Computational models of exchange economies

N. Botta

June 14, 2011

Market economies, exchange economies

- market economy
  ↑
  exchange economy

- market economy (planned, free, . . .)
  - commodities G have prices $\pi \in G \to \mathbb{R}_{>0}$
  - economic agents A have, consume, produce, trade commodities
  - G, A are finite sets

- example: free producer
  - has a "current" stock $q \in G \to \mathbb{R}_{\geq 0}$ of commodities
  - does
    - trade $o < q$ in exchange for i
    - consume i
    - produce p
  - is left with a stock $q - o + i - i + p$ of commodities
  - o and i depend on the producer's offer and demand and on its interactions with the environment ...
  - ...offer and demand depend on its current stock q, on its goals and on its offer and demand policies ...
  - p depends on i and on the producer's production efficiency ...

Theories of market economies: aims

- explain the balance between production and consumption (supply and demand) of commodities:

  > "From the time of Adam Smith's *Wealth of Nations* in 1776, one recurrent theme of economic analysis has been the remarkable degree of coherence among the vast number of individual and seemingly separate decisions about the buying and selling of commodities . . . Would-be buyers ordinarily count correctly on being able to carry out their intentions", Arrow (1974)

- explain how the prices of commodities are established:
  - as rates at which commodities are traded (dynamical prices, traders as price setters, Marshall (1890), Walras (1889))
  - as parameters in optimization problems for price takers (static prices, Pareto (1906), Debreau (1959) . . . Neo-Walrasian GE)

Exchange economies: basic assumptions

- no consumption or production, agents just exchange commodities
- the total amount of each commodity is conserved, all feasible allocations are re-allocations of an initial distribution of commodities $x_0 \in A \to (G \to \mathbb{R}_{\geq 0})$
- exchanges are the outcome of bilateral trades between agents

Computational models of exchange economies: aims

- explain if (by mean of which mechanisms) the economy as a whole converges towards an equilibrium of supply and demand in a decentralized setup ...
- explain how the prices of commodities are established ...
- ... in terms of simple, economically plausible bilateral trade mechanisms

Computational models of exchange economies: difficulties

- goals, motivations for developing CMoEE are often vague
- specifications
  - of what CMoEE are meant to model
  - of how CMoEE work

  are ambiguous, incomplete or just missing
- . . . it is therefore almost impossible to discuss / assess the correctness of CMoEE implementations

H. Gintis' model of decentralized bilateral exchange (2006)

- "An agent-based model [of decentralized bilateral exchange] is a computer simulation of the repeated play of a game in which a large number of agents are endowed with software encoded strategies governing both how they play the game and how they gather information and update their behavior"
- the idea is to explain price formation in terms of iterative imitation & mutation rules for agent-spacific prices
- the rules are driven by a trading fitness of the agent's prices
- the trading fitness is computed in a trading game

Specification framework for CMoEE

- inspired by Gintis (2006)
- Haskell-like notation, only sets, lists, functions and relations
- basic notions:
  - allocations, prices, utility profiles ...
  - Walrasian equilibrium, demand, excess demand ...
  - bilateral exchange, elementary bilateral exchange ...
  - bilateral trade, offer & demand policy, trade resolving policy ...
  - sectors, trading schedule, trading round, trading game ...
- aims:
  - assist the specification, implementation, testing of CMoEE
  - support the formulation of precise questions about a model's behavior

Example: model specification

- agent-specific prices imitation & mutation iteration:

$$\pi\, 0 = \pi_0$$
$$\pi\,(t+1) = \text{fold}\ cm\,(\pi\ t)\,(cms\ t)$$

$$\pi \in \mathbb{N} \rightarrow A \rightarrow (G \rightarrow \mathbb{R}_{>0})$$

$$cms \in \mathbb{N} \rightarrow List\ (A \times A) \times (G \rightarrow [\,0, 1\,])$$

- *cm* is the replicator dynamic rule (Taylor and Jonker, 1978):

$$z' = cm\, z\,((a_1, a_2), \xi) \;\Rightarrow$$
$$z'\, a \neq z\, a \;\Rightarrow\; a = a_1 \vee a = a_2$$
$$\wedge$$
$$f\, a_1 < f\, a_2 \;\Rightarrow$$
$$z'\, a_2 = z\, a_2 \;\wedge\; z'\, a_1\, g = \begin{cases} z\, a_2\, g & \text{if } \xi\, g < mp \\ (z\, a_2\, g)/mf & \text{if } \xi\, g \geq mp \wedge \xi\, g < 0.5 \\ (z\, a_2\, g) * mf & \text{if } \xi\, g \geq mp \wedge \xi\, g \geq 0.5 \end{cases}$$
$$\wedge$$
$$f\, a_1 \geq f\, a_2 \;\Rightarrow$$
$$\ldots$$

- mp, mf are fixed but the trading fitness $f \in A \to \mathbb{R}$ ...

