The application of precedence operations to interactive Canadian income tax calculations

F. J. B. Goddu

1166 Whiteoaks Avenue, Mississauqa, Ontario, L5J 3B5 Canada

Precedence operations provide a method of assigning a sequential priority code to items required to calculate Canadian income tax. Such a priority code is useful to assure that a taxpayer can supply these items interactively in a 'natural' order, thus simplifying the required programming. This paper describes one method of establishing a priority code from a precedence matrix for the relatively large number of items required to calculate Canadian income tax.

(Received October 1974)

1. Introduction

Canadian income tax law requires that anyone earning income in Canada report this income along with appropriate deductions, calculate the tax due, and submit the report on returns and schedules supplied by the Revenue Department. Residents of the province of Quebec must submit a similar but independent return. Taken together, the two basic returns consist of several hundred distinct items of information organised on six pages. The returns are distributed by the Revenue Department in January of each year. The completed returns must be filed by the end of April. Any system designed to process income tax information is thus constrained to a three month period. During this relatively short period, any modifications required from previous years' calculations must be implemented and proved. The complexity of these modifications is dependent on the relationships between the items of information required to complete the returns.

A significant number of these items are used by the Revenue Department to identify the individual taxpayer. This group of items includes the taxpayer's name and address, social insurance number, province of residence, etc. These items do not have computational interrelationships and can therefore be ignored for the purposes of this paper.

Even though these items are excluded, there still remain at least 250 distinct items of information whose value and calculation are dependent on one another. For example, 'Total Employment Earnings' is the sum of 'Total Earnings' and 'Tips and Gratuities'. 'Total Employment Earnings' is then itself used for several calculations and participates along with 'Expense Allowance' in the calculation of 'Net Employment Earnings'.

When an analysis of the feasibility of specifying an interactive algorithm for aiding taxpayers with their income tax calculations was made, it became clear that some method for establishing the precedence relationships of these items would be useful.

A large number of items is involved. The relationships between these items are known and relatively trivial. The items, with manageable exceptions, are of the same rank, i.e. they are directly associated with an individual taxpayer. In developing the application, a 'seat of the pants' technique was used to assign a unique priority to each item included in the precedence matrix. In the hope that a theoretical justification for the technique can be developed, this paper describes the use of the priority code, the procedures used to assign the code to each item, and the assumptions made concerning the characteristics of the items and their interrelationships.

2. The use of a priority code

The interactive system envisioned consists of an individual taxpayer armed with the pertinent records and documents being supplied with a terminal and a worksheet. The worksheet would contain a descriptive list, in priority code order, of the items required to complete the income tax calculations. The system, when activated, prods the taxpayer to supply the value associated with an item it requires for its calculations. Alternatively, the system produces the value associated with a computed item. The taxpayer records all values on his worksheet. When the values of all the items on the worksheet have been established, the taxpayer transposes the results on the return supplied by the government with the aid of page and line references associated with a particular item on the worksheet.

The priority code is an explicit statement of the sequence in which individual items must be processed. Two constraints will apply:

(*a*) The taxpayer should only supply the value of any required item once.

(*b*) The sequence of processing should be comfortable to the taxpayer.

The specific requirement chosen is that computations be grouped by type. For example, calculations of income, calculations of deductions, and calculations of taxes due are the three major groups defined.

To emphasise the user orientation of this design it is assumed, for now, that sufficient computer resources exist to preclude any concern for 'machine' constraints. However, since the priority code is assigned before programming begins, a more efficient program structure should result. The priority code is established by iteratively operating on the precedence matrix (Langefors, 1970) (or the connectivity matrix (Ward, 1974)) defined by the items which make up the system.

