COGNITIVE SCIENCE **18**, 283–323 (1994)

Explaining Emotions

PAUL O'RORKE
University of California, Irvine
ANDREW ORTONY
Northwestern University

Emotions and cognition are inextricably intertwined. Feelings influence thoughts and actions, which in turn can give rise to new emotional reactions. We claim that people infer emotional states in others using commonsense psychological theories of the interactions among emotions, cognition, and action. We present a situation calculus theory of emotion elicitation representing knowledge underlying commonsense causal reasoning involving emotions, and show how the theory can be used to construct explanations for emotional states. The method for constructing explanations is based on the notion of abduction. This method has been implemented in a computer program called AbMaL. The results of computational experiments using AbMaL to construct explanations of examples based on cases taken from a diary study of emotions indicate that the abductive approach to explanatory reasoning about emotions offers significant advantages. We found that the majority of the diary study examples cannot be explained using deduction alone, but they can be explained by making abjuctive inferences. These inferences provide useful information relevant to emotional states.

1. INTRODUCTION

Explaining people's actions often requires reasoning about emotions. This is because experiences frequently give rise to emotional states, which in turn make some actions more likely than others. For example, if we see someone striking another person, we may explain the aggression as being a result of

An earlier abbreviated version of this article appeared in the *Proceedings of the 14th Annual Conference of the Cognitive Science Society.* Paul O'Rorke's work on this project was supported in part by grant number IRI-8813048 from the National Science Foundation. Andrew Ortony was supported in part by grant number IRI-8812699 from the National Science Foundation and in part by Andersen Consulting through Northwestern University's Institute for the Learning Sciences. Terry Turner kindly provided diary study data. Steven Morris, Clark Elliott, Tim Cain, David Aha, Tony Wieser, Stephanie Sage, Patrick Murphy, and Milton Epstein participated in various stages of this work. Anonymous reviewers provided useful criticism and suggestions.

Correspondence and requests for reprints should be sent to Paul O'Rorke, Department of Information and Computer Science, University of California, Irvine, CA 92717-3425.

anger. As well as reasoning about actions induced by emotional states, we can reason about emotional states themselves. In the right context, we might reason that a person was angry because we knew that he or she had been insulted. Explaining emotional states often requires reasoning about the cognitive antecedents of emotions. This article focuses on explanations of this kind.

Although people appear to generate explanations involving emotions effortlessly, the question of how one might compute such explanations remains a difficult open question, just as the more general question of how to automate commonsense reasoning remains open. We present a computational model of the construction of explanations of emotions. The model is comprised of two main components. The first component is a method for constructing explanations. The second component is a situation calculus representation of a theory of emotion elicitation. The representation of emotion-eliciting conditions is inspired by a theory of the cognitive structure of emotions proposed by Ortony, Clore, and Collins (1988). In addition to codifying a set of general rules of emotion elicitation, we have also codified a large collection of cases based on diary study data provided by Turner (1985). We have implemented a computer program that constructs explanations of emotions arising in these scenarios. The program constructs explanations using abduction. We describe the representation in some detail in later sections. In the remainder of the introduction, we provide some background information on the reasoning component, the theory of emotions, and the diary study.

1.1 Abductive Explanation

Our approach to constructing explanations is based on work in artificial intelligence and cognitive science on computational methods employing Charles Sanders Peirces's notion of abduction (Peirce, 1931–1958). Peirce used the term abduction as a name for a particular form of explanatory hypothesis generation. His description was basically:

The surprising fact C is observed;
But if A were true, C would be a matter of course,
hence there is reason to suspect that A is true.

In other words, if there is a causal or logical reason A for C, and C is observed, then one might conjecture that A is true in an effort to explain C.

Since Peirce's original formulation, many variants of this form of reasoning have also come to be referred to as abduction. We focus on a view of abduction advocated by Poole (e.g., Poole, Goebel, & Aleliunas, 1987). In this approach, observations O are explained given some background knowledge expressed as a logical theory T by finding some hypotheses H such that

$$H \wedge T \vdash O.$$

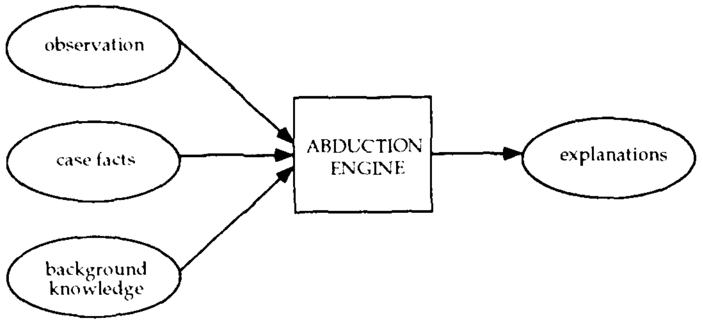

Figure 1. Inputs and Outputs of the Abduction Engine

In other words, if the hypotheses are assumed, the observation follows by way of general laws and other facts given in the background knowledge. Consistency is also desirable so it is usually required that

$$H \wedge T \nvdash \mathit{false}.$$

We use an abduction engine: a computer program that automatically constructs explanations. The explanations of observations are based on general background knowledge and knowledge of particular cases provided to the program in a machine-readable form.

The particular abduction machinery that we use is based on an early approach to mechanizing abduction described in Pople (1973). The abduction engine is part of an explanation-based learning system called AbMaL (Abductive Macro Learner). It is implemented in PROLOG.[1]

An input/output characterization of the program is given in Figure 1. AbMaL takes as input a collection of PROLOG clauses encoding theories. One theory represents background knowledge, another captures the facts of the case at hand. An observation to be explained is given as a query. AbMaL's output includes an explanation of the given observation. AbMaL generates one explanation at a time: It will search for an alternative explanation only if the user rejects the first one found. An explanation can include assumptions made in order to complete the explanation.

In addition to background knowledge, case facts, and a query, AbMaL is also given an operationality criterion and an assumability criterion. The operationality criterion is used to flag queries that should be turned over to the underlying PROLOG interpreter. The intuition is that AbMaL performs explanatory reasoning, whereas the PROLOG interpreter performs lower level reasoning in a more efficient manner without keeping track of explanatory relationships. Separate theories are provided: Explanatory rules

[1] PROLOG was chosen because the basic operation involved in constructing explanations, abductive inference, is related to backward chaining. PROLOG provides basic operations such as unification that are essential parts of backward chaining.

are used to construct explanations and nonexplanatory rules are used for operational inferences. The two types of rules will be distinguished in this article by using the symbol "←" in explanatory rules and the symbol ': -" in nonexplanatory rules (implemented as PROLOG clauses).

The assumability criterion determines whether a hypothesis (a query that could not be proved or disproved) may be assumed. The query may or may not be operational.[2]

A simplified sketch of the procedure followed by the abduction engine is shown in Figure 2.[3] The abduction engine attempts to construct explanations of given observations using general laws and specific facts. In the implementation, explanations are proofs represented as AND trees. Observations to be explained correspond to conclusions of the proofs (roots of the trees). General laws are encoded as rules and these are used to generate the proofs through a process based upon backward chaining.

The mechanization of abduction is comprised of three steps (Figure 2). The first step corresponds to backward chaining as it is implemented in PROLOG interpreters. The observation is treated as a query. Initially, there is only one query but, in general, there may be a number of open questions in the query list Q. The search process attempts to ground the explanation tree in known facts. If a query is operational, AbMaL attempts to identify it with a fact in the database or in its deductive closure. In attempting to prove operational queries, AbMaL does not keep track of an explanation and it does not use "explanatory" clauses. However, it does allow for the possibility that a query may be operational and/or provable in several ways. If one operationalization of the query fails to pan out, backtracking is used to search for another. If the query is not operational, or no direct operational explanation is possible, then explanatory rules may be used to extend the partial explanation, replacing existing queries with new queries. Before queries are allowed to generate new queries in this manner, a test is applied and those deemed inadmissible are disallowed. Several explanatory rules may apply to a single query. In this case, the alternatives are tried if the first rule fails to lead to an explanation.

The second step begins when backward chaining fails. This "identification" or "merging" step is based on the *synthesis* operator advocated by Pople (1973) and justified by him in terms of Occam's razor. In this step, the remaining unexplained queries are examined and some of them are assumed to be "the same" (identical in the sense that they are coreferential

[2] Assumptions involving operational hypotheses are allowed (see the example in Section 3.2).

[3] For more details of the explanation system, see O'Rorke (in press). That article focuses on abduction and explanation-based learning and on lessons learned in case studies involving several other domains.

```
Given: a first-in-first-out queue of queries Q containing observations to be explained;
    Find: explanations for the queries, using abductive inferences.

1. BACKWARD CHAINING:

   While the query list $Q$ is not empty, do:
      (a) Select the first query $q$ and remove it from $Q$.
      (b) If $q$ is operational then compute an answer directly if possible, else
      (c) If $q$ is an admissible goal and is indirectly explainable using a rule,
          then use the rule to generate new queries and add them to $Q$, else
           If $q$ is an admissible hypothesis
             then add $q$ to the list $U$ of unexplained queries,
             else fail and backtrack.

2. IDENTIFICATION/MERGING:

   While there are unifiable pairs of queries in $U$, unify and replace pairs.

3. ASSUMPTION:

   While there are unexplained queries in $U$,
      (a) Select the first query $u$ and remove it from $U$.
      (b) If the truth of $u$ is not known, and it is an admissible hypothesis, and it is ratified by
          the user (optional), then assume $u$, else fail and backtrack.
```

Figure 2. A Procedural Sketch of the Abduction Engine

statements). The idea is that some of the open questions may actually be the same question even though they may have been generated by separate lines of inquiry. Merging them simplifies explanations, reduces the number of assumptions that must be made, and increases the plausibility of the explanation. Another advantage is that identification assumptions often introduce new information. The identification of two previously unrelated statements in different parts of an explanation often causes sharing of information between the separate branches of the explanation. In the implementation, statements are identified by unifying two well-formed formulae. This can cause variables in both formulae to take on new bindings. The new bindings then propagate to other statements that share the same variables.

Merging is implemented using unification. In terms of Figure 2, at the beginning of this stage Q is the empty list, U is a non-nil list of unexplained statements, and the explanation is incomplete. The algorithm continues by first selecting an arbitrary unexplained statement u from U. If u can be identified (unified) with any other statement in U, then the pair is replaced in U with their identification. The identification step ends when no more queries in U are pairwise identifiable. Unlike the previous step, this step is not deductively sound. The assumption that corresponding variables mentioned in merged queries refer to the same individual may be incorrect. When erroneous assumptions of this type are detected at explanation time, they are recoverable through backtracking.

