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)).
```

Foundations of Language Interaction

HANDOUT TWO

May 31, 2001

M. Stone
mdstone@cs.rutgers.edu

1 Introduction

Today we consider the topic of ACTIONS, PLANS and INTENTIONS, developing some key ideas.

(19) Actions are the basic choices that agents make in each step of the cycle of perception and choice.

(20) To start, we'll treat a plan is an argument that demonstrates how performing a sequence of actions in the current circumstances leads to desired effects.

(21) An intention is a plan that the agent is committed to.

An intention guides the agent's deliberation by proposing actions that the agent might choose, focusing the agent's attention to the circumstances in the world that make the action appropriate, allowing the agent to more quickly identify problems and opportunities that arise in carrying out the plan, and suggesting ways that the agent can respond to the unexpected.
Recap and extension: an agent simulator with intentions.

(22)
```
agitate State Intentions :-
    perceive State Beliefs,
    update State Beliefs Intentions NewIntentions,
    !,
    act State NewIntentions.
```

(`act` has to call `agitate` recursively, of course.)
Today's objective is to see how to flesh out this program, by looking at some simple but classic representations of ACTIONS, PLANS and CONDITIONS, and by exploring some fundamental ideas about SEARCH.

2 Actions

The central challenge of describing many kinds of real-world action is finding a good approach to INERTIA.

(23) A state in the world tends to persist "by inertia", unless an action occurs whose effects explicitly disrupt that state.

The STRIPS representation of [Fikes and Nilsson, 1971] is a classic approach, which describes actions in terms of PRECONDITIONS, ADDITIONS and DELETIONS.

(24) a The PRECONDITIONS of an action is a list of facts that must be true in order for the action to be executed (that is, executed at all, executed safely or executed successfully).

b The ADDITIONS of an action is a list of facts that specify the new states in the world that the action institutes—the positive effects of an action.
c The DELETIONS of an action is a list of facts that specify the states in the world that are in progress when the action begins but that the action disrupts.

This description describes an algorithm for simulating the effects of action that builds-in inertia in a convenient way.

(25) a Suppose facts *F* are true and you perform action *E*, with strips description *P*, *A*, *D*.
b If every fact in *P* occurs in *F*, then the state of the world after doing *E* consists of *F* with the facts in *D* removed and the facts in *A* added.

To specify STRIPS in λProlog, we have this background:

(26) a
```
kind fact, action type.
```
b
```
type is_action
action -> list fact -> list fact -> list fact -> o.
```

A door domain.

(27) a
```
type closed, open, empty, keyed, locked, unlocked fact.
```
b
```
type key, turn, pull, unkey action.
```
c
```
is_action turn (keyed::locked::nil) (unlocked::nil)
(locked::nil).
```
d
```
is_action pull (closed::unlocked::nil) (open::nil)
(closed::nil).
```

...and so forth.

3 Plans and intentions

Plans are our first interesting data structure.

(28) a
```
kind plan type.
```
b
```
type finish list fact -> plan.
```
c
```
type step list fact -> action -> plan -> plan.
```

Recall the intuition: a plan is an argument showing that a sequence of actions performed in the current circumstances will lead to a desired effect. Our data structure maps out this argument recursively.

(29) a The simplest plan is a plan for no actions. In this case the desired effect must be true now, and it suffices to spell out what the desired features of the current circumstances are. If this GOAL is `G`, the corresponding plan structure is `(finish G)`.
b A more complicated plan must spell out the first action to be performed, `E`. If we guarantee certain facts `C` to be true when `E` is performed, we will be able to argue that a new set of circumstances obtains after `E` has been executed. Call this `C2`. After we do `E`, then, we will have to continue with a new plan that argues, using only facts in `C2`, that a subsequent sequence of actions will lead to a desired state. If this subplan is `P`, the overall plan structure we need now is `(step C E P)`.

A medium-sized plan:

(30)
```
(step (closed::keyed::locked::nil)
 turn
 (step (closed::unlocked::nil)
  pull
  (finish (open::nil))))
```

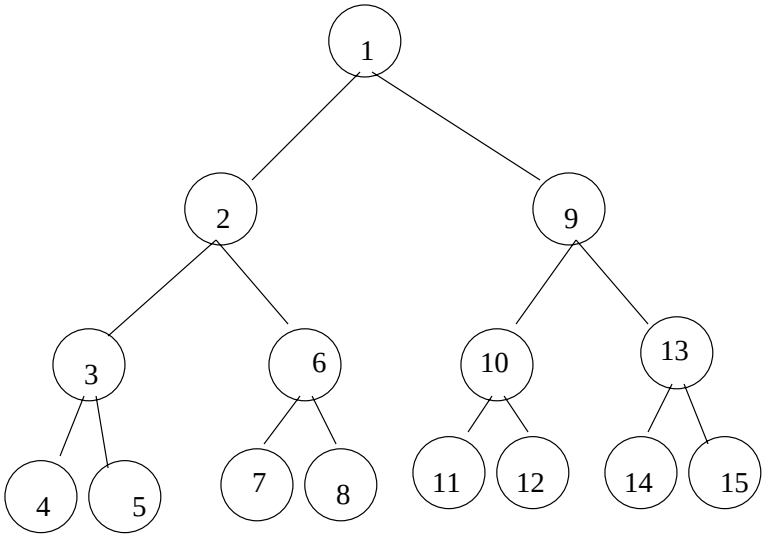
The core of a plan-building routine.

(31) a
```
type bp1 list fact -> plan -> plan -> o.
```
b
```
bp1 F P P :- circ P C, sublist C F.
```
c
```
bp1 F P R :-
    is_action A Pre Post Del,
    circ P C,
    remove Post C PC,
    disjoint PC Del,
    union Pre PC AC,
    bp1 F (step AC A P) R.
```

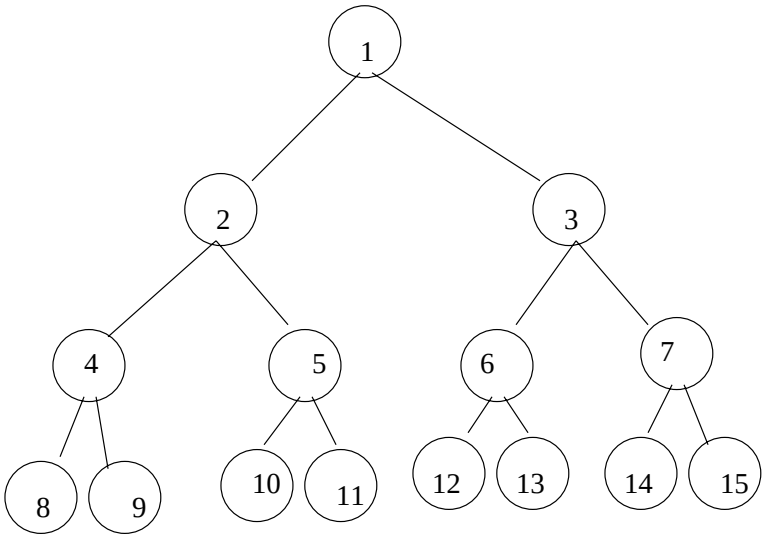
It has the right logic, but it doesn't work because of search.

