연구논문집 제8권 제2호 p.21~24 1977

Automata of Job Control Languages and ANJCL

Kim Heui-Seung
Dept. of Computer Science

〈**Abstract**〉

In certain JCL, for example OS/VS JCL, it was showed that it seems to be possible for JCL itself to select job steps for execution from the presubmitted job steps. I suggest a new kind of JCL type which can control completely its sequence of execution by its own function. This technique can be applied to modular programming so that program segments are automatically executed in certain desired sequence. And it can be also applied to batch job execution.

Job Control Language의 Automaton과 ANJCL

金 熙 昇
電氣工學科

〈요 약〉

처리하고자 하는 job step을 자동적으로 선택할 수 있는 가능성이 포함된 JCL들이 현존하는 JCL들, 예컨대 OS/VS JCL 등에서 보여졌다. 이것을 확장하여 완전히 자동적으로 job step을 선택하여 job step의 처리 순서가 다양하게 제어될 수 있는 형식을 제시하였다. 이 새로운 형식의 JCL로써 modular programming technique을 job step level까지 확장시킬 수 있을 것이며, batch job processing에 응용시킬 수 있을 것이다.

I. Introduction

Most job control languages are such kind of JCL that are processed sequentially as they are arranged. There are no other execution sequences than serial process. If a progammer wants to execute his job steps in another sequence, he must change manually the JCL card sequence.

In a few JCLs, we can find out some special sequence control methods. For example, JCL of IBM OS/VS is not processed only in serial sequence. It can have condition parameter according to certain option. with this parameter each job statement can be executed or not depending on the condition. We can regard this feature as a kind of sequence conrol automaton. The automaton can voluntarily alter its execution sequence, though the way of alteration is limited. we can consider other kinds of sequence control automata which will control its sequence with more different ways.

We can let the JCL have some special function with it so that it can control its sequence. By this function, JCL itself can produce various sequences according to certain sequence input. A controlability will be defined as a total number of various sequences which can be produced by one sequence control method.

I will suggest a few models for JCL sequence

control method which might increase controlability. It is possible to get automata for the JCL sequencing models. Automata will be analyzed to get interrelation between models. Also it will be showed which is the best model for sequencing method and have the best controlability. Control models are reduced from the general control type of usual programming languages.

Ⅱ. Controlability

For convenience' sake, I will define a controlability as follows. Controlability is the total number of possible sequences that certain JCL itself can select N terms from n terms submitted. And a term might be a process or a job step.

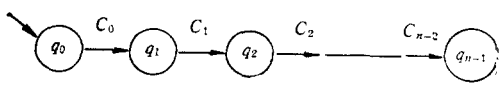
The best controlability or ideal controlability is the total sum of the number of $1 \sim N$ terms selection ways from n terms submitted. That is,

$$A = {}_nP_1 + {}_nP_2 + {}_nP_3 + \cdots + {}_nP_N = \sum_{i=1}^{N} n^i \qquad ①$$

where A is the best controlability.

Ⅲ. JCL models of existing JCL sequencing

Most JCLs control its sequence serially as they are arranged before execution; namely direct sequencing model. This type of model can be figured out as following.



And the automaton for this model is follows.

Terms (or state); $T=\{q_0, q_1, q_2, \cdots q_{n-1}\}$ $(D-1)$

Initial term; q_0 $(D-2)$

Final term; $F \subseteq T$ $(D-3)$

Condition (or Input);

$\Sigma_D=\{C_i \mid C_i=i+1,\ i=0, 1, 2, \cdots n-2\}$ $(D-4)$

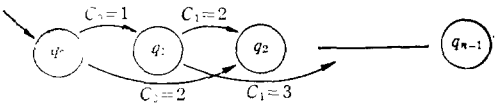
Sequencing function; δ_D

$\delta_D(q_i, c_i)=q_{Ci}$ where $C_i \in \Sigma_D$ $(D-5)$

Automaton

$M=\{T, \Sigma_D, \delta_D, q_0, F\}$ $(D-6)$

Second type of sequencing model is 'Execute or skip sequencing'. As we can see in certain JCL, each job cotrol statement can be executed or not along the serial progress of step execution. This model can be viewed as follows.



