Speech and Natural Language

A Schema-Based Approach to Understanding Subjunctive Conditionals

Wayne Wobcke
Basser Dept. of Computer Science,
University of Sydney,
Sydney, 2006,
AUSTRALIA.

Abstract

We present an AI approach to subjunctive conditionals based on a possible-worlds version of situation semantics, arguing that the connection between the antecedent ϕ and consequent ψ of a conditional 'if ϕ were the case, ψ would be' is an informational relation between types of situations of the kind that can represented in a hierarchy. We present a sound and complete logic of conditionals based on Stalnaker's [1968] logic C2. We discuss methods of evaluating conditionals concerning object types and their properties using inheritance hierarchies, and those concerning actions and events using hierarchies of plans. In the latter case, we specify how to generate the hierarchy of situations from a collection of plan schemas. Our theory explains the traditional 'paradoxes' of conditionals (failure of strengthening the antecedent, failure of transitivity, failure of contraposition) as arising naturally from the fact that our hierarchies use defaults and exceptions.

1 Introduction

A subjunctive conditional is a sentence of the form 'if ϕ were the case, ψ would be too', where ϕ and ψ are propositions. Subjunctive conditional statements invite us to consider a hypothetical scenario, one in which ϕ holds, and ask us whether ψ holds in that scenario. They are often used counterfactually, that is, when ϕ refers to a specific event that did not occur. Conditionals form a vital part of our knowledge, e.g. the justification that the plans of AI actually work can be seen as the truth of a series of conditionals of the form 'if I were to do action A, then condition C would obtain', 'if condition C were to obtain, then I could do action B', etc. Conditionals thus play an important role in prediction of the future and our acting in the world to influence the future.

It is a common criticism of certain philosophical accounts of conditionals, e.g. [Stalnaker, 1968] and [Lewis, 1973], that although these theories provide a rigorous semantics and a sound and complete conditional logic, a crucial aspect of the problem of conditionals is being packaged aside and regarded as a purely pragmatic problem outside of the domain of logical enquiry. In such possible-worlds frameworks, this is usually manifest in the so-called selection function, which forms the basis upon which the semantics of conditionals rests. In these kinds of theories, a subjunctive conditional of the form 'if ϕ were true, ψ would be too' is true relative to a world w if ψ is true in in the world that the selection function picks out for ϕ from w. Nothing much is said about particular selection functions: a set of constraints that selection functions must satisfy is given and related to various axioms of the logics.

The primary aim of this paper is to give a formal account of subjunctive conditionals whose semantics is derived from standard sorts of AI knowledge representation schemes (inheritance hierarchies and collections of planning schemas). As more of an inspiration than a formal basis, we will follow Barwise's [1985] approach to conditionals based on situation semantics, [Barwise and Perry, 1983], in which a subjunctive conditional of the form 'if ϕ were the case, ψ would be too' is true relative to some background context B if there is a constraint $\phi \Rightarrow \psi$ that holds relative to B. However, unlike Barwise, we will take this background context itself to be represented by a situation type, which is specified by a set of possible worlds. For us, a constraint $\phi \Rightarrow \psi$ holds at a situation type σ if in the most general situation type resulting from σ by the addition of information ϕ, ψ also holds. This treatment of conditionals gives us two things: (i) a way of implementing a system that 'understands' subjunctive conditionals – because this 'most general' relation is computable, and (ii) a way of formalizing a logic of conditionals by adapting the Stalnaker-Lewis methods – because the rule of choosing the most general subtype of a situation type can be regarded as a specific selection function. Our theory explains the traditional 'paradoxes' of conditionals (failure of strengthening the antecedent, failure of transitivity, failure of contraposition) as arising naturally from the fact that hierarchies utilize defaults and exceptions. The use of types of situation also enables us (with Barwise) to give a unified account of factual and counterfactual conditionals.