3. A precedence matrix

A precedence matrix, P, is one means of representing the hierarchic structure of items within a system. Another representation is a directed graph (Smyth and Radaceanu, 1974; Langefors, 1970). Though harder to visualise, the matrix representation is more suited to a system with a large number of items. Considering P as an operator, it maps a set of items onto the set of its immediate precedents. The portion of the precedence matrix used in this paper that describes the dependencies of the item 'Total Employment Earning' described above is shown in **Fig. 1.** Since, element $p_{1,4}$ of the

	1	*2*	*3*	*4*	*5*
1	0	0	0	1	1
2	1	0	0	0	0
3	1	0	0	0	0
4	0	0	0	0	0
5	0	0	0	1	0

Fig. 1 Precedence matrix P

matrix is non-zero, the item 'Net Employment Earnings' requires the item 'Total Employment Earnings'. As for all non-zero elements in the matrix, it is an unconditional requirement. The corresponding directed graph is shown in **Fig. 2.** In the terminology of a directed graph, a row of the matrix with only zero elements identifies a 'root' item. A column with only zero elements identifies a 'leaf' item. A non-zero element corresponds to an 'edge'. Following the edges from a leaf item to a root item defines a 'trunk'. The priority code results from the assignment of a sequence to the edges in the system. This implies that the range of the priority code will equal the number of edges.

4. Matrix multiplication

Langefors has defined a procedure for multiplying precedence matrices which has the effect of pushing the edges of a system down the 'trunk' of the hierarchic tree through the immediate succedent items such that items that are second precedents to an item become its immediate precedents. This procedure consists of replacing all columns with non-zero elements by the union of the columns identified by the row number of the non-zero elements. Explicitly, the operation can be described as follows:

Define a matrix

$$P = p_{i,j} \ .$$

Where there are $i = 1, 2, 3, \ldots, n$ rows and $j = 1, 2, 3, \ldots, n$ columns. n corresponds to the number of items represented by the matrix. For each column p_j, find the elements where

$$p_{i,j} = 1 \ .$$

Assume that for the column $j = 1$, this is true for the values $i = a, b, c$. Form the first column, q_1, in a new matrix P^2, $q_{i,j}$ by solving

$$q_1 = p_a \cup p_b \cup p_c \ .$$

The union operator, $\cup$, is accomplished by applying a Boolean OR among the columns p_a, p_b, p_c.

The application of this operation on P results in matrix P^2 as shown in **Fig. 3.** As expected, the item 'Total Employment Earnings' became a 'leaf' item as a result of this operation. The leaf generation associated with this item is therefore two, corresponding to matrix P^2. Items could also become either roots or leaves or neither.

If precedence matrix multiplication is repeated, ultimately all items will become a root and leaf (i.e will have dropped their edges). This occurs when there are only zero elements remaining in the matrix. Maintaining the analogy, the trunk is de-nuded of edges. Thus $P^3 = P^2P$, as shown in **Fig. 4.** Finally, $P^4 = P^3P$ has only zero elements. Knowing when an item became a root or a leaf is the first requirement of establishing the priority code.

5. Circuit errors

Before continuing, as a by-product of precedence matrix multiplication, circuit errors can be detected. A circuit error arises when an item requires itself or requires an item which in turn requires that first item. In complicated systems, the circuit is not always obvious. If a precedence matrix began with a circuit error, it can be detected after the union of precedent columns is done. If the element, $P_{x,x} = 1$ (where $x =$ the number of the column being replaced), then a circuit error has occurred. Unfortunately, the circuit error may not be detected until several generations of precedence matrices have been produced.

6. Priority code

The priority code associated with a particular item is the sequence number that results for that item from the ordering of leaf generations, within root generations and matrix column numbers for all items. Leaf generations and matrix column numbers are ordered in ascending order. Root generations are ordered in descending order. The assigned priority codes for the items of P are shown in **Fig. 5.** The logic behind the priority of leaf generations of the lowest number is that such items are those items which could be required of any subsequent generation. Among items with identical leaf generations, those items that will become roots first should be processed first. Thus those with the highest generation number have priority. To this point, starting from the top of the tree, the priority code assigned to an item corresponds to the sequence resulting from repetitively picking the nearest leaf with the shortest trunk to its root.

