Qualitative Economics in Prolog

R. Berndsen and H. Daniels
QUALITATIVE ECONOMICS IN PROLOG

Ron Berndsen and Hennie Daniels

Faculty of Economics
Institute for Language Technology and AI.
Tilburg University
P.O.Box 90153
5000 LE Tilburg, The Netherlands.
E-mail daniels@htikub5.bitnet.

Abstract.

In this paper we describe a formalism for qualitative reasoning in economics. The framework may serve as a common basis for the intuitive reasoning practised by experienced economists and the more formal qualitative models recently established in the field of artificial intelligence. The emphasis is on representation and implementation aspects of qualitative models. The formalism is illustrated in a well-known Keynesian model. A classification of the qualitative behaviours of the economic system can be generated automatically using a Prolog code. It is shown that the standard economic behaviour put forward by economists corresponds to a homomorphic image of the complete system.

1. INTRODUCTION

The developments in qualitative modelling originate mainly from AI research in the field of electronic circuit analysis, [Davis (1984), Genesereth (1984)] elementary physics [Kuipers (1986), Hayes (1979)] and medical diagnosis [Chandrasekaran and Mittal (1983), Kuipers and Kassirer (1984)]. Only recently have researchers started to consider the exploration of these ideas in economic theory, which has led to some interesting results. For example, the differences between quantitative and qualitative models are illustrated by the classical macro-economic theory of output and employment in Farley (1986).
Constraint propagation techniques are applied to a model concerning the equilibrium of the commodity and labour market in Bourgine and Raiman (1986). In Pau (1986) qualitative arguments occurring in government texts addressing economic subjects are mapped into a formal grammar. The application of qualitative dynamics to a Keynesian model is described in Berndsen and Daniels (1988).

Application of AI methods undoubtedly contributed to the understanding of economic reasoning. However, some of the underlying ideas in qualitative modelling have already been published in the economic literature. The similarity between the theory of confluences [de Kleer and Brown (1984)] and comparative statics [Samuelson (1947)] is pointed out in Iwasaki and Simon (1986a) and a profound treatment of qualitative statics can be found in Greenberg and Maybee (1981).

This reexploration in the application of formal qualitative modelling in economics is mainly due to the tremendous increase in computer power and the proliferation of symbolic programming languages such as Lisp and Prolog. One of the intrinsic reasons for studying qualitative methods is the lack of consistent data that are indispensable in quantitative models. A practical reason is the intractability of the huge amounts of computer output of complex numerical models. Other reasons are the wish to create automatic procedures for tracing causal chains, and to provide procedures to support the validation of the structure of economic models [Fontela (1986), Royer and Ritschard (1984), Bourgine and Raiman (1986), Boutillier (1984)]. In any case we believe that qualitative modelling provides a way of filling the gap between number crunching programs and verbal intuitive reasoning.

In this paper we propose a constraint oriented approach for qualitative modelling. The method can be positioned somewhere between the theory of qualitative reasoning based on confluences [de Kleer and Brown (1984)] and qualitative simulation [Kuipers (1986)]. In the formalism proposed in this article, qualitative dynamic models consist of standard symbolic constraints (e.g. originating from balance sheet equations), constraints representing contemporaneous causality (if two economic entities influence each other directly) and sequential causality (if the influence is unidirectional and there is a time lag involved [cf. Hicks (1979)].

The explicit modelling of causality seems quite natural in economics. In earlier papers [Iwasaki and Simon (1986a, 1986b), de Kleer and Brown (1986)]
causal relations are derived from a static mathematical model. It can be shown that the causality derived from static models by the methods of causal ordering and mythical causality does not reflect the intuitive notion of causality [Iwasaki (1988), Berndsen and Daniels (1989)]. One way to get around this problem is to consider dynamic models [Iwasaki (1988)]. However, we believe that the dynamic model possesses a level of detail which is unnecessary to describe the qualitative behaviour of economic systems. Therefore we start from a declarative representation of causality based on behavioural laws of economics. Similar ideas have been considered in the description of physical devices (Rieger and Grinberg (1977)).

