CONTROL STRUCTURES FOR ACTIONS IN PROCEDURAL TEXTS AND PT-CHART

Yoshio Momouchi

DIVISION OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING, HOKKAIDO UNIVERSITY
SAPPORO 060, JAPAN

This paper describes a partial taxonomy of control structures for actions in procedural texts. On the basis of the taxonomy, we examine natural language expressions for control structures in Japanese procedural texts and present PT (Procedural Text) -chart which represents the structure of a procedural text.

Introduction

Cookbooks, route instructions, machine assembly instructions and algorism descriptions , which are written in natural languages, and sorting programs, which are written in programming languages, are examples of procedural texts. There are many points at which natural language procedural texts and programs may be considered on common ground.
A procedure is composed of a sequence of actions intended to achieve a goal. Control structures determine patterns of behavior for action sequences. An action is something an actor does or can do. An action is enabled by certain states. An action acts on objects and causes the change of states of objects.
In programs, such concepts as objects, states, actions and control structures are defined explicitly. How are these concepts identified in natural language texts? What expressions represent these concepts? We think it most necessary that we have a right understanding of these concepts to understand procedural texts.
In Japanese procedural texts, actions are mostly expressed by verbs and control structures are expressed by nouns, adverbs, auxiliary verbs, postpositional words and so on. The primary verb is a procedure call in programs. Other syntactic structures are used to embed the procedure call information. Control structures in natural language procedural texts are more complex and richer than in programs. The control information is embeded within broad range of syntactic structures. We classify control structures into two groups, temporal and behavioral control structures, which are respectively associated with temporal and behavioral aspects of action sequences, and examine Japanese language expressions for control structures.
In programs, several proposals, e.g. BNF notation and a flowchart technique,[5] have been made for techniques supporting the formal description of the structure or development of a program. Two trials which use BNF notation and PT-chart are made for a formal description of the structure of a natural language procedural text.
A procedural text has the static (organization) structure and the dynamic (control) structure. It is highly desirable that a formal representation framework of a procedural text has the power to represent both the structures.
PT-chart is a graphical representation, which has the power.

L.A.Miller considered syntactic and semantic structures in English recipe texts.[1] He showed that all of the action-encoding mechanisms are based on the concept of a procedure call in the sense of computer programs, various syntactic structures are used to embed the call information and any procedure may be decomposable to successively less complex procedures. He also examined types of verb qualifications in the texts and classified cases controlling the action into seven groups.
On the basis of his studies, we consider further details of text structures, control structures and action-encoding mechanisms in procedural texts, especially in Japanese. This research is an underlying study for the semantic analysis of procedural texts.

Procedural text

We chose texts of a cookbook, "Family Cooking" by K.Okamoto, as procedural texts.
We examined texts of 69 Japanese cookings which are a subset from "Family Cooking". We call them Data. Examples from Data are given in Japanese enclosed by brackets [] and some of them have English translations enclosed by quotation marks ''.
In cookings, objects which actions act on and actors who do actions are as follows,
objects: ingredients, seasonings, cookwares
actors : men and/or women
The framework of a cooking text is roughly constructed by the following components in order of the description.
(a) name (of a cooking)
(b) remarks
(c) ingredients: quantity, notes
(d) preparations (sub) : preparation for each ingredient
(e) preparations (main) : cooking, seasoning, finishing, dishing
(a), (b), (c) and (d) are included in all descriptions of 69 cookings and the contents are almost the same. (e) has many different contents. But the construction of cooking, seasoning, finishing and dishing is most common.
We call a unit delimitted by a period in a text a sentence. Sentences are classified into two groups depending on the finishing style.

(i) sentence finishing with a noun
[raNgiri, ~koto, ~mono (noun)]
(ii) sentence finishing with other words
[yuderu (verb), ~te oku, ~te iru (verb),
yoi (adjective), ~seru (auxiliary verb),
nikui (suffix)]

