

Inference with frames

Minsky's primary idea was that frames provide an instant set of predictions about familiar situations and objects, and this enables us to rapidly deploy the right behavior in those situations or with respect to those objects. Note that this is a form of "tentative inference": The predictions are generally correct, but there can be violations of expectations (which we would notice in "verifying" our expectations, by looking around). Minsky allowed for some of this explicitly, via "default values" -- but it's also reasonable to regard entire slots as tentative, e.g., you might expect a board in a classroom, but wouldn't be "disoriented" if you walk into a classroom lacking a board.

Schank & Abelson's "scripts" had a similar purpose, but oriented more towards familiar patterns of events (e.g., restaurant visits) than familiar entity types.

The description logics that were developed subsequently, like CLASSIC, LOOM, ALC, and OWL-DL (the last for the "semantic web") tended to be more rigid, treating the slots as *requirements*, and didn't focus so much on tentative situational inference as on "subsumption" with inheritance, and "classification". Subsumption means making inferences based on the specialization/generalization relations that can hold between frames or concepts. For example, given that savings accounts and checking accounts at a bank are subsumed under "account", then in addition to inferring the special properties specified for a savings (or checking) account for a given account being opened, we can also infer the properties inherited from "account". We can also generally reason that a given savings account is a financial contract, if we have an inheritance path from savings account to account to financial contract (a subsumption inference). Classification means recognizing that if we are given certain attributes of an entity that match the slots of a frame F and its superframes, while not matching the attributes of any other frames, then it can be classified as an instance of F -- provided that we have a priori reason to think that the entity is definitely within the class of entities covered by the frame hierarchy. This is useful, for example, in meeting a customer's requirements in selecting a product from a large inventory that has the attributes desired by the customer -- we classify the set of requirements, and then look for inventory items in the identified class.

But description logics (or "terminological logics"), because of their limited slot-and-value (or value type) syntax, were often supplemented with an "assertional" component, where one could write down more general logical formulas. This is similar to treating the terminological (frame) component as a specialist method for fast subsumption and classification inference. But a lot of research was invested in making the terminological logics themselves more general, adding negation, disjunction, limited forms of quantification, etc. The odd consequence of this was that the original intent of description logics to enable rapid inference of any of their logical consequences was undermined: Inferences that originally were obtainable in polynomial time became exponentially complex or even undecidable. Description logics like ALC and OWL-DL try to strike a balance between expressivity and efficiency.

My research group's interest is in a general notion of frame-like or script-like knowledge (generally, "schemas"), more in Minsky's and Schank & Abelson's original spirit of supporting "expectation" inference. But the problem with all the traditional formalisms, because of the attribute-value formats they generally assume, is that they make it hard to specify *relationships* among parts or participating entities in a frame or script. For example, if you try to formulate a frame for

a "hammer", it's easy enough to introduce slots for the handle and the head, but there's no way to say that the head is attached to the handle (at right angles)! We make no such limiting assumption, and instead allow arbitrary logical statements about parts or events in our schemas. Some simple example excerpts from schemas we are using:

From the LISSA schema for casual conversation:

```
.....
(event-schema ((set of me you) have-lissa-dialog.v) ** ?e)
  ...
  ?a7 (Me say-to.v you '(What is your favourite class so far?))
      What is your favorite class so far?
      What is your favorite class?
  ?a8 (You reply-to.v ?a7)
      Hmm, I guess it's math.
      Your favorite class is math.
  ?a10 (Me react-to.v ?a8)
      Oh, good. I like math as well.
      I like math.
  ?a11 (Me say-to.v you '(Are you finding it hard?))
      Do you find it hard?
      Do you find math hard?
  ?a12 (You reply-to.v ?a11)
      Not especially.
      You don't find math hard.
  ?a14 (Me react-to.v ?a12)
      It's good to have a gift for math.
      You are very good in math.
  ...
)
```

From the "blocks world" schema for spatial question answering:

```
.....
?a5. (:repeat-until (:context (?a5 finished2.a))
      ; Prompt the user for a spatial question.
      ?a7 (Me say-to.v you
          '(Do you have a spatial question for me?))
      ; User replies with either spatial question, special request,
      ; or smalltalk.
      ?a8 (You reply-to.v ?a7.)
      ?a9 (:if
          ((:equal (ulf-of.f ?a8.) (GOODBYE.GR))
            ?a11. (:store-in-context '(?a5 finished2.a))
            ?a12. (Me react-to.v ?a8.)); this is the answering step
          ...))
```