$SemRepr2 = Situation\,(e2, shipment1 * (Time,$$

$$certn\,moment * (Earlier, \#now\#) : t2)\,(Object1, y1)$$

$$(Recipient, certn\,firm * (Name1, 'Sail') : x3)).$$

10.3.2 Description of the Algorithm "Representation-of-situations"

The algorithm considered below is developed for representing the information about the situations (events) mentioned in the text in the array *Sitdescr*.

External specification of the algorithm

Input:
Rc – array – classifying representation of the input text;

nt – integer – length of the input text (the number of rows in *Rc* and *Matr*);

kindtext – string – designation of the form of the input text;

Matr – MSSR of the text;

Performers – two-dimensional array including the semantic images of the situation paricipants;

maxnumbsit – integer – number of the situations mentioned in the text.

Output:

Sitdescr – array representing the information about the situations mentioned in the input text;

$Used[1 : nt]$ – one-dimensional array for keeping the signs of multiple usage of the structured semantic unit in the rows of the array *Sitdescr*;

$Conj[1 : maxnumbsit]$ – one-dimensional array for keeping the signs of conjunction usage in the rows of the array *Sitdescr.*

$$Algorithm\text{“}Representation-of-situations\text{”}$$

Begin

Comment

The processing of direct semantic connections between the verbal form and a noun or a construct

End-of-comment

$Loop\ for\ j1\ from\ 1\ to\ nt\ Used[j1] := 0$

$End-of-loop$

$Loop\ for\ j2\ from\ 1\ to\ maxnumbsit\ Conj[j2] := 0$

$End-of-loop$

$Loop\ for\ m\ from\ 1\ to\ nt$

Comment

The first passage of the strings in *Rc*

End-of-comment

$Class1 := Rc[m, tclass]$

$If\ (class1 = noun)\ or\ (class1 = constr)$

$Then\ begin\ d := nmasters[m];$

Comment

The quantity of the text units governing the text unit *m* is found

End-of-comment

$If\ d > 0$

$Then\ loop\ for\ q\ from\ 1\ to\ d$

$Begin\ p1 := Matr[m, posdir, q];\ class2 := Rc[p1, tclass]$

$If\ (class2 = verb)\ or\ (class2 = participle)$

$Then\ var2 := Matr[p1, mark];$

Comment

Mark of the situation designated by the governing verbal form

End-of-comment

$numbsit := Numb(var2);$

$role := Matr[m, reldir, q]$

$if\ (class1 = noun)$
$then\ if$ *kindtext* is not included in the set $\{specqs - relat2, specqs - quant1\}$
$then\ if\ Used[m] = 0\ then\ actant := Performers[m]$
$else\ if\ (class1 = noun)$
$then\ actant := Varbuilt(Matr[m, mark])$
$end - if$
$if\ (class1 = constr)\ then\ actant := Rc[m, unit])$
$end - if$
$end - if$
$else$
Comment
If *kindtext* is included in the set $\{specqs - relat2, specqs - quant1\}$
End-of-comment
$if\ ((Used[m] = 1)\ or\ ((Used[m] = 0)\ and\ (Matr[m, mark] = S1))$
$then\ actant := Varbuilt(Matr[m, mark])$
$else\ actant := Performers[m]$
$end - if$
$end - if$
Comment
Kindtext
End-of-comment
$end - if$
Comment
$class1 = noun$
End-of-comment
Comment
$Varbuilt(x_j) = x_j;\ Varbuilt(S_j) = y_j$
End-of-comment
$Sitdescr[numbsit, expr] := Sitdescr[numbsit, expr] + '(' + role + ',' + actant + ')'\ end - if$
end
$end - of - loop\ \{on\ q\}$
$end - if$
end
$end - if$
$End - of - loop$
Comment
End of a loop on m
End-of-comment
Comment

Example 3. Let's consider the question Qs1 = "Where five aluminum two-ton containers came from?" Let's assume that before applying the algorithm "$Representation - of - situations$", the following relationship took place:

$$Sitdescr[1, expr] = Situation(e1, delivery2 *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1).$$

Then after applying the algorithm "$Representation-of-situations$" with a certain choice of the marked-up conceptual basis the following relationship will be true:

$$Sitdescr\,[1, expr] = Situation\,(e1, delivery2 *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Place1, x1)\,(Object1, arbitrary\,container1 *$$

$$(Element, certn\,set * (Numb, 5)$$

$$(Qual-compos, container1 * (Weight, 2/ton)$$

$$(Material, aluminum)) : S1) : y1$$

End-of-comment

loop for k1 from 1 to nt

Comment

Filling in of *Sitdescr* – the second passage of *Rc* – the processing of the text units governed by nouns depending on a verbal form

