Applying RL to Take Pedagogical Decisions in Intelligent Tutoring Systems

Ana Iglesias Maqueda
Computer Science Department
Carlos III of Madrid University

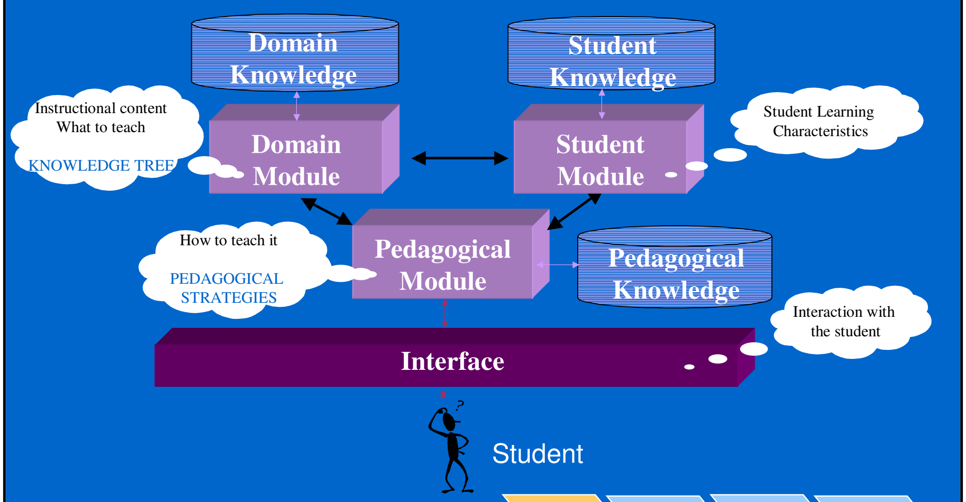
Content

- Intelligent Tutoring Systems (ITSs)
  - Definition
  - Problems
- Aims
- Reinforcement Learning (RL)
- Proposal
  - RL Application in ITSs
  - Working Example
- Conclusions and Further Reseach

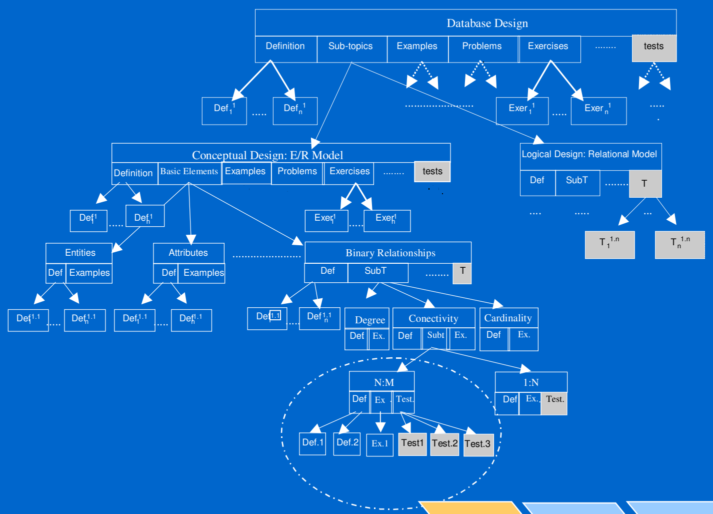
Intelligent Tutoring Systems (ITSs)

- Intelligent Tutoring Systems (ITSs):
  "computer-aided instructional systems with models of instructional content that specify *what* to teach, and teaching strategies that specify *how* to teach" [Wenger, 1987].

