Language Technology II: Dialogue Learning I

Summer 2012

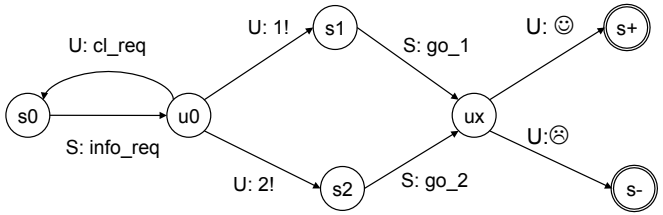
Manfred Pinkal

Dialogue Design and Dialogue Policy

- The general task of designing an dialogue model can be subdivided into two seperate sub-tasks:
- Specification of a fixed framework for the dialogue structure through global design decisions:
  - Specification of a set of dialogue states
  - Specification of the range of available actions of the dialogue system
  - over-all structure of the dialogue (e.g., information collection phase, database look-up, answer generation).
- Specification of a dialogue policy: a decision procedure which decides at a certain point of the global model which concrete action out of a set of alternatives should be taken. Examples:
  - Grounding: Explicit grounding act/ implicit grounding act/ no grounding
  - Selection of presentation mode and modality for alternative user options.

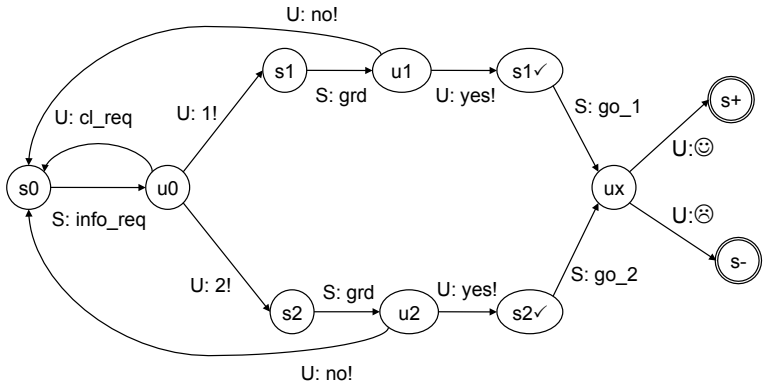
Elevator: FSA with no grounding

S: Where do you want to go?
U: Second floor, please.
S: <goes to 2nd floor>
U: Thank you, that's great!



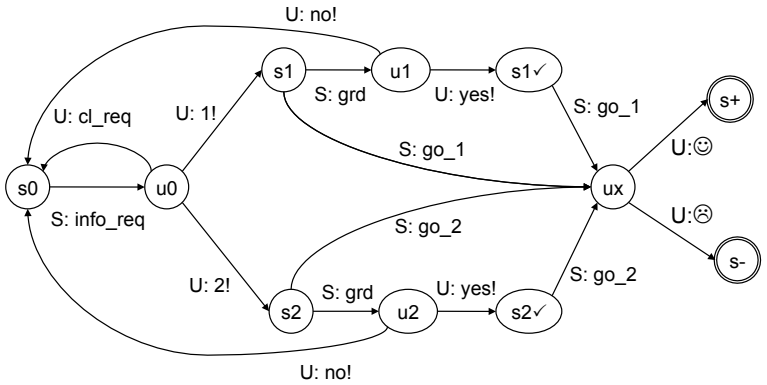
No grounding: Hardwired „optimistic“ policy