The third abduction step tests whether remaining queries can be assumed. The queries are tested to ensure that they are not known to be true (or false). Nonexplanatory theorem proving is allowed in testing whether a hypothesis is known to be true. AbMaL calls PROLOG and if the hypothesis is proven true, then it is not allowed as an assumption. A test against stored negative assertions is used to determine whether a hypothesis is false (we do not use negation as failure). This test is a limited form of the consistency check called for in the formal specification of abduction. Together, these two tests ensure that a hypothesis is not known true or false. Next, an "assumability" test is used to decide whether to assume that a hypothesis is true. The test includes a domain-independent component and a hook that takes advantage of domain-dependent information about admissible hypotheses. This test is applied to each of the queries u in list U. If u is not assumable, then the current attempt to find an explanation is aborted and backtracking is invoked[4] in order to continue the search for acceptable explanations.

Like all methods for constructing explanations, the one just described can spend substantial time searching large spaces of potential explanations. The amount of computation required depends on the domain, the task, and the specific problem at hand. One aspect of the method that favors efficiency

[4] The most recent choice that may be changed is the choice of goals merged in "identication."

is that it does not attempt to compute multiple candidates simultaneously. It does not try to find all possible explanations, it only tries for one. Even so, the depth-first search tends to run away. We keep this tendency in check using depth bounds. We also use other forms of search control that take advantage of both general and domain-specific information. Tests for admissibility are applied to reject inadmissible queries and hypotheses arising in partial explanations. With these constraints on search, the algorithm finds explanations for most examples in a few seconds.[5]

1.2 A Theory of Emotions

The theory of the cognitive structure of emotions employed in the research we describe views emotions as valenced reactions to events, agents and their actions, and objects. It specifies a total of 22 emotion types summarized in an abbreviated form in Table 1. We provide only a brief sketch here. A full description can be found in Ortony, Clore, and Collins (1988).

The emotion types are essentially just classes of eliciting conditions, but each emotion type is labeled with a word or phrase, generally an English emotion word corresponding to a relatively neutral example of an emotion fitting the type. The simplest emotions are the well-being emotions *joy* and *distress*. These are an individual's positive and negative reactions to desirable or undesirable events.

The fortunes-of-others group covers four emotion types: *happy-for, gloating, resentment,* and *sorry-for.* Each type in this group is a combination of pleasure or displeasure over an event further specialized as being presumed to be desirable or undesirable for another person.

The prospect-based group includes six emotion types: *hope, satisfaction, relief, fear, fears-confirmed,* and *disappointment.* Each type is a reaction to a desirable or undesirable event that is still pending or that has been confirmed or disconfirmed.

The attribution group covers four types: *pride, admiration, shame,* and *reproach.* Each attribution emotion type is a (positive or negative) reaction to either one's own or another's action.

The attraction group is a structureless group of reactions to objects. The two emotions in this group are the momentary feelings (as opposed to stable dispositions) of liking or disliking.

The final group is comprised of four compounds of well being $\times$ attribution emotion types. These compound emotions do not correspond to the co-occurrence of their component emotions. Rather, each compound's eliciting conditions are the union of the component's eliciting conditions. For example, the eliciting conditions for anger combine the eliciting conditions for reproach with those for distress.

[5] The examples run in LPA MacPROLOG on a MacIIci.

TABLE 1
Emotion Types

Group	Specification	Types (name)
Well-being	Appraisal of event	pleased (joy) displeased (distress)
Fortunes-of-others	Presumed value of an event affecting another	pleased about an event desirable for another (happy-for) pleased about an event undesirable for another (gloating) displeased about an event desirable for another (resentment) displeased about an event undesirable for another (sorry for)
Prospect-based	Appraisal of a prospective event	pleased about a prospective desirable event (hope) pleased about a confirmed desirable event (satisfaction) pleased about a disconfirmed undesirable event (relief) displeased about a prospective undesirable event (fear) displeased about a confirmed undesirable event (fears-confirmed) displeased aobut a disconfirmed desirable event (disappointment)
Attribution	Appraisal of an agent's action	approving of one's own action (pride) approving of another's action (admiration) disapproving of one's own action (shame) disapproving of another's action (reproach)
Attraction	Appraisal of an object	liking an appealing object (love) disliking an unappealing object (hate)
Well-being/ Attribution	Compound emotions	admiration + joy→gratitude reproach + distress→anger pride + joy→gratification shame + distress→remorse

In general, eliciting conditions are specified in terms of variables that contribute toward increasing the intensity of emotions. The theory specifies global variables that affect all emotions, and local variables that affect subsets of emotions. The variables have values and weights associated with them, and the theory claims that an emotion is experienced only when certain levels, the emotion thresholds, are exceeded.

For anger, the variables affecting intensity are:

- the degree of judged blameworthiness;
- the degree of deviation from personal or role-based expectations, and
- the degree to which the event is undesirable.

The first variable, blameworthiness, is the evaluation of an action against the standards of the judger. The second variable, deviations from expectations, gauges the extent to which the action is unexpected of the agent. The third variable reflects an evaluation of the event (perpetrated by the agent) in terms of its impact upon personal goals.

1.3 A Diary Study of Emotions

We use data taken from a diary study of emotions (Turner, 1985) as the source of examples. Most of the subjects who participated in the study were sophomores at the University of Illinois at Champaign-Urbana. They were asked to describe emotional experiences that occurred within the previous 24 hours. They typed answers to a computerized questionnaire containing questions about which emotion they felt, the event giving rise to the emotion, the people involved, the goals affected, and so on. Over 1,000 descriptions of emotion episodes were collected, compiled, and recorded on magnetic media.

We chose to use this diary study as a source of examples because, although nearly every emotion type is represented, the situations and events described in the entries tend to cluster into a relatively small number of stereotypical scenarios. This is a natural consequence of the importance of examinations, dating, and so on, in the emotional lives of undergraduate students. We were thus able to focus on aspects of the theory and computational model most relevant to emotions, rather than being distracted by problems having to do with representing a wide range of domain-specific knowledge.

2. A SITUATION CALCULUS THEORY OF EMOTION ELICITATION

In this section, we describe a representation language designed to support the construction of explanations involving emotions. The language is based on work on two major contributions to knowledge representation, the situation calculus (McCarthy, 1968) and conceptual dependency (Schank, 1972).

Before we proceed, a few words on methodology may be in order. We use logic-based methods for reasoning and for representing knowledge in this article. There is some controversy over methodological issues associated with whether and how to use logic in artificial intelligence. The logicist approach is presented in Genesereth and Nilsson (1987) and the use of logic in representing commonsense knowledge is discussed in Davis (1990). Good examples of debates about the role of logic are McDermott (1987), Nilsson

(1991), and Birnbaum (1991). We use logic in this article in an attempt to make the presentation of the relevant commonsense knowledge and inference techniques clear, complete, and comprehensible. However, the use of logic-based representations and reasoning methods in this article does not represent a commitment on our part to logicism. Charniak's (1987) quip also applies to us: we are not now, nor have we ever been, logicists.

2.1 Situation Calculus

The situation calculus provides us with a language for expressing causal laws relating actions and physical situations. This first-order logical language was originally invented by McCarthy (1968, 1977). We employ a version of the situation calculus incorporating improvements by Green (1969), McCarthy and Hayes (1969), Fikes and Nilsson (1971), Kowalski (1979), and Reiter (1991).

Situations are represented by terms. Fluents are statements that may or may not be true in a given situation. The negation of a fluent F is (also) a fluent. Partial descriptions of the state of affairs in a given situation S state that fluents such as P hold in S: *holds*(P, S).

Actions are functions that map situations representing the state of the world before the execution of the action into situations representing the state of the world afterward. The situation resulting from applying action A to state S is designated by the term *do*(A, S). We treat the negation of an action A the same as the action of not executing A.

A number of examples that we have studied suggest that people do not make a strong distinction between actions and fluents in the sense that they often want an action to be done without focusing on any explicit effect caused by the action. Actions that are done for their own sake because they are intrinsically enjoyable rather than to achieve other goals are good examples (e.g., chatting on the phone, skiing, watching one's favorite sports team). In response to this observation, we introduced a function *did* that maps actions to fluents and we added a causal law relating corresponding actions and fluents:

causes $(A, did(A), S)$.

The intuition behind this fluent for actions is that, if nothing else, doing an action at least causes it to be done. [If action A is executed in situation S, then it causes the fluent *did(A)* to be true in the resulting situation.] This law allows us to refer to actions through fluents and not just as mappings between situations. This is useful for actions that are done for their own sake but more generally, the *did* fluent is useful whenever we do not wish to focus on a specific effect of an action but rather on the action itself. The importance of this extension in reasoning about emotions is discussed in Section 4.3.

Actions are defined by specifying their preconditions and effects. Preconditions are divided into *action* and *fluent* preconditions following Reiter

(1991). Action preconditions are fluents that imply that the action is possible in a given situation. Fluent preconditions imply that individual effects follow upon execution of an action.

The fact that an action A is possible in a situation S is represented as $poss(A, S)$. If P_1 and P_n are action preconditions for doing A, this is represented:

$$poss(A, S) \leftarrow holds(P_1, S) \wedge \ldots \wedge holds(P_n, S).$$

The effects of actions are represented using something like the "add" and "delete" statements of STRIPS (Fikes & Nilsson, 1971). These statements specify fluents added or deleted upon execution of an action. Both positive and negative effects are encoded in conditional "cause" statements of the form:

$$causes(A, F, S) \leftarrow holds(F_1, S) \wedge \ldots \wedge holds(F_n, S).$$

where each F_i is a fluent condition for action A causing fluent F in situation S. Positive and negative effects are inferred through the following law of direct effects:

$$holds(F, do(A, S)) \leftarrow causes(A, F, S) \wedge poss(A, S).$$

This law states that a fluent holds in the situation resulting from the execution of an action if the action was possible to begin with and if the action causes the fluent.

Our "causes" statements are similar to the "causal associations" of Peng and Reggia (1990) as opposed to their "causation events." A causal association specifies a possible causal relationship whereas a causation event is said to hold iff both the cause and the effect hold and the effect is *actually caused* by the cause. Our statements and rules involving "causes" only capture associations between actions and their effects and the fluent preconditions under which the action, if executed, would lead to the effects. The action preconditions still must be satisfied in order for the action to be possible and actually cause the relevant effect.