ITS Modules (Burns and Capps, 1988)

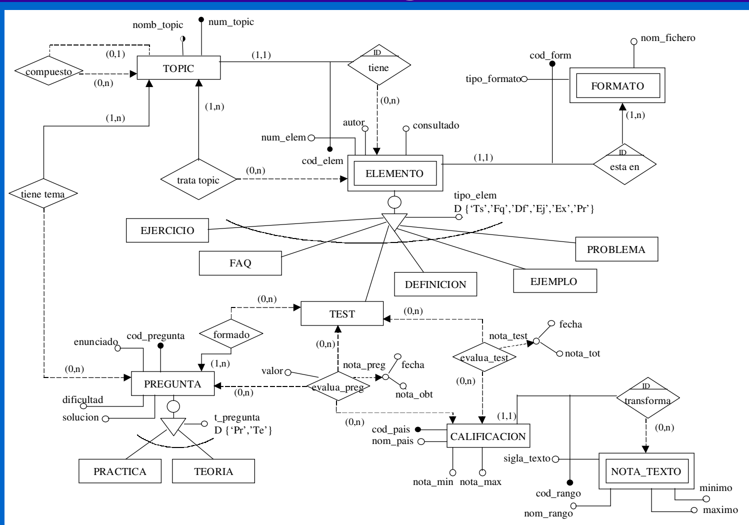


ITS. Knowledge Tree



ITSs Aim RL RL in ITS

ITS. Knowledge Tree. E/R



ITS. Pedagogical Strategies (PS)

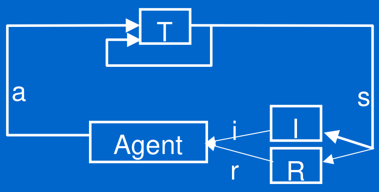
- Specify [Murray, 1999] :
  - how the content is sequenced
  - what kind of feedback to provide,
  - when & how to show information (when to summarise, explain, give an exercise, definition, example, etc.)
- Problems [Beck, 1998]:
  - To encode them
    - A lot of them
    - to incorporate all the experts knowledge
  - ¿ How many strategies are necessary ?
    - Differences among them
    - The moment to apply them
    - ¿ Why they fail ? ¿ how to solve it ?

Aims

- To eliminate the pre-defined PS
- Tutor learn to teach effectively
  - Representing the pedagogical information based on a RL model
  - what, when and how to show the content
    - Adapting to students needs in each moment
    - Based only in adquired experience at the interaction with others students with similar learning characteristics

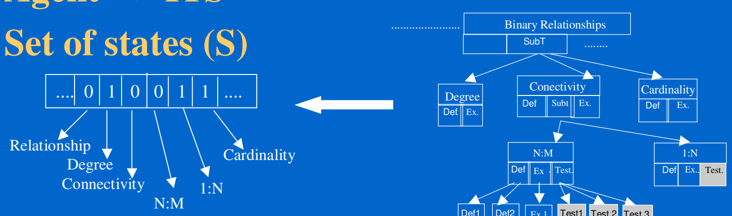
Reinforcement Learning (RL)

- Definition [Kaelbling et al., 1996] :
  - An **agent** is in a determinated **state (s)**
  - The agent execute an **action(a)**
  - The execution produce a **state transaction (T)** to an other state (s')
  - The agent perceive the current state by the **perception module (I)**
  - The environment provide a **reinforcement signal (r)** to the agent
  - The **agent aim** is to maximice the long-run reward



Proposal. RL Components (1/3)

- **Agent --> ITS**
- **Set of states (S)**



- **Set of actions (A): To show items**

A1 = to show Def.1 = {def1}
A2 = {def2}
A3 = {ex1}
A4 = {def1 + ex1}
....

Proposal. RL Components (2/3)

- **Perception of the environment (I: S → S):**
  - How the ITS perceives the knowledge student state.
    - Evaluating his/her knowledge by tests.

- **Reinforcement (R: SxA → R):**
  - Reinforcement signals provided by the environment
    - maximun value upon arriving to the ITS goals.

ITSs | Aim | RL | RL in ITS

Proposal. RL Components (3/3)

- **Value-action function (Q: SxAxΠ → R):**
  - Estimates de usefulness of executing an action when the agent is in a determinated state.
    - ITS aim: to find the maximum value of Q function.

