A Project Report on NP: an Assumption-based NL Plan Inference System that uses Feature Structures

John K. Myers
ATR Interpreting Telephony Research Laboratories
Sanpeidani, Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
myers@atr-la.atr.co.jp

Abstract

This paper presents a project report on NP, a working **N**atural language **P**lan inference system that uses feature structures and is based on assumptions. Input to the system is in the form of feature structures, which can be taken directly from the output of a semantic parser. Plan actions are represented by feature-structure plan schemata with preconditions, hierarchical decompositions, and effects. Output is in the form of a network of believed assertions represented in a knowledge base, and can be reported, used to answer generation-system queries, or drive side-effecting demons. The plan inference component is implemented using a feature-structure-based inference engine and models of plan recognition, prediction, and inference. The inference engine is implemented using a rewriting system for pattern-matching, and an Assumption-based Truth Maintenance System (ATMS) for conjunctions. The ATMS allows pre-instantiation of hypothetically known assertions and implications, which can significantly reduce processing time. The ATMS also permits simultaneous consideration of multiple possible inputs or multiple possible inferred plan outputs; these can be mutually conflicting or supportive. This capability will be important for disambiguation. The NP system is used to infer dialog- and domain-level plans, among other types.

Original contributions include: a plan inference system that works directly from feature structures; a plan inference system that uses an ATMS and plan schema actions with preconditions and effects to infer hierarchical and chained plans; and, an inference engine that works with multiple feature-structure assertions and rules.

Project Goal. This project is aimed toward a dialog understanding system that can be used as part of an automatic interpreting telephone system. Interpretation will be performed by parsing, transferring, and generating utterances. Thus, dialog understanding will be used to recognize speech acts and illocutionary acts, resolve ellipses, and provide required missing information, among other tasks. The understanding system will use the output of the semantic parser, and provide information to the transfer module and generation system. Therefore, feature structures should be used as the basic data representation scheme. Dialog understanding requires a general-purpose plan inference engine that can work with dialog plans, domain plans, commonsense knowledge plans, and so forth. The system must also in the future be able to perform disambiguation of possible utterances.

Background: Assumptions. The plans, intentions, beliefs, etc., of a human are *mental* concepts which cannot be perceived directly: they are *unobservable*[Mye88]. There is insufficient information to represent these concepts with certainty. Therefore, the system must be able to represent concepts in an uncertain manner, using assumptions.

Communication between two people is inherently an assumption-based process. Since it is never completely possible to directly know the concepts of another person, it is necessary in the course of a conversation to take a stance and rely on assumptions about the other person's thoughts [Den87]. Thus, in a dialog understanding system, there are at least two kinds of assumptions that must be represented: assumptions that the two speakers make, which must be modeled by the system, and assumptions that the system makes about the situation,[1] the two speakers and their plans, intentions, etc.

Design. Plan inference and other knowledge-based reasoning tasks require that multiple conjunctive implications be matched against large sets of unordered assertions. The system will have a catalog of world knowledge, common-sense knowledge, and assertions which are believed by the speakers. These must be accessed non-sequentially and used for reasoning. In other words, language understanding should be done by using an "expert system" inference engine.

Computer languages should be used according to their strengths and weaknesses. Feature-structure systems are strong in representing complex, incomplete, or underspecified information, and in performing unification. However, they are extremely inefficient at list processing and numeric calculations (e.g. for evidential reasoning), and don't represent multiple possible worlds. Lisp and other languages can fulfil these needs.

One solution is to build a hybrid system. An inference engine was built which uses a feature-structure language for representation and pattern-matching tasks, while using an ATMS to perform conjunctive implications, represent assumptions, represent possible worlds, and maintain the truth of derived belief networks when nonmonotonic changes occur. The ATMS allows the system to represent, and reason with, all consistent possibilities at the same time–not just the current best choice. In particular, this permits multiple possible inferred plans to be output, and multiple possible observations to be input. This capability will become important for possible utterance disambiguation.

The system interprets the results of the ATMS by using a five-valued uncertainty logic consisting of the uncertain belief values ACTUAL, POSSIBLE, HYPOTHETICAL, INCONSISTENT, or NULL. Each asser-

[1] Currently, most dialog understanding systems start with the assumptions that the hearer and speaker always understand each other perfectly, that they automatically want to cooperate as much as possible, and that they have absolutely no other commitments outside of the conversation. Clearly some of these assumptions can occasionally be incorrect.

```
[[action [[RELN short-answer-negative-set-1]
          [AGEN ?questioner]]]
 [prec1  [[RELN  S-REQUEST]
          [OBJE [[RELN  INFORMIF]
                 [AGEN ?answerer]
                 [OBJE [[RELN  ?verb]
                        ?rest1]]]]]]
 [dec1   [[RELN  Iie-NEGATIVE]
          [AGEN ?answerer]]]
 [dec2   [[RELN  NEGATE]
          [OBJE [[RELN  TeIru-STATIVE]
                 [AGEN @XO1[]]
                 [OBJE [[RELN  ?verb]
                        [AGEN @XO1]
                        [OBJE []]]]]]]] ;(Null)
 [eff1  [[RELN POSSIBLE-EXPECT]
         [OBJE [[RELN  Ta-PERFECTIVE]
                [OBJE [[RELN  Wakaru-1]]]]]]]  ]
```