In section 2 the formal semantics of the constraint language are described. The qualitative behaviour of the economic model consists of all possible sequences of admissible states. This envisionment can be represented as a rooted digraph. In section 3 we discuss the implementation of the algorithm in Prolog. In section 4 the different qualitative behaviours of the Keynesian example are presented. Furthermore, an alternative view is shown in which the envisionment is interpreted as an automaton. In doing so it is possible to abstract from the detailed complete description. In this case states and transitions can be clustered to obtain a homomorphism of the automaton onto itself. It turns out that the standard behaviour usually put forward by economists corresponds to a generalized homomorphism of the complete system in the sense as described in Bavel (1983).

2. QUALITATIVE MODELLING

In this section we describe a formalism for qualitative modelling of economic systems. This formalism is an intermediate form of the method of qualitative simulation (QSIM) given by Kuipers (1986) and the theory of confluences of de Kleer and Brown (1984). The main differences emerge from the fact that they were mainly interested in simulating the qualitative behaviour of physical systems.

2.1 Formalism

In the following, an economic system \( S \) is defined as:
i) a set of variables \( V = \{ v_j \} \) (\( j = 1, \ldots, n \)).

ii) a set of quantity spaces \( QSval_j \) and \( QSdir_j \) for every variable \( v_j \).

iii) a set of constraints \( C \).

Furthermore, time is represented by a finite set of half-open time intervals of uniform length:

\[
T = \{ [t_0, t_1), \ldots, [t_{n-1}, t_n) \} = \{ i_0, \ldots, i_{n-1} \}.
\]

For every variable \( v_j \) two functions on \( T \) are defined:

\[
Qval(v_j): T \rightarrow QSval_j \text{ denoting the qualitative value of } v_j \text{ at } t_k \in T \text{ and } \\
Qdir(v_j): T \rightarrow QSdir_j \text{ denoting the qualitative direction of } v_j \text{ in } i_k \in T.
\]

The interpretation is that \( Qval(v_j) \) is defined at the beginning of time interval \( i_k \) i.e. time point \( t_k \) and \( Qdir(v_j) \) is defined over the complete interval. \( QSval_j \) and \( QSdir_j \) are called quantity spaces. A quantity space is a totally ordered finite set of symbolic values. Various quantity spaces have been proposed in the literature [cf. de Kleer and Brown (1984), Kuipers (1986), Raiman (1986)]. Here we take for \( Qdir \) the quantity space \( QSdir_j = \{ inc, std, dec \} \) and for \( Qval \) either \( QSval_j = \{ +, 0, - \} \) or \( QSval_j = \{ \lambda \} \). In the first case, the set \( \{ +, 0, - \} \) denotes the relative position of a variable \( v_j \) with respect to an important (landmark) value \( \lambda \). For example, if \( v_j \) denotes excess demand, \( \lambda \) could be the value of \( v_j \) at which the corresponding market is in equilibrium. In the second case, \( QSval_j \) is restricted to a single element \( \{ \lambda \} \) which may denote \( <<-\infty, \infty> \) or \( <0, \infty> \). This is the quantity space for variables for which only the qualitative direction is of importance.

A qualitative state \( QS(v_j, i_k) \) of a variable \( v_j \) at \( i_k \) is defined as the pair \( (Qval(v_j, t_k), Qdir(v_j, i_k)) \). A qualitative state of an economic system \( J \) at \( i_k \) is the list of qualitative states of the \( n \) variables \( v_j \):

\[
QS(J, i_k) = QS(v_1, i_k), \ldots, QS(v_n, i_k).
\]

An admissible qualitative state \( QS(J, i_k) \) is a qualitative state of \( J \) such that all constraints are satisfied simultaneously. The corresponding
Assignment of qualitative states to all variables is called a valid interpretation.

Constraints are relations among variables that impose restrictions on combinations of Qval’s or Qdir’s of the variables in the constraint. There are several types of constraints. Some constraints correspond to familiar mathematical operators, such as addition and differentiation, in a qualitative context. Other constraints define monotonic and causal relationships between variables. A constraint is satisfied if the conditions corresponding to the constraint are met. The constraints that apply in the Keynesian model are defined in subsection 2.2.