Frame axioms state that nothing changes unless it is explicitly stated that it changes as a result of some action. We use the following frame axiom schema:

$$holds(P, do(A, S)) \leftarrow \overline{causes(A, \overline{P}, S)} \wedge holds(P, S) \wedge poss(A, S).$$

This frame axiom states that the fluent P will hold after execution of an action if it held before and the action did not cancel it.[6]

[6] In our implementation, queries of the form *not P* are considered to be operational, so the explanatory machinery turns *not(causes (A, not P, S))* over to PROLOG, which attempts to show that *causes(A, not P, S)*. If the attempt to prove this goal fails, negation as failure is used to "prove" the negation. Attempts to prove *causes(A, not P, S)* can match this literal against facts and rules about causes.

The advantage of the frame axiom we have adapted is that there is no need to have a separate frame axiom for every relation. Instead one only needs a single frame axiom. Kowalski (1979) pointed out that earlier versions of frame axioms can be had by forming macros from the very general frame axiom and specific statements about what is deleted.[7]

We provided the operational metalevel predicates *agent, precedes,* and *precondition.* They express important constraints and help to control inference. The *agent* predicate is used to identify or extract the agent of a given action. The *precedes* predicate applies to two arguments:

$$precedes[S,\ do(A,\ S)].$$

The situation S precedes the situation resulting from the execution of action A in situation S. The situation S_0 precedes the result of doing A in S if it precedes S:

$$precedes[S_0,\ do(A,\ S)]\ {:}{-}precedes(S_0,\ S).$$

These rules give sufficient conditions for one situation to precede another.[8] The *precondition* predicate applies to two arguments, an action and a fluent:

$$precondition(A,\ C).$$

This statement is true if C is an action precondition of A. To determine whether it is true, domain-level rules about when actions are possible are consulted. To be exact, *precondition*(A, C) is true if the system finds an explanatory clause with a conclusion of the form *poss*(A, S) and a condition of the form *holds*(C, S).

The following general laws facilitate reasoning about goals:

$$wants[P,\ did(A),\ S] \leftarrow causes(A,\ F,\ S) \wedge diff[F,\ did(A)] \wedge wants(P,\ F,\ S);$$

$$wants(P,\ F,\ S) \leftarrow precondition(A,\ F) \wedge causes[A,\ G,\ S) \wedge wants(P,\ G,\ S).$$

The first rule states that a person may want an action to be done if some effect caused by the action is desired. Note the use of the operational metapredicate *diff,* which ensures that the effect of the action A is different from *did*(A). (This is needed here in order to avoid useless recursion.) The second statement allows for the fact that an action may be directed at satisfying goals that contribute to the eventual achievement of longer term goals. In particular, a person may want a fluent to hold if it is a precondition of an action that causes another desired fluent.

[7] Interestingly, this can be done by explanation-based learning (DeJong & Mooney, 1986; Mitchell, Keller, & Kedar-Cabelli, 1986).

[8] This amounts to something like an induction schema for the *precedes* relation. Because assumptions about this relation are disallowed (see Section 2.4), negation as failure ensures that only the instances covered by these rules are allowed.

2.2 Conceptual Dependency

Primitive actions provided by conceptual dependency (*ptrans, move, atrans, propel, grasp, ingest, expel, mtrans,* and *attend*) are encoded in our situation calculus representation as functions mapping situations and CD roles such as "agent" into new situations. Variants of the functional representations are used when the value of some argument is unknown or unimportant.

For example, the function *atrans* is used to represent transfers of ownership. The most explicit form of *atrans* has arguments for the agent responsible for the transaction, the object in question, the new owner, and the previous owner. In many cases, the previous owner is the agent. We use a three-argument version of *atrans* in such cases. A two-argument version is used when the agent is the recipient and the previous owner is irrelevant.

We show how knowledge about actions is encoded using an example of *ptrans.* It is possible for an agent *P* to move an object *T* from one location *From* to a destination *To* if the thing *T* is initially *at* the location *From:*

$$poss(ptrans(P,\ To,\ From,\ T),\ S) \leftarrow holds(at(T,\ From),\ S).$$

This is an example of an action precondition. The effects of *ptrans* are encoded as follows. A physical transfer of a thing *T* to a destination *To* causes the thing to be *at* the destination:

$$causes(ptrans(P,\ To,\ From,\ T),\ at(T,\ To),\ S).$$

This is a positive effect of the transfer. A negative effect is that the thing is no longer at its original location:

$$causes(ptrans(P,\ To,\ From,\ T),\ \overline{at(T,\ From)},\ S)\ {:}{-}diff(To,\ From).$$

Whereas the positive effect is unconditional, the negative effect has a fluent precondition in this formulation of *ptrans*. The operational metapredicate *diff* is used to ensure that the destination and origin are different locations. This predicate uses PROLOG to ensure that its two arguments cannot be unified. Note that the predicate "causes" should be interpreted with care (see the discussion of "causal associations" vs. "causation events" in Section 2.1). For example, *causes*[*ptrans*(*P, D, F, T*), *at*(*T, D*), *S*] may be true even when *holds*[*at*(*T, F*), *S*], and *poss*[*ptrans*(*P, D, F, T*), *S*] are false so that *do*[*ptrans*(*P, D, F, T*), *S*] is impossible.

Additional actions required by the examples that we have encoded include *abuse, attack, breakup, call, close, die, excel, fight, gossip, hurt, insult, kill, open,* and *score.* Preconditions and effects of actions are encoded using fluents such as *alive, dead, did, have, rested, single,* and *unfaithful.* These actions and their preconditions and effects are represented using general laws similar to those shown previously in the example of *ptrans.*

2.3 Emotion-Eliciting Conditions

Eliciting conditions for emotions are encoded in a collection of rules for all emotion types except likes and dislikes. The rules are an initial attempt to represent the emotion-eliciting conditions proposed by Ortony et al. (1988), and sketched in Section 1.2. The elicitation rules correspond to explanatory rules in the computational implementation.[9] Simplifying assumptions and limitations of this initial representation are discussed in Section 4.4. We present the rules in pairs corresponding to complementary emotions.

People may experience joy over a fluent[10] in a situation if they want it and it holds, but they may experience distress if they want a fluent that holds not to hold:

$$joy(P, F, S) \leftarrow wants(P, F, S) \wedge holds(F, S);$$

$$distress(P, F, S) \leftarrow wants(P, \overline{F}, S) \wedge holds(F, S).$$

A person may experience neither joy nor distress in the event that a fluent holds, if he or she desires neither the fluent nor its negation. Even if we grant the law of the excluded middle for a fluent, it is still possible that a person is indifferent to it.

A person may be happy for another if he or she experiences joy over a fluent presumed to be desirable for the other. We express this in terms of joy over the other's joy:

$$happy\_for(P_1, P_2, F, S) \leftarrow joy[P_1, joy(P_2, F, S_0), S].$$

Note that the desire for the fluent is implicit in the embedded joy. Although the rule does not encode the fact that a person is usually happy for another before or while the other is happy, the rule does reflect the fact that they may be happy in different situations (at different times).

A person may be sorry for another if he or she experiences distress over a fluent presumably undesirable for the other. We express this in terms of distress over the other's distress:

$$sorry\_for(P_1, P_2, F, S) \leftarrow distress[P_1, distress(P_2, F, S_0), S].$$

The undesirability of the fluent is implicit in the embedded distress. The two people may be distressed in different situations and no temporal constraints are placed on these situations in the present formalization.

[9] For the distinction between explanatory and nonexplanatory rules, see Section 1.1.

[10] Note that in this formalization of the theory sketched in Table 2 fluents are used in place of events. In many cases, emotional reactions to events are actual reactions to effects of the events, rather than to the event itself. Even when the focus is on an event, we can cover this case by representing the fact that the event occurred as a fluent. For example, in the case of actions we use fluents of the form *did(action)*.

A person may gloat over a fluent that gives them joy that (they believe) is not wanted by another. We express this in terms of joy over another's distress:

$$gloats(P_1, P_2, F, S) \leftarrow joy[P_1, distress(P_2, F, S_0), S].$$

People may resent another person if they are distressed about an event that pleases the other person. We express this in terms of distress over another's joy:

$$resents(P_1, P_2, F, S) \leftarrow distress[P_1, joy(P_2, F, S_0), S].$$

Again, the desirability of the event for the other is implicit in the embedded distress and joy and we currently do not require any particular temporal order for the relevant situations.

The *hopes* rule captures the idea that people may experience hope if they want a fluent and anticipate it:

$$hopes(P, F, S) \leftarrow wants(P, F, S) \wedge anticipates(P, F, S).$$

People may experience fear if they want an anticipated fluent not to obtain:

$$fears(P, F, S) \leftarrow wants(P, \overline{F}, S) \wedge anticipates(P, F, S).$$

These rules allow for hopes and fears even if, unknown to the person, the hoped-for or feared fluent in fact already holds.

Although many examples of hopes and fears involve expectations, we use the predicate *anticipates* in order to suggest the notion of "entertaining the prospect of" a state of affairs. The purpose of this is to avoid suggesting that hoped-for and feared events necessarily have a high subjective probability. We also want to avoid suggesting that they always occur in the future.

Satisfaction occurs when a hoped-for fluent holds:

$$satisfied(P, F, S) \leftarrow precedes(S_0\ S) \wedge hopes(P, F, S_0) \wedge holds(F, S).$$

Fears are confirmed when feared fluents hold:

$$fears\_confirmed(P, F, S) \leftarrow precedes(S_0\ S) \wedge fears(P, F, S_0) \wedge holds(F, S).$$

We require the fear to precede its confirmation and we expect the hope to occur before it is satisfied.

Relief may be experienced when the negation of a feared fluent holds:

$$relieved(P, \overline{F}, S) \leftarrow precedes(S_0\ S) \wedge fears(P, F, S_0) \wedge holds(\overline{F}, S);$$

$$relieved(P, \overline{F}, S) \leftarrow fears(P, F, S_0) \wedge holds(\overline{F}, S).$$

We give two rules for relief because fear usually occurs before the fluent holds, but sometimes relief occurs in the absence of prior fear (as when a person discovers that a missed flight crashed).

