A formal notation that re-expresses natural language sentence structure

by

Erik Sandewall

UPPSALA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCES

UPPSALA UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCES
REPORT NR 12
APRIL, 1968

A formal notation that re-exp⁻esses natural language sentence structure

by

Erik Sandewall

The research reported here was partly supported by the Swedish Natural Science Research Council (Dnr 2711-2 ans 6826) and by the Advanced Research Projects Agency of the Office of the U.S. Secretary of Defence (SD-183).

Notation.

In this report we shall make heavy use of a formal notation. Formulas refer to items, which may be either:

(a) individual, indivisible elements, or

(b) sets of items, or

(c) tuples (or lists, in the sense of the word used e.g. in the programming language lisp, or ordered sets) of items.

The present section explains the notation, and is intended mainly for reference.

Formulas in text. To distinguish formulas from surrounding English text, we use the following conventions:

(1) if we refer to the meaning of the formula, we enclose it by slashes;

(2) if, on the other hand, we refer to the formula itself, we enclose it by (double) quotes;

Example: We could synonymously write:

this proves that /a $\leq$ b/

or

this proves that "a $\leq$ b" holds .

(3) a formula which occupies its own line is interpreted as if it were surrounded by quotes;

(4) exceptions from (1) - (3) may occur when no misunderstandings can result.

Distinguished objects. /true/, /false/, and /nil/ are three items /nil/ is the empty set.

Set notation. The following infixes are used as usual:

$$\cup \quad \cap \quad \in \quad \subseteq \quad =$$

Besides, expressions of the following types are used as usual:

$$\{\beta_1, \beta_2, \cdots, \beta_k\} \qquad \{\beta \mid \Phi(\beta)\}$$

Logical connectives. The following prefix and infixes are used as usual:

$$\neg \quad \wedge \quad \vee \quad \supset \quad \equiv$$

Tuples. $\langle \beta . \gamma \rangle$ is the ordered pair whose components are $/\beta/$ and $/\gamma/$.
$\langle \beta_1, \beta_2, ---, \beta_k \rangle$ is the ordered k-tuple whose members are $/\beta_1/$, $/\beta_2/$, etc.
Pairs and tuples are interrelated through

$$\langle \ \rangle \quad = \text{nil} \quad \text{(i.e. the empty set is also the unique 0-tuple);}$$

$$\langle \beta_1, --- \beta_{k-1}, \beta_k \rangle = \langle \langle \beta_1, ---, \beta_{k-1} \rangle . \beta_k \rangle$$

Products. $\lhd \phi . \psi \rhd$ is the set of all pairs of members of the sets $/\phi/$ and $/\psi/$.

Formally,

$$\lhd \phi . \psi \rhd = \{ \langle \beta . \gamma \rangle | \ \beta \in \phi \ \wedge \ \gamma \in \psi \}$$

We define $\lhd \phi_1, ---, \phi_{k-1}, \phi_k \rhd$ by analogy.

Forms. If $/\lambda/$ is a set, then $\lambda(\beta_1, \beta_2, --- , \beta_k)$ is the unique element $/\gamma/$ which satisfies

$$\langle \beta_1, \beta_2, ----- \beta_k, \gamma \rangle \in \lambda$$

If no such γ exists, the value is /nil/; if several exist, the value is arbitrary.

Mappings. Let ϕ and ψ be sets. When we write

$$\lambda : \quad \phi \twoheadrightarrow \psi$$

we mean to assert that

(1) $/\lambda/$ is a set and can be used as specified for forms;
(2) $\lambda \subseteq \lhd \phi . \psi \rhd$
(3) $\forall \ \beta \in \phi \ \exists$ exactly one $\gamma \in \psi : \langle \beta . \gamma \rangle \in \lambda$

These three conventions make it possible to use λ as indicated under "Forms". Notice, therefore, that what looks like a function symbol in our formalism always has a second interpretation as a set of tuples.

Quantifiers. Written as $\forall$ and $\exists$.

Additonal comment. Curled brackets are used to enclose literature references as well as sets. Ordinary parentheses are used to indicate the scope of quantifiers and infixed functions. Like in most programming languages, we often use many-letter words for individual constants and variables, e.g. "true", "nil".

Introduction.

In various reports {Bohnert 1967a, McCarthy 1961a, Raphael 1964a} it has been suggested that the language of formal logic be used to re-express the meaning of natural-language(*) sentences. If this can be done, it will have several obvious advantages:

(+1) Semantic theory, developed for formal logic, can be applied to the semantics problems of natural language;

(+2) Question-answering (= fact-retrieval) systems could be constructed as a combination of a routine for translating NL sentences into logical language, and a deduction routine.

The translation process appears straight-forward at first. A sentence like "Peter gives the box to John" could be represented as a relation "give(Peter, box14, John)" where "box14" is the name of this particular box. We must therefore introduce one formal-language predicate (like the three-placed "give") for each NL verb. If some component of the NL sentence is missing, we have to insert a dummy variable. For example, "Peter gives away the box" would go into "$\exists$t: give(Peter, box14, t)".

However, some facts about natural language make this approach quite cumbersome:

(-1) NL makes very heavy use of descriptive phrases (noun-phrases starting with the definite article). To give a reasonably compact translation of these, it appears necessary to extend the formal logic notation with the description operator {McCarthy 1961a} or some equivalent notation. This immediately makes it less trivial to write a deduction routine as outlined in (+2).

(-2) To every natural-language verb, there is a large number of possible sentence components. A sentence with the verb "give" may contain components that indicate

(1) the one who gives
(2) what he gives
(3) to whom he gives
(4) when he gives
(5) where he gives

(*) Natural language (NL) shall refer to natural English or similar languages.

(-2) cont.

(6) why he gives (his psychological motivation)

(7) the (real) cause of his action

(8) his attitude ("reluctantly", "grandly", etc.)

and several others. Of course, we very rarely encounter any single sentence that contains all these components, but each of these components occur in some sentences, and it is therefore necessary to reserve one argument of the formal-logic predicate for each of them. This means that the translation of any ordinary sentence gets messed up with a large number of dummy symbols, with the same function as the t in the above translation of "Peter gives away the box".

(-3) In NL, sentences are used for the dual purpose of (1) stating that a certain activity takes place, and (2) describing the activity. The latter case occurs when one sentence is a component in another, e.g. "Peter therefore suggested that John should go and look for the stolen car"; "The plan failed because the train was on time for once". For such applications, predicate calculus does not work at all.

Let us try to translate the first of the two examples. With the obvious meaning of the symbols, we could have

"suggest(Peter, go-look-for(John, car5))".

The sub-expression "go-look-for(John, car5), has of course a truth-value. The predicate "suggest" therefore takes its first argument to be a person, and (in this case) its second argument to be a truth-value. This is a complication, but a moderate one.

But worse, suppose "go-look-for(John, car5)" has the value <u>true</u>, and the phrase "isyellow(moon)" also has the value <u>true</u>. Then the two phrases

"suggest(Peter, go-look-for(John, car5))" and

"suggest(Peter, isyellow(moon))"

have the same truth-value, as the arguments of "suggest" have the same value in both cases. This situation is obviously untenable.

It seems to have been overlooked that there is an <u>alternative</u> way of looking at NL sentences, which immediately eliminates difficulties (-2) and (-3). We can make the convention that sentences describe <u>activities</u>, which are objects just like Peter or car5. Let the sentence "Peter gives the box to John" describe the activity Pbj.