Figure 1

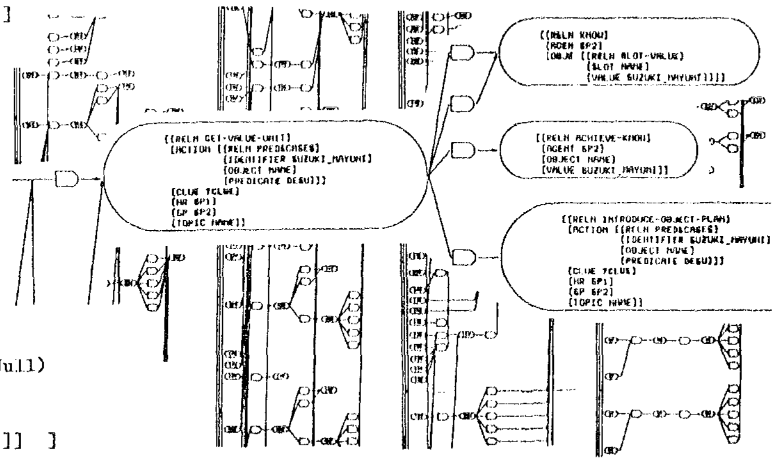

Figure 2

tion is assigned a value. Note that merely because an assertion is believed true in all known consistent possible worlds, this does not mean that it is ACTUAL it might only be POSSIBLE. See [Mye89a] for details.

Plan Schemata. The plan schemata are best explained by an actual example.[2] See Figure 1. Plan schemata are formed from possibly cyclic feature structures and can include variables, co-instance variables and rest variables. A schema has a name, a series of preconditions, a series of decompositions, and a series of effects.

Plan Inference. Plan inference is performed by implementing models for plan recognition, prediction, and inference on top of the inference engine (thus, it also uses feature structures as its main data representation). Recognition is based on *required entailment* and uses a strong model that states that the conjunction of the decompositions plus preconditions implies recognition of the action (in a bottom-up fashion). If a weaker recognition method is desired, the user can specify *sufficiency sets* of particular preconditions and decompositions to recognize the action. (E.g., {prec1, dec2} is sufficient for the previous example.)[3] If the entailment is not *required*, an assumption that the antecedents do in fact imply the recognized action can be incorporated in the conjunction, yielding a conditional recognition.

The model for plan prediction requires that each assertion be duplicated in a parallel top-down network where it is marked as PREDICTED. In this case, a predicted action implies each of the preconditions and decompositions, and an effect implies a predicted action. Plan inference comprises a match between recognized and predicted assertions.

[2] This is an (abridged) plan schema to recognize a short answer interaction set in Japanese, e.g. "Annai-sho wa o-mochi desu ka?" "Iie. Motte-imasen." ("(Do) (you) have the announcement?" "No. (I) don't have." [sic]). Short answers are formed in Japanese by repeating the verb. Although these are semantically and pragmatically well-formed, they cannot be translated literally but must be recognized and transferred, since English forms short answers by repeating the auxiliary ("No. I don't."). The key feature is that the verb is repeated without an object, after an inform-if (yes/no) question.

[3] This is a refinement of Knoblock's necessary and sufficient conditions [Kno88].

Since the inference engine is in turn based on an ATMS, the plan inference method also inherits the capabilities of representing possible (assumed) or actual assertions. The representation carefully distinguishes between actual, possible, predicted, and hypothetical occurrences. Multiple self-consistent possibilities can be represented. This results in inferred plans also being possible or actual.

Naturally, it is possible to implement other models of recognition, prediction, or inference, using the system. The formal philosophical foundations of plan inference are not well understood (e.g., direct, indirect, and interacting causes [Pea88]; or triggering vs. supporting causes), and are being researched. A careful examination of the model specifications shown here reveals that the current system infers plans having monotonic actions (although particular states may be retracted in a nonmonotonic fashion). Nonmonotonic extensions are being investigated.

Execution. There are three kinds of input to the system. First, the plan designer specifies a series of plan schemata. Next, world knowledge assertions and hypothetical utterances, in the form of feature structures, are fed into the system and pre-instantiated. Later, possible (candidate) or actual utterances are input into the system and used for plan inferencing.

The results consist of a network of inferences and recognized plans, represented inside the ATMS. This structured knowledge base can be queried by the transfer or generation systems to provide information to resolve problems. Alternatively, side-effecting demons attached to selected ATMS nodes can process and report plan inference information autonomously. Demons are also used to set and delete processing flags in the ATMS network (e.g., for printing out the results of an inferred plan only once, even though the results are continually true).