4 Search

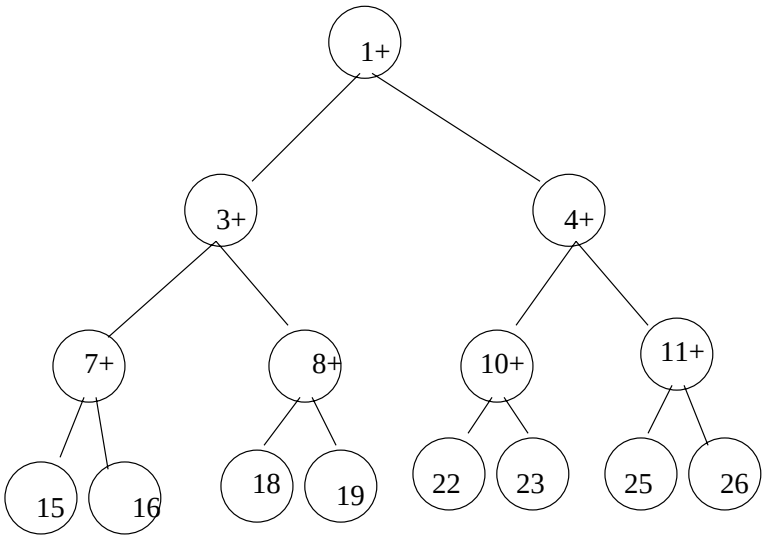
Logic programming languages do DEPTH-FIRST SEARCH.



Bad news if there's infinite paths.
An alternative is BREADTH-FIRST SEARCH.



But this means scrapping the logic programming search all together. Plus it takes an obscene amount of space.
Standard quick compromise ITERATIVE DEEPENING SEARCH.



Do depth-first search, but have a maximum depth that you're willing to consider at any stage. If you find no solution, increase the depth bound.
Nice λProlog higher-order implmentation of iterative deepening.

(32) a
```
type id (int -> o) -> int -> o.
```
b
```
id P D :- P D.
```
c
```
id P D :- N is D+1, id P N.
```

Better plan search:

(33)
```
id (n \ bpn F P R n) 0
```

5 Two final things

Handling unexpected events with intentions.

(34) a
```
type update state -> list fact -> plan -> plan -> o.
```
b
```
update State Facts Plan Plan :-
    circ Plan C, sublist C Facts.
```
c
```
update State Facts Plan Next :-
    id (n \ (build_plan Facts Plan Next (a  true) n;
    repair_plan Facts Plan Next n)) 0.
```

(35) a
```
type repair_plan list fact -> plan -> plan -> int -> o.
```
b
```
repair_plan Facts (step Pre B Post) Next Depth :-
     build_plan Facts Post Next (a \ not (a = B)) Depth.
```
c
```
repair_plan Facts (step Pre B Post) Next Depth :-
    repair_plan Facts Post Next Depth.
```

Using intentions to decide what to do next.

(36) a
```
type act state -> plan -> o.
```
b
```
act State (finish G).
```
c
```
act State (step C A P) :-
    execute State A,
    agitate (do A State) P.
```

An intention guides the agent's deliberation by proposing actions that the agent might choose, focusing the agent's attention to the circumstances in the world that make the action appropriate, allowing the agent to more quickly identify problems and opportunties that arise in carrying out the plan, and suggesting ways that the agent can respond to the unexpected!

Foundations of Language Interaction

HANDOUT THREE

June 7, 2001

M. Stone
mdstone@cs.rutgers.edu

6 Introduction

Today we consider the interaction of KNOWLEDGE and INTENTIONS.

(37) Last week, we characterized a plan as an argument that demonstrates how performing a sequence of actions in the current circumstances leads to desired effects.

A rational agent need not make all its decisions immediately. It can instead defer choices to later steps of deliberation. Plans can and should guide these later choices, but only if they anticipate the new reasons to act afforded by the agent's increased future information. Our new account of plans follows [Stone, 1998]; those of you unsatisfied with [Davis, 1994] as a guide to today's meeting may want to consult this paper.

(38)
```
agitate State Goals Intentions :-
    perceive State Beliefs,
    update State Beliefs Goals Intentions
    NewIntentions,
    !,
    act State Goals NewIntentions.
```

Today's objective is to extend the representations of intentions used by our classic agent simulator, showing how to REPRESENT knowledge and choice, and how to formalize INFERENCES about knowledge and choice in plans.

7 Representing knowledge

Recall that `fact` was the type of statements in the agent's knowledge base; the agent's knowledge base was a list of facts. We now add three new declarations.

(39) a
```
kind agent, object type.
```
b
```
type k agent -> fact -> fact.
```
c
```
type sm (object -> list fact) -> fact.
```

`k` REPRESENTS knowledge:

(40) a If `a` is an agent, `(k a)` is a function from facts to facts; this is called a MODAL OPERATOR.

b use [A] to write `(k a)` in logic.

c `(k a f)` represents the fact that agent `a` knows fact `f`.

`sm` is indispensible for SPECIFYING knowledge.

(41) a `(sm P)` represents the fact that there is an object `x` for which all the facts in `(P x)` are true.
b Use λProlog's function syntax to make this look more like $\exists x P(x)$: `(sm x\ P x)`.
c Illustrates HIGHER-ORDER ABSTRACT SYNTAX. Use λ-abstraction to represent bound variables in the OBJECT LOGIC of fact-expressions as bound variables in λProlog, the META-LOGIC.
d To substitute an object-level term for an object-level bound variable, use meta-level function application.

Some key formulas, from [Hintikka, 1971].

(42) a `type food object -> fact.`
b `sm x\(k self food x)::nil` $= \exists x[\text{SELF}]fx$
c `k self (sm x\ food x::nil)` $= [\text{SELF}]\exists x fx$

(42b) means that there is a specific object *x* about which you know that it's food. (42b) is an INDEFINITE SPECIFICATION of what you know—it constrains what you know but does not say exactly what you know (it doesn't say what that *x* is that you know is food). (42c) means that you know that there is some food. (42c) says exactly what you know, but indicates that you have only INDEFINITE KNOWLEDGE—you don't actually know what the food is, specifically.