There would then be a sentence-to-subject relationship from Pbj to Peter, a sentence-to-direct-object relationship from Pbj to the box, etc. With the obvious notation, the whole phrase could be expressed as

$$\exists \text{Pbj}: [\text{Pbj} \in \text{gives} \wedge \text{subject(Pbj, Peter)} \wedge \text{direct-object(Pbj,box14)} \wedge \text{indirect-object(Pbj, John)}] .$$

"gives" would then of course be the set of all activities of giving.(*)

The advantages of this notation over the notation where verbs go into predicates, are:

(+1) We do not have to keep a fixed number of components for each verb. "Peter gives away the box" translates into

$$\exists \text{Pbj}: [\text{Pbj} \in \text{gives} \wedge \text{subject(Pbj Peter)} \wedge \text{direct-object(Pbj,box14)}]$$

(+2) Adverbials of the type "reluctantly" be handled by sets of activities (e.g. the set of all activities that are performed reluctantly), rather than by putting them as arguments. "Peter reluctantly gives away the box" then goes into

$$\text{Pbj}: [\text{Pbj} \in \text{gives} \wedge \text{Pbj} \in \text{reluctant} \wedge \text{subject(Pbj,Peter)} \wedge \text{direct-object(Pbj,box14)}]$$

This notation easily handles cases like "Peter reluctantly but good-humouredly gives away the box" which make difficulties if we try to shoehorn the whole description of Peter's attitude into one argument position.

(+3) As there is no formal distinction between activities and other objects, we can have sentence-to-direct-object or sentence-to-its-cause relationships from one activity to another. For example, "Peter suggested that John should go and look for the car" could translate into

$$\exists \text{Ps, Jglc}: [\text{Ps} \in \text{suggests} \wedge \text{subject(Ps, Peter)} \wedge \text{direct-object(Ps, Jglc)} \wedge \text{Jglc} \in \text{go-away} \wedge \text{Jglc} \in \text{look-for} \wedge \text{subject(Jglc, John)} \wedge \text{direct-object(Jglc, car5)}]$$

(*)

(*)

The use of the concepts "subject", "direct object", etc. may be disturbing to some readers. Let us explain, therefore, that the two-place relations between activities and their particular objects can in principle be written in a more neutral manner as "R_1", "R_2", etc.
The formula would then be

$$\exists \text{Pbj}: [\text{Pbj} \in \text{gives} \wedge \text{R1(Pbj,Peter)} \wedge \text{R2(Pbj,box14)} \quad \text{R3(Pbj,John)}]$$

The R_i are relations between <u>objects</u> (in a generalized sense of the word) in the real world, not between words or other linguistic entities. - In most examples, we shall rename these relations as "subject", "direct-object", increase legibility. This does not imply that we will slavishly follow patterns of traditional grammar; indeed, the relation symbols derived from there will soon become insufficient.

(+3) cont.

(The reader may wonder what happens if John never actually goes to look för the car. To cope with hypothetical activities, we need two sets for each verb; one set (suggests, looks-for) of the actually occurring activities of the verb, and one set (suggest, look-for) consisting of both actual and hypothetical activities.)

These advantages seem important enough to justify looking closer into the "activity" concept. The activity was introduced as an artificial construct, and it is therefore not clear at this point what the activity really is. This makes it difficult for us to give well-founded answers to questions like

(a) Does the sentence "Peter gives the ball and the box to John" refer to one or two activities?

(b) If it consists of one activity, what is the relationship of that activity to the activity of "Peter gives the box to John"? Is there a part-whole relationship?

(c) On the other hand, if it refers to two activities, how many activities in "Liza sees that Peter gives the ball and the box to John" have "Liza" as subject? If two, are there also two activities for "He did A because of B and C"? If so, we can trace backward through previous causes of any activity, and it will split into infinitely many activities. If not, what is the difference between the two "and"s?

It is of course meaningless to ask what an activity really is. The only valid criterium for answering the questions above must be that of finding a simple and consistent notation. However, we are unlikely to come up with a consistent notation unless we have acquired a more precise idea of this activity concept.

In the present work, we shall (1) make this activity concept more precise, and (2) on that basis introduce a couple of function symbols and predicate symbols that enable us to describe activities within the predicate calculus.

Chapter 1

Patterns and objects in an artificial world

1.0 The overall approach.

Let us first outline the approach in this report.

In chapter 1, we study an artificial world, viz. a two-dimensional universe of cellular automata. This "world" is comparable to the "world" of a chess game in the sense that there is a discrete, two-dimensional "space" (like squares on the checkerboard) and a discrete, one-dimensional "time" (like the sequence of moves). However, in the cellular-automaton world there are no players and no game situation. For our purpose, this is an advantage.

In the cellular world, every square or cell is in one of a finite number of states at any point in time. The state of a square at time /t/ is a function of the state of the same square at time /t-1/, and of the states of neighboring squares at times /t/ and /t-1/. By suitable selection of the next-state function, it is possible to construct a cellular world which can be described through analogy with our physical world. Thus we can construct "paths" of squares which are permanently in the same state; "signals" which "move" along the "paths" and "branch" at "intersections" between paths, etc. An example of such a next-state function, was given in a previous report, {Sandewall 1967d} . Reading of that report is recommended, as many examples will be drawn from there. From now on, it will be referred to as $.

Phenomena in the cellular world of $ can clearly be described by natural-language sentences. In this chapter we shall introduce a formalism for describing such phenomena, which may serve as an alternative description language.

Our first task is to define in a formal manner what we mean by "objects", "activities", and "patterns" in the cellular world. To facilitate reading, let us even now give some examples of what we intuitively mean by those words. The examples are drawn from the cellular world defined in $.

Objects.

Ex. 1 A signal is an object which moves from one cell to the next at the rate of (usually) one cell per time unit, and which "causes" the cell in which it is located at any given time, to be in state /s/.

Ex. 2 A path is an object which does not move around in space. It appears as a non-branched, diagonal sequence of cells in state /x/.

Ex. 3 An intersection is also a non-moving object and appears as the intersection of two paths.

Activities.

Ex. 1 Let us watch one sequence of snapshots, which display one particular signal moving down one particular path. We are then watching one activity.

Ex. 2 Let us watch how one signal reaches one intersection, and splits into two signals which go one branch each. Then, again, we have been watching one activity.

Patterns.

Ex. 1 Let us collect all activities, where some signal moves down some path, into a set. This set of activities is called a pattern.

Ex. 2 Consider the set of all activities where two signals move down the same path, three steps apart. This is also a pattern.

Very roughly, we can say that our formal "objects" correspond to natural-language objects and persons; our formal "activities" correspond to natural-language sentences; and our formal "patterns" correspond to natural-language verbs. It turns out that formal descriptions of phenomena in the cellular world often have a structure similar to the natural-language descriptions.

In chapter 2, this correspondence is put to use. We take ordinary, natural language sentences that describe our physical world, and assume that they too can be translated into the formalism developed for the cellular world. In other words, we assume that the physical world can (for these purposes) be sufficiently well approximated by a (perhaps very complicated) cellular world. The translation from natural-language sentences to the formalism must proceed through analogy with the examples for the cellular world.

Through this approach, we obtain a method to translate natural-language sentences into a well-founded formalism. It also becomes possible to set up and prove laws for the formalism through reference to the cellular-world interpretation. Such laws, when applied to translates from natural language, make it possible to carry out common-sense deductions in a formal manner.

Chapter 1, which we have now started, is rather long, because the definition of patterns is recursive: An activity (= a member of a pattern) may be recognized, either because certain squares in the cellular world are in certain states, or because some other, lower-level activities have previously been recognized in certain cells. It requires some writing to work that out formally. It is essential that we do this, however. One important reason is, it gives us an excuse for approximating the physical world with a cellular one. (Explanation: it becomes possible to consider natural-language sentences about the physical world as descriptions of very-high-level activitites in a cellular world with a quantum-sized grid). Other good reasons will later become apparent.

1.1 Initial definitions.

The purpose of this section is to introduce a systematic notation for our variant of a cellular space. Most terms and conventions can be intuitively understood through refernece to $.

We shall assume two basic sets of elements, and construct further sets from them.

Assumption. /Allint/ is the set of all non-negative integers.(Actually, any well-ordered set will do). "i", "j", "t" will be used as variables for integers.

Assumption. /Allstat/ is a set of elements(*) which serve as states on the board. "stat" will be used as a variable for states. "x", "s", "e", and "o" are constants which denote states in the cellular world defined in $.