For lack of a better idea, among items with identical leaf and

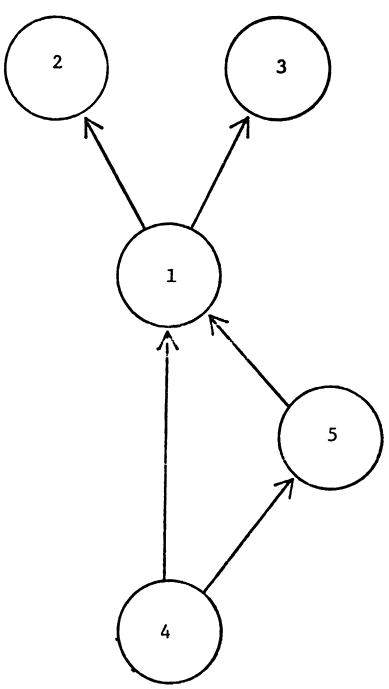

Fig. 2 Directed graph

	1	*2*	*3*	*4*	*5*
1	0	0	0	1	0
2	0	0	0	1	1
3	0	0	0	1	1
4	0	0	0	0	0
5	0	0	0	0	0

Fig. 3 Precedence matrix P^2

	1	*2*	*3*	*4*	*5*
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Fig. 4 Precedence matrix P^3

Row column number	*Item description*	*Generation item became*		*Priority code*
		leaf	*root*	
1	Total employment earnings	2	3	3
2	Total earnings	1	4	1
3	Tips and gratuities	1	4	2
4	Net employment earnings	4	1	5
5	Expense allowance	3	2	4

Fig. 5 Assigned priority code

row generations, the items with the lowest matrix column number will be processed first. If it is assumed that matrix column numbers are assigned to an item and then to its precedents, and it is assumed that items are identified in a 'comfortable' order, then there is some justification for this decision.

In any case, the priority code so established does guarantee uniqueness. In practice it was found that the first assumption could be satisfied. The second will usually not be satisfied. This is especially true as modifications to P are made as a result of structural changes in the system. **Fig. 6** shows a portion of the worksheet produced for the proposed system.

From an examination of the resulting priority code, only the first of the two constraints described in Section 2 has been met automatically. A programmer does have sufficient information, however, to establish a comfortable sequence of processing for the taxpayer. An alternative algorithm is being studied to overcome this limitation. If it can be implemented practically it will be the subject of a future paper.

7. Establishing a precedence matrix

Since the manner in which the items of the system are identified is so critical to the priority code chosen, a formal procedure for identifying items was defined. When analysing business information systems, a method of calculating important items usually is immediately obvious. The complexity of these systems arises from the method required to co-ordinate the arrival of all the necessary items in the correct sequence to accomplish these relatively straightforward computations. Income tax calculations are an ideal basis for proving procedures for recording computational dependencies of items, since, in the worst case, all necessary items must be available at the beginning of the computation. All these items are associated with a single taxpayer. An appropriate key for a Canadian taxpayer is his Social Insurance Number. Best of all, the record so defined need not be maintained. In fact, a strong case can be made for ensuring that the taxpayer be the only one privileged to know the values of the items associated with his return. The system envisioned would delay the submission of items until the last possible moment during the calculations. This is similar to the requirements described by Ward (1974). One important difference is that these procedures are intended for the analysis stage of a system. They ensure that the analyst has submitted sufficient information about an item to enable its place in a precedence matrix to be established accurately. From this perspective, other information given about an item is essentially commentary. The characteristics required can be grouped under four headings: Identification; Basis of typification; Composition; Producing condition.

7.1. *Item identification*

Aside from a 30 character description, a six character identifier is assigned to each item. Six characters were chosen in the hope that programmers would use these identifiers, along with file identifiers, as variable names. How this identifier is used to build-up a precedence matrix is described below. A general

DATE 28/10/74 CALCULATION WORKSHEET JOB SYS PROGRAM SYS04
PERIOD ON REQUEST PAGE 1

FOR VALUES OF THE TYPE INPUT-, YOU WILL BE ASKED TO SUPPLY A VALUE AS FOLLOWS: 999 = ? AN A MEANS THAT VALUES WILL BE ACCUMULATED.
A Q MEANS THAT THE VALUE MUST BE QUALIFIED. YOU MUST SUPPLY INPUT TYPE-E. IN ALL OTHER CASES, YOU MAY REPLY) IF NO VALUE EXISTS.
WHEN THE CALCULATION IS COMPLETE, A VALUE WILL BE PRINTED. AFTER RECORDING THIS VALUE IN THE COLUMN-WORKING VALUE-RESPOND WITH: OK.