Disappointment occurs when the negation of a hoped-for fluent holds:

$$disappointed(P, \overline{F}, S) \leftarrow precedes(S_0, S) \wedge hopes(P, F, S_0) \wedge holds(\overline{F}, S).$$

$$disappointed(P, \overline{F}, S) \leftarrow hopes(P, F, S_0) \wedge holds(\overline{F}, S).$$

The fluent is usually hoped-for in a situation that occurs in advance of the present situation but disappointment (e.g., at a missed opportunity) may occur in the absence of prior hope.

Pride and shame can occur for individuals or groups. An agent experiences pride over praiseworthy actions executed either by the agent or by another member of a "cognitive unit" (Heider, 1958) containing the agent. Agents may experience shame if they do a blameworthy act or if another member of their cognitive unit does a blameworthy act:

$$proud(P, A, S) \leftarrow agent(A, P) \wedge holds[did(A), S] \wedge praiseworthy(A);$$

$$proud(P_1, P_2, A, S) \leftarrow agent(A, P_2) \wedge holds[did(A), S] \wedge praiseworthy(A) \wedge cognitive\_unit(P_1, P_2);$$

$$shame(P, A, S) \leftarrow agent(A, P) \wedge holds[did(A), S] \wedge blameworthy(A);$$

$$shame(P_1, P_2, A, S) \leftarrow agent(A, P_2) \wedge holds[did(A), S] \wedge blameworthy(A) \wedge cognitive\_unit(P_1, P_2).$$

The predicates *praiseworthy* and *blameworthy* are intended to reflect personal standards rather than normative or social standards, except insofar as the judging person subscribes to such standards.

A person may admire another person if the other person does something praiseworthy, but a person may feel reproach toward another if the other does something blameworthy:

$$admire(P_1, P_2, A, S) \leftarrow agent(A, P_2) \wedge holds[did(A), S] \wedge praiseworthy(A);$$

$$reproach(P_1, P_2, A, S) \leftarrow agent(A, P_2) \wedge holds[did(A), S] \wedge blameworthy(A).$$

Compound emotion types are comprised of the eliciting conditions of components taken from the well-being and attribution groups. We do not include the component emotions in the eliciting conditions in order to avoid suggesting that the component emotions are necessarily felt as part of feeling the compound. Instead, we collect the eliciting conditions of the components and simplify them, eliminating redundancies.

Gratitude is a compound of the eliciting conditions of joy and admiration. A person may be grateful toward another person if the other person does a praiseworthy action that causes a desirable fluent to hold:

$$grateful(P_1, P_2, A, S_1) \leftarrow agent(A, P_2) \wedge holds[did(A), S_1] \wedge precedes(S_0, S_1) \wedge causes(A, F, S_0) \wedge praiseworthy(A) \wedge wants(P_1, F, S_1) \wedge holds(F, S_1).$$

The *angry_at* emotion type focuses on anger at other agents. It is a compound comprised of the eliciting conditions of reproach and distress. A person may be angry at another if an undesirable fluent is caused by a blameworthy action taken by the other person:

$$angry\_at(P_1, P_2, A, S1) \leftarrow agent(A, P_2) \wedge holds[did(A), S_1] \wedge precedes(S_0, S_1) \wedge causes(A, F, S_0) \wedge blameworthy(A) \wedge wants(P_1, \overline{F}, S_1) \wedge holds(F, S_1).$$

We distinguish this from angry about, which because it focuses on the undesirability of the situation, is better thought of as a special case of *distress:* frustration (typically at goal blockage).

Gratification is a compound emotion comprised of the eliciting conditions of pride and joy. A person may be gratified if he or she does a praiseworthy action that results in a desirable fluent:

$$gratified(P, A, S_1) \leftarrow agent(A, P_2) \wedge holds[did(A), S_1] \wedge precedes(S_0, S_1) \wedge causes(A, F, S_0) \wedge wants(P, F, S_1) \wedge holds(F, S_1) \wedge praiseworthy(A);$$

$$\text{gratified}(P_1, P_2, A, S_1) \leftarrow agent(A, P_2) \wedge holds[did(A), S_1] \wedge precedes(S_0, S_1) \wedge causes(A, F, S_0) \wedge cognitive\_unit(P_1, P_2) \wedge wants(P_1, F, S_1) \wedge holds(F, S_1) \wedge praiseworthy(A).$$

Because there is a cognitive unit variant of pride, there is also a variant of gratification. This variant of *gratified* is closely related to *grateful.*

Remorse is a compound emotion comprised of the eliciting conditions of shame and distress. People may be remorseful if they do a blameworthy action that results in an undesirable fluent:

$$remorseful(P, A, S_1) \leftarrow agent(A, P_2) \wedge holds[did(A), S_1] \wedge precedes(S_0, S_1) \wedge causes(A, F, S_0) \wedge wants(P_1, \overline{F}, S_1) \wedge holds(F, S_1) \wedge blameworthy(A);$$

$$remorseful(P_1, P_2, A, S_1) \leftarrow agent(A, P_2) \wedge holds[did(A), S_1] \wedge precedes(S_0, S_1) \wedge causes(A, F, S_0) \wedge cognitive\_unit(P_1, P_2) \wedge wants(P_1, \overline{F}, S_1) \wedge holds(F, S_1) \wedge blameworthy(A);$$

The second rule here provides the eliciting conditions of a cognitive-unit variant of *remorseful.* These conditions are derived from the corresponding variant of *shame.* This variant is closely related to *angry_at.*

The eliciting conditions given in this section represent a first attempt at formalizing necessary, but not necessarily sufficient, conditions for the elicitation of the corresponding emotions. (This is why we say "a person *may* experience emotion *x* under conditions *y*.") In some cases, additional conditions may be required to capture more fully commonsense knowledge about emotion elicitation. It may be that a person's disposition towards another person should play a role in explaining the elicitation of "fortunes of others" emotions. For example, perhaps the eliciting conditions of *happy_for* should include a requirement that a person may be happy for another if that person does not dislike the other.

Cognitive units play an important role in some emotions and are provided for in the situation calculus theory of emotions. A special predicate is used for cognitive units in the eliciting conditions of the *attribution* emotions *pride* and *shame.* This predicate is also used in the given "background knowledge" to encode groups that may form cognitive units. This is an attempt to capture the idea that people can form (relatively stable) cognitive units with their family members and close friends.

Studies of a number of examples suggest that many goals are shared by members of the same cognitive unit. People want good things to happen not just to themselves but also to others in their cognitive unit. They want to avoid bad things and they do not want bad things to happen to others in their cognitive unit. For example, everyone wants to excel, and they want people in their cognitive unit to excel, too. People generally do not want to experience harm, and they do not want other members of their group to be harmed either. This sort of general law is represented using conditional *wants:*

$$wants[P, \overline{harmed(Q)}] \leftarrow cognitive\_unit(P, Q).$$

2.4 Finding Plausible Explanations of Emotions Efficiently

In this section, we describe three additional sources of knowledge that contribute to efficiency. The first source, the assumability criterion, specifies which queries are admissible as hypotheses. The second source, the operationality criterion, specifies which queries can be answered without explanation. The third source, a set of rewrite rules, specifies transformations that map alternative representations into a canonical form. In addition to improving efficiency, these three knowledge sources also help limit the search for explanations for emotions to plausible candidates.

In general, many explanations are possible and it is important to constrain the search to avoid large numbers of implausible hypotheses and explanations. In early experiments without constraints we found that the abduction engine conjectured large numbers of implausible causal relationships. This problem was addressed by disallowing assumptions involving metapredicates like *diff* and instances of the following:

$$preconditions(A, F);$$

$$causes(A, F, S).$$

In other words, the abduction engine was not allowed to assume arbitrary preconditions for actions, nor was it allowed to assume unprovable causal relationships between actions and effects.

The operationality criterion provides a second source of knowledge that enables efficient recognition of the truth or falsity of queries. Operational queries can be answered relatively efficiently because they do not involve the additional overhead associated with constructing explanations. Such queries are turned over to the base-level interpreter (PROLOG). In most cases, they are answered by simply attempting to find a matching statement in the database, although in general backward chaining is allowed. The following predicates are considered to be operational:

$diff(X, Y)$;
$member(X, Y)$;
$opposite(X, Y)$'
$action(X)$;
$precondition(A, C)$;
$agent(A, P)$;
$precedes(S_1, S_2)$;
$cognitive\_unit(P, Q)$;
$d\_likes(P, Q)$.

We assume that no explanatory reasoning is involved in answering these queries. For example, the dispositional attitude *d_likes* is operational because we assume that likes and dislikes are not explainable. In addition to these predicates, simple queries present in the database as facts are also considered to be operational.

Rewrite rules enable the system to recognize alternative ways of representing the same expression. For example, since the negation of doing an action is considered to be identical to the execution of the negation of that action, the term $\overline{do(A)}$ is considered to be an alias for $do(\overline{A})$. At key points in the computation, the system uses rewrite rules to transform such expressions into a canonical form. Note that this use of canonical forms does not put us in danger of subscribing to Woods's (1975) "canonical form myth." Woods pointed out that it is provably impossible to find canonical form functions for many formal systems and that such functions should not be expected to map all internal representations of sentences with the same "meaning" into a single canonical form. But our rewrite rules do not carry much inferential burden; instead, they provide representational flexibility.

This allows us to use two ways of associating fluents and situations. One advantage of the *holds* relation introduced by Kowalski (1979) is that it avoids the need for an extra state parameter for all relations. The disadvantage of the use of *holds* (as opposed to including extra arguments for situations) is that it requires an extra predicate in the sense that it requires us to embed fluents in *holds* statements. Besides being aesthetically undesirable in some situations, the use of *holds* can increase the branching factor of some explanatory searches because facts and clauses tend to be indexed and fetched by the top-level predicate. Sometimes it is more convenient to use situations as arguments of fluents. In other cases, it is preferable to associate fluents with situations using the *holds* predicate. The rewrite rules listed here facilitate explanations involving chains of fluents by linking equivalent representations:

$$holds[F(Args), S] \Rightarrow F(Args, S);$$
$$holds\{holds[F(Args), S_1], S_2\} \Rightarrow holds[F(Args), S_1];$$
$$holds[F(Args, S_1), S_2] \Rightarrow F(Args, S_1).$$

The first rule establishes the equivalence between a fluent with a situation as an argument and the corresponding situationless fluent holding in the same situation. The second rule "unwraps" embedded *holds* statements. The truth of a *holds* statement depends only on the situation it applies to. The third rule is a consequence of the first two.