Assumption. /Borderstat/ is a member of /Allstat/.

Definition. /Board/ is the set of all 2-tuples $\langle i,j \rangle$ for which /i+j/ is even. (Every such 2-tuple is called a cell. Clearly,

$$\text{Board} \subseteq \lhd \text{Allint, Allint} \rhd \quad) .$$

Assumption. /ilo/, /ihi/, /jlo/, and /jhi/ are constant integers. We assume

$$\text{ilo} < \text{ihi} \wedge \text{jlo} < \text{jhi}$$

Definition. /Picture/ is the set of all $\langle i, j \rangle \in$/Board/ which satisfy

$$\text{ilo} \leq i \leq \text{ihi} \wedge \text{jlo} \leq j \leq \text{jhi}$$

Definition. /Border/ is the set of all $\langle i, j \rangle$ in /Board/ which satisfy /$i = \text{ilo} \vee i = \text{ihi} \vee j = \text{jlo} \vee j = \text{jhi}$/.

Definition. Space-time = $\lhd$ Picture . Allint $\rhd$.

Definition. A position is a member $\langle i,j,t \rangle$ of /Space-time/. (It is thought about as the "glimpse" of cell $\langle i,j \rangle$ at time /t/).

(*) An element is an item which is neither a set nor a tuple, and for which we do not assume any structure. Cf the introductory section on notation.

<u>Definition.</u> <u>A next-state function</u> is mapping

$$\text{nxtstatfunc} : \langle \text{Allstat}, \text{Allstat}, \text{Allstat}, \text{Allstat}, \text{Allstat} \rangle \rightarrow \text{Allstat}$$

which satisfies the following restriction:

$\forall$ a, b, c, d $\in$ Allstat:

nxtstatfunc(a,b,Borderstat,c,d) = Borderstat

<u>Definition:</u> An <u>initial state function</u> is a mapping

$$\text{initstatfunc}: \langle \text{Picture} \cdot \{0\} \rangle \rightarrow \text{Allstat}$$

which satisfies the following restriction:

$\forall$ $\langle$i,j$\rangle$ $\in$ Border: initstatfunc(i,j,0) = Borderstat

<u>Remark:</u> The restriction on the initial state function guarantess that every cell on the border will initially be in /Borderstat/, and the restriction on the next-state function guarantees that it will then stay so for every t .

<u>Definition:</u> A <u>state function</u> is a mapping

$$\text{statfunc}: \langle \text{Picture} \cdot \text{Allint} \rangle \rightarrow \text{Allstat}$$

which satisfies the following restrictions:

1. /statfunc/ has some initial state function as a subset;
2. $\langle$i,j,t$\rangle$ $\in$ Space-time: statfunc(i,j,t+1) =

```
nxtstatfunc( statfunc(i-1,j-1,t+1),
             statfunc(i+1,j-1,t+1),
             statfunc(i,j,t),
             statfunc(i-1,j+1,t),
             statfunc(i+1,j+1,t) )
```

<u>Comment.</u> For the cellular world described in \$, we have

Allstat = { o ,x,s,e, Borderstat }

/nxtstatfunc/ is given on page 7 of \$, except for the part that deals with /Borderstat/. In this paper, we will assume throughout that /Allstat/ and /nxtstatfunc/ have been chosen like in \$. /ilo, ihi, jlo, jhi, initstatfunc/ are variable and will be chosen anew for each example.

<u>Some additional terminology</u>. The position $\langle i,j,t \rangle$ <u>is in</u> /Picture/ iff / $\langle i,j \rangle \in$ Picture/. The set of members of /statfunc/ whose third component /t/ is the same, is called the <u>snapshot</u> of /statfunc/ at time /t/.

<u>A formal difficulty</u>. According to the definition in the preliminaries, if $\langle i,j,t \rangle$ is not a member of Picture, then /statfunc(i,j,t) = nil/. Therefore, if we select $\langle i,j,t \rangle$ as a member of /Border/, then at least one of the arguments of /nxtstatfunc/ in the second restriction on a state function, is /nil/. In order to make the expression there fully defined, we must make the following, clearly harmless

<u>Assumption</u>. nil $\notin$ Allstat

Notice however that this problem occurs exactly in those cases when /statfunc(i,j,t) = Borderstat/, so that /nxtstatfunc(... ... ...)/ is independent of any but its third argument.

From these definitions, it follows that /initstatfunc/ is the snaphot of /statfunc/ at time zero, and that the snapshot of /statfunc/ at time /t+1/ can be unambgiously and effectively determined from the snapshot of /statfunc/ at time t. Consequently, /statfunc/ is uniquely determined from /initstatfunc/ and /nxtstatfunc/.

1.2 Initial step in the recursive definition of patterns.

The purpose of the present section is to give formal definitions of elementary activities and elementary patterns in the cellular world. The definition of objects is postponed to the next section.

Definition. A stage is an arbitrary subset of /Space-time/.

Definition. An elementary activity on a stage /stag/ is the restriction of a state function /statfunc/ to /stag/, i.e.

$$statfunc \cap \triangleleft stag \; . \; Allstat \triangleright$$

Thus the elementary activity is a set of fourtuples $\langle i,j,t,stat \rangle$, each of which indicates what state some location $\langle i,j,t \rangle$ is in. The corresponding stage is the set of locations where the activity takes place.

Definition. An elementary pattern is a set of elementary activities.

Generating functions. The number of possible patterns is of course very large, but we are only interested in patterns whose members display good similarity. One possible criterium for such a pattern is that there should exist some generating function which can be written down as a short simple expression. The generating function should then be a function which takes some parameters (e.g. integers) as arguments; whose value is always an activity; and whose range of values is identical to the pattern that it is generating. Let us express this formally.

Assumption. A set /Allparam/ of elements is available. We shall use it as a parameter space. Intuitively, it could be thought about as a set of n-dimensional vectors with integer components, or (more general) a set of list structures.

Definition. If /pat/ is a pattern, then any mapping

$$genf: \quad \triangleleft Allparam \triangleright \rightarrow pat$$

which satisfies $\{genf(par) \mid par \in Allparam\} = pat$

is called a generating function of the pattern /pat/.

In the very simplest case, all members of the pattern are congruent. The parameter space could then be a set of vectors $\langle I,J,T \rangle$, and the generating function could use I,J, and T to translate a constant, canonical version of the pattern through time and space.
- In some slightly more complex cases, additional components in the parameter vector may regulate e.g. the length of a path segment, or the number and relative distance of signals along a path.

Fictitious activities. Consider the following situation. We have a fixed /Picture/ and a fixed /nxtstatfunc/, but we use a couple of different /initstatfunc/, and therefore we obtain some different /statfunc/. An activity is a subset of /statfunc/. Clearly, one and the same activity can be a subset of several different /statfunc/. Consequently, patterns are independent of the state function. We can form patterns, not all members of which are subsets of the same /statfunc/. Indeed, it is possible to form patterns, some of whose members are not subsets of any /statfunc/ that can occur with the given /Picture/ and the given /nxtstatfunc/.

This is completely analogous to the possibility of discussing hypothetical activities in natural language. In the cellular world, /statfunc/ describes the state of the world at each point in time. An activity which is not a subset of the given /statfunc/ is therefore an activity that does not "occur" in the world.

Pivots and tokens. Consider an activity /acti/ in the pattern "signal meets signal in intersection". If we look at the series of snapshots which show the collision, we can clearly identify the cell $\langle i,j \rangle$ and the point /t/ in time where the collision takes place. They correspond to a fourtuple $\langle i,j,t,s \rangle$ in /acti/ (remember that a cell is in state /s/ when a collision takes place). However, it would be impossible to let this single fourtuple characterize the activity, as it is in no way different from any fourtuple $\langle i',j',t',s \rangle$ where one signal is just passing by, on a path. Instead, we have chosen to define an activity as a set of fourtuples located around $\langle i,j,t \rangle$ in time and space. Together, they contain enough information to assert that a collision takes place then and there.