DESCRIPTION	TYPE		SEQUENCE		WORKING VALUE	GOVERNMENT FORM, PAGE, AND LINE TO WHICH VALUE MUST BE POSTED		
TOTAL EARNINGS PER T4 SLIP	INPUT-I	A	001	=		FD,02,01		F001
TIPS AND GRATUITIES	INPUT-I		002	=		FD,02,04		F002
EMPLOYMENT EXPENSE PERCENT	COMPUTE		003	=				T001
EARNINGS MAXIMUM	COMPUTE		004	=				T002
ALLOWABLE EXPENSES	INPUT-I		005	=		FD,02,10		F010
ALIMONY RECEIVED	INPUT-I		006	=		QB,02,18		F022
OTHER INCOME	INPUT-I	A	007	=				F055
UNEMPLOYMENT INSURANCE BENEFIT	INPUT-I		008	=		FD,02,18		F015
OLD AGE SECURITY PENSION	INPUT-I		009	=		FD,02,14	QB,02,13	F016
QPP BENEFITS	INPUT-I	A	010	=		FD,02,15		F017
OTHER PENSIONS	INPUT-I	A	011	=		FD,02,16		F018
MO 18 OR DISABILITY PEN. CEASE	INPUT-I		012	=				F037
MO 70 OR DIED	INPUT-I		013	=				F038
LITERAL 12	COMPUTE		014	=				W037A
NET RENTAL INCOME	INPUT-I		015	=		FD,02,21		F209
EARNED INCOME - F	INPUT-I		016	=				F217
PENSION PLAN INDICATOR	INPUT-I		017	=				F057
EARNED INCOME RATE FOR RRSP	COMPUTE		018	=				T008
ADULT TRAINING ALLOWANCES	INPUT-I		019	=		QB,02,07		F004
TIPS AND GRATUITIES-P	INPUT-I		020	=		QB,02,08		F005
CODED PENSIONABLE WAGES	INPUT-I	A	021	=				F040
INSURABLE EARNINGS MAXIMUM	COMPUTE		022	=				T006
CODED INSURABLE EARNINGS AT .6	INPUT-I	A	023	=				F122
CODED INSURABLE EARNINGS AT 1%	INPUT-I	A	024	=				F220
QPP MAXIMUM	COMPUTE		025	=				T003
OTHER PROVINCIAL INCOME	INPUT-I	A	026	=		QB,02,19		F023
QPP EXEMPTION	COMPUTE		027	=				T004
PENSION CONTRIBUTION-CURRENT	INPUT-I	A	028	=				F053
TAXABLE AMOUNT OF DIVIDENDS	INPUT-I	A	029	=		FD,02,19		F020
INSURANCE PREMIUM 1% RATE	COMPUTE		030	=				T007?
INSURANCE PREMIUM .6% RATE	COMPUTE		031	=				T0071
PENSION CONTRIBUTION-PAST SERV	INPUT-I	A	032	=				F052
TUITION FEES ACTUAL	INPUT-I		033	=				W060
TUITION MINIMUM	COMPUTE		034	=				T040
INTEREST	INPUT-I	A	035	=		FD,02,20		E021
OTHER DEDUCTIONS CODED	INPUT-I	A	036	=				E036
QPP PERCENT	COMPUTE		037	=				T005
RRSP CONTRIBUTIONS	INPUT-I	A	038	=				F054
ALIMONY PAYMENTS	INPUT-I		039	=		QB,02,29		E059
MOVING EXPENSES	INPUT-I		040	=		QB,02,33		F062
TAXABLE GAIN OR LOSS	INPUT-I		041	=		FD,02,23		F208
EARNED INCOME - P	INPUT-I		042	=				F218
DUES	INPUT-I		043	=		QB,02,28	FD,02,39	F058
CHILD CARE EXPENSES	INPUT-I		044	=		FD,02,41	QB,02,32	F061
OTHER DEDUCTIONS - P	INPUT-I	A	045	=		QB,02,34		F063
DEPENDENT NET INCOME - F	INPUT-I		046	=		FD,04,18		E077
MEDICAL EXPENSES	INPUT-I		047	=		FD,02,51	QB,02,39	E091
MEDICAL DEDUCTIONS %	COMPUTE		048	=				T018
MARRIED CLAIM - F	COMPUTE		049	=				T016
EXEMPTION INDICATOR	INPUT-I		050	=		FD,04,15	FD,04,16	E074
YEAR OF BIRTH	INPUT-E		051	=		FD,01,09	QB,01,06	E198
BIRTH YEAR FOR INELIGIBILITY	COMPUTE		052	=				T010