    - Algorithm: Q-learning (determinist) [Watkins, 1989]:

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a')$$

where γ is the discount parameter in future actions

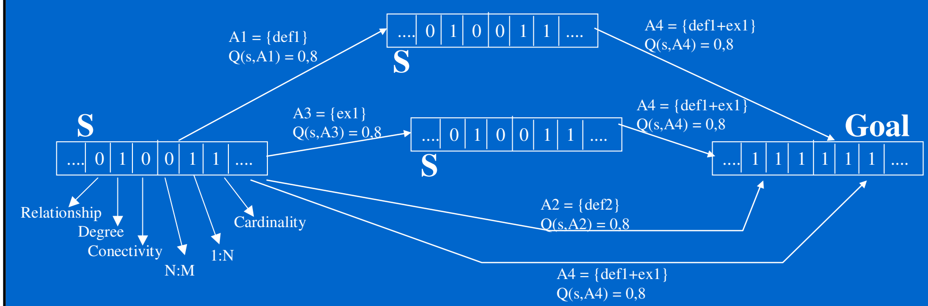
ITSs | Aim | RL | RL in ITS

Proposal. Q-learning

Q-learning Algorithm	Q-learning adapted to ITS domain
▪ For each pair (s∈ S, a∈ A), initialise the table entry Q(s,a).	▪ For each pair (s∈ S, a∈ A), initialise the table entry Q(s,a).
➢ Observe the current state, s	➢ Test the current student knowledge, obtaining *s*
➢ Do forever	➢ Do forever
- Select an action, a, and execute it	- Select a set of knowledge tree leaves, *a*, to show to the student.
- Receive the immediate reward, r	- Receive the immediate reward, r. A positive reward is received when the ITS goal is achieved. A null reward is obtained in any other case.
- Observe the new state, s'	- Test again the current student knowledge, *s'*
- Update the table entry for Q(s,a) as follows: $Q(s,a)=(1-\alpha)\,Q(s,a)+\alpha\{r+\gamma\, max_{a'}\, Q(s',a')\}$	- Update the table entry for Q(s,a), that estimates the usefulness of executing the *a* action when the student is in a particular knowledge state: $Q(s,a)=(1-\alpha)\,Q(s,a)+\alpha\{r+\gamma\, max_{a'}\, Q(s',a')\}$
- Set s to s'	- Let us *s* the current student knowledge state, *s'*.

Table 1. Q-learning algorithm

Proposal. Example (1/2)



Q(s,a)	A1	A2	A3	A4
S	0,8	0,8	0,8	0,8
Goal	0.0	0.0	0.0	0.0

ITSs | Aim | RL | RL in ITS

Proposal. Example (2/2)

- Let us suppose
  - $r = \begin{cases} 1 \text{ if } s' = \text{goal} \\ 0 \text{ if } s' \neq \text{goal.} \end{cases}$
  - $\gamma = 0{,}9$
  - $Q(s,a) = \gamma^{(size(a)-1)} r + \gamma^{size(a)} \max_{a'} Q(s',a')$ (1)
- Example
  - Student 1:
    - A*1* action is randomly chosen:
      $Q(S, A1) = 0 + 0.9^1 * \max_{a'}\{0.8, 0.8, 0.8, 0.8\} = 0.72$ (2)
    - A*4* is executed next:
      $Q(S, A4) = 0.9^{2-1} * 1 + 0.9^2 * \max_{a'}\{0,0,0,0\} = 0{,}9$ (3)
  - Student 2:
    - A*2* is randomly chosen:
      $Q(S, A2) = 0.9^{1-1} * 1 + 0.9^1 * \max_{a'}\{0,0,0,0\} = 1$ (4)

ITSs | Aim | RL | RL in ITS

Conclusions

- To eliminate the pre-defined PS
- System adapts to student
  - in real time: by trial and error,
  - based only on <u>previous</u> information of <u>interactions</u> with other students with similar characteristics
- General technique
  - domain independent

Further Research

- Experiments
  - Implement the theorical model
  - Test the ITS with real students
  - Validate the model
- Others
  - Classify students
  - Use hierarchical RL algorithms
  - Use planning