Sentences are classified into six groups depending on the contents.
(i) action sentence, which describes actions
[ika wa mizuaraisi, omote no kawa o muku]
'wash and peel the squids'
(ii) state sentence, which describes states
[zeNbu de yottu ni wakareru]
'it has separated into four pieces'
(iii) action-and-state sentence, which describes actions and states
[nitatte kure ba sato to miriN o kuwaeru]
'when the mixture comes to a boil, add the sugar and mirin'
(iv) feeling sentence, which describes writer's feeling
[soko no fukai kobati ga yoi]
'one had better use the small deep dish'
(v) explanation sentence, which describes an explanation about ingredients or cookwares
[unerigusi to iu]
'it is called a winding spit'
(vi) mixed sentence, which describes more than two of actions, states, feelings and explanations (except for (iii))

Most sentences from Data are grouped into (i) or (iii). Some examples of action-and-state sentences are given below. States are generally conditions for actions. The condition is given in parentheses after each example.
[mi ga yawarakaku nare ba hi o kesite]
'when the flesh becomes tender, turn off the fire'
(start condition)
[nebari ga deru made mazeru]
'mix until it is gluteneous'
(continuation termination condition)
[ne no bubuN wa eNtookei ni naru yoo ni muku]
'peel the part of the root as it'll become cylindrical'
(ongoing behavior condition)
[hitasite oku to siN ga naku nari moto no 3 bai gurai ni fukuramu]
'soak it, and it'll become pithless and swell three times larger than the origin'
(result condition)
(__: state description)

Conditions are grouped into the following three;
precondition: condition which a state satisfies before an action
ongoing condition: condition which a state satisfies during an action
postcondition: condition which a state satisfies after an action

If a precondition is satisfied, an actor can do an action. In the above examples, start condition is a precondition, continuation termination condition is a postcondition, ongoing behavior condition is an ongoing condition and result condition is a postcondition.
In Japanese procedural texts, language expressions for actors are always omitted and more than one action is often described in one sentence.

Control structures

Language expressions which are used to embed the action call information are the following in Japanese procedural texts.
(A) verb
(B) participial adjective in a noun phrase
(C) constituent of a noun phrase or compound noun

In Japanese, a participial adjective is placed in pre-nominal position. A participial adjective indicates an action that should be taken place some time prior to the cooking step in which it occurs. Constituent of a noun phrase or compound noun also indicates an action.
The primary verb is an action call, in the sense of computer programs. The mechanism to control a sequence of actions is called a control structure. Many various control structures are found in procedural texts, and so it is very important to consider control structures systematically.
A partial taxonomy of control structures and Japanese expressions for them are given in the following;

[I] Temporal control structure
(1) Start
(a) Start time
(i) Time point
[hi kara orosu magiwa ni naganegi o kuwae]
'add the spring onions just before one takes the pot off the fire'

(ii) Time interval
[siagari tikaku natta koro sisitoogarasi o kuwae]
'add the green pepper about the time of finishing'

(b) Start condition
(i) Independent of preceding actions
[futoi mono wa haNgetugiri ni suru]
'cut radishes in semicircle slices if thick'

(ii) Dependent on preceding actions (I)
[siru ga fukiagare ba hi o tomeru]
'if the soup boils over, put out the fire'

(iii) Dependent on preceding actions (II)
[daikoN ga yawarakani narikakeru koro gohaN o kuwaeru]
'add the boiled rice about the time when the radish becomes tender'

(2) Continuation
(a) Continuation time interval
(i) Definite time interval

[30 puNkaN nekaseru]
'ferment it for 30 minutes'

(ii) Indefinite time interval
[sibaraku oku]
'let it stand for a while'

(iii) Approximate time interval
[yaku 2 fuNkaN yuderu]
'boil it for about two minutes'

(iv) Upper limit
[saikoo 1 jikaN kurai yuderu]
'boil it for less than one hour'

(v) Lower limit
[saitei 1 jikaN wa tukekoNde oku]
'leave it pickled for more than one hour'

(b) End time
[nabe ni ireru tyokuzeN made mizu ni tukekoNde oku]
'leave it soaked in water until one puts it into the pot'

(c) Continuation condition
[katai aida niru]
'boil it while it is hard'

(d) End condition
[azayakana midoriiro ni naru made sizukani yuderu]
'boil it gently until it becomes a bright green color'

(3) Repetition
(a) Continuous repetition
(i) Definite times
[kore o 2,3 kai kurikaesu]
'repeat this two or three times'

(ii) Indefinite times
[kawa ni fuooku de suukasyo ana o akete oku]
'open the several holes in the face of it with a fork'

(b) Intermittent repetition
[tokidoki mizu o kaenagara]
'making water afresh occasionally'