- ...is recomputed at each iteration in a trading game
- the game is played at fixed prices $\pi\ t$ and consists of n rounds
- each round is defined by a random sequence of bilateral trades

$$f\ a = \frac{1}{n} * (f_1\ a + \ldots + f_n\ a)$$

$$(f_k, x_k) = \text{fold}\ tr\ (f_0, x_0)\ (ts\ k)$$

$$x_0, x_k \in A \to (G \to \mathbb{R}_{\geq 0})$$
$$ts\ k \in List\ A \times A$$

- the game is based on a number of assumptions:
  - agents are partitioned in $|G|$ equally populated sectors:

$$s \in A \rightarrow G$$
$$\forall\, g \in G\ |s^{-1}\, g| = |A| / |G|$$

  - trading takes place across sectors:

$$(a_1, a_2)\ \text{elem}\,(ts\,k) \ \Rightarrow\ s\,a_1 \neq s\,a_2$$

  - tr yields an elementary bilateral exchange of α units of $g_1 = s\,a_1$ and β units of $g_2 = s\,a_2$ ...

$$tr\,(f, x)\,(a_1, a_2) = (f', x')$$
$$\Rightarrow$$
$$x'\,a \neq x\,a \ \Rightarrow\ a = a_1 \vee a = a_2$$
$$-\,\alpha = (x' - x)\,a_1\,g_1 = -(x' - x)\,a_2\,g_1 \leq 0$$
$$+\,\beta = (x' - x)\,a_1\,g_2 = -(x' - x)\,a_2\,g_2 \geq 0$$

- …and an update of f given by an agent-specific utility function $u \in A \rightarrow ((G \rightarrow \mathbb{R}_{\geq 0}) \rightarrow \mathbb{R})$

$$
\begin{aligned}
& tr\,(f, x)\,(a_1, a_2) = (f', x') \\
& \;\Rightarrow \\
& f'\,a \neq f\,a \;\;\Rightarrow\;\; a = a_1 \vee a = a_2 \\
& f'\,a_1 = f\,a_1 + u\,a_1\,(x'\,a_1) \\
& f'\,a_2 = f\,a_2 + u\,a_2\,(x'\,a_2)
\end{aligned}
$$

α and β are obtained by applying an agent-independent trade resolving policy trp to the offers and demands of a_1, a_2. These are given by agent-specific offer & demand policies:

$$
\begin{aligned}
(\alpha, \beta) &= trp\,(\pi\,a_1)\,(\pi\,a_2)\,g_1\,g_2\,(o_1, d_1)\,(o_2, d_2) \\
(o_1, d_1) &= odp\,a_1\,(x\,a_1)\,(\pi\,a_1)\,g_1\,g_2 \\
(o_2, d_2) &= odp\,a_2\,(x\,a_2)\,(\pi\,a_2)\,g_2\,g_1
\end{aligned}
$$

- can one put forward policy specifications which are to hold independently of the economic interpretation of *trp*, *odp* ?
  - $(o, d) = odp\, a\, (x\, a)\, (\pi\, a)\, g\, g' \Rightarrow o \leq x\, a\, g$
  - $(\alpha, \beta) = trp\, (\pi\, a_1)\, (\pi\, a_2)\, g_1\, g_2\, (o_1, d_1)\, (o_2, d_2)$

    $\Rightarrow$

    $\alpha \leq o_1 \wedge \beta \leq o_2$
- can one put forward specifications for *trp*, *odp* on the basis of their economic interpretation, empirical data ?
  - $d_1 \leq o_2 \wedge d_2 \leq o_1$

    $\Rightarrow$

    $trp\, (\pi\, a_1)\, (\pi\, a_2)\, g_1\, g_2\, (o_1, d_1)\, (o_2, d_2) = (d_2, d_1)$
  - $d_1 * (\pi\, a_2\, g_2) > o_1 * (\pi\, a_2\, g_1)$

    $\Rightarrow$

    $trp\, (\pi\, a_1)\, (\pi\, a_2)\, g_1\, g_2\, (o_1, d_1)\, (o_2, d_2) = (0, 0)$

- the dynamic of prices at large times ($t > 20000$) critically depends on the trading policies *trp*, *odp*
  - are there economically sound policies which ensure the agents' total demand and supply tending towards some balance ?
  - are there economically sound policies which ensure the agents-specific prices to tend towards some equilibrium prices ?

Example: model implementation (c++)

- lack of explicit language support for
  - higher order functions
  - constrained genericity
  - non-integer value genericity

  suggest a simple, pragmatical DBC approach based on asserting pre- and postconditions at run time

- this approach can be made appealing to the practitioner by providing a small set of domain specific type aliases and validity queries

- Example: elementary bilateral trade operator()

```
/*-----------------------------------------------------------------------
operator()
------------------------------------------------------------------------*/
void operator()(       Allocation& x,
                 const Private_Prices& p,
                 const Allocation& xe,
                 const Agent& a1,
                 const Agent& a2,
                 const Good& g1,
                 const Good& g2) const {

  REQUIRE(is_valid_allocation(x));
  REQUIRE(is_valid_private_prices(p));
  REQUIRE(is_valid_allocation(xe));
  REQUIRE(is_valid_agent(a1));
  REQUIRE(is_valid_agent(a2));
  REQUIRE(is_valid_good(g1));
  REQUIRE(is_valid_good(g2));
  REQUIRE(is_odp_initialized());
  REQUIRE(is_trp_initialized());

  LET(const Allocation y(x));

  // ...

  ENSURE(is_elementary_bilateral_exchange(x, y, a1, a2, g1, g2));
}
```

Conclusions

- we can provide precise mathematical specifications of some computational models of exchange economies in terms of a few elementary notions: trading policies, elementary bilateral trade and exchange, . . .
- these notions can be applied to:
  - derive domain-specific type(def)s and DBC constructs for model implementation in mainstream programming languages
  - design "crucial" experiments for testing programs
  - study models, reason about numerical results
- however, we are far from having precise specifications of problems, aims and conjectures in exchange economy modeling and . . .
- . . . of the relationships between prices and allocations of specific CMoEEs and prices and allocations in market equilibrium theories