Fig. 6

format code distinguishing among amounts, dates, characters, and strings is required along with an initial (default) value for the item. Document references are allowed, e.g. page and line or card columns. Provision has been made for recording two comment lines. Finally, the type of computation must be identified. The possibilities allowed are that the item value be computed, that the item value be supplied by the taxpayer or processing will terminate, that the item value will be used, or that the item value be supplied by the taxpayer or the value will be computed.

7.2. *Basis of typification*

Any classification of taxpayers is based on characteristics of these taxpayers as denoted by the items of information submitted by a taxpayer. One item of information required of a taxpayer is income level. As for all items, there will be a valid range of values this item may be given. In this case, the possible values are the range from zero upward. Bases can be used to identify the 'income level' types into which it is desired to split the class of taxpayers. A base is created by subdividing the range of acceptable values for an item into groups denoting the characteristics which constitute the basis of typification. It is important that each base which is defined exhaust all the values of an item. Thus, a base to typify taxpayers by income level could be defined as follows:

1. Taxable income less than $1,000.
2. Taxable income exceeds $999 and less than $5,000.
3. Taxable income exceeds $4,999.

An appreciation of the powerful utility of this technique results from its consistent application to all items. This is especially true when an analyst is defining relational edits among items. The procedures defined allow for 30 groups of valid ranges for one particular item. Three logical operations are allowed:

1. Equality to value
2. Inequality to value
3. Greater than value 1 and less than value 2.

7.3. *Composition*

The composition of an item involves assigning a value to that item by completing some operation on a set of precedent items which have correct values. For this tax system, up to three precedent items can participate in any one of a relatively simple set of operations. These operations are addition, multiplication, division, subtraction, minimum, maximum, equality (takes the same value as) or literal. These operations must be defined such that they will be completed successfully. That is, the item being produced must have an acceptable value as defined by the basis of typification for that item. If this requirement is adhered to, the assumption can be made, when defining these operations, that the precedent values will have correct

```
    DATE 28/10/74              ELEMENT DEFINITION AS OF 741023                    JOB SYS PROGRAM SYS04
    PERIOD ON REQUEST                                                                      PAGE

E001    100 DESCRIPTION: TOTAL EARNINGS PER T4 SLIP      PRIORITY:   1   CHANGED: 740623          01-43
  Y     200 TYPE IS - 3: IF NOT INPUT USE INITIAL VALUE
  I     300 FORMAT  - 2: ADDED AMOUNT RIGHT JUSTIFIED                    INITIAL: 0000000000
  I     401 DOCUMENT ID: PAGE FD  LINE 02  POSITION 01
  I
  V
E001

E002    100 DESCRIPTION: TIPS AND GRATUITIES             PRIORITY:   2   CHANGED: 740623          01-43
  Y     200 TYPE IS - 3: IF NOT INPUT USE INITIAL VALUE
  I     300 FORMAT  - 1: SIMPLE AMOUNT RIGHT JUSTIFIED                   INITIAL: 0000000000
  I     401 DOCUMENT ID: PAGE FD  LINE 02  POSITION 04
  I
  V
E002

E003    100 DESCRIPTION: TOTAL EMPLOYMENT EARNINGS       PRIORITY: 105   CHANGED: 740623          02-42
  Y     200 TYPE IS - 2: COMPUTE AS SHOWN BELOW
  I     300 FORMAT  - 1: SIMPLE AMOUNT RIGHT JUSTIFIED                   INITIAL: 0000000000
  I     401 DOCUMENT ID: PAGE FD  LINE 02  POSITION 05
  I     601 SOURCE  - 4: USE E001   +E002
  I
  V
E003

E004    100 DESCRIPTION: ADULT TRAINING ALLOWANCES       PRIORITY:  19   CHANGED: 740623          01-34
  Y     200 TYPE IS - 3: IF NOT INPUT USE INITIAL VALUE
  I     300 FORMAT  - 1: SIMPLE AMOUNT RIGHT JUSTIFIED                   INITIAL: 0000000000
  I     401 DOCUMENT ID: PAGE QB  LINE 02  POSITION 07
  I
  V
E004

E005    100 DESCRIPTION: TIPS AND GRATUITIES-P           PRIORITY:  20   CHANGED: 740623          01-34
  Y     200 TYPE IS - 3: IF NOT INPUT USE INITIAL VALUE
  I     300 FORMAT  - 1: SIMPLE AMOUNT RIGHT JUSTIFIED                   INITIAL: 0000000000
  I     401 DOCUMENT ID: PAGE QB  LINE 02  POSITION 08
  I
  V
E005

E006    100 DESCRIPTION: TOTAL EARNINGS PER TP4 SLIPS    PRIORITY: 109   CHANGED: 740623          02-34
  Y     200 TYPE IS - 1: IF NOT INPUT THEN COMPUTE IT
  I     300 FORMAT  - 2: ADDED AMOUNT RIGHT JUSTIFIED                    INITIAL: 0000000000
  I     401 DOCUMENT ID: PAGE QB  LINE 02  POSITION 06
  I     601 SOURCE  - 1: USE E001
  I
  V
E006

E007    100 DESCRIPTION: TOTAL EMPLOYMENT EARNINGS-P     PRIORITY: 135   CHANGED: 740623          03-33
```