(4) Selection
(a) Obligatory selection
[utuwa ni moru ka, aruiwa oobati ni ikki ni akete]
'fill it in a dish or empty it into a big pot at a breath'

(b) Optional selection
[konomi de gomasio o furikakeru]
'sprinkle salt with parched sesame over it according to one's preference'

(5) Sequence
(a) Successive sequence
[nabe ni nidasijiru (nibaNdasi) o irete hi ni kakeru]
'put the stock into the pot and put it over the fire'
[sayuu ni hiraki, ago no bubuN o hootyoo de tatakikiru]
'open it right and left and chop off the chin part with a kitchen knife'
[itido kireini mizuaraisita noti, mizuke o fukitori]
'after one washes it, wipe out the moisture'

(b) Sequence with time delay
(i) Definite time delay
[1 jikaN go ni hi ni kakeru]
'put it over the fire after one hour'

(ii) Indefinite time delay
[kurogoma to sio wa betubetu ni iri, ato de mazeawasu]
'parch sesame and salt separately, then lately mix them'

(iii) Order
[saigo ni yaku 20 byookaN tuyobi ni site]
'lastly, cook it over the hot fire for about 20 seconds'

(6) Parallel
(a) Simultaneous parallel (one action)
(i) Action by one actor
[goboo o kaiteNsasenagara raNgiri ni si]
'cut the burdock turning over it'

(ii) Action by one actor and continued action
[gutugutu ninagara taberu]
'eat simmering it'

(b) Concurrent parallel (some actions)
A good example is doing actions for ingredients in sub preparations. Some actors can do actions for ingredients concurrently.
There are two cases.
(i) an action is independent of another action
(ii) an action is dependent on another action

[II] Behavioral control structure
(1) Ongoing condition
[sizukani arumihaku o hagasu]
'take the alminum foil off gently'
(__: expression for control structure)

There is not always one to one correspondence between a control structure and a language expression. We have examined Japanese procedural texts. We have a table of relations between control structures and corresponding Japanese expressions.

Start	:koro, toki, noti, syuNkaN, magiwa ni, tokoro de, toki wa, mae ni, uti ni, to dooji ni, no baai wa, tara, nara, kara, ba, wa
Continuation	:kaN, kaN ijoo, kaN hodo, yaku~kaN, saitei~kaN, 1 tyuuya, 1 baN, made, kurai, kurai ni, sibaraku, tuzukeru, te iru, te oku
Repetition	:kore o ~kai kurikaesu, tugitugi to,

	tokidoki, 2,3 do, mooitido, suukasyo, zutu
Selection	:ka, aruiwa, baai ni yotte wa, konomi de
Sequence	:te, kara, ato, ato de, ato kara, tugi ni, tuzuite, saigo ni, saisyo wa, saigo wa, mazu, tadati ni, sugu, saki ni
Parallel	:te, kara, nagara, yoo ni, mama
Ongoing	:sizukani, yukkuri to, yoo ni, mama

Conditions (states) which must be true in order for the action to start and/or proceed are described by several different expressions as known from examples.
There are some cases in which a condition is expressed by a part of a noun phrase or compound noun.

(1) Participial adjective in a noun phrase
[marumete oita nikudaNgo]
'a hand-rolled quenelle'
[5,6 cm nagasa ni kirisoroeta fuki]
'a butterbur cut the pieces to 5,6 cms in length'
[saki no togatta hotyoo]
'a sharp-pointed knife'

(2) Constituent of a noun phrase
[yuzu no wagiri]
'a round slice of a citron'
[unagi no kabayaki]
'broiled eels'

(3) Constituent of a compound noun
[ajituke-siitake]
'a seasoned mushroom'
[arai-gome]
'washed rice'
[net-tou]
'boiling water'
(__:state description)

States result by doing actions. But the above expressions about states don't always call actions. If an expression for calling the action to cause the state is included in the preceding context, the action needn't be called. Relations between the context and the action to cause the state are grouped as follows;

(1) Noun phrase, which includes a verb (in the case of Japanese)
(a) context has an expression for an action to cause a state
[mizu de sarasu-----mizu ni sarasite atta udo]
'bleach in the water-----an udo bleached in the water'

(b) context doesn't have an expression for an action to cause a state
[aratta mame]
'washed beans'

(2) Noun phrase or compound noun, which doesn't include a verb
(a) context has an expression for an action to cause a state
[wagiri ni site oku-----yuzu no wagiri]
'leave it cut in round slices-----a citron cut in round slices'

(b) context doesn't have an expression for an action to cause a state
[nure-fukiN]
'wet dishcloth'
(__:action description)

In (1-b) and (2-b), it is necessary to call an action to cause a state. (1-b) corresponds to (B) and (2-b) corresponds to (C).