In earlier work, [Wobcke, 1988b], we used (our version of) situation semantics as a formalization of ordinary 'script-based' inference, treating both what Schank and Abelson [1977] call 'script-inference' (filling in the gaps in a causal chain) and 'script-activation' (determining the relevance of a script), as entailment over classes of situation types. This we called plan recognition, following Kautz. However, that theory was monotonic in that if from input Φ it followed some conclusion X, from inputs *Φ* and , % also followed. This was plainly an oversimplification, and so the second aim of this paper is to further reconstruct Script-activation' inferences as conditional reasoning, in which monotonicity fails. Furthermore, insofar as the theory is based on standard planning representations, we claim that our approach to conditionals provides a natural account of (simple kinds of) planning. The theory presented here serves also to formalize the intuitions appealed to in our discussion of inheritance, [Wobcke, 1988c].

2 Conditionals and Hierarchies of Situations

Our theory follows in the same vein as the truth-theoretic accounts of conditionals developed by Stalnaker [1968] and Lewis [1973]. But rather than defining truth relative to a possible world, our underlying semantic framework is a hierarchy of types of situations. Suppose we know that some situation *s* is of type a. We know that *s* could turn out a number of ways depending on what further information about *s* we could obtain. Intuitively, the situation types that are recorded in the hierarchy below a represent all the subtypes of a that *s* could possibly be an instance of. So with the standard script-type examples from [Schank and Abelson, 1977], we can imagine a situation type 'restaurant' with subtypes 'cafeteria' and 'fast-food'. With an inheritance hierarchy, we can imagine an object type 'bird' with subtypes 'penguin' and 'emu'.

A conditional 'if ϕ were the case, ψ would be' concerning a situation *s* is evaluated with respect to a situation type σ that we know *s* to be an instance of, by seeing if $\phi \Rightarrow \psi$ is a constraint that holds at σ. A constraint $\phi \Rightarrow \psi$ holds at σ if however σ develops into a most general subtype σ' of σ such that ϕ holds of σ', ψ also holds of σ'.

The determination of the base type σ with respect to which a conditional is evaluated is a pragmatic concern which we will largely ignore. In many cases, there is no prior context, and so the most general subtype of σ that satisfies ϕ is just the most general type in the hierarchy that satisfies ϕ. For example, consider the situation type σ consisting of my holding a loaded gun that is pointed at you. Consider in this context, the conditional

> If I were to pull the trigger, you would be injured.

Supposing that all the situation types that are subtypes of *a* in which I pull the trigger satisfy your being injured, the conditional will be true. On the other hand, if it is admitted that the gun may equally well not fire if I pull the trigger, the conditional will be false.

We now motivate our analysis by discussing three ideas which make schemas and hierarchies of situations a useful basis for a theory of conditionals.

First, a conditional is not considered true in isolation of other related true conditionals. This argument comes from Hanson's [1961] discussion of the explanation of everyday events in relation to causal statements. Hanson claims that there are as many true causal statements (and so true conditionals) as there are explanations, and that these explanations are dependent on the point of view adopted. Thus of a car crash, it is said that

> The car crashed because the driver tried to avoid the pedestrian.
> The car crashed because the brakes failed.
> The car crashed because the tyres skidded on a patch of ice.

Here, what happened was that in the trying to avoid the pedestrian, the driver applied the brakes which failed to grip the icy road. All the conditions and events play some role in the crash, but each causal statement isolates one of them. The point is that each causal statement is true. So each of the corresponding counterfactuals is true:

> If the driver hadn't tried to avoid the pedestrian, the car wouldn't have crashed.
> If the brakes hadn't failed, the car wouldn't have crashed.
> If the tyres hadn't skidded on a patch of ice, the car wouldn't have crashed.

To handle this phenomenon, we will assume that the set of causal statements forming the complex explanation is represented in the one plan schema. All the conditionals will be true because the situations which are used in evaluating the conditionals are determined from this one schema.

The second argument for a hierarchy of schemas is that this allows the representation of default assumptions and exceptions. The use of defaults is realized in the theory of conditionals by the failure of strengthening the antecedent For example, the first conditional is true, the second false:

> If I had recharged the battery, the car would have started.
> If I had recharged the battery and left it disconnected, the car would have started.