At any n-th step (or term), the condition for this step, C_n can have one value m which is the next executable term number. Automaton for this model is $(E-1)$ through $(E-7)$.

Terms; $T=\{q_0, q_1, q_2, \cdots q_{n-1}\}$ $(E-1)$

Initial term; q_0 $(E-2)$

Final terms; $F \subseteq T$ $(E-3)$

Condition;

$\Sigma_E=\{C_i \mid C_i=i+1$ or $i+2$ when $i=0, 1, 2, \cdots n-3$ and $C_i=i+1$ when $i=n-2\}$ $(E-4)$

Condition generator; g_E

$$g_E(C_i)=\begin{cases} C_i \leftarrow i+1 \text{ or } i+2 & \text{when } i=0,1,2,\cdots n-3 \\ C_i \leftarrow i+1 & \text{when } i=n-2 \end{cases} \quad (E-5)$$

Execute or skip sequencing function: δ_E

$\delta_E(q_i, C_i)=q_{Ci}$ where $C_i \in \Sigma_E$ $(E-6)$

Automaton

$M_E=\{T, \Sigma_E, \delta_E, g_E, q_0, F\}$ $(E-7)$

Former automaton or serial sequencing model have condition input C_i which has value of $i+1$. Thence controlability can be deduced. For $n<N$, controlability is n. For $n \geq N$, controlability is N. Controlability for this model is extremely small.

Later automaton or the execute or skip sequencing model have some higher controlability than the former. Controlability for this model is follows.

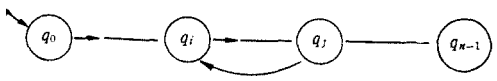
$A_E=\sum_{i=1}^{M} {}_nC_M$ where $M=n$ when $n<N$,

$M=N$ when $n \geq N$

In comparison with the best controlability $A(\sum_{i=1}^{N} n^i)$, this valaue is less than for $N>1$ and same for $N=1$.

Ⅳ. New kinds of JCL model and comparative study.

One of the new JCL type which I will consider is 'conditional looping sequencing' model as following figure.



Some part of the JCL steps can be repeated as many as finite number of times. It behaves like looping of the high level programming lanages.

Automaton for this model is as following.

Terms; $T=\{q_0, q_1, q_2, \cdots q_{n-1}\}$ $(L-1)$

Initial term; q_0 $(L-2)$

Final term; $F\subseteq T$ $(L-3)$

Condition;

$\Sigma_L=\{C_i|0\leq C_i\leq i+1,\ C_i$ is integer when $i=1, 2, 3, \cdots n-2\}$ $(L-4)$

Labels;

$L=\{l_k|k=0, 1, 2, \cdots n-1\}$ $(L-5)$

Condition generator; g_L

$g_L(C_i)=(C_i\leftarrow m)$ where m is integer, $0\leq m\leq i+1$ when $i=1, 2, 3, \cdots, n-2$ $(L-6)$

Conditional looping function; δ_L

$\delta_L(q_i, c_i)=q_{Ci}$ where $C_i\in\Sigma_L$ $(L-7)$

Automaton; M_L

$M_L=\{T, \Sigma_L, L, \delta_L, g_L, q_0, F\}$ $(L-8)$

Another type of JCL model that we can think out is 'Conditional branch sequencing' model as following.

When reached at one term of the series, it is always possible to branch to any term on the series if N selection is not terminated.

Automaton for this is as follows.