PT-chart

We describe a formalized framework of a procedural text. The fundamental style of the following descriptions is borrowed from BNF notation of Algol60. The braces{} denote none or more repetitions of the enclosed material. The brackets[] denote none or one occurrence of the enclosed material.

<procedural text>::=(<name>[<declaration>]<condition><action>)
<action>::=[<statement-list>]<action body>[<statement-list>]
<action body>::=<simple action>|<sequential action>|<selective action>|<repeated action>|<parallel action>
<simple action>::=(<name><condition>(<act>))
<sequential action>::=(<name><condition>(<action>{;<action>}))
<selective action>::=(<name><condition>(<action>{,<action>}))
<repeated action>::=(<name><condition>(<action>{.<action>}))
<parallel action>::=(<name><condition>(<action>:<action>{:<action>}))
<condition>::=([<temporal control>][<behavioral control>][<modal condition>])
<temporal control>::=<sequence>|<selection>|<repetition>|<parallel>|<start>|<continuation>
<behavioral control>::=<ongoing>
<declaration>::=(<declaration unit>{<declaration unit>})
<declaration unit>::=<data declaration>|<action declaration>
<data declaration>::=(<name><data>)
<action declaration>::=(<procedural text>)

On the basis of the above descriptive framework, we construct a representation framework, PT-chart. PT-chart has the power to represent both the organization and control structures of a procedural text.
PT-chart has a tree structure composed of nodes and edges. A pattern is a subchart of PT-chart. Nodes are Point, Name, Condition, Declaration, Statement, Action and Data. Edges are Line, Arrow, Sequence, Selection, Repetition and Parallel. Some important pattern types, which mainly have the double purpose of the organization structure

and control structure, are Sequence, Selection, Repetition and Parallel.

Nodes;

—●— : Point node, which represents a junction

—N— : Name node, which represents a name

—C— : Condition node, which represents a condition

—D— : Declaration node, which represents to be a declaration

—A : Action node, which represents actions

—T : Data node, which represents data

—S : Statement node, which represents a statement which is mentioned at this point of a procedure

Edges;

—— : Line edge, ——→ : Arrow edge

——▷ : Sequence edge, —+— : Selection edge

—ꝏ— : Repetition edge, ══ : Parallel edge

Patterns;

Sequence pattern:
doing procedures 2,3,...and n in that order under a condition C_1

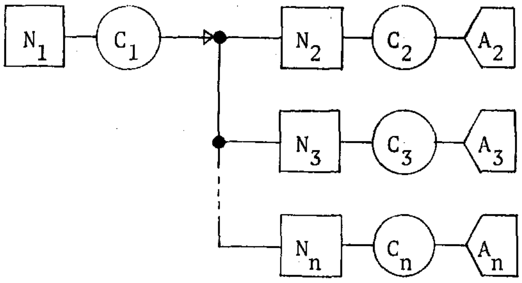
Selection pattern:
doing one of procedures 2,3,...and n under a condition C_1

Parallel pattern:
doing all procedures 2,3,...and n concurrently under a condition C_1

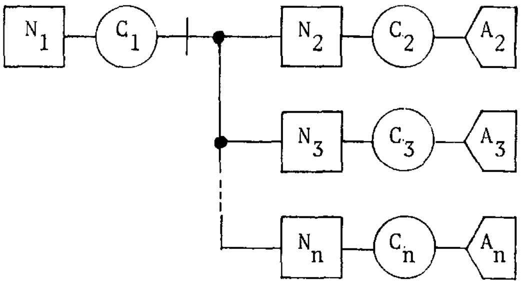
Repetition pattern:
repeating procedures 2,3,...and n in that order under a condition C_1

Declaration pattern:
declaration of procedures and/or data (procedures=>actions)

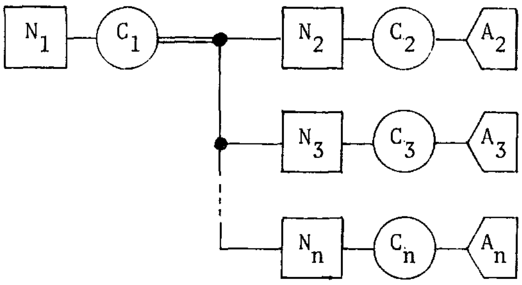
Name or condition node may be omitted in each pattern.