Fig. 7

values. The precedent items are defined by means of their six character identity.

7.4. *Producing conditions*
An item can be computed in as many as 30 different ways depending on the satisfaction of any one of 30 mutually exclusive producing conditions. A producing condition is defined by combining up to four bases for precedent items. All specified bases must be true for the producing condition to be activated.

Using this procedure, an item can be defined with, theoretically, up to 210 precedent items. The trivial circuit of defining the item itself as a precedent item is immediately obvious and can be corrected. A sample item definition is shown in **Fig. 7.**

8. Defining the matrix

From the above information, the matrix, *P*, can now be defined. The information coded above is ordered using the value of the item identity. By searching the composition and producing conditions of the first item, a list of precedent item identities is produced in the order the items were coded. Items used in more than one operation or condition are only placed in the list the first time they occur. The first column of matrix *P* is now created. The item associated with the second row (and column) corresponds to the first precedent item in the list produced above. Successive rows and columns are created from this list. The information about the next item in the file is then read and a new precedent list is established. However, when columns and rows are now posted to matrix *P*, the columns or rows assigned for all previous items must be examined to ensure that a column or row has not already been established for that item. When all items in the file have been examined a square precedence matrix with as many columns as there are items will have been produced.

9. Limitations

In working with test results, two limitations were uncovered which require further analysis. The system will request an item of the taxpayer even though it is easily deduced that it is not required. Similarly, the system will produce all referenced values even though they have no value, i.e. have been assigned the default value. In certain situations, this is unnecessarily tedious for the taxpayer.

10. Operating characteristics

The procedures described in this paper are operational. The necessary programs have been written in COBOL or BAL, as appropriate, using an IBM 360-65. Thirty-three generations of the income tax precedence matrix were required to assign the priority code and produce the element definition list and worksheet. A significant effort was required to assure that this processing could be accomplished in a reasonable amount of computer time (10 minutes). A simulation of the taxpayer calculations is now (August 1974) being tested.

11. Conclusion

A procedure for assigning a sequential priority code to income tax calculations has been presented. The procedure seems suited to systems composed of items whose representation by means of a directed graph could be called a forest of tall trees. It would seem unsuited for systems where hierarchic relations are best represented by a forest of short squat hedges. Further research will be done to attempt to expand the procedure to multi-level files.

References

BARRATT, J. (1970). Bases of Typification, *DPM* 5, IOS Computer Systems S.A., Nyon, Switzerland.
LANGEFORS, B. (1970). *Theoretical Analysis of Information Systems*, Lund University, Stockholm, Sweden.
SMYTH, W. F., and RADACEANU, E. (1974). A Storage Scheme for Hierarchic Structures, *The Computer Journal*, Vol. 17, No. 2, pp. 152-156.
WARD, R. G. (1974). A Variable Delay Method for Improving Programs, *The Computer Journal*, Vol. 17, No. 2, pp. 157-164.