For the moment, we therefore do not have any possibility to say, in our formal apparatus, that the activity (signal meets signal) takes place at one precise point in time, and in one precise cell. We shall now introduce pivots and tokens, which will make it possible to express such things.

Assumption. A set ⌊Alltoken⌉ of elements is available. Its members are called tokens.

Assumption. Some mapping assigns one token to each pattern, in such a way that no token is assigned to two different patterns.

Above, we agreed to use a generating function

genf: $\triangleleft$ Allparam $\triangleright$ → pat

for each pattern /pat/. Now that we have a token for each pattern, it is more satisfactory to assume one general generating function

actgenf: $\triangleleft$ Alltoken, Allparam $\triangleright$ → $2^{\triangleleft \text{Allint, Allint, Allint, Allstat} \triangleright}$

If ⌊tok $\in$ Alltoken⌉, and ⌊par $\in$ Allparam⌉, then /actgenf(tok,par)/ is the activity, determined by the parameter ⌊par⌉, in the pattern associated with the token ⌊tok⌉. In other words.

λ((par) actgenf(tok,par))

is a generating function of the pattern associated with the token ⌊tok⌉.

Assumption. Let ⌊pat⌉ be a pattern, which has had the token ⌊tok⌉ assigned to it. With each ⌊act⌉ in ⌊pat⌉, we associate a set of fourtuples on the form $\langle$i,j,t,tok$\rangle$. This set of fourtuples is written "pivots(tok,act)". Each fourtuple is called a pivot of the activity /act/.

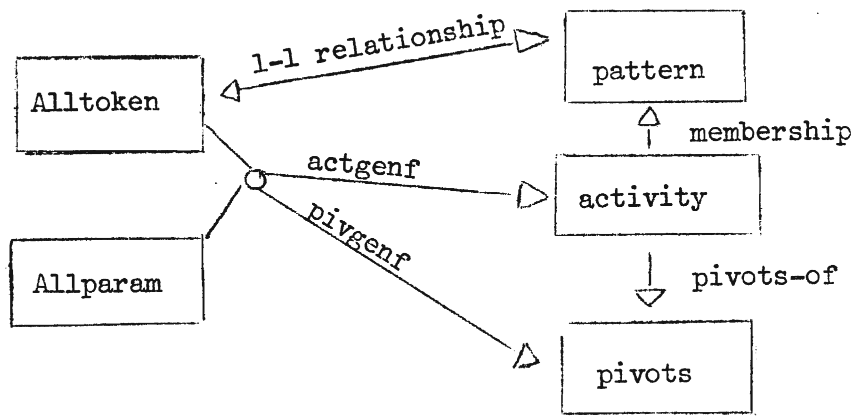
The function ⌊pivots⌉ must of course be specified for each pattern, so that it selects the right i, j, and t. In the case of the pattern "signal meets signal in intersection", it is natural to let ⌊pivots(tok,act)⌉ be a set with one member $\langle$i,j,t,tok$\rangle$, where $\langle$i,j,t$\rangle$ is the location where the collision "takes place". For the pattern "signal proceeds along line", on the other hand, it is natural to let ⌊pivots(tok,act)⌉ be a set of $\langle$i,j,t,tok$\rangle$ for each location $\langle$i,j,t$\rangle$ where the signal appears.

For simple patterns, whose members are congruent or almost congruent, the location of the pivots can clearly be determined from the parameter used by the generating function. It is therefore natural to introduce a single name for the function

$$\lambda((tok,par)\ pivots(tok,\ actgenf(tok,par))\ )$$

Let us call this function "pivgenf", to indicate that it is a generating function (i.e. a function with a parameter among its arguments) and generates pivots.

We have now introduced so many sets and functions that a diagram is necessary to illustrate their relationships:



We finally define the function /verb-with/ in such way that /verb-with(tok,statfunc)/ is the set of all fourtuples $\langle i,j,t,tok \rangle$ which are pivots of activities that occur in /statfunc/. We have:

$$\text{verb-with}(tok,statfunc) = \bigcup_{\substack{par \in Allparam \\ actgenf(tok,par) \subseteq statfunc}} pivgenf(tok,par)$$

or equivalently:

$$\text{verb-with}(tok,statfunc) =$$

$$\{ v \mid \exists\ par \in Allparam : actgenf(tok,par) \subseteq statfunc \wedge v \in pivgenf(tok,par) \}$$

Clearly, /verb-with(tok,statfunc)/ contains information to indicate where in /statfunc/ activities in the pattern associated with /tok/ occur.

Remark: The reason why /tok/ must be an argument of the function /pivots/ is that the same activity can be a member of several patterns.

1.3 Objects and nominals

By an object in the cellular space, we mean such a thing as a signal, a path, or an intersection. Every such object will be considered a special kind of activity. The set of all signals is therefore a pattern, as is the set of all paths, etc.

It follows that each object is a set of fourtuples $\langle i,j,t,tok \rangle$, where /tok/ is the token for the object class = pattern. Each such fourtuple indicates that the object is in cell $\langle i,j \rangle$ at time /t/. We shall call the fourtuple a momentary object.

For each activity /act/ in an ordinary pattern (= a pattern that corresponds to a natural-language verb), we construct one or several nominals. (These may correspond to the subject, direct and indirect object, time and space adverbial, etc. of natural language). Each nominal shall be the restriction of an object (or a union of objects) to the stage of /act/.

For example, if /act/ is in the pattern "signal meets in intersection", then the "subject" nominal of /act/ shall be a subset of an activity in the "signal" pattern. In fact, it shall consist of those fourtuples in the "signal" activity, which are on the form $\langle i,j,t,tok \rangle$, where $\langle i,j,t \rangle$ is a member of the stage of /act/.

In analogy with the pivot generating function, we assume a nominal generating function, /nomgenf/.

To each activity, there may be several nominals, corresponding to natural-language subject, direct object, etc. We therefore introduce a set of placers (term borrowed from {Bohnert 1967a}) to distinguish the nominals. Placers occur as the third argument of /nomgenf/. Let us proceed to putting this down formally.

Assumption. A set [Allplacer] of elements is available. Its members are called placers.

Some members of this set are: [by] (∿ subject); [od] (∿ direct object); [oi] (∿ indirect or dative object); [from] , [to] , [in] (used as spatial prepositions); [since] , [until] , [at] (used as temporal prepositions).

<u>Notation</u>. In analogy with the preceeding section, we introduce the following two functions:

nominal: /nominal(tok,act,plac)/ is the nominal of activity /act/ under the placer /plac/ in the pattern associated with with the token /tok/.

nomgenf = λ((tok,par,plac) nominal(tok, actgenf(tok,par),plac))

These functions are of course analogous to /pivots/ viz. /pivgenf/.

In most respects, the nominals of an activity can be considered as second, third etc. sets of pivots. Notice however that the following property holds for pivots, but can not be generalized to nominals:

$$\langle i,j,t,tok2\rangle \in pivgenf(tok,par) \supset tok = tok2$$

Formally, no distinction is made between objects and other activities. In particular, nominals may be subsets of any kind of activities, and are not restricted to be subsets of objects. This corresponds of course to our problematic natural-language situation in the introduction: "Peter suggested that John should go and look for the car".

1.4 Recursive definitions of patterns. The function /history/.

In section 1.2 we defined an elementary activity as a set of fourtuples on the form

$\langle i,j,t,tok \rangle$

where /tok ∈ Alltoken/ . We now make the transition to a more general

Definition. An activity is a set of fourtuples on the form

$\langle i,j,t,s \rangle$

where /s ∈ Allstat U Alltoken/. Such a fourtuple is called an event. It is also convenient to use

Definition. Allstatok = Allstat ∪ Alltoken.
The definitions of pattern etc. are in the obvious way generalized to account for these non-elementary activities.