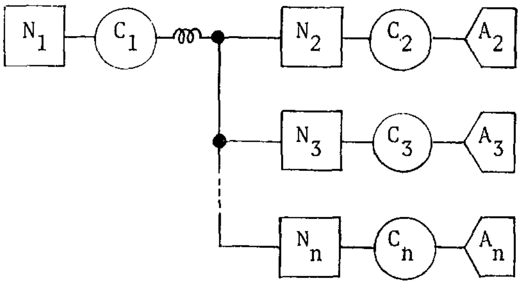
Sequence pattern



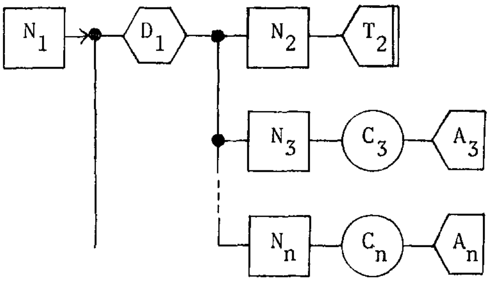
Selection pattern



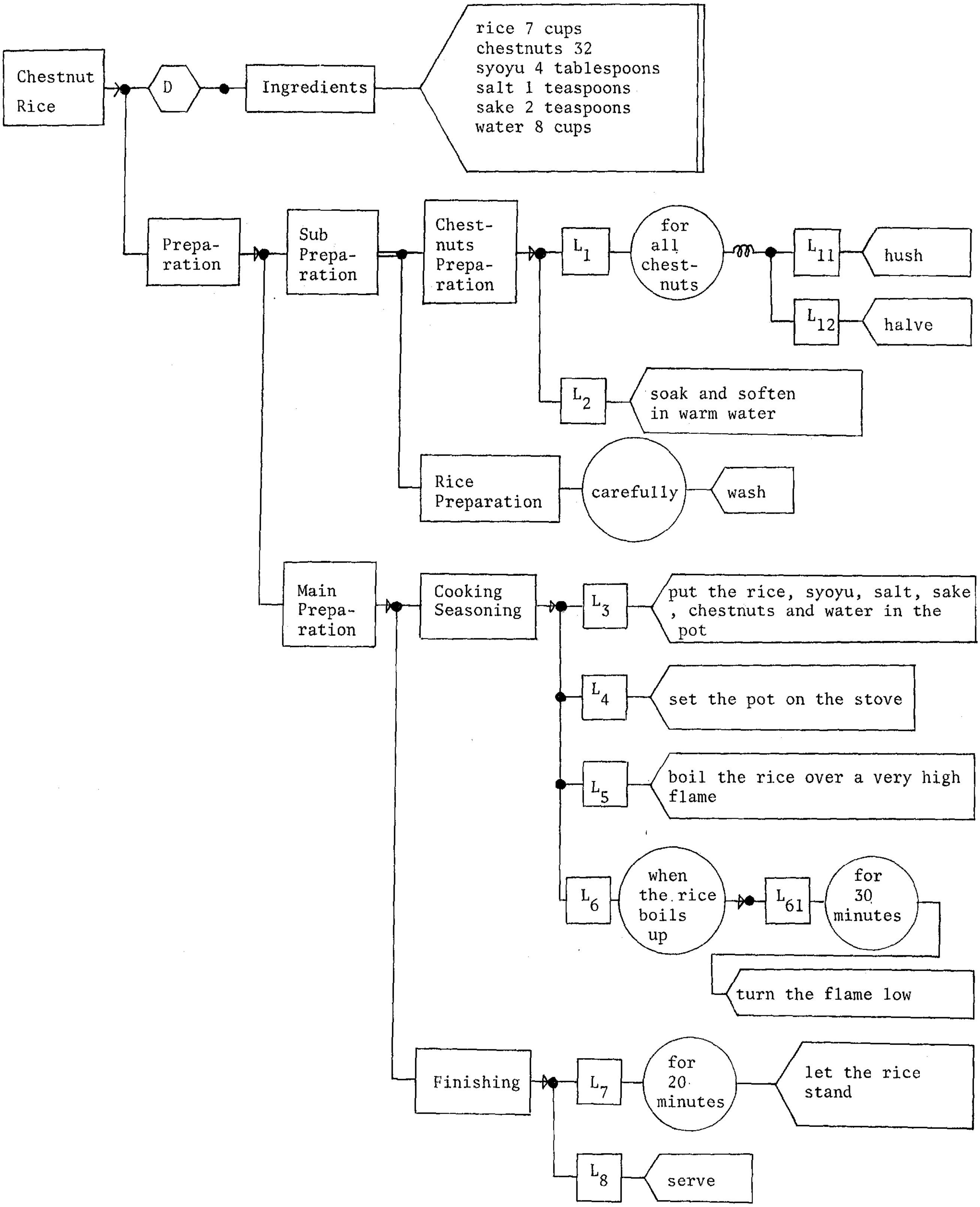
Parallel pattern



Repetition pattern



Declaration pattern



An example of PT-chart

Discussions

Procedural texts of cookings are mostly composed of descriptions of actions that act on concrete objects, ingredients or cookwares, and descriptions of states of them. So we can regard the semantics of a procedural text as state transformation from an initial state before a cooking to a final state after a cooking. Then it is related with algorithmic logics which are considered as logics of programs[4]. A way will be open to consider logics for general procedural texts. In such logics, a problem is how to incorporate a procedure into logics. By studies of algorithmic logics, three methods have been found.

(1) A method which regards a procedure as a logical formula. The meaning of a procedure α as a logical fomula is 'α terminates'. Suppose that p is a logical fomula and α is a procedure, $\alpha;p$ is also a logical formula. The meaning of the formula is 'α terminates and p is true'.
(2) A method Which regards a procedure as a modal operator. Suppose that p is a logical formula and α is a procedure, $\langle\alpha\rangle p$ is also a logical formula. $\langle\alpha\rangle$ is a modal operator. The meaning of the formula is 'if α terminates, then p is true'. We can define another modal operator $[\alpha]$ as $[\alpha]p= \neg\langle\alpha\rangle\neg p$.
(3) A method which regards properties of a procedure as axioms. The axioms give the semantics of a procedure.