A valid state transition is a ordered pair \( \{x_i, i_k\}, \{x_i, i_{k+1}\} \) in such a way that \( x_j \) \( \{x_j, i_k\}, \{x_j, i_{k+1}\} \) is a valid variable transition and \( \{x_j, i_{k+1}\} \) is admissible. The set of valid transitions can be divided into two disjoint subsets \( QD \) and \( QS \) (listed in Table 1 and 2 respectively). If the quantity space for a particular variable \( QSval_j = \{\lambda\} \) then only transitions of the qualitative directions need to be taken into account. These transitions are called \( QD \)-transitions. Otherwise, \( QSval_j = \{+,-,0\} \) and so-called \( QS \)-transitions apply.

**TABLE 1  QD-transitions**

<table>
<thead>
<tr>
<th>QDi</th>
<th>Qdir(x_i,i_k)</th>
<th>Qdir(x_i,i_{k+1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>QD1</td>
<td>Any</td>
<td>std</td>
</tr>
<tr>
<td>QD2</td>
<td>Any</td>
<td>inc</td>
</tr>
<tr>
<td>QD3</td>
<td>Any</td>
<td>dec</td>
</tr>
</tbody>
</table>

**TABLE 2  QS-transitions**

<table>
<thead>
<tr>
<th>QSi</th>
<th>QS(x_i,i_k)</th>
<th>QS(x_i,i_{k+1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>QS1</td>
<td>(0,std)</td>
<td>(0,Any)</td>
</tr>
<tr>
<td>QS2</td>
<td>(0,inc)</td>
<td>(+,Any)</td>
</tr>
<tr>
<td>QS3</td>
<td>(0,dec)</td>
<td>(-,Any)</td>
</tr>
<tr>
<td>QS4</td>
<td>(+,dec)</td>
<td>(0,Any)</td>
</tr>
<tr>
<td>QS5</td>
<td>(+,Any)</td>
<td>(+,Any)</td>
</tr>
<tr>
<td>QS6</td>
<td>(+,dec)</td>
<td>(-,Any)</td>
</tr>
<tr>
<td>QS7</td>
<td>(-,dec)</td>
<td>(0,Any)</td>
</tr>
<tr>
<td>QS8</td>
<td>(-,Any)</td>
<td>(-,Any)</td>
</tr>
<tr>
<td>QS9</td>
<td>(-,inc)</td>
<td>(+,Any)</td>
</tr>
</tbody>
</table>

where each \( QDi \) is a subset of 3 transitions with \( Any \in QSdir = \{inc, std, dec\} \). Analogously, \( QSi \) is a subset of 3 or 9 (\( QS5 \) and \( QS8 \)) transitions.

A qualitative behaviour of a variable \( v_j \) from \( i_k \) to \( i_{k+n} \) is a sequence of qualitative states with valid transitions between them:

\[ QS(v_j,i_k), \ldots, QS(v_j,i_{k+n}) . \]
Accordingly, a qualitative behaviour of the system \( \mathcal{S} \) from \( i_k \) to \( i_{k+n} \) is the corresponding sequence of admissible qualitative states of \( \mathcal{S} \).

The environment of \( \mathcal{S} \) with initial state \( QS(\mathcal{S},i_0) \) is a rooted directed graph \( E \) with the following properties:

a) \( QS(\mathcal{S},i_0) \) is the root.

b) the set of nodes of \( E \) contains all admissible qualitative states of \( \mathcal{S} \) that are reachable from the root by valid state transitions.

c) there is a link between two nodes of \( E \) iff there exists a valid state transition between them.

A path from the root to another node corresponds to some qualitative behaviour of the system.

2.2. Application to the Keynesian model

In this subsection we formulate a Keynesian model in a constraint representation. The model has \( \lambda \) variables \( \{C,I,Y,M_1,M_2,M_d,r\} \) and \( \lambda \) constraints. The quantity space \( QS_{val} \) for \( M_d \) is \( \{*,0,-\} \) and for the other variables \( \{\lambda\} \) where \( \lambda \) stands for \([0,\infty)\). The constraints are given by:

\[
\begin{align*}
\text{ADD}(C,I,Y) & \quad (1) \\
\text{ADD}(M_1,M_2,M_d) & \quad (2) \\
M_1^+(M_1,Y) & \quad (3) \\
\text{DERIV}(r,M_d) & \quad (4) \\
\text{SC}^+(Y,C) & \quad (5) \\
\text{SC}^-(r,I) & \quad (6) \\
\text{SC}^-(r,M_2) & \quad (7)
\end{align*}
\]