This generalization can perhaps easiest be understood from the viewpoint of a recognition program. Suppose we have a grogram that "looks" at a cellular world, recognizes activities, and generates the corresponding pivots. After the above generalization, the program is permitted to "look" at the previously generated pivots, as well as to "look" at the universe itself. Recognition of one "low-level" pattern may be an essential step in recognizing another, "higher-level" one. (The analogy with higher-order perceptrons is clear).

Specifications of patterns may be recursively interdependent. Suppose, for example, that patterns /p1/ contains some elementary activities, plus some activities which contain pivots of pattern /p2/ in a certain lay-out; and that all activities in /p2/ contains pivots of pattern /p1/. Then it is at least in principle possible that the above pattern recognition function should recognize one activity in /p1/, therefore recognize an activitiy in /p2/, therefore recognize another activity in /p1/, etc.

Let /statfunc/ be a given set of events, e.g. the state function of the cellular world. We shall define the function /history/ such that /history(statfunc)/ is the set of all events that can recursively be recognized in /statfunc/. To make the definition simple, it is natural to include /statfunc/ itself as a subset of /history(statfunc)/.

(1) statfunc $\subseteq$ history(statfunc)

(2) actgenf(tok,par) $\subseteq$ history(statfunc) $\supset$

pivgenf(tok,par) $\subseteq$ history(statfunc)

(3) all members of /history(statfunc)/ can be reached through (1) and (2).

At the end of section 1.2, we defined the function /verb-with/ in such a way that /verb-with(tok,statfunc)/ is the set of all events $\langle$i,j,t,tok$\rangle$ which can be (directly) recognized in statfunc. This definition must now be generalized to include recursively recognized events. This could be done with a new recursive definition, similar to the one for /history/, but it is more convenient to rely on the definition for /history/ and define:

verb-with(tok,statfunc) =
history(statfunc) $\cap$ $\langle$Allint, Allint, Allint, {tok}$\rangle$

Notice that the following implication does not hold:

* actgenf(tok,par) $\subseteq$ history(statfunc) $\supset$
nomgenf(tok,par,plac) $\subseteq$ history(statfunc)

By analogy, the fact that a statement in natural language is true in the physical world does not imply that its nominals exist there. For example, it may be true that "they were discussing a green moon" or "Peter suggested that John should go and look for the car", even if the green moon does not exist, and John never actually goes to look for the car.

1.5 Examples of higher-order patterns.

The examples in this section shall illustrate how higher-order patterns may be used in the cellular world. We will get additional motivation for the use of higher-order patterns in the next chapter, where by contrast the real (physical) world is considered.

Example 1 In our cellular world, we have agreed to call a straight diagonal sequence of locations in state /x/, a path. The definition of the pattern of paths is then quite straightforward; it can be described by a generating function which takes parameters for (say) the x- and y-coordinates of the lower end of the path; the length of the path; a logical variable which indicates whether the path goes upward-left or uward-right; and the time or time interval at which the path occurs.

However, a path defined that way fails to be recognized whenever there is a signal on it, i.e. some location on the path is in state /s/. This is counter-intuitive, and we would like to remedy it. Thus an activity such as depicted here should be a member of the generalized path pattern:

```
8   x
7     s
6       s
5         s
4           x
3             x
2               s
1                 x
    2 3 4 5 6 7 8 9
```

Formally, this is the set

$$\{\langle 2,8,t,x\rangle \text{ , } \langle 3,7,t,s\rangle \text{ , } \langle 4,6,t,s\rangle \text{ , } \ldots\ldots \langle 9,1,t,x\rangle\}$$

How should the extended pattern be defined formaly? One way is to modify the generating function, so that it also takes a parameter component, which is a list of cells that should be in state /s/, rather than /x/. This approach is clumsy. We can do better if we permit higher-order patterns, i.e. patterns defined from previous patterns.

Let the token of the extended path pattern be /P/. We shall now introduce an auxiliary pattern with token /A/. The auxiliary pattern occurs whereever a location is in state /x/ or /s/. Definitions:

(1) actgenf(A, $\langle$ i,j,t,log $\rangle$) =

if log **then** { $\langle$i,j,t,x$\rangle$ } **else** { $\langle$i,j,t,s$\rangle$ }

Comment: For any given fourtuple $\langle$i,j,t,x$\rangle$ or $\langle$i,j,t,s$\rangle$, we can select a suitable parameter vector /par/ such that /actgenf(a,par)/ is the set that has this fourtuple as sole member. /par/ should be $\langle$ i,j,t,log$\rangle$, where /log = **true**/ if the location $\langle$i,j,t$\rangle$ is in state /x/, and /log = **false**/ otherwise.

(2) pivgenf(A, $\langle$i,j,t,log$\rangle$) = { $\langle$i,j,t,A$\rangle$ }

Comment: From (1) and (2), it follows that for every statfunc, i,j, and t,

$\langle$i,j,t,x$\rangle$ $\in$ history(statfunc) V $\langle$i,j,t,s$\rangle$ $\in$ history(statfunc) $\supset$
$\langle$i,j,t,A$\rangle$ $\in$ history(statfunc)

(3) actgenf(P, $\langle$i,j,t,n,dir$\rangle$) =

if n $\leq$ 2 **then** nil **else** acp(i,j,n,A)

where

acp(i,j,n,q) = **if** k=0 **then** nil **else**
{ $\langle$i,j,t,q$\rangle$ } U acp(**if** dir **then** i+1 **else** i-1,
j+1, n-1, q)

Comment: /acp(i,j,n,q)/ is therefore a path consisting of /n/ fourtuples which contain token /q/. The path starts in cell $\langle$i,j$\rangle$ and goes upwards right, if /dir = **true**/, and upwards left otherwise. All locations in the path are at time /t/. "dir" and "t" are free variables. - /actgenf(P, ...)/ is such a path if /n > 2/ and the empty set otherwise.

(4) pivgenf(P, $\langle$i,j,t,n,dir$\rangle$) = **if** n $\leq$ 2 **then** nil **else** acp(i,j,n,P)

<u>Comment 1</u>: From /n $\leq$ 2/ follows /actgenf(.......) = nil _ history(statfunc)/ for any/statfunc/, and therefore it is necessary that also /pivgenf(.......)/ should be /nil/ in this case (otherwise /history(statfunc)/, according to its definition, would be enriched with a lot of pivots that we do not desire).

<u>Comment 2</u>: In any /statfunc/, any location $\langle i,j,t \rangle$ which "has" pivot /P/ in it, also "has" pivot /A/. The reverse is not true; a location "has" pivot /P/ only if it is part of a sequence of more than two locations that "have" pivot /A/ (i.e. more than seven locations in state /x/ or /s/).

<u>Comment 3</u>: There is also another difference between the pivots with /A/ in them, and pivots with /P/ in them: a path of e.g seven fourtuples $\langle i,j,t,A \rangle$ consists of seven different activities, whereas a path of seven fourtuples $\langle i,j,t,P \rangle$ is one single activity, or possibly a subset of an activity. This is important. In the latter case, if we define the lowest location of the path as the /from/ nominal of the activity(*) , then we can look at an arbitraty fourtuple $\langle i,j,t,P \rangle$ on the path and say: "this is a pivot, which is a member of an activity, whose /from/ nominal is...". Translated, we are then saying "this pivot is in a path that goes <u>from</u> the location ...". Such expressions will be used in the next chapter. The corresponding trick is not possible for $\langle i,j,t,A \rangle$ pivots, as each such pivot forms its own activity and therefore only "knows" about its immediate environment.

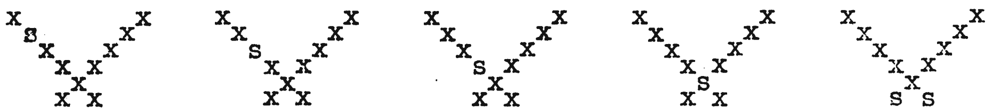
From these comments, it follows that the token P is indeed associated with the improved path pattern.