8 Plans and proof

Suppose you want to achieve a goal *G* at some point.

(43) a You need to choose a specific action *x*, based on your knowledge. Your knowledge should tell you that *x* brings it about that (BIAT) you know *G*.
b Prove $\exists x[\text{SELF}](x \text{ BIAT } [\text{SELF}]G)$.

Suppose you want to achieve a goal *G*, and you get to choose two actions. You need to choose the first now, but you don't need to choose the second until your next cycle of perception and action.

(44) a You need to choose a specific action *y*, based on your knowledge. Your knowledge should tell you that *y* brings it about that you can achieve *G* in the sense of (43) afterwards.
b Prove $\exists y[\text{SELF}](y \text{ BIAT } \exists x[\text{SELF}](x \text{ BIAT } [\text{SELF}]G))$

I can't resist a peek ahead to collaboration! Suppose you and a friend want to achieve *G*; you act first, then the other goes.

(45) a You need to choose a specific action *y*, based on your knowledge. Your knowledge should tell you that *y* brings it about that the other can achieve *G* afterwards.
b Prove $\exists y[\text{SELF}](y \text{ BIAT } \exists x[\text{OTHER}](x \text{ BIAT } [\text{SHARED}]G))$

We will presume that you COORDINATE on starting: as part of the collaboration you both know the proof and have agreed to act as it lays out. At the end [SHARED] ensures that you can coordinate on stopping.

9 Formalizing Deductions

(46) a Suppose you have a bunch of premises `Fs` including `sm x\(C x)`, and you're trying to prove `G`. And let `w` be a symbol that doesn't occur in `Fs` or `G`. If you can prove `G` from `Fs` together with `C w` then you can prove `G` from `Fs`.

b If the subproof with `w` is `Plan w`, then the overall proof is `whatever Fs C Plan`.

c
```
type whatever
list fact -> (object -> list fact) -> (object -> plan) ->
plan.
```

(47) a Suppose you have a bunch of premises `Fs` which all take the form `k Agent F`, and you're trying to prove `k Agent G`. If you can prove `G` from `Fs` then you can prove `k Agent G` from `Fs`.

b If the subproof is `Plan`, then the overall proof is `know Agent Fs Plan`.

c
```
type know
agent -> list fact -> plan -> plan.
```

(48) a If you want to prove an existential statement, just prove an instance. We will only have to prove existential statements which quantify over the action the agent chooses.

b If the subproof is `Plan` and the action selected is `Action`, the overall proof is `find Action Plan`.

c `type find action -> plan -> plan.`

Here's a plan that assumes that we know of something that it's food. We plan to eat it, and thereby to sate our hunger.

(49)
```
(whatever ((sm x\ k self (food x)::nil)::nil)
 (x\ k self (food x)::nil)
 (f\(find ((k self (food f))::nil)
     (eat f)
     (know self ((k self (food f))::nil)
      (step ((food f)::nil)
       self (eat f)
       (know self ((k self full)::nil)
        (finish (full::nil)))))))
```

As always, we keep only the premises we need to continue. We also assume that if anyone knows something, it's true. (Remember we're using this inference to guide deliberation, so if we have reason to disbelieve something that a plan may depend on, we should thrash out the discrepancy now rather than assess what we would otherwise expect about mental states.)

(50)
```
(find look
 (know self
  ((k self (sm x\((food x)::nil)))::nil)
  (whatever ((sm x\((food x)::nil))::nil)
   (x\((food x)::nil))
   (f\(step ((food f)::nil)
       self look
       (find (eat f)
        (know self ((k self (food f))::nil)
         (step ((food f)::nil)
          self (eat f)
          (know self ((k self full)::nil)
           (finish (full::nil)))))))))))
```

(50) shows the kind of reasoning involved in (49) would appear as a subplan of a larger plan that describes first getting more information, then what you'll do in the future once you have that information.

10 Using Plans

Finding the next action:

(51) Scan down into the plan structure until you find the first `step` operation. (The agent should be `self`.) Do the specified action.

Finding the intention for the next round of deliberation

(52) Scan down into the plan structure until you find the first `know` operation after your current `step`. (The agent should be `self`.) Save this as your intention for next time.

Handling indefinite information.

(53) When you reach a substructure of the form `whatever _ _ PF`, create a new, unspecified value with some variable `X` and process `PF X`.

For the case of plan (50).

(54) a Next action is `look`

b Subplan is
```
(know self ((k self (food X1))::nil)
 (step ((food X1)::nil)
  self (eat X1)
  (know self ((k self full)::nil)
   (finish (full::nil)))))
```

c Logic variable `X1` replaces bound variable `f` in plan instance.

Monitoring plan execution—actually, same as before:

(55)
```
update _ Facts _ Plan Plan :-
    circ Plan C, entail_all Facts C.
```

In this case, `Facts` gives the specific knowledge you have

(56)
```
know self food o7
```

And the circumstances `C` is an indefinite specification of this knowledge, involving variables:

(57)
```
know self food X1
```

As in any query processing, when you establish that (56) entails (57), you compute the substitution `X1 = o7`. Thus after the call to `update` succeeds in clause (55), our current intention is as in (58).

(58)
```
(know self ((k self (food o7))::nil)
 (step ((food o7)::nil)
  self (eat o7)
  (know self ((k self full)::nil)
   (finish (full::nil)))))
```

From (58) we get

(59) a Next action: `eat o7`

b Next plan:
```
(know self ((k self full)::nil)
        (finish (full::nil)))
```

Foundations of Language Interaction

HANDOUT FOUR

June 14, 2001

M. Stone
mdstone@cs.rutgers.edu

11 Introduction

Today we turn from FORMING INTENTIONS to RECOGNIZING INTENTIONS.

12 Recap

Intentions in deliberation:

(60) a Any PLAN is a representation that maps out what choices the agent makes and what actions it does;
b it shows that the choices would lead to certain hypothetical effects under certain hypothetical circumstances.
c If the plan effects what the agent wants —
d If the agent believes the circumstances obtain —
e The agent can commit to the plan – this is the agent's intention – and act to follow it.

The examples got a bit ugly...