In the emotion-eliciting conditions, the emotion types occurring in the heads of the rules are expressed as fluents with situational arguments. The bodies of the rules contain *holds* statements whose arguments are fluents eliciting emotional reactions. It is important to ensure that the alternative representations are viewed as equivalent so that inferences are not lost. We enforce the equivalence by mapping complex representations to relatively simple canonical forms using rewrite rules. This facilitates explanations involving emotional chains (as shown in Sections 3.2 and 4.3).

3. EXPLAINING EMOTIONS

In this section, we show how our situation calculus representation of emotion elicitation can be used to explain emotional states. We describe the process of codifying examples. We show how explanations are produced by the abduction engine sketched earlier. We provide assumptions and explanations produced by the computer program and show how its outputs are interpreted. The examples discussed in this section were chosen to show some of the breadth of the reasoning and representation methods. In a later section, we will use these examples to illustrate strengths and weaknesses of the methods.

3.1 An Explanation for Joy

The first example is based on the following "case," a simplified version of a scenario taken from Turner's (1985) diary study:

Mary went home to see her family.
She ate a home-cooked meal.
She wanted to stay home.
Mary was happy.

We hand-coded the case into the inputs shown in Figure 3. The first input states that Mary wants to be home in the situation that follows after she went home and ingested a home-cooked meal. Abbreviations at the bottom of the table are used for the relevant situations. Note that this codification of the example is crude in the sense that we have not atempted to capture

```
Case Facts
    wants(mary, at(mary, home), s2).
Query
    why(joy(mary, _, s2))
Explanation
    joy(mary, at(mary, home), s2)
        wants(mary, at(mary, home), s2)
        holds(at(mary, home), s2)
            not causes(ingest(mary, home_cooked_meal), not at(mary, home), s1)
            holds(at(mary, home), s1)
                causes(ptrans(mary, home), at(mary, home), s0)
                poss(ptrans(mary, home), s0)
            poss(ingest(mary, home_cooked_meal), s1)
                ┌─────────────────────────────────────────┐
                │holds(have(mary, home_cooked_meal), s1)│
                └─────────────────────────────────────────┘
Abbreviations
    s1=do(ptrans(mary, home), s0)
    s2=do(ingest(mary, home_cooked_meal), s1)
```

Figure 3. An Explanation for Joy

much of the information associated with Mary visiting her family. The underscores and the use of a constant for "home_cooked_meal" also signify simplification aimed at avoiding having to deal with issues related to quality of food and different methods of food preparation. Such subtleties are lost. We strive only to capture basic facts of the case. The case specifies that Mary is happy in a situation following certain actions. It specifies that she wants to be at home in this situation but it does not specify the fluent that she is happy about. In the query about why Mary is experiencing joy, the situation is specified but the fluent is left blank (using a "don't care" variable designated in PROLOG as an underscore "_").

Figure 3 also shows an explanation produced by the abduction engine. The explanation is in the form of a tree. The first line is the root of the tree; indented lines are branches. The first level of indentation shows the propositions immediately supporting the root. The second level of indentation shows their supporters, and so on. The deepest levels of indentation correspond to leaves of branches of the explanation tree.

This explanation for Mary's happiness is interpreted as follows. In the first line, we see that the fluent she is happy about has been identified as the fact that she is at home in the given situation. Mary's location prior to going home was not specified in the given case fact and neither is it determined during the construction of the explanation. Her joy is supported by the second line, which was part of the input. A case fact stating that Mary wanted to be at home was given. The fat that she is at home is explained by the remainder of the tree. Mary is a home because she went there and the fact that she ingested a meal did nothing to cancel this result. The explanation that Mary is at home after she ate the meal rests on an assumption that it was possible to eat because she had it after she went home. Abductive inferences (assumptions) are distinguished from leaves of explanation trees that are known to be true by enclosing them in boxes as in Figure 3.

The explanation was constructed by the abduction engine using the situation calculus theory of emotion elicitation described earlier. The system is not allowed to explain the initial query (why was Mary happy?) directly, even though it is a known fact. Instead, it finds reasons by backward chaining on rules of situation calculus and emotion elicitation rules. In this case, the rule specifying the eliciting conditions for joy applied. Backward chaining on this rule generated two new queries: Does Mary want something—something that holds in the given situation? A given case fact stated that Mary wants to be at home. This fact was used to "ground out" one of the queries. The query about how the desired state of affairs came to pass (how Mary came to be at home) was answered by backward chaining on a law of situation calculus. The frame axiom stating that effects of earlier actions persist unless canceled by later actions was applied to infer that Mary was at home because she went there earlier and nothing she did in the meantime canceled this effect. Most of the remaining queries grounded out in known facts provided as facts of situation calculus and conceptual dependency (e.g., Mary was at home because she did a *ptrans* and physically transferred herself home). But an assumption was made in order to complete the explanation. An action precondition of *ingest* states that it is possible to ingest something if one has it first. It was not stated in the input whether Mary had possession of the meal, but the abduction engine assumed that she did.[11]

3.2 An Explanation for Happy_For

This example illustrates the fortunes-of-others emotion, *happy_for:*

Mary's roommate is going to Europe.
Mary is happy for her.

Here, Mary's roommate is going to Europe and thus Mary is happy for her. The explanation constructed by the abduction engine is shown in Figure 4. The first reason translates the emotion to be explained into joy over another's joy. The explanation states that Mary is happy for her because she will be happy in Europe and Mary wants her roommate to be happy because she likes her. In the construction of this part of the explanation, a new query is generated in an effort to explain Mary's roommate's joyful reaction to being in Europe. The query is initially in the form:

holds(*joy*(*roommate*(*mary*), *at*(*roommate*(*mary*), *europe*), *s*2), *s*1);

but this is immediately simplified, using rewrite rules mapping fluents into a canonical form (Section 2.4), to strip off a superfluous *holds* predicate and an unnecessary situation *s*1:

joy(*roommate*(*mary*), *at*(*roommate*(*mary*), *europe*), *s*2).

This simplified version of the query invokes the emotion eliciting rule for *joy* producing the explanation shown.

Note that the predicate *d_likes* is to be interpreted as dispositional liking as opposed to momentary liking. Also, it is *assumed* that Mary likes her roommate.[12] Her roommate will be happy in Europe because she is there, assuming that she wants to be there. She will be in Europe as a result of going there.

3.3 An Explanation for Gloating

In our third example (Figure 5), the parenthetical comments were not encoded, but they are included to retain fidelity to the original report from the diary study:

[11] See Section 4.1 for a discussion of several issues associated with the abductive inference of preconditions.

[12] This is an example where an operational predicate (dispositional liking) is assumed in spite of the fact that an attempt to prove it fails because there is nothing in the list of known facts about whether Mary likes her roommate.

```
Case Facts
    NONE
Query
    why(happy_for(mary, roommate(mary), at(roommate(mary), europe), s1))
Explanation
    happy_for(mary, roommate(mary), at(roommate(mary), europe), s1)
        joy(mary, joy(roommate(mary), at(roommate(mary), europe), s2), s1)
            wants(mary, joy(roommate(mary), at(roommate(mary), europe), s2), s1)
                d_likes(mary, roommate(mary))
            joy(roommate(mary), at(roommate(mary), europe), s2)
                wants(roommate(mary), at(roommate(mary), europe), s2)
                holds(at(roommate(mary), europe), s2)
                    causes(ptrans(roommate(mary), europe), at(roommate(mary), europe), _29530)
                    poss(ptrans(roommate(mary), europe), _29530)
Abbreviations
    s1=do(ptrans(roommate(mary), europe), s0)
    s2=do(ptrans(roommate(mary), europe), _29530)
```

Figure 4. An Explanation for Happy_For

```
Case Facts
    NONE
Query
    why(gloats(john, guy, did(expel(guy,vomit)), s2))
Explanation
    gloats(john, guy, did(expel(guy, vomit)), s2)
        joy(john, distress(guy, did(expel(guy, vomit)), do(expel(guy, vomit), _21375)), s2)
            wants(john, distress(guy, did(expel(guy, vomit)), do(expel(guy, vomit), _21375)), s2)
            distress(guy, did(expel(guy, vomit)), do(expel(guy, vomit), _21375))
                wants(guy, did(not expel(guy, vomit)), do(expel(guy, vomit), _21375))
                holds(did(expel(guy, vomit)), do(expel(guy, vomit), _21375))
                    causes(expel(guy, vomit), did(expel(guy, vomit), _21375))
                    poss(expel(guy, vomit), _21375)
Abbreviations
    s1=do(expel(guy,vomit), s0)
    s2=do(hear(john, expel(guy,vomit)), s1)
```

Figure 5. An Explanation for Gloating

John heard one of the guys who lived below him throwing up
(the morning after a party).
(The guy got what he deserved.)
John gloated.

The explanation illustrating explanatory reasoning involved *gloating* shown in Figure 5 states that John gloats over his neighbor's retching because John takes pleasure in his neighbor's distress. The system is forced to assume that John wanted his neighbor to experience distress. John's neighbor felt distress over vomiting because it is an undesirable event. Deeper inferences might explain how this unpleasant action occurred. These would require additional causal connections (e.g., between the party, excessive drinking, and retching). Another line of inference currently outside the scope of our implementation is: John may wish his neighbor ill because the neighbor's party had been noisy and disturbed John. This explanation would require the addition of knowledge about parties and noise, and about needs associated with sleep.

3.4 Explanations for Relief and Fear

The following case provides examples of *relief* and *fear:*

Mary wanted to go to sleep.
Karen returned.
T.C. finally left her place.
Mary was relieved.

The case is encoded as shown in Figure 6. The case facts say Mary wants sleep. The query asks why Mary is relieved that T.C. is not at her home in the situation that results after T.C.'s departure. T.C.'s departure occurred in the situation resulting from Karen's return.

The automatically constructed explanation assumes that Mary fears T.C.'s presence in her home because she does not want T.C. to be in her home but she anticipates that he will be there. A deeper explanation connecting this desire and anticipation to Mary's desire for restful sleep should be possible. For example, the presence of T.C. might interfere with Mary's sleep.