Notice that the auxiliary pattern whose token is A, can be compared to a natural-language substance noun (like "water", "steel") wheras the path pattern P can be compared to an object noun (like "drop", "hammer").

(*)

Notice that objects are considered as special cases of activities. There is therefore nothing to prevent us from attaching nominals to (whatever we intuitively think is) an object.

Example 2. Consider now the definition of the pattern 'signal arrives in intersection'. A typical activity in this pattern could be illustrated through the following snapshots:

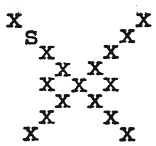


Like in the preceeding example we have some complications to the intuitive concepts. We would like to include in the pattern activities where

(a) there occur some other signal, which may arrive in the intersection earlier or later than the studied one;

(b) there occur some location which is in the "excited state" /e/;

(c) there are one or two 'by-paths' in the intersection, as discussed in $, i.e., e.g.,



Such intersections display a different signal reproduction pattern from intersections without by-paths.

If we want to define this as an elementary pattern, we will have to use a very complicated parameter. It is more convenient to proceed in the following steps:

(1) Introduce an auxiliary pattern for the states /x/, /s/, and /e/, similarly to the previous example.

(2) Define the 'intersection' pattern using the auxiliary pattern. Suppose it has the token I.

(3) Define the 'signal arrives in intersection' pattern as indicated by the following snapshots:

```
s        .        .        .
 .        s        .        .
  .        .        s        .
   I        I        I        I
```

The points stand for unspecified locations. Thus each activity in this pattern has seven members, on the form $\langle i,j,t,s\rangle$ or $\langle i,j,t,I\rangle$. One activity in the pattern is the set

$$\{\langle 0,4,9,s\rangle, \langle 3,1,9,I\rangle, \langle 1,3,10,s\rangle, \langle 3,1,10,I\rangle, \langle 2,2,11,s\rangle, \langle 3,1,11,I\rangle, \langle 3,1,12,I\rangle\}$$

In fact, the definition of the pattern 'signal arrives in intersection' may be simplified further. Suppose we have previously defined a pattern for 'signal proceeds in straight line'. Each activity in this pattern isaa set

$$\{\langle i\pm k,\ j-k,\ t+k,\ s\rangle\}_{k=0}^{n}$$

for arbitrary i,j, and n, with n greater than (e.g.) 3 . Of course, we must use either 'i+k' for all k, or 'i-k' for all k. Let the token of this pattern be /L/, and let there be a pivot in each location where the signal proceeds, i.e. for all k from 0 to n. The pattern 'signal arrives in intersection" can then be defined as the set of all activities

$$\{\langle i,j,t,I\rangle, \langle i\pm 1,\ j+1,\ t-1,\ L\rangle, \langle i,j,t,L\rangle\}$$

for arbitrary i,j,t, and arbitrary sign in /±/. Notice: it is not recommended to define the pattern as the set of all

$$\{\langle i,j,t,I\rangle, \langle i,j,t,L\rangle\}$$

because then the pattern would include activities where a signal starts exactly in the intersection at 'the hour of creation', i.e. for /t=0/.

Chapter 2

Natural-language sentences as descriptions of activities

2.0 Introduction.

Let us review the argument in the Introduction to this report. We said essentially:

(A) We would like to have a method for translating natural language (e.g. English) into a formal language (e.g. predicate calculus);

(B) This might be done in various ways. One approach, which has some distinctive advantages, is to consider sentences as descriptions of "activities", which are generalized "objects" in the world. We would then translate sentences into expressions of the type

$$\exists \, Pbj \; : \; [Pbj \in gives \wedge subject(Pbj,Peter) \wedge \; \ldots \;]$$

(C) This intuitive "activity" concept is a little bit unclear, which was exemplified by some difficult-to-answer questions. A strict definition is needed.

The primary purpose of this report is to satisfy the request in (C). Thus in chapter 1 we have defined a formal construct, called "activity", and introduced a formalism to describe activities. In this chapter, we shall demonstrate that the formal activity concept is indeed applicable to many natural-language sentences. Our approach will be to select some sample sentences in natural language, to assume that they are descriptions of activities in a cellular world[(*)], and to re-express them in our activity-describing formalism.

In sections 2.3 and 2.4, we shall also discuss more in general how certain common natural-language situations can be treated. However, we will only attain the restricted goal of applying the results from chapter 1 to some, carefully selected situations in natural language. This is sufficient for an answer to (C) above. In particular, the three questions posed on page 6 of the Introduction will be answered in section 2.6.

(*) Thus we will effectively approximate the physical world with a cellular world.

One thing which we would like to do, but which we do not do in this report, is to demonstrate that a given, reasonably large subset of (or approximation to) natural language can be translated into an activity-describing formalism without loss of semantic information. Work on a translation procedure which does exactly that, is presently going. If successful, it will prove that our activity-describing formalism is an answer to problem (A).

2.1 Notation.

Some additional notation will be needed besides what was introduced in chapter 1.

Natural-language sentences refer to the physical world, which (according to our assumptiom) can be interpreted as a cellular world. We let /statfunc/ be the state function of that world, and define the additional constants

reality = history(statfunc)

imagination = ◁Allint, Allint, Allint, Allstatok▷

Thus we have / reality $\subseteq$ imagination / .

Small-letter constants will be used to denote sets of events. It is convenient to characterize activities (including objects) by the set of their pivots, rather than by the set of events which constitute the activity proper. One activity may be a member of several patterns, so the functions /pivots/ and /nominal/ take a token as one of their arguments. The corresponding functions from pivot-sets to activities and to nominals do not need a token as argument:

A(piv) = the activity whose pivot-set /piv/ is;

N(piv,plac) = the nominal of /A(piv)/ under the placer /plac/.

A formal definition would say:

$\forall$ tok $\in$ Alltoken, par $\in$ Allparam, plac $\in$ Allplacer :

A(pivgenf(tok,par)) = actgenf(tok,par)

N(pivgenf(tok,par),plac) = nomgenf(tok,par,plac)

It is convenient to have a predicate /ispivots/ such that /ispivots(piv)/ is true iff /piv/ is a pivot-set, i.e. is a member of the domain of /A/ and /N/. Formal defintition:

ispivots(piv) $\equiv$ $\exists$ tok $\in$ Alltoken, par $\in$ Allparam :
piv = pivgenf(tok,par)

The intuitive meaning of most small-letter constants will be clear from context. Examples: /john/, /box22/. (/box22/ is one particular box, not the set of all boxes). Let us repeat again that unless otherwise noted, these are the pivot set.

Patterns will be denoted by capital-letter constants:
/BOY/, /MALE/, /GIVE/, /IRON/, /RED/. Especially f r mass nouns and adjectives, it is convenient to use the corresponding small-letter constant to denote the set of all pivots of activities in the pattern. Thus

$$\text{red} = \text{verb-with(RED,imagination)};$$

$$\text{iron} = \text{verb-with(IRON,imagination)};$$

etc.

Finally, we need a two-place relation to indicate that a pivot-set is in a pattern:

$$\text{piv} \underline{\leftarrow} \text{pat} \equiv A(\text{piv}) \nleftarrow$$

It follows:

$$\text{piv} \underline{\leftarrow} \text{pat} \equiv$$
$$\text{ispivots(piv)} \wedge \text{piv} \subseteq \text{verb-with(pat,imagination)}$$

e.g.

$$\text{john} \underline{\leftarrow} \text{BOY} \equiv \text{ispivots(john)} \wedge \text{john} \subseteq \text{boy}$$

2.2 Some simple sentences.

In this section, we shall indicate through three examples how simple sentences with verb, subject, direct-object, and time and space adverbials can be translated. We are restricted to cases where subject etc. are in some sense names of objects ("Peter", "the ball").