(61)
```
(find look
 (know self
  ((k self (sm x\((food x)::nil)))::nil)
  (whatever ((sm x\((food x)::nil))::nil)
   (x\((food x)::nil))
   (f\(step ((food f)::nil)
       self look
       (find (eat f)
        (know self ((k self (food f))::nil)
         (step ((food f)::nil)
          self (eat f)
          (know self ((k self full)::nil)
           (finish (full::nil)))))))))))
```

But you can read out the important bits:

(62) a The choices and actions are, in the first step, to `look`; and in the next step, when you know about some food `f`, to `eat f`.
b The hypothetical effects are that the agent feels `full`; the hypothetical circumstances are that the agent knows that there is some food—`k self (sm x\((food x)::nil))`.
c If there are goals that this plan achieves:

```
goals Plan PlanGoals,
member G AgentsGoals,
entail_one PlanGoals G
```

d If the agent believes these circumstances obtain:

```
circ Plan PlanCirc,
entail_all AgentsBeliefs PlanCirc
```

e The agent commits to the plan and performs the first action

```
choose Plan Action Next
```

13 Recognizing Intentions

When you recognize an agent's action as intentional, you describe all these steps.

(63) a You build a plan that is a consistent representation of choice, action and effect.
b The actions in the plan must be consistent with the actions you observe.
c The circumstances that the plan supposes must be consistent with what you know of the agent's beliefs.
d The effects that the plan achieves must be consistent with what you know of the agent's goals.

For each of these constraints, you can point to a step of the agent's deliberation which could not have used this plan if the constraint was not met.

(64) a Sandy says "I'm hungry." Later you observe Sandy in the kitchen opening cabinets and peering inside.
b You attribute to Sandy the intention in (61).

Note how consistency is the right condition.

(65) a You may assume that the agent has done or will do more actions that you don't know about. For example, you may assume a later event where Sandy eats the food she hopes to find.
b You may assume that the agent believes something when you might not otherwise have any reason to attribute any opinion to the agent one way or another. For example, you might know yourself that there is no food in the kitchen, and so you might have no idea whether Sandy thinks this is a typical kitchen, with food, or whether she knows the deal.
c In other cases, the agent may have goals that you don't know about ahead of time, as well.

Note also that, in plan recognition, there's one right answer, and you have to entertain all the possibilities. By contrast, planning is a satisficing problem. You can choose any plan that works.

(66) Chris is standing on the sidewalk with his hand in his pocket. Is he

- Getting his car keys to drive away?

- Getting his house keys to go inside?
- Getting some change to put into a parking meter?
- About to light a cigarette?

We can't just assume one—the situation is really ambiguous.

14 Intention Recognition in Collaboration

Playing catch.

(67) a Kim throws the frisbee.
b Sandy attributes an intention to Terry. Here is the content of this intention. Terry throws, Sandy catches, Sandy throws, Terry catches. As a result of these actions, the frisbee goes from Terry to Sandy and back again, achieving goal of fun.
c Sandy thinks this intention can be realized, so Sandy commits to it herself.
d Sandy performs two steps of deliberation and action guided by her current intentions. That is, Sandy catches and then Sandy throws. Meanwhile, Terry recognizes Sandy's action as meeting his expectations, and Terry remains committed to these intentions, too.
e Finally, Terry catches, completing their joint execution of the original intention.

Questions and answers.

(68) a Terry asks whether *p* is true.
b Sandy attributes an intention to Terry. Here is the content. Terry asks whether *p* is true, Sandy replies with the answer. As a result of these actions everybody knows whether *p* is true.
c Sandy thinks this intention can be realized, because she knows whether *p* is true (say it's true). So Sandy commits to this intention herself.
d Sandy deliberates, and decides to reply `yes` based on matching up the information required in Terry and Sandy's intention with the information Sandy actually has now.
e Terry recognizes Sandy's reply as meeting his expection and completing their joint execution of the original intention. Everyone knows *p* is true.

Foundations of Language Interaction

HANDOUT FIVE

June 21, 2001

M. Stone
mdstone@cs.rutgers.edu

15 Introduction

Today—at last—we can look at INTENTIONS IN COMMUNICATION.

16 Grice on Meaning

In [Grice, 1957], Grice is trying to do TWO things, corresponding to two definitions on page 385.

(69) "*A* meant_{NN} something by *x*" is (roughly) equivalent to "*A* intended the utterance of *x* to produce some effect in an audience by means of the recognition of this intention.

(69) is a characterization of communicative reasoning and particular processes in conversation. Thomason refines this as in (70) in next time's reading [Thomason, 1990].

(70) To mean *p* is to intentionally reveal an intention to make *p* asserted through the hearer's recognition of the status of an intention or plan of the speaker's.

More on this soon.

(71) "*x* means_{NN} (timeless) that so-and-so" might as a first shot be equated with some statement or disjunction of statements about what "people" (vague) intend (with qualifications about "recognition" to effect by *x*.

(71) is a proposal about where our intuitions about linguistic meaning come from. In unpacking linguistic meanings into generalizations about what speakers do, (71) suggests a way of demystifying knowledge of meaning, learning of meaning, and so forth.

(72) I must disclaim any intention of peopling all our talking life with armies of complicated psychological occurrences.

This is 1957—the same year as *Syntactic Structures* and Newell and Simon's *General Problem Solver*. Grice is writing to the behaviorists that are still running around with dumb ideas about grounding the meanings of words and sentences in learned associations between sounds and referents. Yet Grice would not accept, nor even recognize, the modern conception of knowledge of meaning realized as a computational system in the semantic component of a speaker's grammar. (71) may be as outdated as (72).

17 Linguistic Intentions

From last time.

(73) a Terry asks whether *p* is true.
b Sandy attributes an intention to Terry. Here is the content. Terry asks whether *p* is true, Sandy replies with the answer. As a result of these actions everybody knows whether *p* is true.
c Sandy thinks this intention can be realized, because she knows whether *p* is true (say it's true). So Sandy commits to this intention herself.
d Sandy deliberates, and decides to reply `yes` based on matching up the information required in Terry and Sandy's intention with the information Sandy actually has now.
e Terry recognizes Sandy's reply as meeting his expection and completing their joint execution of the original intention. Everyone knows *p* is true.

A theory of replying:

(74) If *Q* is a question we are considering at this point in the dialogue, and the answer to *Q* is *A*, and the expression *E* can mean *A*, then *replying to Q by saying E* can achieve the result that everybody knows that the answer to *Q* is *A*.

A strips formalization:

(75)
```
is_action (reply Q E)
          (k all (want Q) ::  answer Q E ::
                    k all (o_name A E) ::  nil)
          (k all (answer Q A) ::  nil)
          nil.
```

Aside: linguistic action and grammatical theory.

(76) a Utterance = action.
b Precondition = presuppostion.
c Addition = assertion.
d Deletion = change in salience in discourse? Or not meaningful?

Another aside.