We interpret this as follows. In the first example, there is an implicit assumption that the battery is connected. This assumption is denied in the antecedent of the second conditional, and the informational (and causal) chain between antecedent and consequent is broken. We assume that there is a planning schema consisting of 'recharge(battery) —> connect(battery) —> starts(car)'. Thus the most general situation which satisfies *recharge(battery)' also satisfies 'starts(car)', but the most general situation that also satisfies '~ connect(battery)' will not satisfy 'starts(car)'. This is an example of a default assumption, but defaults may also appear in the consequent of a conditional, e.g. with the familiar

> If that thing were a bird, it would be able to fly.

This conditional is true, although it allows exceptions. With the standard network, the most general 'bird' situation satisfies 'flies', but some of its subtypes (penguins) do not. This kind of example will be used to explain the failure of contraposition (see section 4).

Third, conditionals often come in pairs, e.g.

> If I had recharged the battery, the car would have started.

> If I hadn't recharged the battery, the car wouldn't have started.

To handle such pairs (and some of the examples above), we must further interpret what a plan actually says about when one of its actions fails to happen. In doing so, we follow Mackie [1965], who argues that each cause of some event is a necessary component of a collection of actions and/or conditions which together are sufficient for the occurrence of the event. Now in the context of a specific plan, we can consider each link *A* —> *B* to mean that in that plan, A is a necessary component of a collection of actions (all those *A* such that *A* -> *B)* which together are sufficient for B's executability. Thus when *A* does not happen, *B* cannot (hence the truth of the conditional). We will build in to the theory (see section 4) a way of generating from the one schema, a hierarchy of situations that accounts for such pairs.

3 The Logic SC

3.1 Situated Conditionals: Syntax

The axiomatization of our logic *SC*, for *situated conditionals*, derives from the S4-based version of Stalnaker's [1968] logic C2. We use the binary operator > to denote the subjunctive conditional; '$\phi > \psi$' can be read as 'if ϕ were the case, ψ would be too'. Each conditional is evaluated with respect to a context c. More concretely, a formula $c: \phi > \psi$ is true if ψ holds in the most general situation type that is a subtype of c that satisfies ϕ. We need only consider a single most general situation type because when there would otherwise be a collection of σ' that were most general subtypes of σ, we will assume that the disjunction of the σ' is more general than each of the σ' and hence is *the* situation type satisfying ϕ with respect to which ψ is tested.

Syntactically, SC differs from C2 in two respects. First, the partiality of truth assignments in situations means that the law of conditional excluded middle:

$$c: (\phi > \psi) \vee (\phi > \sim\psi)$$

is not valid. Second, disjunction distribution, i.e.

$$c: \phi > (\psi \vee \chi) \rightarrow (\phi > \psi \vee \phi > \chi)$$

fails, due to the fact that situations can satisfy disjunctive properties without satisfying either property in the disjunction. However, the above formula is valid when ψ is a literal:

Definition. A *literal* is a formula of the form $\Box\phi$, $\sim\Box\phi$, $\Diamond\phi$, $\sim\Diamond\phi$, $\phi > \psi$ *or* $\sim(\phi > \psi)$, where ϕ and ψ are propositions.

The following scheme defines the axioms of the system SC:

(P1) all the PC axioms written with formulae $c: \phi$,
(P2) $c: \phi$, for all c and all PC axioms ϕ,
(P3) $c: (\phi \rightarrow \psi) \rightarrow (c: \phi \rightarrow c: \psi)$, for all PC formulae ϕ and ψ,
(P4) $c: \sim\phi \rightarrow \sim c: \phi$,
(M1) $c: \Box\phi \rightarrow \phi$,
(M2) $c: \Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$,
(M3) $c: \Box\phi \rightarrow \Box\Box\phi$,
(M4) $c: \Box(\phi \rightarrow \psi) \rightarrow (\Diamond\phi \rightarrow \Diamond\psi)$,
(M5) $c: \Box\phi \rightarrow \Diamond\phi$,
(M6) $c: \Diamond\phi \rightarrow \sim\Box\sim\phi$,
(M7) $c: \sim\Diamond\phi \rightarrow \Box\sim\Diamond\phi$,
(M8) $c: \sim\Diamond\phi \rightarrow \Box\sim\phi$, ϕ a conjunction of literals,
(M9) $c: \phi \rightarrow c: \Diamond\phi$,
(M10) $c: \Diamond(\phi \vee \psi) \rightarrow (\Diamond\phi \vee \Diamond\psi)$, ϕ a literal,
(M11) $c: \phi \vee \psi \rightarrow c: \phi \vee c: \psi$, ϕ a literal,
(M12) $c: \sim\Diamond(\phi \;\&\; \sim\Diamond\phi)$,
(C1) $c: \Diamond\phi \;\&\; \Box(\phi \rightarrow \psi) \rightarrow (\phi > \psi)$,
(C2) $c: (\phi > \psi \rightarrow \chi) \rightarrow (\phi > \psi \rightarrow \phi > \chi)$,
(C3) $c: (\phi > \sim\psi) \rightarrow \sim(\phi > \psi)$,
(C4) $c: (\phi > \psi) \;\&\; (\psi > \phi) \rightarrow (\phi > \chi \rightarrow \psi > \chi)$,
(C5) $c: \Diamond(\phi \;\&\; \psi) \rightarrow (\phi > \Diamond\psi)$,
(C6) $c: (\phi > \psi) \rightarrow \Diamond(\phi > \psi)$,
(C7) $c: \Diamond(\phi > \psi) \rightarrow (\Diamond\phi \;\&\; \Diamond\psi)$,
(C8) $c: (\phi > \psi) \rightarrow (\phi > (\chi \leftrightarrow (\psi > \chi)))$,
(C9) $c: \phi \rightarrow (c: \psi \leftrightarrow c: \phi > \psi)$,
(C10) $c: (\phi > \psi \vee \chi) \rightarrow (\phi > \psi \vee \phi > \chi)$, ψ literal.

The inference rules in SC are modus ponens, necessitation and conditionalization, i.e.,

(MP) From α and $\alpha \rightarrow \beta$ infer β.
(N) If SC $\vdash c: \phi$, infer $c: \Box\phi$.
(C) If SC $\vdash c: \psi$, infer $c: \Diamond\phi \rightarrow (\phi > \psi)$.

Note that $\Diamond$ can be defined in terms of the > operator, using the theorem $c: \Diamond\phi \leftrightarrow (\phi > \phi)$. However, $\Box$ cannot be defined in terms of >, as it can in Stalnaker's system, because Stalnaker's definition depends on the fact that worlds assign truth values to every proposition. Specifically, he uses the fact that when $\sim\phi$ fails to hold at any world accessible to a given world, ϕ must hold at all worlds accessible to the given world. This property does not hold when situations are used in place of worlds.

3.2 Situated Conditionals: Semantics

The semantics of the logic SC is based on a hierarchy of situation types. A situated conditional *c*: Φ > y is true if y holds in the most general situation type which satisfies Φ that is a subtype of the type *[c*]. We follow Stalnaker [1968] in using a selection function, which for each situation type and antecedent of a conditional, picks out a situation type with respect to which the consequent of the conditional is tested. Our logic SC is complete with respect to models which satisfy certain conditions on this selection function. It is important to note that *the restrictions we place on selection functions do not guarantee that the most general subtype of the base type is selected.* But for any model with selection function meeting our requirements, there is a corresponding model with selection defined according to the 'most general subtype' rule, which satisfies exactly the same set of propositions. Thus the logic SC can be regarded as the logic of situated conditionals.

Definition. A *selection function f is* a partial function defining for a situation type a and proposition Φ, a subtype of a which meets the following conditions:

(i) if ϕ holds at some subtype of σ, then $f(\sigma, \phi)$ is defined,
(ii) ϕ holds at $f(\sigma, \phi)$,
(iii) if ϕ holds at σ, then $f(\sigma, \phi) = \sigma$,
(iv) if ψ holds at $f(\sigma, \phi)$ and ϕ holds at $f(\sigma, \psi)$, then $f(\sigma, \phi) = f(\sigma, \psi)$,
(v) if ψ holds at any subtype of σ that satisfies ϕ, then $\Diamond\psi$ holds at $f(\sigma, \phi)$.