Example 2.2.1: Peter is giving the ball to John

Constants: peter, GIVE, ball3, john

Formulation:

$$\exists\, p \,\underline{\leftarrow}\, \mathrm{GIVE} \;:\; N(p,by) \subseteq peter \;\wedge\; N(p,od) \subseteq ball3 \wedge N(p,oi) \subseteq john \;\wedge\; p \subseteq reality$$

Remark: That formulation does not contain anything about time. If the natural-language sentence is taken to indicate that /p/ occurs at a context-specified /now/, we would let the last literal be

$$\ldots \;\wedge\; p \subseteq reality \cap now$$

Example 2.2.2: Peter is sitting in the chair

Constants: peter, SIT, chair2

Remark: In this case, it would not be reasonable to say that the chair is a nominal of the activity, because "in the chair" is only one out of many possible indications of location. Instead, we would like to consider "in the chair" as an area, and indicate that Peter's sitting takes place in that area. As /chair2/ is after all the pivot-set of an activity, we prefer to consider "in /chair2/" as a nominal of the chair and to be the set of all $\langle i,j,t,s \quad \rangle$ for arbitrary s, which are (intuitively) located "in" the chair. This approach gives us the following

Formulation:

$$\exists\, p \,\underline{\leftarrow}\, \mathrm{SIT} \;:\; N(p,by) \subseteq peter \wedge p \subseteq N(chair2,in) \wedge p \subseteq reality$$

Remarks: Time can be added like in example 2.2.1. Notice that time adverbial and space adverbial are handled alike.

<u>Example 2.2.3</u>: Peter is sitting in a chair

<u>Remark</u>: In the previous example, he was sitting in <u>the</u> chair.

Constants: peter, SIT, CHAIR

Formulation: $\exists\, p \underline{\leftarrow} \text{SIT},\quad c \underline{\leftarrow} \text{CHAIR} \;:\; N(p,by) \subseteq \text{peter} \;\wedge$

$p \subseteq N(c,in) \;\wedge\; p \subseteq \text{reality}$

The philosophy used in these sample translations are intuitively obvious, and we shall not attempt to put them down formally.

2.3 Objects (redefined).

In section 1.3, we made the convention that objects shall be considered as special cases of activities. For example, each individual boy would be an activity in the pattern of all boys. This idea has the advantage that natural-language objects and sentences become interchangable (which was one basic idea of this report), but the disadvantage of not explaining phrases like "the male bear" or "the red table". The purpose of this section is to modify and extend the definition of objects in such way that the advantage of the old definition is retained, whereas the disadvantage is eliminated.

Let us notice at first that some objects do not correspond to natural-language objects. In example 1 of section 1.5, we had an example of a "mass noun" in the cellular world, a pattern characterized by the token /A/. We assume that natural-language mass nouns correspond to such patterns. Patterns for mass nouns will be called amorphous, and patterns for ordinary nouns will by analogy be called morphous. The same adjectives are used on their member activites. Only morphous activites can correspond to natural language objects.

The morphousness of a pattern is of course not an inherent property, but instead dependent on what natural language we are translating to or from.

Patterns that correspond to natural-language verbs will be considered as morphous or amorphous, dependent on whether their member activities are intuitively thought about as individual or not. For example, "to meet" would be a morphous pattern, whereas "to rain" would be amorphous.

Let us then turn to the question whether all natural-language objects can be described as activities.

Objects that do not have any pattern at all seem to be (literally speaking) unthinkable. Our concern is therefore restricted to objects which are members of several patterns. Such objects are very common. For example, the object /peter/ would be in the patterns BOY, HUMAN, MAMMAL, etc. More precisely, each of those patterns recognize an activity on the stage where Peter is located.

These various "aspects" of Peter, these various activities may or may not be identical or overlapping sets of events. However, the pivot-sets of the various aspects are surely disjoint, because each pivot-set is a set of $\langle i,j,t,tok \rangle$, where /tok/ is the token of the respective pattern.

Under these circumstances, it seems the best way out to define an object as the union of several pivot-sets with a common stage. To put this precisely, we make the following

Definition: A set /a2/ of events is said to be an extension of a set /a1/ of events iff

$$[a1 \subseteq a2] \wedge [\forall \langle i,j,t,s2 \rangle \in a2 \quad \exists \langle i,j,t,s1 \rangle \in a1]$$

Formally, we express this as

$$a1 \underset{\sim}{\subset} a2$$

Definition: Let /pat/ be a morphous pattern. A set /ob/ of events is said to be "an object from /pat/" or simply "a pat" iff

$$\exists piv : \; piv \underline{\leftarrow} pat \wedge piv \underset{\sim}{\subset} ob$$

Formally, we express this as

$$ob \underset{\sim}{\leftarrow} pat$$

and

$$piv = aspect(ob,pat)$$

Also, a set of events is said to be an object iff it is an object from some pattern.

With these definitions, we have e.g.

$$john \underset{\sim}{\leftarrow} BOY$$

$$john \underset{\sim}{\leftarrow} HUMAN$$

etc. as desired. Notice also that with the above definitions, an object may well contain some events is an amorphous pattern. This enables us to re-express e.g. "a red table". Such an object is interpreted as an object from the pattern /TABLE/, i.e. a pivot-set from the pattern /TABLE/, enriched by some pivots from the pattern /RED/.

As a consequence of the re-definition of objects, we have to change another old convention. In section 1.3, it was said that nominals of an activity shall be restrictions of(other)objects to the stage of that activity. We also assumed a nominal generating function, /nomgenf/. However, we can not let the nominal generating function for an activity like "Peter is giving the ball to John" generate all events in the object /peter/ on that stage, because then a complete specification of the nominal objects would have to be in the parameter.

Instead, we let the nominal generating function generate restrictions of activities in the most general possible pattern which can occur in the nominal. For example, if we claim that only humans are capable of giving, then the "subject" nominal of an activity of the pattern /GIVE/ shall always be a restriction of an activity in the pattern /HUMAN/. For use in our translations, we then define a function /restriction/ and relations Ψ and Φ as follows:

restriction (a1, a2) =

$$\{x \in a1 \mid x = \langle i,j,t,s \rangle \wedge \langle i,j,t,s2 \rangle \in a2\}$$

(All unbound variables in the set-expression are existentially quantified).

$$\Psi(\text{piv},\text{obpart},\text{plac}) \equiv$$

$$\text{ispivots}(\text{piv}) \wedge N(\text{piv},\text{plac}) \subseteq \text{obpart}$$

$$\Phi(\text{piv},\text{ob},\text{plac}) \equiv \exists\, \text{pat} : \text{ob} \precsim \text{pat} \wedge$$

$$\Psi(\text{piv}, \text{restriction}(\text{ob}, N(\text{piv},\text{plac})), \text{plac})$$

For example, consider the sentence "John gives the ball to Mary". We have an object /john/ on the form

$$\text{john} = \text{johnhuman} \cup \text{johnboy} \cup \ldots$$

where johnhuman = aspect(john,HUMAN) is an activity $\in$ HUMAN,

johnboy = aspect(john,BOY) etc.

If /piv/ is the pivot-set of the sentence, we shall have

$$N(\text{piv},\text{by}) \subseteq \text{johnhuman}$$

$$\exists\, \text{johnpart} \subseteq \text{john} \quad : \quad \Psi(\text{piv}, \text{johnpart}, \text{by})$$

$$\Phi(\text{piv},\text{john},\text{by})$$

Here, /johnpart/ is the restriction of /john/ to the stage of N(piv,by), but /johnpart/ overlaps both with /johnhuman/, with /johnboy/, and with the other aspects of /john/. In other words, /john/ is the entire person throughout time, whereas /johnpart/ is the slice of /john/ during the interval in time where this giving takes place.

For another example, notice that the existentially quantified /c/ in example 2.2.3 is an aspect of the chair in which Peter is sitting, not the chair in extenso.

2.4 Structures.

The section will be short. Its purpose is to create a notation for handling what we call structures, i.e. objects defined as assemblies of other objects.

The new functions do not have any fundamental importance in our system. We introduce them as examples of how various situations can be described in a precise manner, starting ultimately from a small number of basic concepts and basic functions: /actgenf/, /pivots/, and /nominal/.