Elevator: FSA with explicit Grounding



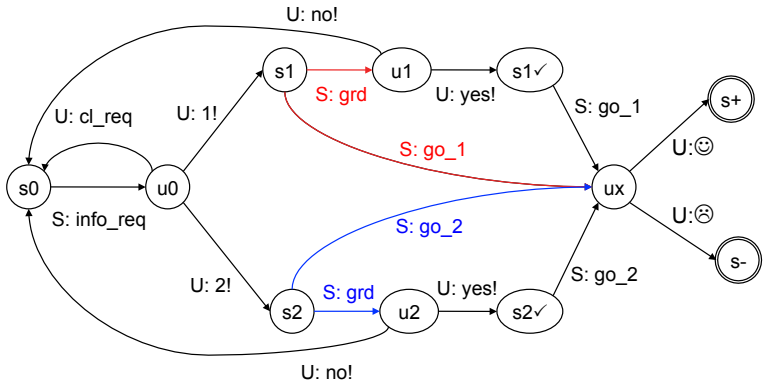
Explicit grounding: Hardwired „cautious“ policy

A combined FSA model



Combination of the two models

A combined FSA model



Combination of the two models: Non-deterministic with respect to grounding strategy!

Dialogue Policy

- Global design decisions include the specification of
  - a set of dialogue states S, the State Space (a set of nodes in a FSA, or a set of structured information states)
  - a set A of possible system actions, the Actions Set (transitions, ISU operations)
  - a range of admissible possible actions A(s) for each state s
- A dialogue policy is a decision procedure that selects specific actions a∈A for possible dialogue states s∈S, more technically: a function
  - π: S → A
- The objective of dialogue design:
  - Find a policy π which is optimal with respect to the purposes of the dialogue.

Form Filling/ ISU Dialog Model

S1: What kind of music do you want to hear?
U1: Pop music.
S2: Which artist do you want to hear?
U2: Madonna.

State s_0:	genre: empty artist: empty genre_conf: false artist_conf: false	Action a_1:	request_info genre: ?
		Update u_1:	genre := pop
State s_1:	genre: pop artist: empty genre_conf: false artist_conf: false	Action a_2:	request_info artist: ?
		Update u_2:	artist := madonna
State s_2:	genre: pop artist: madonna genre_conf: true artist_conf: false		

Form Filling/ ISU Dialog Model

S1: What kind of music do you want to hear?
U1: Pop music.
S2: By which artist do you want to hear pop music?
U2: Madonna.

State s_0:
genre: empty
artist: empty
genre_conf: false
artist_conf: false

Action a_1: request_info genre: ?

Update u_1: genre := pop

State s_1:
genre: pop
artist: empty
genre_conf: false
artist_conf: false

Action a_2: Implicit_conf genre: pop
request_info artist: ?

Update u_2: artist := madonna
genre_conf := true

State s_2:
genre: pop
artist: madonna
genre_conf: true
artist_conf: false

Determining Dialogue Policies

How do we find the optimal dialogue policy?

- Set alternative parameters by hand; examples:
  Confidence threshold for grounding, dependent on the importance of the decision
  Maximum number of items for which graphical display is appropriate (dependent on actual user situation)
- Run full implemented system or WoZ experiment with human users, evaluate, modify or refine.

Comment:

- This is at present the typical solution in real-world applications.
- Adaptive dialogue behaviour requires to take a large set of feature combinations into account. Costs are high, results not always optimal.
- Is machine learning an alternative?

Supervised Dialogue Learning

Supervised learning of dialogue policies:

- Collect data from WoZ experiments, which are set up in a novel way:
- Work with several wizards. Don't impose a specific strategy on the wizards' behavior, but just give them a general instruction ("Help the user reach its goal!")
- Derive a n-gram based statistical model from the data that proposes the most probable system move as the appropriate system reaction.

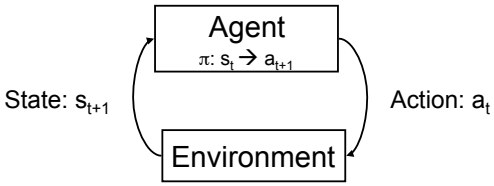
Difficulties:

- Learnt dialogue behaviour is locally consistent, global optimisation is not supported (due to n-gram constraint).
- System reproduces the average wizards' behavior; it cannot assess the value of the decisions, or include novel, unrealized decisions.
- Data sparseness.