Condition (i) is obviously a reasonable requirement. Conditions (ii), (iii) and (iv) are Stalnaker's conditions on selection functions which guarantee that the selected subtype should in fact satisfy ϕ and that the current situation type should be given the highest priority when it comes to making the selection. Condition (iv) implies that if ψ holds at $f(\sigma, \phi)$, then $f(\sigma, \phi \,\&\, \psi) = f(\sigma, \phi)$. Condition (v) captures part of the idea that $f(\sigma, \phi)$ is defined as the most general subtype of σ that satisfies ϕ. This requirement is equivalent to the condition (v'), which implies (but is not implied by) condition (v).

(v') if ϕ holds at a subtype σ' of s, then σ' is a subtype of $f(\sigma, \phi)$.

SC provides the logic of situated conditionals in the sense that no additional propositions hold of the restricted class of models where selection functions are defined to respect (v') and not just (v). Call these latter models the *proper models.*

Lemma. For any SC model M, there is a proper model M' equivalent to M in the sense that exactly the same propositions holding of M hold in M'.

An SC interpretation consists of a set of situations together with a reflexive, transitive accessibility relation on that set. The interpretation of a context constant symbol c is a situation type, written $[c]$. A formula $c : \phi$ where ϕ is an *atomic* proposition (i.e. contains no modal or conditional operators) is true just if ϕ holds in the situation type $[c]$. These simple situation types must satisfy a consistent and deductively closed set of atomic PC formulae – to guarantee this, $[c]$ is taken to be a non-empty set of PC interpretations (worlds). Thus for atomic propositions, this truth condition is equivalent to van Fraassen's [1966] supervaluations. A formula $c : \Box\phi$ is true if ϕ holds in all subtypes of $[c]$, and a formula $c : \Diamond\phi$ is true if ϕ holds in at least one subtype of $[c]$. A formula $c : \phi > \psi$ is true if ψ holds in $f([c], \phi)$.

More formally, let I be an SC interpretation. Then we define the truth conditions for formulae in I in two stages, first for the atomic SC formulae, then for the complex SC formulae. The truth conditions are defined for formulae in conjunctive normal form. Let $[c]$ be the set of worlds that is the interpretation of c and consider the hierarchy of situation types I_c rooted at $[c]$. Then, for the atomic SC formulae $c : \phi$:

$I_c \models \phi$ if $[c] \models \phi$, for all atomic propositions ϕ,
$I_c \models \sim\phi$ if $I_c \not\models \phi$, for ϕ non-atomic,
$I_c \models \phi \vee \psi$ if $I_c \models \phi$ or $I_c \models \psi$, for ϕ or ψ non-atomic
$I_c \models \phi \,\&\, \psi$ if $I_c \models \phi$ and $I_c \models \psi$, for ϕ or ψ non-atomic,
$I_c \models \Box\phi$ if all subtypes of $[c]$ in I_c satisfy ϕ,
$I_c \models \Diamond\phi$ if some subtype of $[c]$ in I_c satisfies ϕ,
$I_c \models \phi > \psi$ if $f([c], \phi)$ is defined and satisfies ψ.

The truth conditions for complex SC formulae are as follows:

$I \models c : \phi$ if $I_c \models \phi$,
$I \models \sim c : \phi$ if $I_c \not\models \phi$,
$I \models c : \phi \vee c' : \psi$ if $I_c \models \phi$ or $I_{c'} \models \psi$,
$I \models c : \phi \,\&\, c' : \psi$ if $I_c \models \phi$ and $I_{c'} \models \psi$.

Proposition. SC is a sound and complete inference system.

Proof. (Sketch) The proof uses the standard Henkin method of constructing a model for any consistent sentence from a maximal consistent set of SC sentences. However, there are some peculiarities of our particular proof stemming from the supervaluation semantics used for atomic propositions.

As in [Stalnaker and Thomason, 1970], we construct a model for a consistent set of sentences Γ from a maximal consistent set Γ^* containing Γ. However, where in their proof any maximal consistent set containing Γ will do, we require some further properties of this set. Given a set of SC formulae Σ, define Σ_c to be the set of ϕ such that $c : \phi \in \Sigma$. Then we require Γ_c^* (which will not in general be maximal consistent, although it will be consistent since Γ^* is) to satisfy (i) deductive closure, (ii) definiteness, and (iii) coherence, where the latter two of these are defined as follows.