Constraint (1) denotes the national accounting identity in a closed economy without a government \( (Y = C + I) \) and in (2) the total money demand is defined as the sum of \( M_1 \) and \( M_2 \). The relationship between \( M_1 \) and \( Y \) is modelled by a monotonicity constraint. This corresponds to the formal representation of contemporaneous causality [Hicks (1979)]. Constraints (5), (6) and (7) are the constraints representing sequential causality. They impose a relation on the direction of change of the first variables and the direction of change of the second variable in the next time interval. In the \( \text{SC}^+ \)-constraint both variables point in the same direction whereas in the \( \text{SC}^- \)-constraint the
directions are opposite. Constraint (4) reflects the adjustment mechanism of the money market.

In the following, we describe the restrictions induced on the variables by each constraint.

ADD-constraint

ADD(a,b,c) defines the variable c as the qualitative sum of the variables a and b. Depending on the particular application at hand, it is possible to take both Qval and Qdir into account or only Qdir. The former case applies only if for all variables joined by an ADD-constraint QSval = {+,-,0}. It is assumed that the ADD-constraint holds for the tuple (0,0,0). A tuple of qualitative values of the variables a,b and c satisfy the ADD-constraint at \( i_k \) if:

\[
\text{Qval}(a,i_k) \oplus \text{Qval}(b,i_k) = \text{Qval}(c,i_k)
\]

Where \( \oplus \) (qualitative addition) and \( \simeq \) (weak equality sign) are defined by the following tables:

<table>
<thead>
<tr>
<th>@</th>
<th>+</th>
<th>-</th>
<th>0</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>?</td>
<td>+</td>
<td>?</td>
</tr>
<tr>
<td>-</td>
<td>?</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>+</td>
<td>-</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \simeq )</th>
<th>+</th>
<th>-</th>
<th>0</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>-</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>0</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>?</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The weak equality sign \( \simeq \) is a two-place predicate. Here we will not go into details of qualitative algebra, the interested reader is referred to Dormoy and Raiman (1988) and Williams (1988).

Furthermore, the ADD-constraint puts also a restriction on the Qdir's of a,b and c, which is equivalent to the restriction on the Qval's.

\( M^+ \) and \( M^- \)-constraint

The monotonicity constraints \( M^+(a,b) \) and \( M^-(a,b) \) define a monotonic functional relationship between a and b. \( M^+ \) is appropriate if the relationship between a and b is monotonic and increasing. Conversely, if the relationship is
decreasing and monotonic the $M^-$-constraint applies. The monotonicity constraint puts a restriction on the $Qdir$'s of $a$ and $b$, namely for the $M^+$-constraint: $Qdir(a,i_k) = Qdir(b,i_k)$, and similarly with a minus sign for $M^-$.  

**DERIV-constraint**

The derivative relation between two variables is represented by the DERIV-constraint. DERIV($a$, $b$) is satisfied at $i_k$ iff the pair ($Qdir(a,i_k), Qval(b,i_k)$) matches one of the entries in the table below:

<table>
<thead>
<tr>
<th>DERIV($a$, $b$)</th>
<th>$Qdir(a,i_k)$</th>
<th>$Qval(b,i_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>std</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>inc</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>dec</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

In the Keynesian model the adjustment mechanism on the money market is represented by DERIV($r,M_d$), so $QSval(M_d)$ must be $\{+,0,-\}$. This quantity space denotes the relative position of $M_d$ with respect to the exogenous money supply $M_s$.

**SC$^+$- and SC$^-$-constraint**

The causal constraints SC$^+(a,b)$ and SC$^-(a,b)$ denote the relation of sequential causality between $a$ and $b$. SC$^+(a,b)$ holds if $a$ influences $b$ positively. If the influence of $a$ on $b$ is negative, then SC$^-(a,b)$ holds. The constraint SC$^+(a,b)$ puts a restriction on the pair ($QS(a,i_{k-1}), QS(b,i_k)$) as follows:

$Qdir(a,i_{k-1}) = Qdir(b,i_k)$ and similarly with a minus sign for SC$^-$.  