Reinforcement Learning

- System learns optimal dialogue policy by executing dialogues and getting feedback on its performance: Reinforcement Learning.
- Reinforcement learning builds on the concept of Markov Decision Process, a framework for modeling decision-making by an agent in an environment whose behavior is (partly) random.
- The agent selects an action based on the current state (plus reward information).
- The environment emits information about the state it has adopted (and assigns a reward for the agent's last action).

Decision Process



- Agent interacts with a stochastic environment:
  - At time t, agent picks an action a_t (on the basis of its current state s_t and strategy π)
  - Execution of a_t (non-deterministically) influences the subsequent behaviour of the environment, which assumes state s_{t+1}.
  - According to strategy π, agent (deterministically) selects and executes an action: $\pi(s_{t+1}) = a_{t+1}$

Dialog Manager: Form Filling/ ISU

S1: What kind of music do you want to hear?
U1: Pop music.
S2: By which artist do you want to hear pop music?
U2: Madonna.

State s_0:
genre: empty
artist: empty
genre_conf: false
artist_conf: false

Action a_1: request_info genre: ?

Update u_1: genre := pop

State s_1:
genre: pop
artist: empty
genre_conf: false
artist_conf: false

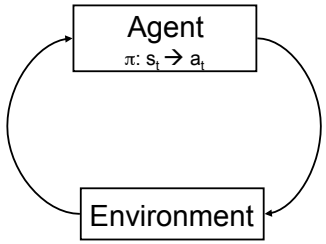
Action a_2: Implicit_conf genre: pop
request_info artist: ?

Update u_2: artist := madonna
genre_conf := true

State s_2:
genre: pop
artist: madonna
genre_conf: true
artist_conf: false

Decision Process: Example

State s_t:
genre: pop
artist: empty
genre_conf: false
artist_conf: false

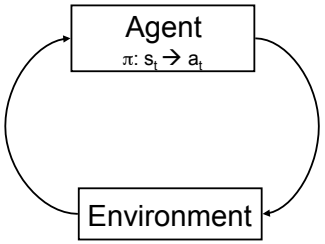


Action a_t:
Implicit_conf genre: pop
request_info artist: ?

S1: What kind of music do you want to hear?
U1: Pop music.
S2: By which artist do you want to hear pop music?

Decision Process: Example

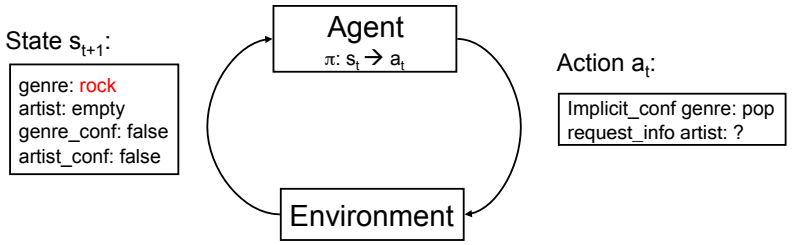
State s_{t+1}:
genre: pop
artist: madonna
genre_conf: true
artist_conf: false



Action a_t:
Implicit_conf genre: pop
request_info artist: ?

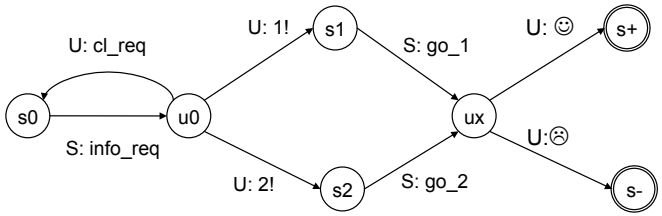
S1: What kind of music do you want to hear?
U1: Pop music.
S2: By which artist do you want to hear pop music?
U2: Madonna.

Environment is non-deterministic!

