Linguaggi di Programmazione (A.A. 2018-19)

Implementazione in Java di un Social Game System

Specifiche del progetto

Implementare in Java un *Social Game System* (SGS), ovvero un'implementazione della teoria dei giochi evolutivi basato sullo scambio di messaggi fra giocatori. La descrizione di un SFS è riportata di seguito. Sono consentite semplificazioni del modello o "variazioni sul tema" (auspicabilmente ben motivate). Il sistema prodotto dovrebbe consentire quantomeno l'implementazione del *gioco della vita* di Conway. È possibile, ma non necessario, corredare il sistema di un'interfaccia grafica che consenta di visualizzare l'evoluzione del gioco mediante un'animazione.

Si richiede la consegna di un programma *funzionante*, scritto in Java, accompagnato da uno o più casi studio, anche semplici, che ne mettano in luce gli aspetti più significativi. Scelte progettuali, implementative e caso studio andranno discusse in una breve relazione, alla quale andrà allegato il sorgente Java, estesamente corredato di commenti.

Buon lavoro!

1 A Social Game System

We shall present a game model called *Social Game System* (SGS). Players belong to a set $\mathcal{U}$, called *universe*, which we assume numerable and ranged over by the metavariables d, $e \ldots$. At any time only a finite subset of players are active in the game, each connected with a subset of the currently active players, called the player's *acquaintances*. Acquaintance is a reflexive, symmetrical relation on players, not necessarily transitive. A player d's acquaintances form an array of length k, called d's *neighborhood*, the indices of which are called *positions*. A player may hold more than one position in another player's neighborhood. A position may be held by at most one player, but it may also be *empty*, in which case we write $\perp$ its value. The first position of d's neighborhood is, by convention, always held by d itself. The length k, which is the same for all players, is therefore an upper bound to the number of acquaintances a player may have at any time. In the following we shall take $\mathcal{U}$ and k as given.

The above description can be formalised as follows: We write $\mathcal{N}$ the set of natural numbers. We write $dom\ (f) \subseteq A$ the *domain* of a partial function $f : A \rightharpoonup B$. Given $k \in \mathcal{N}$ and a set A, we write $A^k_\perp$ the set $(A + \{\perp\})^k$ of *partial* k-tuples of elements of A. Clearly, $A \subseteq B$ implies $A^k_\perp \subseteq B^k_\perp$. We say that $\vec{a} \in A^k_\perp$ and $\vec{b} \in B^k_\perp$ have *equal structure* when $a_i = \perp$ if and only if $b_i = \perp$, for all $1 \leq i \leq k$.

A *relational board* (of dimension k on $\mathcal{U}$) is a partial function $\beta : \mathcal{U} \rightharpoonup (\mathcal{U} \times \mathcal{N})^k_\perp$ such that, if $\beta(d) = \vec{v}$ is defined and $v_i = (e, j)$, then $\beta(e) = \vec{u}$ and $u_j = (d, i)$. We also postulate that $v_1 = (d, 1)$. A board represents the net of acquaintances of the active players at a certain time of the game, the domain of β being the set of currently active playes. When $\beta(d) =< v_1, \ldots v_k >$ is defined and $0 \leq i \leq k$, then either $v_i = \perp$, that is d's i-th position is empty, or $v_i = (e, j)$, that is the position is held by e, while d, by symmetry an acquaintance, holds e's j-th position. The condition $v_1 = (d, 1)$ makes acquaintance reflexive.

A *profile* of a board β consists of a partial function $\sigma : \mathcal{U} \rightharpoonup \mathcal{N}^k_\perp$ with same domain $D \subseteq \mathcal{U}$ as β and such that $\sigma(d)$ and $\beta(d)$ have equal structure, for all $d \in D$. A profile represents a *state* of the board at some time of the game. The numbers in the k-tuple $\sigma(d)$, when defined, are called *messages*, the latest messages received by d from its acquaintances. In particular, $\sigma(d)_1$, the message last received from d itself, is called d's *wealth* and may be used to represents d's cumulative payoff at that time.

2 SGS dinamics

Board profiles change as result of players exchanging messages. A player receives messages from its acquaintances and reacts by sending messages back to them according to a feedback function. Every $d \in \mathit{dom}\ (\beta)$ is endowed with it's own structure preserving *feedback function* $\phi_d : \mathcal{N}^k_\perp \rightarrow \mathcal{N}^k_\perp$. Preserving structure means that if $\phi_d(\vec{v})$ has same structure as $\vec{v}$. Given feedback functions ϕ_d, one for each $d \in \mathit{dom}\ (\beta)$, a state *transition* of β is a pair (σ, σ') of profiles of the board such that, if $\beta(d)_i = (e, j)$ then $\sigma'(e)_j = \phi_d(\sigma(d))_i$. Transitions are written $\sigma \rightarrow \sigma'$ and read "from σ to σ'".

TO BE CONTINUED...