Definition. A set Σ_c (as defined above) is *definite* if whenever Σ_c contains $\phi \vee \psi$ where ϕ is a modal or conditional literal, either Σ_c contains ϕ or Σ_c contains ψ (or both).

Definition. A set Σ_c (as defined above) is *coherent* if whenever Σ_c contains ϕ, Σ_c contains ψ iff Σ_c contains $\phi > \psi$.

It is clear from (M11) and (C9) that these properties are required ((M9) is covered by (C9)). Then once it has been established that every consistent set of sentences Γ can be extended to a maximal consistent set Γ^* such that Γ_c^* is deductively closed, definite and coherent, the model drops out.

The full proof may be found in [Wobcke, 1988a]. □

4 Evaluating Subjunctive Conditionals

The theory of conditionals presupposes that the knowledge represented in a hierarchy of plan schemas and in an inheritance hierarchy can be used to construct a hierarchy of situation types. In the case of the inheritance network, the hierarchy just is the network. In the case of planning knowledge, the construction is derived from a hierarchy of planning schemas that is augmented to handle negated action descriptions. We define/ (a, Φ) for any situation type a and proposition Φ, when a corresponds to a plan, to be the most general subtype of a that satisfies Φ.

As an example of how the theory works for our favourite inheritance hierarchy, suppose birds have short legs and can fly, penguins have short legs but can't fly, and emus have long legs and also can't fly. Then the following are all true:

> If that bird were a penguin, it wouldn't be able to fly.
> If that bird were an emu, it wouldn't be able to fly.
> If that bird couldn't fly, it would be a penguin or an emu.
> If that bird couldn't fly, it might or might not have long legs.

but

> If that bird couldn't fly, it would have long legs.

is false (i.e. allowing for penguins). The most general subtype of *bird* that satisfies '~ flies* satisfies the disjunction 'has(long-legs) V ~has(long-legs)\ but satisfies neither disjunct.

We can now illustrate why contraposition for conditionals fails. Consider 'if that bird had short legs, it would be able to fly'. This conditional is true in the above network because birds by default have short legs, and they can fly. But for the contrapositive to be true, we would need the most general subtype of *bird* that satisfies '~ flies' to satisfy the property of not having short legs. In the above hierarchy, birds that can't fly are either emus or penguins, one of which has long legs, the other of which doesn't, so the contrapositive isn't true.

Now we consider the case of plan schema hierarchies. The straightforward definition of a hierarchy of plans in which one plan is a subtype of another if the latter could be elaborated into the former is inadequate because it fails to handle conditionals whose antecedents are the negations of actions, for example,

> If I had recharged the battery, the car would have started.
> If I hadn't recharged the battery, the car wouldn't have started.

From the discussion in section 2, each link *A* —> *B* in a plan means that A is a necessary component of a collection of actions (all those *A* such that *A* —> *B)* which together are sufficient for *B* to be executable. Therefore when an action *A* does not happen, all those actions in the plan which depend on A's being done will also fail to eventuate. We now make this idea precise.

We associate with each schema a collection of *failure* schemas, which say what happens when some action(s) in the schema fail. To do this, we start with the schemas ordered according to the simple definition proposed above. Now starting from the most general situation types and working down to the most specific ones, we define the failure schemas associated with each schema. First, there are no failure schemas for the primitive actions. So consider a schema with a minimal number of actions (perhaps just two actions and one link). Consider the partially ordered actions in the schema in reverse order. Take an action *A* in the plan connected to a collection of following actions $B_{i;}$ and suppose the failure schemas for all the B_i have already been constructed. Now construct further failure schemas for *A* by replacing *A* by ~A in all those failure schemas that contain any one of the negated B_is. This gives a collection of failure schemas for all the simplest schemas. Now for any more specific schema, repeat the process but not using the (reverse of) the partial order on the actions contained in that schema, but using instead the more restrictive partial order obtained by coalescing the partial orders on the actions in the more general schemas from which the specific schema was derived.