End-of-comment

$class1 := R_c\,[k1, tclass]$

if $(class1 = noun)$ *or* $(class1 = constr)$

Comment

The examples of combinations: "containers with Indian ceramics" (class1 = noun), "with the lamps of 60 watt" (class1 = constr)

End-of-comment

then $posmaster := Matr\,[k1, posdir, 1]$

Comment

The position of the word "containers" or "lamps" for above combinations

End-of-comment

if $posmaster = 0$ *then output* ("$Wrong\,text$")
else $class2 := Rc\,[posmaster, tclass]$
  if $class2 = noun$
  then $rel1 := Matr\,[k1, reldir, 1]$
    $varmaster := Matr\,[posmaster, mark]$
    if $(class1 = constr)$ *then* $arg2 := Sembase\,[k1]$
    end − *if*
    if $(class1 = noun)$
    then $vardep := Matr\,[k1, mark]$
      if $Used\,[k1] = 0$ *then* $arg2 := Performers\,[k1]$
      else $arg2 := vardep$
      end − *if*
    end − *if*
    letter := the first symbol of the string *varmaster*

Comment

The variable *varmaster* corresponds to the noun governing the text unit in the position $k1$, moreover this noun is governed by the verbal form – the verb or participle

End-of-comment

$$if\ letter = 'x'\ then\ descr1 := rel1\,(varmaster, arg2)$$
$$else$$

Comment

i.e. if letter = "S"

End-of-comment

$$semhead := \text{the first element of the } Sembase\,[posmaster]$$
$$descr1 := rel1 + '(arbitrary' + semhead + ' * (Element, ' + varmaster + '), ' + arg2 + ')'$$
$$end - if$$

Comment

Then the expression $descr1$ is being added with the help of conjunction $\wedge$ to the right of the expression $Sitdescr\,[numbsit, expr]$ characterizing the considered situation with the number $numbsit$.

End-of-comment

Example 4. Let Qs2 = "What writers from Tomsk did participate in the conference?" and after the first loop (for m from 1 to nt) the array *Sitdescr* has the following configuration:

Table 10.3 The structure of the array *Sitdescr* after the loop with m changing from 1 to nt

mrk	expr
e1	SemRepr1

where

$$SemRepr1 = Situation\,(e1, participation * (Time,$$
$$certn\,moment\,[*\,(Earlier, \#now\#) : t1)\,(Agent1,$$
$$arbitrary\,writer\,[*\,(Element, certn\,set *$$
$$(Qual - compos, writer) : S1) : y1)$$
$$(Event1, certn\,conference : x1))\,.$$

In the second loop (for $k1$ from 1 to nt) the operator

$$Descr1 := Geographic\_localization$$

$$(y1, certn\,city * (Name, 'Tomsk') : x2)$$

is being executed , and then the array *Sitdescr* acquires the new configuration

mrk	expr
e1	SemRepr2

where

$$SemRepr2 = Situation\,(e1,\ participation * (Time,$$
$$certn\,moment\,[*\,(Earlier,\ \#now\#)\ :\ t1)\,(Agent1,$$
$$arbitrary\,writer * (Element,\ certn\,set *$$
$$(Qual-compos,\ writer)\ :\ S1)\ :\ y1)$$
$$(Event1,\ certn\,conference\ :\ x1))\wedge$$
$$Geographic\_localization\,(y1,\ certn\,city *$$
$$(Name1,\ 'Tomsk')\ :\ x2)\,.$$

Comment
End of example
End-of-comment

$possit := Matr\,[posmaster,\ posdir,\ 1]$
$varsit := Matr\,[possit,\ mark];\ numbsit := Numb\,(varsit)$
$Sitdescr\,[numbsit,\ expr] := Sitdescr\,[numbsit,\ expr] + '\wedge' + descr1$
$If\ Conj\,[numbsit] = 0\ then\ Conj\,[numbsit] := 1$
$end-if$

Comment
The sign of using the conjunction in the string $Sitdescr\,[numbsit,\ expr]$
End-of-comment

$end-if$
$end-if$
$end-if$
$end-of-loop$
end
Comment
End of the algorithm
End-of-comment

10.4 Final Stages of Developing an Algorithm of Semantic Representation Assembly

Let's consider the final stages of developing the algorithm *BuildSem*1 constructing a K-representation of an input NL-text, proceeding mainly from its matrix semantic-syntactic representation *Matr*.