The explanation in Figure 6 states that Mary is relieved because T.C. is no longer at her home. The explanation of his absence does not include the possibility that he may have been driven away by Karen's return. But it is interesting for another reason. It illustrates the use of causal laws to infer negative fluents relevant to emotional reactions. In this case, because T.C. moved from Mary's home to another location, he is no longer *at* Mary's home.

```
Case Facts
    wants(mary, sleep(mary), _)
Query
    why(relieved(mary, not at(tc, home(mary)), s2))
Explanation
    relieved(mary, not at(tc, home(mary)), s2)
        precedes(s1, s2)
        fears(mary, at(tc, home(mary)), s1)
            ┌──────────────────────────────────────────┐
            │ wants(mary, not at(tc, home(mary)), s1)  │
            ├──────────────────────────────────────────┤
            │ anticipates(mary, at(tc, home(mary)), s1) │
            └──────────────────────────────────────────┘
        holds(not at(tc, home(mary)), s2)
            causes(ptrans(tc, _29887, home(mary), tc), not at(tc, home(mary)), s1)
            poss(ptrans(tc, _29887, home(mary), tc), s1)
                holds(at(tc, home(mary)), s1)
                    not causes(ptrans(karen, home(mary)), not at(tc, home(mary)), s0)
                    ┌───────────────────────────────┐
                    │ holds(at(tc, home(mary)), s0) │
                    └───────────────────────────────┘
                    poss(ptrans(karen, home(mary)), s0)
Abbreviations
    s1=do(ptrans(karen, home(mary)), s0)
    s2=do(ptrans(tc, _29887, home(mary), tc), s1)
```

3.5 An Explanation for Anger

The following example, based on a diary study case involving a dating scenario, illustrates the *angry_at* emotion type. The example involves the breakup of a couple of college students, Kim and John:

Kim wanted to break up with John.
John didn't want to break up with Kim.
They broke up.
John is angry at Kim.

The example is encoded as shown in Figure 7. The query encodes the question "Why is John angry with Kim over the breakup?" The explanation states that John is angry with Kim because Kim initiated the breakup and it caused John to be single. The fact that John doesn't want to be single was given but the remainder of the explanation involves two assumptions. The first is that Kim and John were a couple prior to the breakup. The second assumption is that John views Kim's breaking up with him to be *blameworthy.*

4. DISCUSSION

The previous sections described a representation for knowledge about emotion elicitation and a computer program that constructs explanations based on cases taken from a diary study about emotion episodes. In this section, we discuss some of the strengths and weaknesses of our explanatory reasoner and our representation of knowledge about emotion elicitation.

4.1 Advantages of Abductive Reasoning about Emotions

We claim that abduction is superior to deduction as a basis for explanatory reasoning about emotions because it subsumes deduction which, on its own, will fail when a proof cannot be derived from a given set of facts. The primary advantage of abduction is that it allows for the possibility that assumptions may be required to complete explanations, so that an explanation of a given observation can be proposed even when it does not follow logically from given facts.

It is unreasonable to expect all the information needed to construct explanations involving emotions to be provided in advance. It is particularly unlikely that all the relevant information about mental states will be provided. Indeed, we would like to acquire this sort of information by inference, and abductive inference allows us to do so during the construction of explanations. Abduction may be viewed as a search for plausible hypotheses that help explain observations.

The majority of the cases in the diary study data require assumptions. In this regard, the examples that we have discussed are representative. The kinds of assumptions needed included missing preconditions, goals, prospects, and judgments.

```
Case Facts
    wants(kim,single(kim),_)
    wants(john,not(single(john)),_)
Query
    why(angry_at(john,kim,breakup(kim,kim,john),s1))
Explanation
    angry_at(john, kim, breakup(kim, kim, john), s1)
        agent(breakup(kim, kim, john), kim)
        holds(did(breakup(kim, kim, john)), s1)
            causes(breakup(kim, kim, john), did(breakup(kim, kim, john)), s)
            poss(breakup(kim, kim, john), s)
                holds(couple(kim, john), s)
        precedes(s, s1)
        causes(breakup(kim, kim, john), single(john), s)
        blameworthy(breakup(kim, kim, john))
        wants(john, not single(john), s1)
        holds(single(john), s1)
            causes(breakup(kim, kim, john), single(john), s)
            poss(breakup(kim, kim, john), s)
                holds(couple(kim, john), s)
Abbreviations
    s1 = do(breakup(kim, kim, john), s)
```

Figure 7. An Explanation for Anger

Examples of preconditions inferred by abductive inference included the following. In the example of *joy* (Section 3.1) the explanation was completed with an assumption that Mary had possession of a home-cooked meal. This explained how it was possible for her to ingest it. In the example of *gloating* (Section 3.3) it was assumed that it was possible for John's neighbor to vomit. In Section 3.5 it was assumed that Kim and John were a couple before she broke up with John and he became angry.[13]

Examples of abductive assumptions about goals occur frequently. The example in Section 3.4 required an assumption that Mary wanted T.C. to go somewhere else in order to explain Mary's fear that T.C. would be at her home. Assumptions about others' goals occur in explaining fortunes-of-others emotions. In the example for *happy_for* (Section 3.2), an assumption was made about Mary's roommate's desire to be in Europe. The example of *gloating* (Section 3.3) required an assumption that John wanted his neighbor to be distressed.

Abductive assumptions about other mental states include assumptions about whether agents anticipate events. In the example of *relief* (Section 3.4), it was necessary to assume that Mary anticipated T.C.'s continued (unwelcome) presence in her home.

Assumptions about judgments of blameworthiness and praiseworthiness are important in explaining attribution emotions and compound emotions. For example, in the *anger* case, the assumption that Kim's breaking up with John was blameworthy was made in order to explain why John was angry at Kim.

None of the explanations constructed in these examples could have been constructed by the abduction engine without its abductive inference capability, given the background knowledge and codifications of the cases provided with the observations to be explained. Given the same information, a purely deductive PROLOG-style interpreter would have failed to find an explanation.

Admittedly, the knowledge base could conceivably be extended so that some assumptions could be eliminated and replaced by deductive inferences. Knowledge of ethics and standards of behavior could be provided, reducing the number of assumptions in explanations requiring judgments of blameworthiness and praiseworthiness. Some additional inferences could be made deductively, rather than abductively. For example, we saw several instances of necessary preconditions in the emotion cases: It is necessary for

[13] The reader may wonder how it is possible to make these assumptions given that the abduction engine is not allowed to assume that an arbitrary fluent might be a precondition for anaction (Section 2.4). The reason is that assumptions of the form *preconditions*(A, F) are prohibited, whereas assumptions of the form *holds*(F, S) are permitted. In other words, we allow a conjecture that a condition is true in a given situation but we disallow a conjecture that the condition is a precondition of an action.

two people to form a couple in order to break up, it is necessary to have food prior to eating it, and so on. If we provide the reasoning system with facts stating that such preconditions are necessary, it should be possible to replace these abductive inferences with deductive inferences.

This is special case of a more general idea. Recent research on the relationship between abduction and other forms of reasoning shows that there is a close relationship between abduction and an alternative deductive approach based on closure and minimization (Konolige, 1992). It is possible to translate abduction into the alternative approach by rewriting a logical theory and adding "closure statements," for example, statements to the effect that the known causes or preconditions are the only ones. This is obviously appropriate if the known preconditions are necessary and not just sufficient. However, it is not likely that all relevant preconditions, causes, desires, prospects, and judgments can be provided in advance.

The abductive approach is well suited to the domain of emotion-relevant reasoning because it does not require complete knowledge of causation, and causal closures need not be computed and asserted. The ability to generate hypotheses and make assumptions

- that implicit preconditions held in an effort to explain how an action led to an effect,
- that agents had certain desires in order to explain their emotional reactions or actions,
- that agents anticipated certain events in order to explain emotional reactions, and
- that actions are considered praiseworthy or blameworthy on the basis of emotional reactions to those actions

gives abduction advantages that are important for explanatory reasoning about emotions.

4.2 The Problem of Evaluating Explanations

The most important limitation of the method implemented here is that it does not address questions associated with evaluating the plausibility of multiple explanations. Such questions include the following:

How can one avoid a combinatorial explosion of explanations, many of which are completely implausible? Evaluation of plausibility cannot wait until all possible explanations have been produced. Sometimes infinitely many explanations are possible, so some evaluation must be done during explanation construction.

The machine-generated explanations we have presented here were generated using depth-first search. Most were the first acceptable explanations generated but, in some cases, the initial explanations were rejected by the user. Alternative explanations for a given example are often compatible but, in general, alternate explanations will include mutually inconsistent competitors. This raises the question of how one should decide what to believe and what to disbelieve when conflicts arise between alternate explanations? Methods for weighing the evidence would help decide which explanation is more plausible in many cases. But in other cases, it might be prudent to delay making a decision (Josephson, 1990). Or one could take some action aimed at acquiring new information that might help resolve the conflict.

Finally, how should one decide when to assume a hypothesis that would explain given observations? Currently, we rely on simple heuristics that specify inadmissible assumptions (see Section 2.4). After applying these heuristics, the abduction engine falls back on the user. The user is asked to validate each assumption and to accept each explanation. If the user rejects an assumption or explanation, backtracking occurs and the abduction engine seeks the next alternative.

4.3 Advantages of the Situation Calculus for Emotions

In this section, we list some desiderata for representations of theories of the cognitive structure of emotions. We show how the features of our situation calculus representation address goals relevant to representing and reasoning about emotion elicitation.

One of the most basic tenets of the theory of emotions sketched in Section 1.2 is the view that emotions are positively or negatively valenced reactions. Each emotion is paired with an emotion with a complementary valence. In addition, pairs of opposing extrinsic predicates play an important role in reasoning about emotions. For example, the opposing predicates *praiseworthy* and *blameworthy* play a crucial role in the emotion theory. Negations and opposites are important in the emotion-eliciting conditions. Our representation language provides support for these aspects of the theory by allowing for both positive and negative fluents and actions.

The situation calculus account for actions captures important causal information clearly needed in constructing explanations involving emotions. Situation calculus provides for a causal theory of actions that includes both positive and negative effects and preconditions. Emotion types (represented as fluents) are not caused directly by actions, in the sense that they do not appear as direct effects encoded in *causes* statements. Instead, they are caused indirectly; the theory specifies eliciting conditions that contain actions and other fluents. We saw several instances of actions causing effects that resulted in emotional reactions. In the first emotion example, Mary was happy to be at home. The fact that she was at home was caused by the fact that she went there. This is an instance of a positive effect of *ptrans*. We saw an example of a negative consequence of an action engendering an emotional reaction in the case of Mary's relief when T.C. vacated her home. Given that T.C. moved to another location, a negative effect was used to infer (and explain) the fact that T.C. was no longer at Mary's home.

Frame axioms capture the notion that fluents persist unless explicitly altered by actions. The emotion-eliciting conditions use this to advantage: They do not require fluents to be caused by the action most recently executed. Frame axioms are employed to propagate fluents caused by one action through successive actions into later states (provided they are not canceled by intervening actions). In the initial example, the reason Mary is at home is that she went there earlier and the fact that she ingested a meal did nothing to cancel this result.