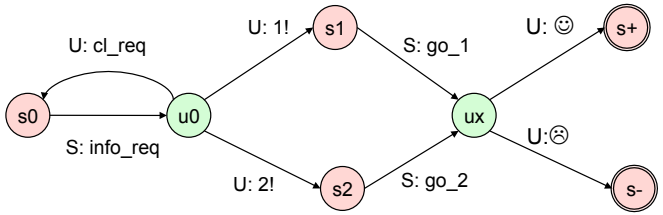


S1: What kind of music do you want to hear?
U1: Pop music.
S2: By which artist do you want to hear pop music?
U2: Not pop, rock!

A FSA Model with no grounding

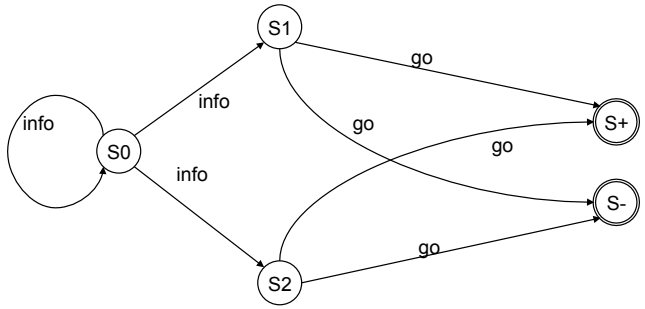


Decision Process and FSA

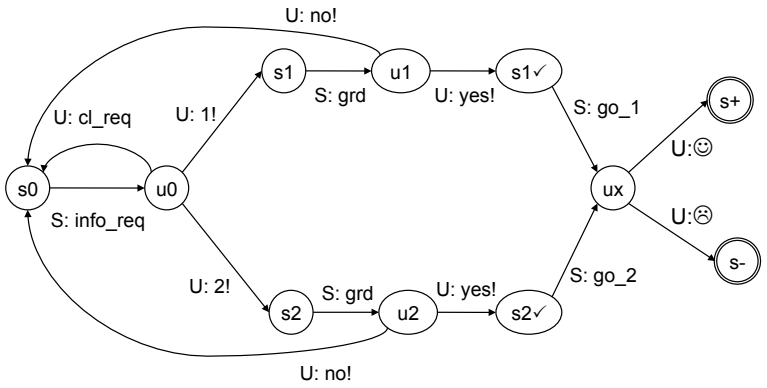


Decision Process and FSA

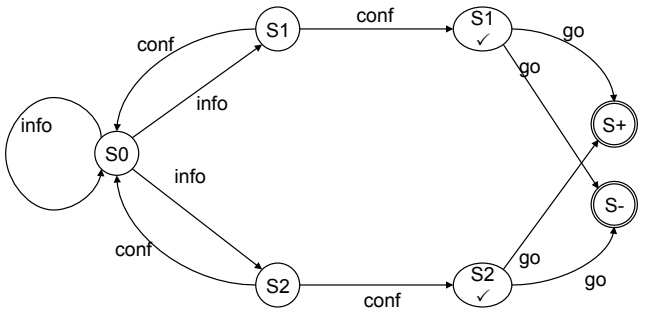
Outgoing edge labels from s indicate action of agent π(s).
Nodes represent states of environment (observed by the agent).
Choice of action π(s) is deterministic (in this automaton!)
Transition to result state is non-deterministic.