10.4.1 External Specification of the Algorithm "Final-operations"

The considered algorithm is designed for reflecting the information conveyed by the situations description array *Sitdescr* in the final value of the string *Semrepr* (semantic representation of the input text).

Input and Output Data

Input:

Rc – array – classifying representation of the input text;

Rm – array – morphological representation of the input text;

kindtext – string characterizing the form of the input text;

mainpos – integer – position of the interrogative word in the beginning of the text;

Matr – MSSR of the text;

Performers – two-dimensional array containing the semantic images of the situation's participants;

numbsit – integer – number of the situations mentioned in the input text, i.e. the number of filled-in rows of the array *Sitdescr*;

numbqswd – integer – number of interrogative words in the input text;

Sitdescr – array representing the information about the situations mentioned in the input text;

Semrepr – string – initial value of the semantic representation of the text.

Output:

Semrepr – string – final value of the semantic representation (SR) of the input text.

Descriptions of auxiliary algorithms

Function "*Right*"

Arguments:

*pos*1 – the number of a row of the classifying representation *Rc* of the input text, i.e., the position number of the input text unit;

*class*1 – string – designation of the class of the text unit.

Value:

*pos*2 – position of the closest from right (to the position *pos*1) text unit of the class *class*1.

Function "*Stringvar*"

Arguments:

R – letter of Latin alphabet,

N – an integer.

Value:

String of the form *RS*, where *S* is the string denoting the integer *N*.

Example 1. If *R* is the letter "t," and *N* is the number 2, then *Stringvar* (R, N) is the string $t2$.

Algorithm

*Begin pos*2 := *pos*1
loop − *until pos*2 := *pos*2 + 1; *class*2 := *Rc* [*pos*2, *tclass*]
exit − *when* (*class*2 = *class*1)
end

10.4.2 Algorithm "Final-operations"

Algorithm

begin loop for k from 1 *to maxnumbsit*
event := *Sitdescr* [*k*, *expr*]
if Conj [*k*] = 1 *then event* := ′(′ + *event* + ′)′
end − *if*
if k = 1 *then situations* := *event*
else situations := *situations* + ′ ∧′ + *event*
end − *if*
end − *of* − *loop*
if maxnumbsit > 1 *then situations* := ′(′ + *situations* + ′)′
end − *if*
Comment
The string *situations* describes the situations mentioned in the input text
End-of-comment
Case of kindtext from
Stat : *Semrepr* := *Situations*
Comment
Example 2. Let T1 = "Professor Igor Novikov teaches in Tomsk."
Then initially

$$Semrepr := the\,empty\,string.$$

After the termination of the algorithm *BuildSem*1

$$Semrepr = Situation\,(e1, teaching * (Time, \#now\#)$$
$$(Agent1, certn\,person * (Qualif, professor)$$
$$(Name, 'Igor')\,(Surname, 'Novikov') : x2)$$
$$(Place3, certn\,city * (Name1, 'Tomsk') : x3)).$$

End-of-comment
Imp : *Semrepr* := *Semrepr* + ′ ∧′ + *Situations* + ′)′
Comment

Example 3. Let T2 = "Deliver the case with details to the warehouse 3."
Then initially

$$Semrepr := '(Command\,(\#Operator\#, \#Executor\#, \#now\#, e1)'.$$

After the termination of the algorithm *BuildSem*

$$Semrepr = (Command\,(\#Operator\#, \#Executor\#, \#now\#, e1)$$
$$\wedge Goal\,(e1, delivery1 * (Object1, certn\,case *$$
$$(Content1, certn\,set * (Qual - compos, detail)) : x1)$$
$$(Place2, certn\,warehouse * (Number, 3) : x2)))$$

End-of-comment
$Genqs:\ Semrepr := Semrepr + Situations + '))))'$;
Comment
Example 4. Let T3 = "Did the international scientific conference 'COLING' ever take place in Asia?"
Then initially

$$Semrepr := 'Question\,(x1, (x1 \equiv Truth - value\,('.$$

After the termination of the algorithm *BuildSem*

$$Semrepr := 'Question\,(x1, (x1 \equiv Truth - value$$
$$(Situation\,(e1, taking\_place * (Time, certn\,moment$$
$$* (Earlier, \#now\#) : t1)\,(Event, certn\,conference$$
$$* (Type1, international)\,(Type2, scientific)$$
$$(Name, "COLING") : x2)\,(Place, certn\,continent *$$
$$(Name1, 'Asia') : x3))))).$$