(77) a To account for Grice's means$_{NN}$ we should also represent the MECHANISM in the theory of (74) and (75).
b Those of you who struggled through Pollack's definition of Conditional Generation of actions know one way this might work: (74) is part of a THEORY or CONTEXT that appeals to an implicit CONDITION or set of circumstances. These conditions describe the prerequisites for the process of intention-recognition and collaboration in conversation.
c [CONVERSATION]: If *Q* is a question we are considering at this point in the dialogue, and the answer to *Q* is *A*, and the expression *E* can mean *A*, then *replying to Q by saying E* can achieve the result that everybody knows that the answer to *Q* is *A*.

A plan to answer a question:

(78)
```
(find (reply Q E)
 (know Agent (k Agent (k all (want Q)) ::
              k Agent (answer Q A) ::
              k Agent (k all (o_name A E)) ::
              nil)
  (step (k all (want Q) ::
        answer Q A ::
        k all (o_name A E) ::
        nil)
   Agent (reply Q E)
   (finish all (k all (answer Q A) ::  nil)))))
```

Recall as always.

(79) a When the agent commits to the plan, the agent makes sure it applies.
b This means proving `answer Q A` and `k all (o_name A E)`.
c In logic programming this will set `A` to an answer that the agent knows and, by grammatical reasoning, set `E` to any expression that could refer to `A`

18 Linguistic Intention Recognition

Since we've put off questions of search for plans, recognition means finding a match between a template you have, like the template in (78), the action you observe, and your current information. To start, that's at least:

(80) a Setting `E` to the observed answer, say `yes`.
b Setting `Q` to the observed question, if there was one. (Realistically there isn't—and the same story goes if we just use the action `reply E`—but we will assume `Q=q 1`.)
c Proving or assuming instances of the circumstancs of the plan that the agent who did the action must have checked before carrying out the plan.

A plausible starting assumption in step (80c) is that you must PROVE facts that attribute shared knowledge to the participants in the conversation (because you should really share your mutual beliefs). On the other hand, it is relatively painless to assume that your partner is acting on the basis of information you don't have. Here then:

(81) a Prove: `k all (want (q 1))`
b Prove: `k all (o_name A yes)`
c Assume: `k partner (answer (q 1) A)`

Observation one:

(82) a As always in logic, proving `k all (o_name A yes)` will instantiate `A` to a specific value.
b Hence an account of PRESUPPOSITION AS ANAPHORA [Kripke, 1991, van der Sandt, 1992]—resolve presuppositions by hypothesizing specific instances as mental representations behind an utterance.
c In this case, the candidate interpretations correspond to the things that `yes` means.

19 Intended Recognition

The key new thing from Grice is that in communication the speaker will INTEND this plan to be recognized.

(83) Committing to a communicative plan requires two checks. You must make sure that the plan applies. And you must make sure that the plan will be recognized as intended.

This means assessing the inference in (18) in a suitable context, and rejecting plans that don't work.

(84) a In this case, since `yes` only means `truth`, the plan is recognized.
b If you had a word `mmm` that could mean `truth` and could mean `falsehood`, then the plan to answer with this word wouldn't recognized. Therefore the word wouldn't be used.

20 Questions for Discussion and Research

(85) a What does it mean to recognize a communicative plan based on misconceptions? If the misconceptions are in what your partner assumes privately about the world? If the misconceptions are in what your partner takes to be shared?
b Once you have recognized a plan, what do you do with it? For example, does the plan achieve its effects?

These interdisciplinary problems are of central interest in current research, and are studied under headings like COOPERATIVE AGENCY (in computer science), ACCOMMODATION (in linguistics and philosophy) and GROUNDING (in psychology).

Foundations of Language Interaction

HANDOUT SIX

July 12, 2001

M. Stone
mdstone@cs.rutgers.edu

21 Introduction

Today's topics are ACCOMMODATION and GRICEAN REASONING in utterance interpretation. The goal is to show how theories of the SEMANTICS–PRAGMATICS INTERFACE in linguistics may benefit from formal models of interpretation.

22 Recognizing Assumptions in Plans

We begin by contrasting three scenarios involving plan-recognition for real-world action. The scenarios take place in the following setting.

(86) Picture a corridor with a door at one end. (Either corridor near the elevator on the third floor of Core where my office is will do.) The door has a metal frame but the panels are glass; you can see through the door. The door is sometimes locked—always at night. John is approaching the door from the outside, while Mary walks down the corridor on the insider of the door.

Here are the different scenarios.

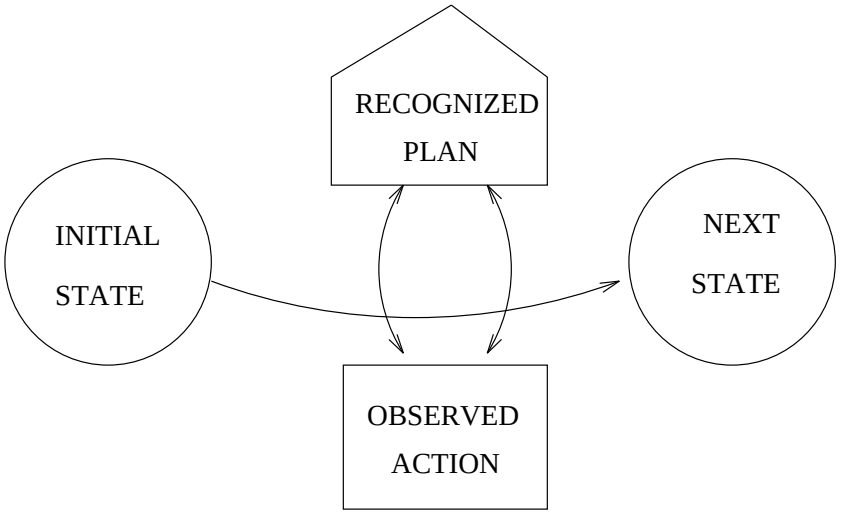
(87) Mary watches John pull the handle and open the door.

(88) Mary watches John pull the handle. John makes a puzzled face and nothing happens.

(89) Mary watches John pull the handle. John is looking at Mary expectantly; nothing else is happening with the door.

From our point of view, these scenarios all unfold by the same process, which is very similar to conversational process. I abstract it in the following picture.

(90)



These categories apply to the three scenarios as follows.

(91) a INITIAL STATE. The door is unlocked; John believes the door is unlocked; John wants to go through the door.
b OBSERVED ACTION. Pulling the handle.
c RECOGNIZED PLAN. Pull the handle when the door is unlocked to open the door and go through.
d NEXT STATE. John no longer has goal of opening the door; John will go through it and Mary knows this.