Researches are left to implement a computer program for processing natural language procedural texts. Natural language procedural texts may have many incomplete, ambiguous expressions. The program must process them using context or knowledge. In transformation from natural language texts to complete, unambiguous texts which are written in a formal language, the program must also understand descriptions of objects, actions, states and control structures. The programs for processing natural language algorithm descriptions are considered by several researchers. [6,7] They discuss that the right understanding of control structures in natural language algorithm descriptions is an important function to the program. So the identification of control structures in procedural texts is an underlying study for implementing the computer program.

The author would like to express his thanks to Dr.E.Miyamoto and Mr.T.Maeda of Hokkaido University, and Mr.H.Sawamura of IIAS-SIS for their helpful discussions and to Dr.L.A.Miller for sending papers to him.

References

1) Miller,L.A.:'Natural language procedures: guides for programming language design', Paper presented at the Comp. Prog. Symp., Sixth cong. of IEA(1976)
2)Chodorow,M.S. and L.A.Miller:'The interpretation of temporal order in coordinate conjunction ', IBM Thomas J. Watson Res. Center Res. Report RC6199(1976)
3)Miller,L.A.:'Naive programmer problems with specification of transfer-of-control', Proc. of NCC(1975)
4)Sawamura,H.:'Algorithmic logics', Information Processing Vol.20, No.8, IPSJ(1979) (in Japanese)
5)Ferstl,O.:'Flowcharting by stepwise refinement ', SIGPLAN Notices Vol.13, No.1(1978)
6)Wile,D., R.Balzer and N.Goldman:'Automated derivation of program control structure from natural language program descriptions', SIGART Newsletter 64, ACM(1977)
7)Hobbs,J.R.:'What the nature of natural language tells us about how to make natural-language-like programming languages more natural', SIGART Newsletter 64, ACM(1977)