3. **IMPLEMENTATION ISSUES**

In this section we describe some implementation issues of the qualitative reasoner. This algorithm performs two tasks: 'envisioning' and cycle detection.
3.1 Envisionment

To determine the envisionment, the algorithm takes three clauses as input: 'economic_system', 'qs' and 'node'. The clause 'economic_system' specifies a particular model. For example, the Keynesian model is given by the clause:

```
ecological_system(keynes,  
  var([c,i,m1,m2,md,r,y]),  
  qsval(md,[plus,zero,minus]),  
  constraints([splus(y,c),  
                scmin(r,i),  
                scmin(r,m2),  
                deriv(r,md),  
                mplus(m1,y),  
                add(c,i,y),  
                add(m1,m2,md)]).
```

The third argument qsval specifies the quantity spaces. The qualitative value zero in the quantity-space Qsval of the total money demand (md) corresponds with the value of the exogenous money supply. The other variables have quantity spaces Qsval = {~} and Qsdir = {+, -, 0} but are not shown above. The other two clauses 'qs' and 'node' are as follows:

```
qs(N,Time,Qs_variables).  
node(N,Label,Predecessors).
```

The clause 'qs' represents a qualitative state of the system QS( ~, i_k). The first argument N is a unique number. The domain of the variable Time is the set of time intervals i_k. The third argument is a list of qualitative states of all variables.

The clause 'node' represents a node in the envisionment and its incoming links. The first argument N is a unique number. The second argument of 'node' is a label marking special states. These states are discussed in subsection 3.2. The last variable Predecessors is the list of immediate predecessors that represent incoming links of node N in the envisionment. The variable N in 'qs' and 'node' is used to identify corresponding occurrences of both clauses. For
the initial state $N = 1$ and new states are numbered onwards. The output of the algorithm is the set of all occurrences of 'node' and 'qs'.

In the following, two main parts of the algorithm are discussed. The first part defines the complete envisionment of the system $\mathcal{F}$. This top-level predicate is given by:

```
envisionment([]).
envisionment([N|Open]) :-
    all_admissible_successors(N,Suc),
    concat(Open,Suc,Newsuc),
    envisionment(Newsuc).
```

The variable $N$ is the node of the envisionment to expand. 'Open' is a list of nodes to be expanded later on. The variable 'Suc' is a list of successor nodes of $N$. The variable 'Newsuc' is the concatenation of the lists Open and Suc. The first clause of the procedure given above defines the envisionment of an empty list as true. The second clause states that the envisionment of $[N|Open]$ is true if all admissible qualitative states that are a successor of node $N$ are in Suc and the envisionment of the list Newsuc is true. The nodes of the envisionment are generated breadth-first because the list of successors is appended at the end of the list of open nodes.

The second part of the algorithm is the constraint satisfaction process that determines new admissible states. Constraint satisfaction takes place at two levels. At the first level assignments of qualitative states to variables in a particular constraint are made in such a way that the constraint is satisfied. This is called constraint consistency filtering. At the second level assignments that satisfy individual constraints are compared mutually to verify if these assignments agree on common variables. This is called global consistency filtering. An efficient way of global consistency filtering is Waltz filtering [Waltz (1975)]. In Waltz filtering only pairs of adjacent constraints are considered. A pair of constraints is adjacent if they share one or more variables. For every constraint $C_i$ there is a list $L_i$ of tuples of qualitative states that satisfy $C_i$. Consider an adjacent pair of constraints $(C_i,C_j)$. Tuples on $C_i$ that assign a qualitative state to the common variable which is not in any tuple of $L_j$, are deleted. If it is not possible to delete more tuples from $C_i$ then another pair $(C_i,C_k)$ ($k \neq j$) is filtered where $C_k$ is
adjacent to $C_i$. This process terminates if no more tuples can be deleted from any list $L_i$.