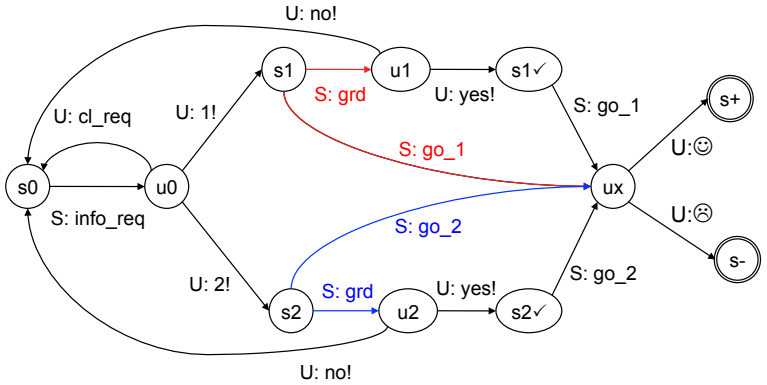
FSA Model with explicit grounding



FSA Model with explicit grounding

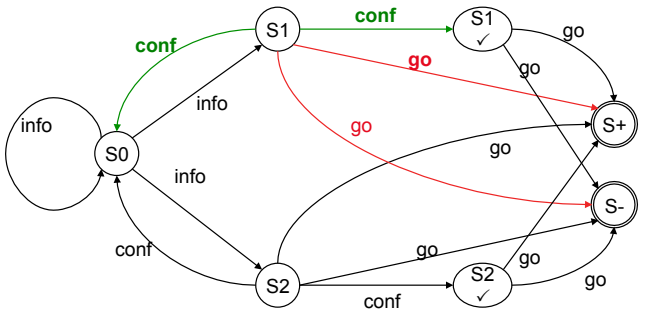


Combination of Models



Combination of Models

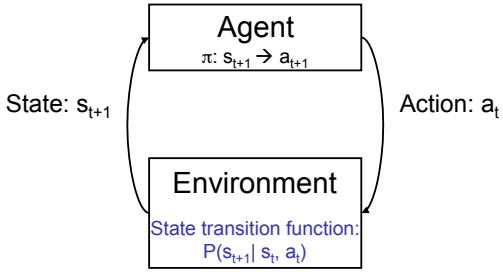
Combination of Models (1) and (2):
π(S1) is not uniquely defined.
Flexible/ underspecified grounding policy.



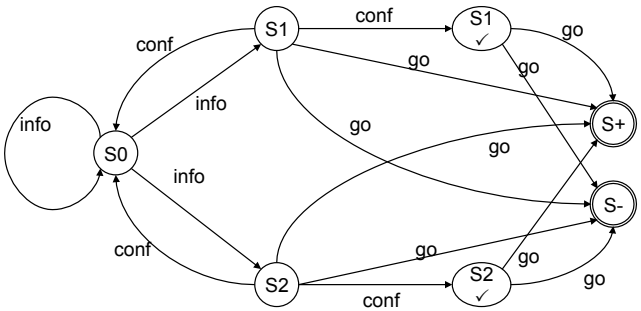
Completing the Picture of an MDP

- Dialogue policy is not part of the combined model, but must be separately determined.
- We have to assess the usefulness or utility of the alternative dialogue moves (Confirmation Request or Direct Execution).
- To this purpose, we first need a probabilistic model for the behaviour of the environment: the state transition function.

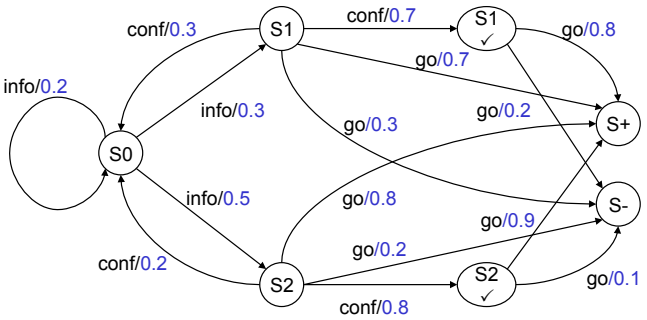
Completing the Picture of an MDP



The basic elevator model



Adding Transition Probabilities

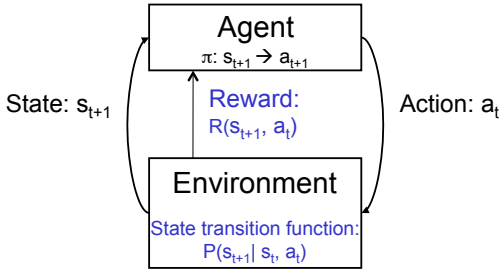


State transition function reflects the experience of the agent with its environment: Highly expected/ probable/ possible/ improbable reactions