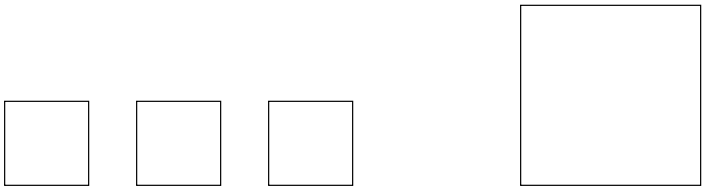
(92) a INITIAL STATE. The door is locked but John believes the door is unlocked; John wants to go through the door.
b OBSERVED ACTION. Pulling the handle.
c RECOGNIZED PLAN. Pull the handle when the door is unlocked to open the door and go through.
d NEXT STATE. John still has the goal of opening the door and going through it; John is debugging his intention and trying to find another that will work in the current circumstances.

(93) a INITIAL STATE. John and Mary both know that the door is locked, but John and Mary share an assumption that each will come to the other's assistance.
b OBSERVED ACTION. Pulling the handle.
c RECOGNIZED PLAN. Pull the handle when the door is unlocked to open the door and go through.
d NEXT STATE. John still has the goal of opening the door and going through it; and Mary has recognized this goal from John's action. Mary, if she's as cooperative as expected, now plans to help with this goal by opening the door from the inside.

THE OBSERVED ACTION AND THE RECOGNIZED PLAN ARE THE SAME IN ALL CASES. The difference depends on the initial state, which determines the effects that the plan causes and the reactions that the agents have to the plan. This difference thereby percolates into the next state.
EXAMPLE (89) AS ANALYZED IN (93) IS AN EXAMPLE OF ACCOMMODATION. ACCOMMODATION [Lewis, 1979] describes a process by which participants in a collaborative relationship can collude with one another with the effect that a flawed intention manages to contribute to the goal for which it is exhibited despite its flaws. Accommodation in pragmatics is tricky because—unlike in example (89) where you could watch Mary open the door—you can't watch the steps that conversational participants take to accommodate one another. Accommodation just looks like a mysterious exception to the ordinary rules of conversation [Thomason, 1990].

23 Presupposed Standards for Vague Predicates

Now let's consider a similar case that's an explicit example of language use. The physical context is as in (94).

(94) 

The utterance is (95).

(95) I want the large square.

The point of this example is to show how the intention behind (95) can be recognized in the context provided by (94), even if we assume that:

(96) a Vague adjectives presuppose a standard of comparison.
b The context does not inherently supply a standard of comparison.

In context, then, (95) is associated with a flawed communicative intention—a plan whose presuppositions are not met. Nevertheless, simply by being recognized, (95) can achieve all the intentions we would normally associate with it; and it can, in addition, update the context to include a standard of comparison for large squares—by accommodation.
By (96), the formal presuppositions of (95) are given in (97).

(97) `square X` $\wedge$ `size X S` $\wedge$ `standard size i+D` $\wedge$ `in S i+D`

In words, `X` is a square, the size of `X` is `S`, the interval `i+D` lower bounded by the value `D` and without an upper bound provides the standard for large size in the context, and `S` lies within the interval `i+D`. To link up explicitly with what we have done before, these presuppositions arise as part of a communicative intention such as that represented in (98).

(98)
```
(find "i want the large square"
 (know self
  (k self (want X) ::
   k self (k all (square X)) ::
   k self (k all (size X S)) ::
   k self (k all (standard size i+D)) ::
   k self (k all (in S i+D)) ::
   nil)
  (step (want X ::
         k all (square X) ::
         k all (size X S) ::
         k all (standard size i+D) ::
         k all (in S i+D) ::
         nil)
   agent "i want the large square"
   (finish all (k all (want X))))))
```

The presupposition shows up in the intention as the condition required for the utterance to achieve its effect. The speaker intends all to recognize the specific intention matching (98) that is behind utterance (95). For the presupposition that means that the speaker and the hearer are coordinating on the instance of the presupposition that figures in this intention. Specifically, we will want the instances in (99).

(99) a X is square number 4 from (94).
b S is the size of square number 1 from (94), say one inch square.
c D is an arbitrary (or underspecified) representation of a "vague" standard that must lie somewhere between the size of squares 1 through 3 from (94) (at half an inch square, say) and the size of square number 4 from (94).

By the way, this way of looking at things involves a rational pragmatic reconstruction of the theory of presupposition as anaphora [Kripke, 1991, van der Sandt, 1992]—and we'll see that it also builds in Beaver's observation that all accommodation occurs at top level [Beaver, 2001].

24 Recognizing the Plan

Let's consider the hearer's inference in recognizing the plan in (98) as instantiated in (99). The hearer is tracking the speaker's deliberation, and knows:

(100) a The speaker is acting as though a certain context obtains. This pretend context is different from the actual context only in certain potentially predictable ways; in particular, the pretend context may supply standards for vague predicates that the actual context does not.
b In this pretend context, the instance of (98) intended by the speaker applies.
c In this pretend context, the instance of (98) intended by the speaker can be recognized as intended.

Here are the steps of inference that the hearer can make:

(101) a By (100a) and (100b), the hearer can infer that X is one of the four squares of (94), and that S is the size of X.
b By (100a) and (100b), the hearer can infer that the pretend context specifies a standard of size smaller than the size S of X.
c At this point, there remain two qualitatively different standards (less than half inch square; or between half inch square and one inch square). But (100c) eliminates the smaller standard, since it provides no way to recognize which of the four squares is X: the presupposition of the plan can be satisfied in the pretend context with any of the four possible squares. On the other hand, (100c) confirms the larger standard, since the presupposition now has only the resolution where X is square number 4.

At this point, it's up to the hearer how to respond to the recognized plan. The following strategy is sensible.

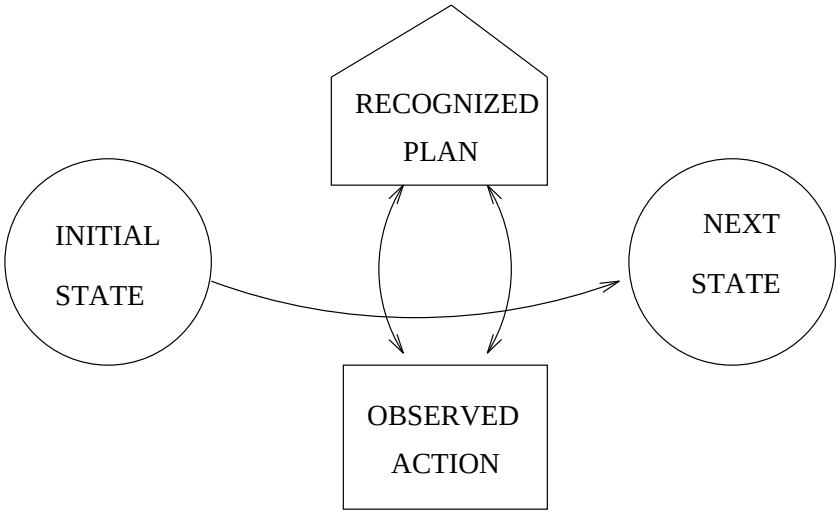
(102) a First, update the representation of the actual context, to provide a standard for size somewhere vaguely between half inch square and one inch square. This is a step of accommodation.