Many examples of emotional reactions defined in the emotion-elicitation rules in terms of fluents are naturally expressed as responses to actions or other events.[14] Consider the example of *gloat*ing. John's unfortunate neighbor vomited. The ejection of contents of the neighbor's stomach through his mouth resulted in a relocation of said contents. But John's neighbor's distress was in reaction to his vomiting rather than its effect (the new location of the contents of his stomach). The function *did* provides us with fluents that enable us to refer to actions when we wish to focus on the action itself rather than on a specific effect of the action.

Chains of emotional reactions occur frequently. Our representation provides for such chains because emotions are represented as fluents that take fluents as arguments and because fluents appear in the elicting conditions of emotions. Like other fortunes-of-others emotions, *happy_for* is an emotional chain reaction. Fortunes-of-others emotions are reactions to events, but instead of focusing exclusively on the event, they also focus on another's emotional reaction to that event. In the example of *happy_for,* the fact that Mary's roommate was going to Europe was not so important to Mary as her roommate's happiness. Mary's roommate's joy engenders Mary's joy.

Goals play a large role in emotion elicitation. Our representation supports reasoning about goals in constructing explanations involving emotions. (See Section 2.1.) A good example of reasoning about chains of desires occurs in a case involving John's gratification over a high score on the graduate record examination (GRE). In that example, the system is "told" that John wants to be enrolled in grad school. John's gratification is explained in terms of his desire to be admitted to graduate school. The rules for desires are used to infer that John wants to be admitted because this is a precondition of matriculation and matriculation results in achievement of enrollment.

[14] In fact (as discussed in Section 2.3) the original informal theory defined emotion types in terms of events rather than fluents.

Situation calculus provides support for temporal reasoning. A situation

$$do\{a_n, do[a_{n-1}, \ldots, do(a_1, s_0)]\}$$

defines a temporal sequence of situations, $s_0, s_1, \ldots s_n$ where for $i = 1$ to n, $s_i = do(a_i, s_{i-1})$ and s_{i-1} *precedes* s_i. Temporal precedence is used in eliciting conditions for the prospect-based emotions, *satisfied, fears_confirmed, relieved,* and *disappointed.* These emotions are reactions to the confirmation or disconfirmation of a hoped-for or feared fluent. The precedence constraints apply when the relevant fluents are hoped-for or feared in a situation prior to confirmation or disconfirmation. The compound emotions *grateful, angry_at, gratified,* and *remorseful* also employ temporal constraints. Each of these emotions is a reaction to an action and a fluent caused by the action. Two situations are relevant in these emotion types, the situation when the action causes the fluent, and the ensuing situation associated with the emotional reaction to the fluent. The eliciting conditions use the predicate *precedes* to ensure that the temporal precedence constraint between these situations is satisfied. We saw examples in the cases involving anger (Section 3.5), and relief (Section 3.4). In the situation prior to his leaving, Mary feared that T.C. would be at her home. She experienced relief after he left.[15].

Note that the situation calculus does not force a temporal ordering on all events. This is an advantage in the context of emotions. In the eliciting conditions of *happy_for,* there is no time constraint between the situations when the two agents are happy. In the case of *happy_for* (Section 3.2), Mary's roommate's emotional reaction and Mary's reaction are allowed to occur in different situations. Using separate situations is useful, for example, if some intervening action results in someone being informed of another's earlier good fortune. Avoiding temporal contraints on the situations is also useful because in some cases the usual temporal order is reversed. For example, one might be happy for another in anticipation of the other's happiness (e.g., upon learning that the other won a lottery even before the lucky winner knew it).

4.4 Limitations of the Situation Calculus of Emotion Elicitation

In this section, we discuss the main limitations of the situation calculus of emotion elicitation. These include limitations inherent in the situation calculus itself and limitations in the theory of emotion elicitation.

A major limitation of this study is that we did not attempt to represent or reason about intensities of emotions. It is important to extend the represen-

[15] As discussed in Section 2.3, in some cases of relief and disappointment, the attendant fears and hopes violate the constraint requiring them to occur prior to the relief and disappointment. Additional eliciting conditions allow for this, but because it is the exception rather than the rule, priority is given to the interpretation that includes the temporal constraint.

tation presented here to include intensities. Many emotion types are represented in natural language by a number of emotion tokens. Many tokens indicate specific relative intensities of a particular emotion type; for example, "annoyance," "exasperation," and "rage" denote increasingly intense subtypes of anger. Ortony et al., (1988) suggested that emotions only occur when their intensities are driven above thresholds. The approach taken here is to use predicates that are true or false in place of these continuous, real-valued variables. This approach may be viewed as a crude first approximation. We speculate that methods developed in AI research on qualitative reasoning about physical systems (e.g., Forbus, 1984; Kuipers, 1986; Weld & de Kleer, 1990) could be applied to the problem of representing and reasoning about intensities of emotions.

Another limitation of the emotion-eliciting conditions is that they are phrased in terms of facts rather than beliefs. This is because we wanted to avoid having to reason about beliefs and knowledge. But such reasoning is clearly relevant to emotions. Consider the eliciting condition for satisfaction. It states that a person is satisfied if the person hoped for a fluent earlier and now it holds. It seems clear that the eliciting condition is too simple. It should be phrased in terms of the person's epistemic state. For example, the rule for satisfaction might be rewritten: A person may be satisfied if that person *believes* that some hoped for fluent holds. Logics of belief and knowledge should help address such issues, but they are beset with their own complexities (e.g., referential opacity, computational intractability) and they might introduce more problems than solutions.[16]

The most important difficulty for the situation calculus underlying our representation is the famous qualification problem (McCarthy, 1977). The problem is that it is difficult, if not impossible, to specify all the preconditions relevant to the successful execution of an action. We do allow assumptions about the possibility of actions, which means that we can explain how an action occurred without knowing all the preconditions that might have made it possible. At present, we do not attempt explanations of inaction, so we do not have to deal with the difficult problem of inferring preconditions that failed, thus preventing an action from occurring. We do not claim to have solved the qualification problem but we believe our representation and reasoning methods are no more limited by it than are other approaches.

Some examples in the diary study are beyond the scope of our current methods because they require reasoning about actions not taken and the resulting negative effects. In one case, a woman's roommate fails to pay their phone bill. This triggers anger and fear. She is afraid that she will get a bad credit rating and that her phone will be disconnected. In another example, a student expresses anger because his mother failed to send his records

[16] For examples of AI approaches to the difficult problems of reasoning about belief, knowledge, and action, see Konolige (1985) and Moore (1985).

to a dentist and he can't get his teeth cleaned without them. Several cases involve students worrying about poor grades caused by not studying for examinations. The general principle in these examples is that the failure to do an action can often be said to be the reason that an effect does not hold. If we could capture this intuition in a "causal law of non-action" we would have something of direct importance for the attribution emotions because they often involve attributing praiseworthiness or blameworthiness to non-action. Such a law would also have indirect impacts, by combining with existing laws such as the rules for indirect goals, for example, to capture the idea that one may not want an action to be done if it causes an undesirable effect. The current situation calculus does not cover such cases because it says little about negative actions.

Time is important in reasoning about many emotions, especially the prospect-based emotions *hope, satisfaction, relief, fear, fears-confirmed,* and *disappointment.* Our situation calculus deals with temporal precedence but ignores all other temporal relationships such as simultaneity. In the situation calculus, information is stored in an initial state and then propagated to later states via frame axioms. This is a problem in reasoning about emotional reactions to ongoing events. For example, a woman can be grateful to her husband for giving her a massage while he is giving it to her, but limitations of the situation calculus prevent a formulation of the eliciting conditions for gratitude during an ongoing action. The use of representation and reasoning methods suggested by Allen (1981) might help address this limitation.

Actions are viewed as discrete, opaque transitions. Situation calculus provides no tools for describing what happens while an action is in progress; it provides no tools for describing continuous processes like the gradual dissipation of anger. Again, representation and reasoning techniques developed in research on qualitative physics (e.g., Bobrow, 1985; Hobbs & Moore, 1985) might help overcome this limitation.

Strict logical implication often fails to capture the reality of relationships among events, actions, and possible effects. Many contributions of actions toward the achievement of goals involved in examples drawn from the diary study are uncertain in the sense that the action is not guaranteed to achieve the goal. Often, actions facilitate or increase the probability that the goal will be achieved. Several examples in the case study data describe student's emotional reactions to the granting of extensions on due dates of assignments. Besides temporal reasoning, these examples require probabilistic reasoning, in the sense that the granting of extensions increases the likelihood of successful completion of projects. Instead of encoding these weak causal relationships as implications, qualitative representations of conditional probabilities could be associated with cause-effect relationships. Plausibility information such as probabilities could enter more directly into the eliciting conditions of several emotions. In particular, the *hopes* and *fears* emotion types depend on entertaining the possibility that the hoped-for or feared fluent will occur. The intensity of hope or fear (and its plausibility) depends in part on the subjective likelihood of the prospective event (Ortony et al., 1988).

5. RELATED AND FUTURE WORK

A similar approach to formalizing commonsense reasoning about emotions is presented in Sanders (1989). However, that work takes a deductive approach, using a deontic logic of emotions. The logic focuses on a cluster of emotions involving evaluations of actions—including what we have called admiration, reproach, remorse, and anger. The evaluation of actions is ethical, and involves reasoning about obligation, prohibition, and permission. The logic was used to solve problems involving actions associated with ownership and possession of property (e.g., giving, lending, buying, and stealing) by proving theorems. For example, the fact that Jack will be angry was proved given the following:

Jack went to the supermarket.
He parked his car in a legal parking place.
When he came out, it was gone.

It is not clear whetheer the theorems were proved automatically or by hand, so questions of complexity of inference and control of search in the deontic logic remain unanswered. We have argued that abduction offers advantages over deduction alone when applied to the task of constructing explanations involving emotions. Furthermore, our situation calculus of emotion elicitation is more comprehensive than the deontic logic for emotions in that it covers more emotion types. But our approach could benefit from the treatment of ethical evaluations. A detailed comparison and integration of the best parts of the two approaches would be worthwhile.

A number of theories of the cognitive determinants of emotions exist (e.g., Roseman, 1984). In principle, situation calculus could be used to codify these alternative theories. The abductive method we propose for explanatory reasoning about emotions does not depend on the particular emotion theory used.