Terms; $T=\{q_0, q_1, q_2, \cdots q_{n-1}\}$ $(B-1)$

Initial term; q_0 $(B-2)$

Final term; $F\subseteq T$ $(B-3)$

Condition;

$\Sigma_B=\{C_i|0\leq C_i\leq n-1,\ C_i$ is integer where $i=0, 1, 2, \cdots, n-2\}$ $(B-4)$

Labels;

$L=\{l_k|k=0, 1, 2 \cdots n-1\}$ $(B-5)$

Condition generator; g_B

$g_B(C_i)=(C_i\leftarrow m)$, m is integer, $0\leq m\leq n-1$

$C_i\in\Sigma_B$ $(B-6)$

Conditional branch sequencing function; δ_B

$\delta_B(q_i, C_j)=q_{Ci}$ where $C_i\in\Sigma_B$ $(B-7)$

Automaton

$M_B=\{T, \Sigma_B, L, \delta_B, g_B, q_0, F\}$ $(B-8)$

First, let's compare M_D with M_E. Comparing $(D-4)$ with $(E-4)$, we can get clearly $\Sigma_D\subset\Sigma_E$. With this result in mind we can compare $(D-5)$ with $(E-6)$ to conclude $\delta_D\subset\delta_E$. This result can be applied to comparison $(D-6)$ with $(E-7)$. Thence we can finally obtain

$$M_E\supset M_D \qquad ②$$

Second, let's consider the relation between M_E and M_B. Comparing $(E-4)$ with $(B-4)$ directly, we can get $\Sigma_E\subset\Sigma_B$. This can be applied to $(E-6)$ and $(B-7)$ and we get $\delta_E\subset\delta_B$. Then next, we can easily find the relation $g_E\subset g_B$. Applying these results to $(E-7)$ and $(B-8)$ shows us the next conclusion

$$M_E\subset M_B \qquad ③$$

Third, we will consider the relation between M_L and M_B. Comparing $(L-4)$ and $(B-4)$ we get $\Sigma_L\subset\Sigma_B$. Applying this to $(L-7)$ and $(B-7)$ we can find $\delta_L\subset\delta_B$. From this result we can recognize next relation.

$$M_L\subset M_B \qquad ④$$

Then, from the results ②, ③ and ④, we can find interrelation between the automata.

$$M_D\subset M_E\subset M_B$$

and

$$M_L \subset M_B$$

That is, M_B is superset of the automata M_D, M_E and M_L. Conditional branch sequencing model for JCL sequence control includes all other models which we have assumed.

Ⅴ. Controlability of M_B

The conditional branch sequencing automaton selects one of the n terms when control reached to any term. Therefore, the total number of possible sequences which consist of N terms is as following A_B

$$A_B = {}_nP_1 \cdot {}_nP_2 \cdots {}_nP_N = \sum_{i=1}^{N} n^i \qquad ⑤$$

A_B is the controlability of the conditional branch sequencing automaton.

Comparing A with A_B, we find out that both A and A_B have same value. That is, controlability of the conditional branch sequencing automaton is the same value with the best controlability which is previously defined.

Ⅵ. Conclusion

Conditional branch sequencing automaton M_B not only includes all proposed automata but also has the best controlability.

Since conditional branch sequencing automaton is represented as following expression,

$$M_B = \{T, \Sigma_B, L, \delta_B, g_B, q_0, F\}$$

then we can construct a new JCL which consists of some elements of M_B.

These element can be follows.

$$L, T, g_B, \delta_B, F$$

L is label of the job control statement

T is the body of the statement, *i.e.* function itself of the statement.

g_B is the condition generator of each statement.

δ_B is branching function, which will let branch to another statement by the condition generated by condition generator. We will call this new JCL by the name of ANJCL.

It is possible that we split any application program into modules and arrange them in order of job steps. If we contruct these job steps by ANJCL, then we can control its execution sequence at will without manual handling. It will needs only condition input for certain sequence.

Another application we may consider is batch job control. And batch job programs have to be executed in certain sequence. If the programmer use ANJCL with certain condition input, he can select his job steps without manual job card handling.

Reference

1. Operating system, Madnick and Donovan, McGraw-Hill.
2. Formal languages and their relations to automata, Hopcraft and Ullman, Addison Wesley.
3. Switching and finite automata theory, Kohavi, McGraw-Hill.
4. Algebraic theory of automata, F.Gecseg and I. Park, 1969, Publishing house of Hungarian Academy of Science.
5. Sequential Machines and Automata theory, Willey, New York.
6. IBM OS/VS 1 JCL Services, VS 1 release 3, IBM.