End-of-comment
$Specqs - relat1,\ Specqs - relat2:$
$if\ Semrepr = 'Question\,(x1, '$
Comment
Example 5. Let T4 = "What publishing house has released the novel 'The Winds of Africa'?"
End-of-comment
$then\ Semrepr := Semrepr + Situations + ')'$
Comment
Example 6. For the question T4

$$Semrepr := 'Question\,(x1, Situation\,(e1, releasing1 *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Agent2, certn\,publish-house : x1)\,(Object3, certn\,novel1$$

$$*(Name1, 'The\,Winds\,of\,Africa') : x2)))'$$

End-of-comment

else

Comment

i.e., if $Semrepr = 'Question\,(S1, (Qual-compos\,(S1, '$

End-of-comment

$posmainnoun := Right\,(mainpos, noun)$

Comment

Example 7. Let T5 = "What foreign publishing houses the writer Igor Somov is collaborating with?"

Then $mainpos = 1$ (the position of the question word "what"),

$posmainnoun := 3$ (the position of the word group "publishing houses")

End-of-comment

$sem1 := Sembase\,[posmainnoun]$

if $sem1$ does not include the symbol $'*'$

then $semhead := sem1$

else $loc1 := Matr\,[posmainnoun, locunit]$

$semhead := Arls\,[loc1, sem]$

$end-if$

Comment

Example 8. For the question T5

$sem1 := publish-house * (Type-geographic, foreign)$

$semhead := publish\_house$

End-of-comment

$end-if$

$Semrepr := Semrepr + sem1 + ') \wedge Description\,(arbitrary' + semhead + ' * (Element, S1) : y1, ' + Situations + '))'$

Comment

Example 9. For the question T5 = "What foreign publishing houses the writer Igor Somov is collaborating with?"

$$Semrepr := Question\,(S1, (Qual-compos\,(S1, publish-house$$

$$*(Type-geographic, foreign)) \wedge Description$$

$$(arbitrary\,publish-house * (Element, S1) : y1,$$

$$Situation\,(e1, collaboration * (Time, \#now\#)$$

$$(Agent1, certn\,person * (Occupation, writer)\,(Name, 'Igor')$$

$$(Surname, 'Somov') : x1)\,(Organization1, y1))))))).$$

End-of-comment

$Specqs - role$:

Comment

Example 10. Let T6 = "Who produces the medicine 'Zinnat'?"

Then initially

$$Semrepr := Question(.$$

End-of-comment

$Unknowns := 'x1'$

$If\ numbqswd > 1$

$Then\ loop\ for\ k\ from\ 1\ to\ numbqswd - 1$

$Vrb := Stringvar('x', k)$

$Unknowns := Unknowns + '\wedge' + vrb$

$end - of - loop$

$Unknowns := '(' + Unknowns + ')'$

$end - if$

$Semrepr := Semrepr + Unknowns + ',' + Situations + ')'$

Comment

End of $Specqs - role$

End-of-comment

Comment

Example 11. After the termination of the algorithm *BuildSem*1 for the question T6 = "Who produces the medicine 'Zinnat'?"

$$Semrepr = Question(x1, Situation(e1, production1$$

$$*(Time, \#now\#)\,(Agent2, x1)$$

$$(Product2, certn\,medicine1 * (Name1, 'Zinnat') : x2)))$$

End-of-comment

Comment

Example 12. Let T7 = "For whom and where the three-ton aluminum container has been delivered from?"

Then initially

$$Semrepr := 'Question('.$$

After the termination of the algorithm *BuildSem*1

$$Semrepr = Question((x1 \wedge x2), Situation(e1, delivery2$$

$$*(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Recipient, x1)\,(Place1, x2)\,(Object1, certn\,container *$$

$$(Weight, 3\,ton))\,(Material, aluminum) : x3))).$$

End-of-comment

$Specqs - quant1$

$posmainnoun := Right\,(mainpos, noun)$

Comment

Example 13. Let T8 = "How many people did participate in the creation of the textbook on statistics?".