b The remainder of the response is exactly the response that the hearer would have taken to the recognized plan in the accommodated context. For example, if the hearer would have handed the large square to the speaker at this juncture if both had already agreed that this was the large square, the hearer now hands the large square to the speaker.

Note however that this kind of strategy probably diverges from actual practice in face-to-face conversation, where more work would be expected of conversational participants over and above (102) in order to seek and provide evidence of mutual understanding—to ACKNOWLEDGE and GROUND the accommodated context change [Clark and Schaefer, 1989, Brennan, 1990].

25 Summary

Let's return to the big picture.



(103)

As in examples (91), (92) and (93), our discussion contrasts two parallel instances of (103) at issue with utterance (95)

(104) a INITIAL STATE. The square is definitely large and John wants it.
b OBSERVED ACTION. I want the large square.
c RECOGNIZED PLAN. Identify that square with reference to a standard of size and establish getting that square as a shared goal.
d NEXT STATE. Mary and John are committed to the new shared goal and Mary is going to give John the square.

(105) a INITIAL STATE. John wants the square but there's no context for whether that square is large.
b OBSERVED ACTION. I want the large square.
c RECOGNIZED PLAN. Identify that square with reference to a standard of size and establish getting that square as a shared goal.
d NEXT STATE. The context provides a standard of size by which the square is large. Mary and John are committed to the new shared goal and Mary is going to give John the square.

Apparently, this story offers a clear distinction among

(106) a LINGUISTIC KNOWLEDGE which outlines the form that recognized intentions must have.
b COMMUNICATION KNOWLEDGE which determines which candidate intentions are plausible in context.
c COOPERATIVE REASONING which describes how we move from one state of the conversation to the next based on the intentions we recognize.

And yet—plenty of play remains in the theory for competing explanations of the same pragmatic competence. Is this the usual case of a theory that has to be judged based on its overall coherence? Or is this something more pernicious and murky that is peculiar to pragmatics?

Foundations of Language Interaction

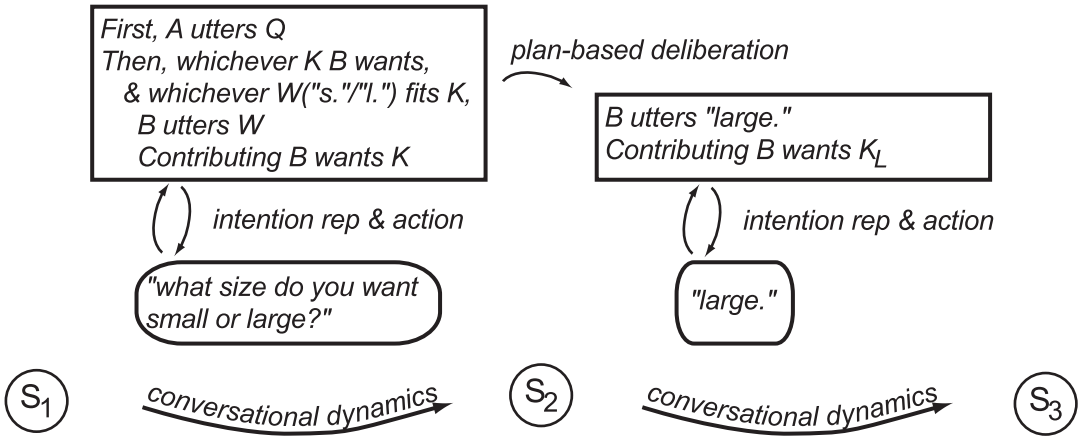
HANDOUT SEVEN

July 19, 2001

M. Stone
mdstone@cs.rutgers.edu

26 Introduction

Today we look at describing CONVERSATIONAL DYNAMICS from the point of view of modern computer science and artificial intelligence research. Recall the kind of framework we're using:



For the moment, we'll abstract away from the communicative intention representations and plan-based deliberation we've been investigating in detail: it's summarized at the level of conversational dynamics. How do we describe conversational dynamics itself, using empirical methods?

27 Some mathematical ideas

First principles might suggest modeling conversational dynamics as a PARTIALLY-OBSERVABLE STOCHASTIC GAME.

(107) A GAME is simply a mathematic formalization of a problem of coordination or competition among agents, where the success of any one agent depends not only on what it does but also on what other agents do.

(108) This game is STOCHASTIC, or governed by probabilistic rather than deterministic laws, because it takes place in a noisy environment in which the effects of actions depend on unreliable components like speech recognizers.

(109) The same noise makes the state of the environment PARTIALLY-OBSERVABLE. Agents must choose their actions based on incomplete information about the world.

There has been some limited work on representing computational problems directly in these terms, such as [Koller and Pfeffer, 1997]. A solution is an equilibrium—a strategy for each participant where the play of each can't be improved if the other plays the strategy. (A wrinkle is that coordination problems typically have multiple equilibria, so players must know what equilibrium to play.) However, practical dialogue research has backed off from these models, which require a lot of parameters and quickly become intractable.

(110) A standard assumption for dialogue research is that the whole community "plays" more or less the same policy, defined by conventions of English. Finding your equilibrium means matching the strategy.

Assumption (110) reduces the problem to a PARTIALLY-OBSERVABLE MARKOV DECISION PROCESS (POMDP), in which you don't have to anticipate other agents' strategic decision-making, and you can analyze dialogue using plain decision theory [Kaelbling et al., 1998]. But POMDPs are intractable too, and further simplification is required.

(111) Walker uses simplified, structured state-representations to finesse the problem of partial observability [Walker, 2000]. This MDP approach emphasizes emphasizes strategy and unpredictability.

Other simplifications are possible.

(112) Horvitz and Paek reason about the value of information in a way that's more faithful to the partial observability of dialogue state, but they accomplish this by giving up on solving strategic problems and using local decision-theoretic heuristics [Horvitz and Paek, 1999, Horvitz and Paek, 2001].

Anyway, from the perspective of computational linguistics, the important thing is not the mathematical formalism you use to solve the problem but the assumptions that you make about linguistic representations in building your model and the techniques that you use to bring the model into correspondence with human behavior.