The constraint filtering process is specified by the clause:

```
valid_interpretation(N,Transitions,[C|Cs],New) :-
    filter(C,N,Transitions,Valid_tuple),
    compatible(Valid_tuple,Old,New),
    valid_interpretation(N,Transitions,Cs,Old),
    valid_interpretation(N,Transitions,[],New).
```

The clause 'filter' specifies the constraint consistency filtering part. In this clause the variable $C$ denotes a constraint and $N$ is the current node in the envisionment. Transitions is a list of possible transitions of all variables and Valid_tuple is a tuple of qualitative states that satisfies constraint $C$. Global consistency filtering is specified in the clause 'compatible'. The clause compatible is true if all qualitative states in Valid_tuple do not conflict with the qualitative states on $O.ld$. $O.ld$ is a list of qualitative states of variables that satisfies some but possibly not all of the constraints. In the clause compatible the variable New is the same as $O.ld$ except that New is updated with Valid_tuple. Thus, New is a valid interpretation that satisfies an additional constraint compared to $O.ld$. In the clause 'valid_interpretation' the list of constraints is denoted by $[C|Cs]$. An interpretation is valid if the list $[C|Cs]$ is empty or if the variable Valid_tuple which satisfies constraint $C$ is compatible and the other constraints Cs have a valid interpretation.

3.2 Behaviour in the envisionment

The envisionment of an economic model is the description of all possible behaviours of the model. Among these behaviours only a few categories are interesting. Economists usually look for tendencies towards equilibrium or unstable paths. In the following we define three types of 'interesting' behaviour. To do this first some special states in the envisionment are defined: equilibrium and no_change states. An equilibrium state is a state with $Qdir(v_j) = std$ for every variable $v_j$. A no_change state is a state that is a successor of itself. The three types of interesting behaviour in the
envisionment are defined as follows. Equilibrium behaviour is a path from the root to an equilibrium state. No_change behaviour is a path from the root to a no_change state. Finally, cyclic behaviour is a cyclic path. We restrict cyclic paths to distinct elementary cycles only because other cycles can be thought of as composed of elementary cycles (An elementary cycle is a path where no node but the first and last appears twice and two cycles are distinct if one is not a cyclic permutation of the other). The first two types of behaviour are found easily in the envisionment. The third type of behaviour is generated by a well-known algorithm that finds all distinct elementary cycles of a directed graph.

An efficient algorithm is developed and implemented in Algol W by Johnson (1975). An overview of similar algorithms can be found in Mateti and Deo (1976). The algorithm we implemented in Prolog reads only a few lines of code.

4. RESULTS

In this section, we present the results of the envisionment of the Keynesian model. Furthermore, a classification of the cycles in the envisionment graph is given.

4.1 The envisionment of the Keynesian model

The input of the envisionment algorithm consists of an economic system, an input state 'qs' and the corresponding node in the envisionment. The economic system was given in subsection 3.1. The clauses 'qs' and 'node' are given by:

```
qs(1,i(1),qs_variables([[c,l,std],
                        [i,l,std],
                        [m1,l,std],
                        [m2,l,std],
                        [md,minus,std],
                        [r,l,dec],
                        [y,l,std]])).
```

```
node(1,[root],[]).
```
The root represents a situation where a positive money supply shock is given at $t_0$. The complete envisionment of the Keynesian model consists of 50 states. In Table 3 the qualitative states in the first three intervals are shown. The corresponding part of the envisionment graph is depicted in Figure 1. Dotted arrows denote incoming and outgoing links of states not depicted. No change states are shown as squares.

**TABLE 3  Qualitative states of the Keynesian model**

<table>
<thead>
<tr>
<th>j</th>
<th>$x_j$</th>
<th>$QS(x_j, i_1)$</th>
<th>$QS(x_j, i_2)$</th>
<th>$QS(x_j, i_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>c</td>
<td>$\lambda, \text{std}$</td>
<td>$\lambda, \text{std}$</td>
<td>$\lambda, \text{inc}$</td>
</tr>
<tr>
<td>2</td>
<td>i</td>
<td>$\lambda, \text{std}$</td>
<td>$\lambda, \text{inc}$</td>
<td>$\lambda, \text{inc}$</td>
</tr>
<tr>
<td>3</td>
<td>$m_1$</td>
<td>$\lambda, \text{std}$</td>
<td>$\lambda, \text{inc}$</td>
<td>$\lambda, \text{inc}$</td>
</tr>
<tr>
<td>4</td>
<td>$m_2$</td>
<td>$\lambda, \text{std}$</td>
<td>$\lambda, \text{inc}$</td>
<td>$\lambda, \text{inc}$</td>
</tr>
<tr>
<td>5</td>
<td>$m_d$</td>
<td>$-, \text{std}$</td>
<td>$-, \text{inc}$</td>
<td>$0, \text{inc}$</td>
</tr>
<tr>
<td>6</td>
<td>r</td>
<td>$\lambda, \text{dec}$</td>
<td>$\lambda, \text{dec}$</td>
<td>$\lambda, \text{std}$</td>
</tr>
<tr>
<td>7</td>
<td>y</td>
<td>$\lambda, \text{std}$</td>
<td>$\lambda, \text{inc}$</td>
<td>$\lambda, \text{inc}$</td>
</tr>
</tbody>
</table>