Then $mainpos := 1$ (the position of the question word group "how many"),

$posmainnoun := 2$ (the position of the word "people")

End-of-comment

$sem1 := Sembase\,[posmainnoun]$

$if\ sem1$ does not include the symbol $'*'$

$then\ semhead := sem1$

$else\ loc1 := Matr\,[posmainnoun, locunit]$

$semhead := Arls\,[loc1, sem]$

Comment

Example 14. For the question T8 $sem1 := person$, $semhead := person$

End-of-comment

$end - if$

$Semrepr := Semrepr + sem1 + ') \wedge Description\,(arbitrary' + semhead + ' * (Element, S1) : y1, ' + Situations + '))'$

Comment

Example 15. For the question T8 = "How many people did participate in the creation of the textbook on statistics?"

$$Semrepr := 'Question\,(x1, ((x1 \equiv Numb\,(S1))$$

$$\wedge Qual - compos\,(S1, person) \wedge Description$$

$$(arbitrary\,person * (Element, S1) : y1,$$

$$Situation\,(e1, participation1 *$$

$$(Time, certn\,moment * (Earlier, \#now\#) : t1)$$

$$(Agent1, y1)\,(Type - of - activity, creation1 *$$

$$(Product1, certn\,textbook1 * (Field1, statistics) : x2)))))'.$$

End-of-comment

$Specqs - quant2 :\ Semrepr := Semrepr + Situations + ')))'$

Comment

Example 16. Let T9 = "How many times Mr. Stepan Semenov flew to Mexico?"

Then initially

$$Semrepr := Question\,(x1, ((x1 \equiv Numb\,(S1))$$

$$\wedge Qual - compos\,(S1, sit) \wedge$$

$$Description\,(arbitrary\,sit * (Element, S1) : e1, .$$

After the termination of the algorithm *BuildSem*

$$Semrepr = Question\,(x1,\,((x1 \equiv Numb\,(S1))$$

$$\wedge\, Qual - compos\,(S1,\,sit) \wedge$$

$$Description\,(arbitrary\,sit * (Element,\,S1) : e1,$$

$$Situation\,(e1,\,flight * (Time,\,certn\,moment *$$

$$(Earlier,\,\#now\#) : t1)\,(Agent1,\,certn\,person *$$

$$(Name,\,'Stepan')\,(Surname,\,'Semenov') : x2)$$

$$(Place2,\,certn\,country * (Name,\,'Mexico') : x3))))).$$

End-of-comment
end

10.4.3 An Algorithm of Semantic Representation Assembly

Combining the auxiliary algorithms developed above in this chapter, let's construct the algorithm *BuildSem*1 ("Assembly-of-SemRepr").

*External specification of the algorithm BuildSem*1

Input:
Rc – array – classifying representation of the input text *T*;
Rm – array – morphological representation of the text *T*;
nt – integer – length of the text *T* (the quantity of rows in *Rc* and *Matr*);
kindtext – string characterizing the form of the input text;
mainpos – integer – position of the interrogative word in the beginning of the text;
Matr – MSSR of the text;
Arls – projection of the lexico-semantic dictionary *Lsdic* on the input text.
Output:
Performers – array;
Sitdescr – array;
Semrepr – string – representation of the input text (semantic representation of the input text which is an expression in a certain standard K-language).

*Algorithm BuildSem*1

Begin Preparation – to – constr – SemRepr
(Rc, Rm, Matr, kindtext, mainpos, Semrepr)
Begin – of – constr – SemRepr
(Rc, Rm, Matr, kindtext, numbqswd, Arls, Performers, Sitdescr, Semrepr)

$Representation-of-situations$
$(Rc, Matr, Performers, Sitdescr)$
$Final-operations$
$(Rc, kindtext, numbqswd, Matr, Performers, Sitdescr, Semrepr)$
end

Thus, the algorithm *BuildSem*1 ("Assembly-of-SemRepr") developed in this chapter transforms an MSSR of a question, command, or statement from the practically significant sublanguages of English, German, and Russian languages into its semantic representation being an expression of the SK-language determined by the considered marked-up conceptual basis – a component of a linguistic basis.

10.5 A Multilingual Algorithm of Semantic-Syntactic Analysis of Natural Language Texts

Let's define the algorithm of semantic-syntactic analysis of NL-texts (the questions, commands, and statements) as the composition of the developed algorithms *BuildMatr*1 and *BuildSem*1.