Techique. Input plan schemata are fed to the plan-schema interpreter, which breaks them up and creates an *instruction rule* for each precondition or decomposition in the schema. An instruction rule consists of a pattern (with variables) to be recognized as an antecedent, plus instructions (with variables) as the consequent. The instruction rules are fed to the nonmonotonic rewriting system [Has89] and used as rewriting rules.

Preinstantiation assertions are input to the rewriting system. One assertion may match several instruction rule patterns. The recognized consequents consist of instructions with instantiated variables. These are fed to the instruction interpreter, which follows the instructions and instantiates hypothetical nodes and implications in the ATMS corresponding to instantiated components of the plan schema. The resulting action and effect assertions are fed back to the rewriting system as more hypothetical input, to instantiate whole networks bottom-up from single input facts. Since the ATMS uses "uniquification" and never reinstantiates existing assertions, this process eventually terminates.

Run-time utterances are submitted to a feature-structure hash test which checks to see whether they have identically been instantiated in the ATMS before (e.g., hypothetically, etc.). If they have, no further pattern matching is required, and the corresponding node is asserted. Otherwise, the input utterance is submitted to the rewriting system for pattern matching and hypothetical instantiation propagation, as before. Following this, the corresponding node is asserted as possible or actual. Assertion triggers a fast spreading activation in the truth maintenance network (using bitvectors) which maintains all consistent "possible worlds". A demon attached to a particular node fires when that node first becomes possible or actual. Demons can reset themselves by deleting support node flags.

Current Status. NP version 2.0, reported here, has been finished and demonstrated. Currently instantiation is done in a bottom-up fashion. A graphic output program allows display of the ATMS network. Nodes can be moused for input or examination. A browsing editor needs to be improved to allow relevant parts of the network to be examined. The system currently works stand-alone and is ready to be integrated with a parser, a transfer system, and a language generator when they have been finished.

Applications. The plan inference system understands ongoing task-oriented conversations between two people, on the subject of registering for a conference. The system serially processes the utterances, maintaining a representation of the currently believed concepts as the conversation progresses.

Currently, input to the system is a corpus of five conversations (20 utterances each, on average) representing the expected feature-structure output of the ATR parsing system, as generated by the parsing researchers. To date, 53 plan schemata have been written, dealing with conversation opening and closing sequences, "inform-if" (yes/no) questions and answers, short answers to questions, ability utterances, inferred knowledge, wants and intentions, domain plans, idioms, and common-sense knowledge. The system understands portions of all five conversations. Research is ongoing in this area. In addition, a separate set of 46 plans duplicating the "four-layer" recognition model [AI89] in feature structures has been implemented and used to understand an entire surface-speech-act-level input conversation with 20 speech-acts (see Figure 2); 232 feature-structure assertions were instantiated.

Future Work. A manual is being written. The instantiation method must be improved. Representing multiple possible nonmonotonic plans is being researched. It will also be necessary to represent and infer plans containing conditional branches, for information-gathering plans in the sample dialogs[4]. However, the main future research deals with the design of a disambiguation system. Disambiguation is an evidential reasoning problem, and will probably require that a causal reasoning system be built and integrated with the multiple-world ATMS.

Naturally, parallel research in illocutionary and perlocutionary force is also continuing. One of the first things that must be done is to build a module that uses possible plans to resolve zero pronouns. Lack of a fully resolved utterance is hindering plan recognition. Other work involves representing and reasoning with intentions found in a dialog.

Conclusion. A working plan inference system that uses feature structures has been described. The system can represent and reason with assumptions, multiple possible inputs, and multiple possible results. Preinstantiation of hypothetical assertions allows inference by fast spreading activation. The resulting system can be used to infer plans directly from the output of a semantic parser, and provide information directly to a transfer or generation system, as part of an automatic interpretation system.

References

[AI89] Hidekazu Arita and Hitoshi Iida. *Tri-Layered Plan Recognition Model for Dialogue Machine Translation.* Technical Report TR-1-0067, ATR, Kyoto, Japan, 1989. (in Japanese).

[Den87] Daniel C. Dennett. *The Intentional Stance.* The MIT Press, Cambridge, Mass., 1987.

[Has89] Toshiro Hasegawa. *The Feature Structure Rewriting System Manual.* TR-1-0093, ATR, Kyoto, Japan, 1989. (in Japanese).

[KA86] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *5th AAAI'86*, pages 32–37, Philadelphia, PA., 1986.

[Kno88] Craig A. Knoblock. Data-driven plan recognition. March 1988. CS Dept., CMU, Pittsburgh, PA

[Mye88] John K. Myers. The necessity of intentions under fallible execution. 1988. Unpublished manuscript.

[Mye89a] John K. Myers. An assumption-based plan inference system for conversation understanding. In *WGNL Meeting of the IPSJ*, pages 73–80, Okinawa, Japan, June 1989.

[Mye89b] John K. Myers. *The ATMS Manual (Version 1.1).* Tech. Rep. TR-1-0074, ATR, Kyoto, Japan, 1989.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers, Inc., Los Altos, CA., 1988.

[4]E.g., "Do you have the announcement?" "No." "Then I will give you the details:..."