Recent research on abduction addresses the issues of search control and plausibility mentioned earlier. Stickel (Hobbs, Stickel, Appelt, & Martin, 1993; Hobbs, Stickel, Martin, & Edwards, 1988) has suggested a heuristic approach for evaluating explanations in the context of natural language processing. Subsequent work by Charniak and Shimony (1990) gave Stickel's weighted abduction a probabilistic semantics. Still more recent work (Poole,

1991) incorporates probability directly into a logic-based approach to abduction. These methods promise to provide significantly improved abduction engines that could be used to construct explanations involving emotions, taking plausibility into account.

Related work on natural language comprehension (Dyer, 1983a, 1983b; Lehnert, 1981) argues that emotion words occur frequently in natural language discourse because emotions play a substantial role in our lives. Natural language systems encountering text involving emotions will need to identify and reason about emotions felt by characters in the text. Important subtasks involved in comprehension such as motivation analysis and plan recognition will often require reasoning about emotions (Charniak & McDermott, 1986).

This work focuses on explaining emotions in terms of eliciting situations. But, although situations give rise to emotional reactions, emotions in turn give rise to goals and actions that change the state of the world. Applications such as plan recognition will require the representation of knowledge for causal connections between emotions and subsequent actions. For a brief description of a system for recognizing plans involving emotions, see Cain, O'Rorke, and Ortony (1989). Cain et al. also described how explanation-based learning techniques can be used to learn to recognize such plans. For a fuller discussion of reasoning about emotion-induced actions, see Elliott and Ortony (1992).

This work is based upon a collection of 22 emotion types. In Ortony, Clore, and Foss (1987) about 270 English words are identified as referring to genuine emotions from an initial pool of 600 words that frequently appear in the emotion research literature. In another study, 130 of these emotion words were distributed among the 22 emotion types discussed here. Some emotion words map to several different types, for example, "upset" is compatible with at least *angry_at, distress,* and *shame.* In addition, many words map to the same type. Encoding the relationship between the affective lexicon and the emotion types is an important topic for future research aimed at automatically processing natural language text involving emotions.

Lehnert (1981) argued that it is important to embed a proposed method for representing and reasoning about affect into a larger information-processing system so that the method can be evaluated in terms of the effectiveness of the larger system. We have done this, in a limited sense, by embedding our situation calculus of emotion elicitation into an abductive reasoning system. However, it is still desirable to incorporate our method into a narrative understanding system comparable to BORIS (Dyer, 1983a). One way to do this would be to embed the system we describe here in TACITUS, a natural language processing system that uses abduction as the basis for comprehension (Hobbs et al., 1993).

6. CONCLUSION

We have presented a representation of a theory of the cognitive antecedents of emotions and we have described an abductive method for explaining emotional states. We sketched a computer program, an abduction engine implemented in a program called AbMaL, that uses the theory of emotion elicitation to construct explanations of emotions. We presented explanations of examples based on cases taken from a diary study of emotions. We examined the strengths and weaknesses of both the representation of knowledge of eliciting conditions and the method for constructing explanations.

The most important advantage of our approach to explanatory reasoning about emotions is that abduction allows us to construct explanations by generating hypotheses that fills gaps in the knowledge associated with cases where deduction fails. We found that the majority of the diary study examples could not be explained using deduction alone because they do not follow logically from the given facts. The abduction engine explained the emotions involved in these cases by making assumptions including valuable inferences about mental states such as desires, expectations, and the emotions of others.

REFERENCES

Allen, J.F. (1981). Maintaining knowledge about temporal intervals. *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 221-226). San Francisco, CA: Morgan Kaufman.

Birbaum, L. (1991). Rigor mortis: A response to Nilsson's "Logic and artificial intelligence." *Artificial Intelligence, 47,* 57-77.

Bobrow, D.G. (Ed.). (1985). *Qualitative reasoning about physical systems.* Cambridge, MA: MIT Press.

Cain, T., O'Rorke, P., & Ortony, A. (1989). Learning to recognize plans involving affect. In A.M. Segre (Ed.), *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 209-211). San Francisco, CA: Morgan Kaufmann.

Charniak, E. (1987). Logic and explanation. *Computational Intelligence, 3,* 172-174.

Charniak, E., & McDermott, D.V. (1986). *Introduction to artificial intelligence.* Reading, MA: Addison-Wesley.

Charniak, E., & Shimony, S.E. (1990). Probabilistic semantics for cost-based abduction. *The Eighth National Conference on Artificial Intelligence* (pp. 106-111). Cambridge, MA: AAAI Press/MIT Press.

Davis, E. (1990). *Representations of commonsense knowledge.* San Francisco, CA: Morgan Kaufmann.

DeJong, G.F., & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning, 1,* 145-176.

Dyer, M.G. (1983a). *In-depth understanding: A computer model of integrated processing for narrative comprehension.* Cambridge, MA: MIT Press.

Dyer, M.G. (1983b). The role of affect in narratives. *Cognitive Science, 7,* 211-242.

Elliott, C., & Ortony, A. (1992). *Point of view: Reasoning about the concerns of others.* In J. Kruschke (Ed.), *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

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

Forbus, K.D. (1984). Qualitative process theory. *Artificial Intelligence, 24,* 85–168.

Geneserth, M.R., & Nilsson, N.J. (1987). *Logical foundations of artificial intelligence.* San Francisco, CA: Morgan Kaufmann.

Green, C.C. (1969). Applications of therem proving to problem solving. In D.E. Walker & L.M. Norton (Ed.), *International Joint Conference on Artificial Intelligence.* San Francisco, CA: Morgan Kaufmann.

Heider, F. (1958). *The psychology of interpersonal relations.* Hillsdale, NJ: Erlbaum.

Hobbs, J.R., & Moore, R.C. (Eds.). (1985). *Formal theories of the commonsense world.* Norwood, NJ: Ablex.

Hobbs, J.R., Stickel, M.E., Appelt, D.E., & Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence, 63,* 69–142.

Hobbs, J.R., Stickel, M.E., Martin, P., & Edwards, D. (1988). Interpretation as abduction. *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics* (pp. 95–103).

Josephson, J.R. (1990). On the "logical form" of abduction. *Working notes of the AAAI spring symposium on automated abduction.* University of California, Irvine, Department of Information and Computer Science.

Konolige, K. (1985). Belief and incompleteness. In J.R. Hobbs, & R.C. Moore (Eds.), *Formal theories of the commonsense world.* Norwood, NJ: Ablex.

Konolige, K. (1992). Abduction vs. closure in causal theories. *Artificial Intelligence, 53,* 255–272.

Kowalski, R. (1979). *Logic for problem solving.* New York: Elsevier.

Kuipers, B.J. (1986). Qualitative simulation. *Artificial Intelligence, 29,* 289–338.

Lehnert, W.G. (1981). Affect and memory representation. *Proceedings of the Third Annual Conference of the Cognitive Science Society* (pp. 78–82).

McCarthy, J. (1968). Programs with common sense. In M. Minsky (Ed.), *Semantic information processing.* Cambridge, MA: MIT Press.

McCarthy, J. (1977). Epistemological problems of artificial intelligence. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (pp. 1038–1044). San Francisco, CA: Morgan Kaufmann.

McCarthy, J., & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (Eds.), *Machine intelligence* (Vol. 4). Edinburgh, Scotland: Edinburgh University Press.

McDermott, D. (1987). A critique of pure reason. *Computational Intellligence, 3,* 151–160.

Mitchell, T.M., Keller, R.M., & Kedar-Cabelli, S.T. (1986). Explanation-based generalization: A unifying view. *Machine Learning, 1,* 47–80.

Moore, R.C. (1985). A formal theory of knowledge and action. In J.R. Hobbs & R.C. Moore (Eds.), *Formal theories of the commonsense world.* Norwood, NJ: Ablex.

Nilsson, N.J. (1991). Logic and artificial intelligence. *Artificial Intelligence, 47,* 31–56.

O'Rorke, P. (in press). Abduction and explanation-based learning: Case studies in diverse domains. *Computational Intelligence, 10.*

Ortony, A., Clore, G.L., & Collins, A. (1988). *The cognitive structure of emotions.* New York: Cambridge University Press.

Ortony, A., Clore, G.L., & Foss, M.A. (1987). The referential structure of the affective lexicon. *Cognitive Science, 11,* 361–384.

Peirce, C.S.S. (1931–1958). *Collected papers of Charles Sanders Peirce (1839–1914).* Cambridge, MA: Harvard University Press.

Peng, Y., & Reggia, J.A. (1990). *Abductive inference models for diagnostic problem solving.* New York: Springer-Verlag.

Poole, D. (1991). Representing diagnostic knowledge for probabilistic horn abduction. *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (pp. 1129–1135). San Francisco, CA: Morgan Kaufmann.

Poole, D.L., Goebel, R., & Aleliunas. R. (1987). Theorist: A logical reasoning system for defaults and diagnosis. In N. Cercone & G. McCalla (Eds.), *The knowledge frontier: Essays in the representation of knowledge.* New York: Springer-Verlag.

Pople, H.E. (1973). On the mechanization of abductive logic. *Proceedings of the Third International Joint Conference on Artificial Intelligence* (pp. 147–152). San Francisco, CA: Morgan Kaufmann.

Reiter, R. (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz (Eds.), *Artificial intelligence, and mathematical theory of computation: Papers in honor of John McCarthy.* San Diego, CA: Academic.

Roseman, I.J. (1984). Emotions, relationships, and health. In P. Shaver (Eds.), *Review of personality and social psychology* (Vol. 5). Beverly Hills, CA: Sage.

Sanders, K.E. (1989). A logic for emotions: A basic for reasoning about commonsense psychological knowledge. In E. Smith (Ed.), *Proceedings of the Annual Meeting of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Schank, R.C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology, 3,* 552–631.

Turner, T.J. (1985). [Diary study of emotions: Qualitative data]. Unpublished raw data.

Weld, D.S., & de Kleer, J. (Ed.). (1990). *Readings in qualitative reasoning about physical systems.* San Francisco, CA: Morgan Kaufmann.

Woods, W.A. (1975). What's in a link: Foundations for semantic networks. In D.G. Bobrow & A.M. Collins (Eds.), *Representation and understanding: Studies in cognitive science.* New York: Academic. (Also appears in *Readings in Knowledge Representation* (1985), R.J. Brachman & H.J. Levesque (Eds.). San Francisco, CA: Morgan Kaufmann.