10.5.1 Description of the Algorithm SemSynt1

*External specification of the algorithm SemSynt*1

Input:
Lingb – linguistic basis;
lang – string variable with the values *English*, *German*, *Russian*;
T – text from the set $Texts(Tform(Lingb))$, where $Tform$ is the text-forming system being one of the components of the linguistic basis *Lingb*.
Output
Semrepr – string – K-representation of the input text (a semantic representation of the input text being an expression of a certain SK-language).

*Algorithm SemSynt*1

Begin
*BuildMatr*1 $(T, Rc, Rm, Arls, kindtext, Matr, mainpos, numbqswd)$
*BuildSem*1 $(Rc, Rm, Arls, kindtext, Matr, mainpos,$
$numbqswd, maxnumbsit, Semrepr)$
End

Example. Let T1 be the question "What Russian publishing house released in the year 2007 the work on multi-agent systems 'Mathematical Foundations of Repre-

senting the Content of Messages Sent by Computer Intelligent Agents' by professor Fomichov?", T1germ be the same question in German "Welcher Russischer Verlag im Jahre 2007 die Arbeit ueber Multi-Agenten Systeme 'Mathematische Grundlagen der Representierung des Sinnes der Nachrichten Schickt von Intelligenten Computeragenten' von Professor Fomichov veroeffentlicht hat?", and let T1russ be the same question in Russian (we use here a Latin transcription of Russian texts) "V kakom rossiyskom izdatelstve byla opublikovana v 2007-m godu rabota po mnogoagentnym sistemam "Matematicheskie Osnovy Predstavleniya Soderzhaniya Poslaniy Komputernykh Intellektualnykh Agentov?" Professora Fomichova?"

Then a linguistic basis *Lingb* can be defined so that at the different stages of processing T1 or T1germ or T1russ the algorithm *SemSynt*1 will create

- two-dimensional arrays *Rc* and *Rm* – the classifying and morphological representations of the input text (see the examples for T1 in Sect. 8.1);
- the two-dimensional arrays $Arls$ – the projection of the lexico-semantic dictionary *Lsdic* on the input text, $Arvfr$ – the projection of the dictionary of verbal – prepositional semantic-syntactic frames Vfr on the input text, and $Arvfr$ – the projection of the dictionary of verbal – prepositional semantic-syntactic frames Vfr on the input text (see the examples for T1 in Sect. 8.2);
- a matrix semantic-syntactic representation (MSSR) $Matr$ of the input text (see the example in Sect. 8.3).

At the final stage of processing T1, the procedure *BuildSem*1 will assemble a K-representation of the text T1 or T1germ or T1russ, this SR can be the K-string *Semrepr* of the following form (independently on the input sublanguage of NL):

$$Question\,(x1,\, Situation\,(e1,\, releasing1 * (Agent2,$$

$$certn\,publish\_house * (Country,\, Russia) : x2)$$

$$(Time,\, 2007/year)\,(Product1,\, certn\,work2 *$$

$$(Name1,\, Title1)\,(Field1,\, multi\_agent\_systems)$$

$$(Authors,\, certn\,person * (Qualif,\, professor)$$

$$(Surname,\, 'Fomichov') : x4) : x3)))),$$

where $Title1$ = "Mathematical Foundations of Representing the Content of Messages Sent by Computer Intelligent Agents."

10.5.2 Some Possible Directions of Using the Algorithm SemSynt1

An important peculiarity of the algorithm *SemSynt*1 is that it directly discovers the conceptual relationships between the meanings of the text's fragments without constructing a pure syntactic representation of an input text.

The adaptation to a concrete sublanguage of natural language is achieved with the help of the string variable *lang* with the values *English*, *German*, *Russian*, this variable is one of the input data of the algorithm *SemSynt*1.

The formal model of a linguistic database described in Chap. 7 and the algorithm of semantic-syntactic analysis *SemSynt*1 can be used, in particular, as a basis for designing NL-interfaces to

- recommender systems of various companies offering the goods and services by means of Web-portals;
- intelligent databases with the information about the manufactured, exported, and imported products;
- autonomous intelligent transport-loading systems (robots);
- intelligent databases with the information about the objects stored in certain sections of an automated warehouse;
- advanced question-answering systems of digital libraries;
- intelligent databases with the information about the scientists, their publications, and projects with their participation;
- the advanced computer systems transforming the NL-annotations of various documents into their semantic annotations being the expressions of SK-languages;
- intelligent databases with the information about the sportsmen and sport competitions.