Foundations of Language Interaction

HANDOUT ONE

May 24, 2001

M. Stone

mdstone@cs.rutgers.edu

Over the next few weeks we'll be developing a FORMALIZATION of LANGUAGE USE as RATIONAL JOINT ACTIVITY.

(1) Computer implementations require a mathematical perspective, and for this we'll be adapting ideas from computational logic [Miller, 1998].

(2) Our first objective is to account for task-oriented dialogue, in which participants in a conversation communicate in the service of developing or carrying out a plan for a specific real-world goal [Allen et al., 1995, Ferguson and Allen, 1998].

(3) Action is joint when carried out by an ensemble of people acting in coordination with each other [Clark, 1996].

(4) For our purposes, action is rational when it arises through our ordinary general processes of deliberation and choice [Bratman, 1987, Pollack, 1992].

Concrete motivation: COOPERATIVE QUESTION ANSWERING—reconstruct the questioner's intentions, and respond to them.

(5) a Have we lost the daemon?
  b WHICH ONE??? The line printer daemon (lpr doesn't work)? The cron daemon (your at files don't get atrun)? The mail daemon (biff doesn't inform you of new mail)?

(6) a How do you change mode for a dotted file (such as .login)?
  b The problem might be that you don't own your .login file. To save space, most of the .login files are linked together (i.e., they're all one file). To fix this, type the following:

```
cp .login .login.copy
rm .login (or rm -f .login, if that doesn't work)
mv .login.copy .login
```

Then you can chmod it, edit it, or whatever.

(7) a Whenever I attempt to use the script command I get a permission denied response. How do I correct this?
  b Script is broken. When the problem has been corrected, script will be made executable.

(Human-human data drawn from the Berkeley UNIX domain of [Wilensky et al., 1988].)

(8) How can we represent intentions?

An AGENT is computational system that acts in the real world.

(9) a A computational system moves through a series of discrete states.
  b The transitions between these states are determined algorithmically.
  c System's states are meaningful, in that they show a causal correspondence with an external reality.
  d The system's transitions between states respect the states' meanings.

REAL-WORLD BEHAVIOR involves a PERCEPTION-ACTION CYCLE.

(10) a Agent's behavior consists of a sequence of steps or cylces of decision-making.
  b Each step begins with the agent getting new information about the state of the environment: PERCEPTION
  c The agent then uses all the information at its disposal to decide what to do next
  d Finally, the agent takes the ACTION it has selected; and the next cycle begins.

Our strategy will be to specify agents with LOGICAL RULES.

(11) a State of the agent is specified as a DATA STRUCTURE, which in our case will be an expression of a formal language.
  b Transitions are specified by DECLARATIVELY MATCHING state data structures, and constructing new state data structures (using matched patterns).

Specify data structures for an agent's state (in λProlog):

(12)
```
kind percept, action, state type.
```

Introduce the percepts and actions you need (for today, a video game world).

(13)
```
type monster percept.
type jewel percept.
type nothing percept.
```

(14)
```
type shoot action.
type pickup action.
type move action.
```

Encode state-sequences as a recursive structure

(15)
```
type start state.
type see percept -> state -> state.
type do action -> state -> state.
```

Here are some expressions representing states in the formal language we've just defined.

(16)
```
a  start.
b  (see nothing start).
c  (do move (see nothing start).
d  (see nothing (do move (see nothing start))).
e  (do move (see nothing (do move (see nothing start)))).
f  (do pickup (see jewel (do shoot (see monster start)))).
```

Here's how we might formalize transitions in our perception-action loop.

(17)
```
a  type choose state -> action -> o.
b  choose (see monster S) shoot.
   choose (see jewel S) pickup.
   choose (see nothing S) move.
```

Operationalize things with a SIMULATOR

(18)
```
a  type perceive state -> percept -> o.
   type execute state -> action -> o.
   type agitate state -> o.
b  agitate State :-
       perceive State Percept,
       choose (see Percept State) Action,
       !,
       execute (see Percept State) Action,
       agitate (do Action (see Percept State)).
```

References

[Allen et al., 1995] Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Miller, B. W., Poesio, M., and Traum, D. R. (1995). The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48.

[Bratman, 1987] Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. Harvard University Press, Cambrdige, MA.

[Clark, 1996] Clark, H. H. (1996). *Using Language*. Cambridge University Press, Cambridge, UK.

[Ferguson and Allen, 1998] Ferguson, G. and Allen, J. F. (1998). TRIPS: An intelligent integrated problem-solving assistant. In *AAAI*.

[Miller, 1998] Miller, D. (1998). λprolog: An introduction to the language and its logic. Manuscript, Penn State.

[Pollack, 1992] Pollack, M. E. (1992). The uses of plans. *Artificial Intelligence*, 57:43–68.

[Wilensky et al., 1988] Wilensky, R., Chin, D., Luria, M., Martin, J., and Wu, D. (1988). The Berkeley UNIX consultant project. *Computational Linguistics*, 14(4):35–84.