**FIGURE 1  A part of the envisionment graph of the Keynesian model**
As it stands, only two of the three kinds of behaviour as described in subsection 3.2. are possible in the envisionment of the Keynesian model. It can be shown that equilibrium is not a solution of the set of constraints of the Keynesian model (see Berndsen and Daniels (1989)). In the complete envisionment graph, there are 8 no_change nodes, summarized in table 4 with a view on the three variables y, r and md.

TABLE 4 Characterization of no change nodes in the Keynesian model

<table>
<thead>
<tr>
<th>node</th>
<th>Qdir(y)</th>
<th>Qdir(r)</th>
<th>QS(md)</th>
<th>characterization of the money market</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>inc</td>
<td>dec</td>
<td>-,inc</td>
<td>trend to equilibrium</td>
</tr>
<tr>
<td>7</td>
<td>inc</td>
<td>inc</td>
<td>+,std</td>
<td>excess money demand</td>
</tr>
<tr>
<td>8</td>
<td>inc</td>
<td>inc</td>
<td>+,inc</td>
<td>trend divergence</td>
</tr>
<tr>
<td>10</td>
<td>inc</td>
<td>inc</td>
<td>+,dec</td>
<td>trend to equilibrium</td>
</tr>
<tr>
<td>26</td>
<td>dec</td>
<td>inc</td>
<td>+,dec</td>
<td>trend to equilibrium</td>
</tr>
<tr>
<td>29</td>
<td>dec</td>
<td>dec</td>
<td>-,std</td>
<td>excess money supply</td>
</tr>
<tr>
<td>32</td>
<td>dec</td>
<td>dec</td>
<td>-,inc</td>
<td>trend to equilibrium</td>
</tr>
<tr>
<td>33</td>
<td>dec</td>
<td>dec</td>
<td>-,dec</td>
<td>trend divergence</td>
</tr>
</tbody>
</table>

The third kind of behaviour is cyclic behaviour. Although the number of cycles is large, it is possible to classify these cycles in a way that is meaningful from an economic point of view. This can be done by exploring the correspondence between the envisionment and the concept of a finite-state automaton.

4.2 Homomorphism, views and classification

Here, we adopt the definition of a finite state automaton from Bavel (1983). A complete finite-state automaton is a triple $\mathcal{A} = \{S, \Sigma, \delta\}$ where

(i) $S$ is a finite set of states
(ii) $\Sigma$ is a nonempty set
(iii) $\delta : S \times \Sigma \rightarrow S$ is a transition function satisfying

$\delta(s, xy) = \delta(\delta(s, x), y)$ and $\delta(s, \varepsilon) = s$ for all $s \in S$ and $x, y \in \Sigma^*$,

where $\Sigma^*$ is the set of finite sequences of members of $\Sigma$ and $\varepsilon \in \Sigma^*$ is the empty sequence.
A finite-state automaton is incomplete if the domain of δ is a proper subset of S x Σ [Bavel (1983)].

Consider an envisionment E of an economic system. From E we may construct an automaton as follows. The finite set of qualitative states of E is taken as the set S of condition (i). The non-empty set Σ is defined as the set of valid state transitions τ in E. Here, τ is a n x 1 vector [t_j] where t_j is a valid transition of variable v_j (j = 1,...,n). Let δ: T → S be a transition function satisfying δ(i,τ) = j (where i,j ∈ S, τ ∈ Σ and T a proper subset of S x Σ) iff there exists a link in the envisionment from node i to node j with valid state transition τ. It is easily seen that the tuple E_a = {S,Σ,δ} with S,Σ and δ as specified above is an incomplete finite-state automaton.