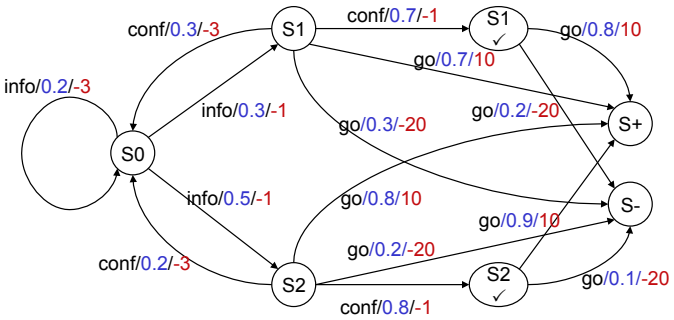
Completing the Picture of an MDP

- Dialogue policy is not part of the combined model, but must be separately determined.
- We have to assess the usefulness or utility of the alternative dialogue moves (Confirmation Request or Direct Execution).
- To this purpose, we first need a probabilistic model for the behaviour of the environment: the state transition function.
- Second, a reward function, giving feedback about the advantage of an action and its resulting state (usually in terms of a real number).

Completing the Picture of an MDP



Adding Reward



Motivation:		
	Each move makes dialogue longer:	Reward -1
	Recognition failures are annoying:	Reward -3
	Final task success is good:	Reward +10
	Final task failure is very bad:	Reward -20

Reward

- Immediate reward *R(s, a)* is locally assigned to state-action pairs.
- To make an informed choice between two actions, local comparison of immediate rewards is insufficient.
- The utility of an action strongly depends on its long-term effects: E.g., the utility of a dialogue contribution depends on the question whether it contributes to an over-all successful execution of the dialogue.
- We have to assess the expected value of an action, by estimating its expected cumulative reward *Q(s, a),* which combines immediate effect and long-term effects of action a, executed in state s.

Expected cumulative reward and optimal policy

- We do not know the outcome of the full interaction at t because the environment has non-deterministic behaviour: We do not know the result state of our action.
- To estimate the cumulative reward $Q(s, a)$ of an action s given a state a, we take the alternative future developments of the interaction into account, and weight them by their probability.
- The optimal policy in a given state s is the one selecting the action which maximises the expected cumulative reward:

$$\pi^*(s) = \arg\max_{a' \in A} Q(s,a')$$

- How do we compute $Q(s, a)$?

Computing Cumulative Reward

- The expected cumulative reward of taking action *a* in state s is defined by the "Bellman equation" (where γ is a discounting factor, $0 \leq \gamma \leq 1$).

$$Q(s,a) = \sum_{s' \in S} P(s' \mid s,a) * [R(s',a) + \gamma * \max_{a' \in A} Q(s',a')]$$

- Questions:
- How can we estimate the state transition function *P*?
- How do we get at the reward *R* ?
- How do we practically determine Q, in order to identify the optimal policy $\pi^*(s)$?

Estimating *P*

- Produce a dialogue corpus using WoZ experiments.
- Learn n-gram probabilities of state transitions (user dialogue moves) from the corpus.

Instantiating *R*

Options:

- Determine immediate reward by hand, using intuition and experience.
  - Arbitrary at least to some degree.
- Assess reward for the dialogue as a whole via (SASSI style) user questionnaires. Only final state-action pairs get non-zero reward.
  - Sparse data problem: Human users assess quality of the full dialogue sequence they have gone through; no assessment of sequences which have not occurred in the data.
- Approximate user assessment through measurable features.
  - PARADISE