28 An experiment we're planning for the fall

People answer questions in the terms in which they are posed.

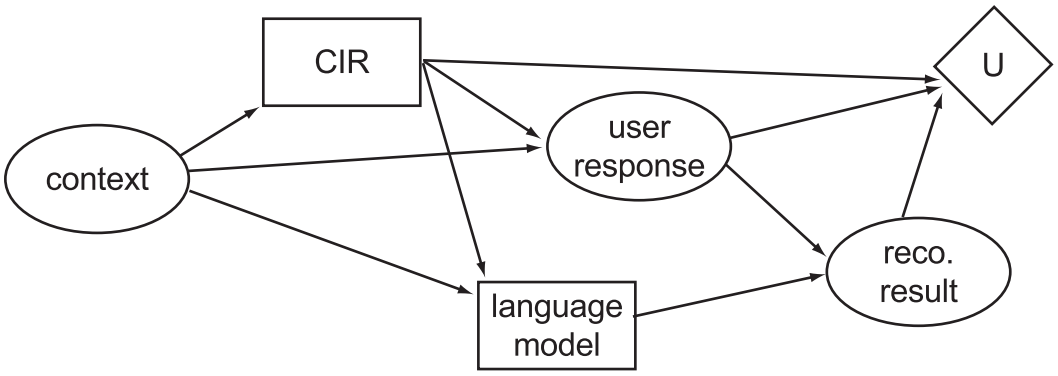
(113) a Q: What size do you want: small or large?
b A: Large.

An example of entrainment. See [Brennan and Clark, 1996].
Dialogue sytems can use this fact to get better performance. The smaller the grammar or the more constrained the vocabulary, the better they are at recognizing what has been said. And good recognition is the most important thing for user satisfaction.
On the other hand, it can be equally frustrating to have to listen to a long list of options when you know how to say what you want—and pretty unnatural to always have to "barge in" to get your task done quickly with a dialogue system. So the system should try to make its questions short whenever possible.

(114) There is a tradeoff of speed and naturalness vs. recognition accuracy between asking short questions (*what size do you want*) and asking verbose questions (*what size do you want: small or large*).

It's an interesting problem because entrainment means you don't always have to be explicit in a context to get the answer you expect.

Here's a simple mathematical model.



The model is represented as an influence diagram [Shachter, 1986]. Circles represent random variables. Squares represents choices that we can make. (So we're matching strategies here, not finding an equilibrium.) The diamond gives the overall payoff associated with the event. Arrows into circles and diamond indicate probabilistic dependencies. Arrows into squares indicate information avialable for a decision. Here we have

(115) a Context: the value of the dialogue state relevant to entrainment, for example, how have we identified the potential answers previously in the conversation.
b CIR: the communicative intention representation we choose for the question (short form or long form). We get to choose this based on the context, and in general we can specify the dependence nondeterministically in the terms of linguistic theory and plan-based deliberation that we've been developing so far.
c Language model: the grammar or description of possible utterances that we provide to the speech recognizer to describe the ambiguities it should consider in recognizing the user's utterance. For our toolkit we have to pick a symbolic grammar, and we have a few to choose from that we can select based on the context and the question we ask.
d User response: what the user actually says (realized, we might suppose, as a communicative intention representation).
e Recognition result: what we understand, though in practice what suffices is probably what the user meant, something the user didn't mean, or total recognition failure.
f Utility: depends on the difficulty the user has understanding the system, the difficulty the system has understanding the user, and the overall ability of the user to solve her task more easily with the system (as in Walker's PARADISE framework).

What we'd like to do is fit the parameters for this model based on what people actually say and use the model to design a strategy for asking questions in a dialogue system.

(116) With four conditions for running subjects—entrainment from context or no entrainment from context, long question or short question—we can collect the distribution of user responses. Of course we have to code this by hand! Otherwise dialogue systems would work already.

(117) We can reuse the corpus of user responses to understand the error rate of the speech recognizer under different grammars (and, effectively, under different distributions of user response).

(118) As Walker does, we can ask the users what they thought of the system and build a correlation model to assess what's important to them.

That gives a full model. Solving it is then actually pretty easy. Hopefully we'll get the solution we expect – but we'll get it through a quantitative model that says how good it is and that we can validate by running the final system. If so, we'll have learned something about dialogue as well as making a better system.

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.

[Beaver, 2001] Beaver, D. (2001). *Presupposition and Assertion in Dynamic Semantics*. CSLI Press. Revision of 1995 PhD Thesis, University Of Edinburgh.

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

[Brennan, 1990] Brennan, S. E. (1990). *Seeking and Providing Evidence for Mutual Understanding*. PhD thesis, Stanford University.

[Brennan and Clark, 1996] Brennan, S. E. and Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 22(6):1482–1493.

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

[Clark and Schaefer, 1989] Clark, H. H. and Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, 13:259–294.

[Davis, 1994] Davis, E. (1994). Knowledge preconditions for plans. *Journal of Logic and Computation*, 4(5):721–766.

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

[Fikes and Nilsson, 1971] Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

[Grice, 1957] Grice, H. P. (1957). Meaning. *The Philosophical Review*, 66(3):377–388.

[Hintikka, 1971] Hintikka, J. (1971). Semantics for propositional attitudes. In Linsky, editor, *Reference and Modality*, pages 145–167. Oxford.

[Horvitz and Paek, 1999] Horvitz, E. and Paek, T. (1999). A computational architecture for conversation. In *User Modeling Conference*, pages 201–210.

[Horvitz and Paek, 2001] Horvitz, E. and Paek, T. (2001). Harnessing models of users' goals to mediate clarification dialog in spoken language systems. In *User Modeling Conference*.

[Kaelbling et al., 1998] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101.

[Koller and Pfeffer, 1997] Koller, D. and Pfeffer, A. (1997). Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1):167–215.

[Kripke, 1991] Kripke, S. (1991). Presupposition and anaphora: Remarks on the formulation of the projection problem. transcript of a lecture given at Princeton.

[Lewis, 1979] Lewis, D. (1979). Scorekeeping in a language game. In *Semantics from Different Points of View*, pages 172–187. Springer Verlag, Berlin.

[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.

[Shachter, 1986] Shachter, R. D. (1986). Evaluating influence diagrams. *Operations Research*, 34:871–882.

[Stone, 1998] Stone, M. (1998). Abductive planning with sensing. In *AAAI*, pages 631–636, Madison, WI.

[Thomason, 1990] Thomason, R. H. (1990). Accommodation, meaning and implicature. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pages 325–363. MIT Press, Cambridge, MA.

[van der Sandt, 1992] van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(2):333–377.

[Walker, 2000] Walker, M. A. (2000). An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.

[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.