The observed correspondence can be useful to centre the attention on the behaviour usually put forward by economists. For example a view of the envisionment on y and r can be defined formally as follows. Let ~ denote the equivalence relation on S, where two states s_1 and s_2 are equivalent if they coincide on r and y i.e. QDIR(r) = QDIR(y) for s_1 and s_2. Clearly, ~ is an equivalence relation and the corresponding projection denoted by α_S is a mapping from S to the set of equivalence classes S'. Similarly, we define α_Σ:Σ → Σ'. It can be shown that the pair α = (α_S,α_Σ) induced by a view for any subset of variables is a generalized homomorphism in the sense of Bavel (1983, Ch. 5). The homomorphic image of E_a under α is E_b = {S',Σ',δ'}. The transition table δ' is given in table 5. The elements of S' are given in the first column and the first line contains elements of Σ'.

TABLE 5 A homomorphic image of the envisionment graph

<table>
<thead>
<tr>
<th>E_b</th>
<th>δ'</th>
<th>1 2 3 4 5 6 7 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDIR(r,y)</td>
<td>inc,inc</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>inc, std</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td>inc, dec</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>std, inc</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>std, dec</td>
<td>e</td>
</tr>
<tr>
<td></td>
<td>dec, inc</td>
<td>f</td>
</tr>
<tr>
<td></td>
<td>dec, std</td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>dec, dec</td>
<td>h</td>
</tr>
</tbody>
</table>

The envisionment graph of E_b embodies a coarser image of the behaviour of δ'. No_change behaviour is only possible when both r and y are not steady. This category includes the no_change nodes of E where the money market is not
clearing i.e. changes in \( y \) dominate changes in \( r \) in such a way that total money demand moves away from money supply. In this kind of behaviour the interest elasticity of money demand is low. Oscillating behaviour is characterized by oscillations in both \( y \) and \( r \). Also cycles in which \( y \) is only increasing or decreasing while the interest rate is oscillating are possible. In this kind of behaviour the interest elasticity of investment is low. This implies that changes in investment which depend on \( r \) are not large enough to induce a decrease in national income.

It is interesting to compare these kinds of behaviour with the standard economic behaviour put forward in Dennis (1981). The standard economic behaviour is the oscillating type, where initially \( y \) and \( r \) are not steady and eventually a new equilibrium state is reached. This behaviour is captured in one cycle of the automaton \( E_b \) namely, \( a \rightarrow c \rightarrow h \rightarrow f \rightarrow a \). Nonetheless, standard economic reasoning arrives at a unique behaviour instead of the three categories described above.

The reasons for discarding the other kinds of behaviour are as follows. Firstly, the interest elasticity of investment is assumed to be relatively high. This assumption rules out the behaviour where \( y \) is monotonically increasing or decreasing. The second reason is the market clearing view of the money market. This corresponds to a kind of stability assumption. It is assumed that the interest elasticity of money demand is high enough to ensure that the system is moving towards equilibrium eventually. Thirdly, steady values of variables are ignored unless it is an equilibrium situation. In that case it is possible to disregard states with \( Qdir(r) = \text{std} \) or \( Qdir(y) = \text{std} \). Thus the assumptions underlying the economic reasoning in Dennis (1981) rule out a large number of qualitative behaviours so that only stable oscillating cycles remain.
5. CONCLUSIONS

In this paper we presented a framework for qualitative reasoning in economics. It is shown that relevant economic conclusions can be drawn from a simple qualitative model in a formal way. Furthermore, some issues about the implementation of this framework in Prolog are described. Clearly, the analysis here is only a first step and the methods have to be refined considerably to deal with more realistic models.

Future research may enhance the applicability of qualitative reasoning methods in economics. Several lines should be followed. Firstly one could incorporate other (fixed) quantity spaces and apply techniques like order of magnitude reasoning. This will reduce the huge number of nodes in the envisionment graph of complex models, compared to simulations where only sign information is used. Another improvement can be found in the enrichment of the formal language. In this paper economic laws are transformed into simple constraints and heuristic economic knowledge (or more sophisticated constraints) cannot be represented. Finally, it would be interesting to develop a methodology to couple qualitative and quantitative methods. In this way a qualitative reasoner could also be used for explanation and validation of complex quantitative models.

REFERENCES