A typical example of a structure is that of the pattern CAR. A car shall be defined as a collection of wheels, motor, seats, horn, etc. which (important!) have been assembled in a certain pattern. Needless to say, the idea of non-elementary patterns is ideal here. We first define the patterns WHEEL, MOTOR, etc., and then use them to define the one-step-higher-order pattern CAR.

For our translation purposes, we need a compact way of writing e.g. "the wheels of the car", i.e. to establish the relationship between a lower-level and a higher-level activity. We shall use the predicate /iscomponent/, and write e.g.

iscomponent(wheel32,car3)

The formal definition is

iscomponent(x,y) $\equiv$

$$\exists\, px, py \;:\; px \underset{\sim}{\subseteq} x \;\wedge\; py \underset{\sim}{\subseteq} y \;\wedge\; px \subseteq A(py)$$

With that definition, the entire wheel throughout time is a component of the car. If we merely want to indicate that componentship holds in a time-slice, we use the more general relation ispart(xv,y) $\equiv$

$$\exists\, px,py,x,v \;:\; px \underset{\sim}{\subseteq} x \;\wedge\; py \underset{\sim}{\subseteq} y \;\wedge$$
$$xv = restriction(x,v) \;\wedge\; restriction(px,v) \subseteq A(y)$$

Notice that both /iscomponent/ and /ispart/ permit /car3/ in the example to have various wheels during various parts of its life. The difference is that with /iscomponent/, no such wheel may have any life "outside" the car, whereas with /ispart/ it may. In such case, /xv/ is the restriction of the wheel to the stage where it is on /car3/.

2.5 Some further examples.

The purpose of this section is to exemplify the conventions from the last two sections.

Example 2.5.1: John and Peter saw a little red cottage

Constants: john, peter, SEE, little, red, COTTAGE

Remark: We consider this as two activities: John saw a little red cottage, and Peter saw a little red cottage.

Formulation:

$$\exists\, aj \underset{\sim}{\leftarrow} \mathrm{SEE},\ ap \underset{-}{\leftarrow} \mathrm{SEE},\ lrc \underset{\sim}{\leftarrow} \mathrm{COTTAGE} :$$

$$\Phi(aj,john,by) \ \wedge\ \Phi(ap,peter,by) \ \wedge$$

$$aj \cup ap \subseteq reality \ \wedge$$

$$lrc \cap little \neq nil \ \wedge\ lrc \cap red \neq nil \ \wedge$$

$$\Phi(aj,lrc,od) \ \wedge\ \Phi(ap,lrc,od)$$

Example 2.5.2: John and Peter met on the bridge.

Constants: john, peter, MEET-INTRANS, bridge2.

Remark: We must have two different patterns, for transitive and intransitive 'meet'.

Discussion: The treatment of the previous example obviously can not be applied here. Furthermore, it is unwise to consider /john/ and /peter/ as different nominals (like in "John met Peter on the bridge") because then we get into trouble with sentences like "John, Peter, Liza, and their eighteen cousins met on the bridge".

Instead, we shall consider "John and Peter" as a structure, similar to e.g. a "car" as discussed in section 2.4. The structure pattern shall be /GROUP/, and it is assumed to recognize arbitrarily large groups of humans during one moment in time.

Formulation: $\exists$ a $\underline{\leftarrow}$ MEET-INTRANS, g $\underline{\leftarrow}$ GROUP, j $\subseteq$ john, p $\subseteq$ peter :

Φ(a,g,by) $\wedge$ [a $\subseteq$ reality $\cap$ N(bridge2,on)] $\wedge$

ispart(j,g) $\wedge$ ispart(p,g) $\wedge$ [j $\cup$ p $\neq$ nil]

Remark: It is essential that the group is momentary, otherwise we could have a group which meets on the bridge, and where John and Peter take part at some other time.

<u>Example 2.5.3</u>: Peter sat in the chair, reading a book.

Constants: peter, SIT, chair2, READ, BOOK

Remark: Here, we have another situation where two patterns are recognized on the same stage. Cf. section 2.3.

Formulation: $\exists$ p $\underline{\leftarrow}$ SIT, q $\underline{\leftarrow}$ READ, b $\underset{\sim}{\leftarrow}$ BOOK :

Φ(p,peter,by) $\wedge$ p $\subseteq$ reality $\cap$ N(chair2,in) $\wedge$

Φ(q,peter,by) $\wedge$ q $\subseteq$ reality $\wedge$ Φ(q,b,od) $\wedge$

restriction(p,q) = p $\wedge$ restriction(q,p) = q

<u>Example 2.5.4</u>: Peter hit the chair, so it fell and made noise.

Constants: peter, HIT, chair2, FALL, MAKE, noise.

Remark: The <u>effect(s)</u> of an activity naturally constitute a nominal . We shall use the placer /so/ for this purpose.
Clearly, we have

$$[\text{piv} \subseteq \text{reality}] \supset [N(\text{piv},\text{so}) \subseteq \text{reality}]$$

Formulation: $\exists\, h \in HIT,\ \ f \in FALL,\ \ m \in MAKE$:

$\Phi(h,peter,by) \ \wedge\ \Phi(h,chair2,od) \ \wedge$

$\Phi(f,chair2,by) \ \wedge$

$\Phi(m,chair2,by) \ \wedge\ [N(m,od) \cap noise \neq nil] \ \wedge$

$[h \cup f \cup m \subseteq reality] \ \wedge\ [f \cup m \subseteq N(h,so)]$

2.6 Answers to the introductory questions.

On page 6, we asked three questions which illustrated our uncertainty about the activity concept. These questions were of course randomly selected, but we shall nontheless go to the trouble of answering them.

(a) Answer: Whatever we choose (depends on how the pattern GIVE is defined).

(b) Answer: If the sentence refers to one activity, then the direct-object nominal of that activity is a structure, consisting of the ball and the box. The relationship between that structure and its components (as expressed by the relations /iscomponent/ and /ispart/) is the only connection to the activity "Peter gives the box to John".

(c) Answer: The pattern SEE may or may not be defined such that "A sees, B and C" means "A sees B, and A sees C". This is the same arbitraryness as we had in the answer to question (a). It is possible because B and C can be in the activities in the pattern, i.e. can be used to define and to recognize the pattern. However, in the case of "A does B so C and D", where we have a nominal which indicates effects, C and D can not be used in the definition of the pattern,so this nominal has to be tucked on the activity after it has been recognized. It follows, then, that we can not distribute the sentence into "A does B so C" and "A does B so D".

2.7 Conclusion.

In chapter 1 and section 2.3, we have given formal definitions of a few basic terms like "pattern", "activity", and "object". We have also defined three basic functions for the description of activities: /actgenf/, /pivots/, and /nominal/. A large number of further functions and relations have been defined from these basic functions, and we have thus enriched the formalism for activity description.

In chapter 2, we have demonstrated through examples how the formal concepts from chapter 1 can be applied to natural-language sentences, and how these sentences can be translated into the formalism for activity description. We have thus made the intuitive "activity" concept precise.

The formal concepts used in this report has proved powerful enough to re-express natural-language nouns, verbs, and adjectives. In the use of these concepts, we have recognized formal counterparts of several natural-language distinctions (e.g. mass nouns vs. ordinary nouns). We have also found a method to re-express some semantic restrictions used in the construction of natural-language sentences (e.g. the restriction that only a human can be a giver*).

The sample translations given in this report would make it likely (without proving it, however) that a large class of natural-language sentences can be translated (without loss of information) into a formalism for activity description. The expressiveness discussed in the previous paragraph makes such translation particularly attractive. There is work towards a routine for automatic translation from a simplified English to a pattern describing formalism. When such a routine becomes available we can consider it as proved that activity-describing formalisms are powerful enough to re-express natural language.

(*) Naturally, it is not our purpose to discuss the specific question who and what can be a giver. The important thing is that activity-describing formalisms can express restrictions such as this one.