A simple example will make this procedure clear. Consider two schemas A -> *B* and *C ->D.* We will get failure schemas ~A -> ~*B* and ~C -> ~*D* (the '-V is a plan link, not an implication). Now suppose the two simple schemas are combined into a schema A -> C -> *B* ->Z). When C doesn't occur, we don't want to say that *B* doesn't, because *B's* occurrence depends only on that of A. We do, however, want to say that *D* fails to occur. Constructing the failure schemas by reference to the original schemas yields these results.

We can now state formally the definitions of the hierarchy of plan plan descriptions, assuming a given hierarchy of plan schemas. First, the definition without taking into account negated actions, then the modified definition:

Definition. (Type Hierarchy of Plans) A plan P_1 is a sabfypeofaplan P_2, written $P_1 < P_2$, if

> (i) either P_1 or P_2 is primitive, P_1 has P_2 as an ancestor in the given hierarchy,
> or (otherwise)
> (i) the set of role variables of P_1 contains the set of role variables of P_2,
> (ii) corresponding role types in P_1 are subtypes of those in P_2 using an inheritance hierarchy,
> (iiia) the expansion of P_1 contains a subtype of P_2, or
> (iiib) each action in the expansion of P_1 is a subtype of an action in the expansion of P_2.

Definition. (Type Hierarchy of Plan Schemas - Modified for Failure Schemas) A plan schema P_1 is a *subtype* of a plan schema P_2, written $P_1 < P_2$, *if* $P_l < P_2$ under the above definition with an added clause stating that if action A is a subtype of an action *B,* then negated action ~A is a subtype of the action *B* and of the negated action *~ B.*

So now, intuitively, a schema P_1 is a subtype of a schema P_2 if P_1 contains a collection of actions that are subtypes of counterparts in P_2 and all the counterparts of actions negated in P_2 are negated in P_1.

Define a *simple plan* to be a (possibly partial) instantiation of a single plan schema. The hierarchy of situations used for our theory of conditionals is a hierarchy of *plan descriptions* (c.f. [Wobcke, 1988b]), consisting of disjunctions of those formulae that completely describe the simple plans. First, define a *complete* description of a simple plan to be a conjunction of atomic formulae such that the conjuncts are in one-one correspondence with the actions in the plan. These conjunctions can be identified with the simple plans themselves. We take as the situation types in the hierarchy the disjunctions of such complete descriptions (except that with regard to failure schemas, we allow only disjunctions of complete descriptions of *completely instantiated* failure schemas). In the hierarchy of plan descriptions, the disjunction of two plan descriptions D_1 and D_2 is placed directly above D_1 and D_2, and if $D_1 \vee D_2$ is a disjunctive description of simple plans, we define $f(D_1 \vee D_2, \Phi)$ to be $f(D_1\ \Phi)\ V/(D_2, \Phi)$.

We can now explain why conditionals with negated antecedents Φ where ($\Phi > y$ is a constraint, always seem to be uncategorically true. Our explanation turns on the use of such conditional statements. The use of the conditional is often counterfactual, in which case it is presupposed that the antecedent actually occurred. So for example, the earlier conditional is being evaluated with respect to a situation which includes the battery being recharged. Now in such a context, the recharging is a necessary condition on the car's starting, and by our definition of the hierarchy of complete plan descriptions, the conditional will be true at such a situation (by reference to an appropriate failure schema). Conversely, suppose I didn't recharge the battery: I didn't this morning, for example, and the car started as usual. If today I say 'if I hadn't recharged the battery, the car wouldn't have started', then I am saying something false. But in doing so, I am violating the precondition on the use of the conditional.

Note that as a result of our definitions, it is not a restriction on our theory that conditionals must always look forwards in time. For example, the following two statements are both true:

If he had struck the match, it would have lit.
If the match had lit, he must have struck it

But this does not force us to accept contraposition for conditionals. Suppose there are two ways to light a cigarette, using a match and using a lighter. Then the following are true:

If he had used a match, the cigarette would have lit.
If he had used a lighter, the cigarette would have lit.
If the cigarette had lit, he could have used a match.
If the cigarette had lit, he could have used a lighter.

but the contrapositives of the first two statements (the third and fourth with 'must' replacing 'could') are both false.

5 Conclusion

In this paper, we presented an AI theory of conditionals that emphasizes the judgement of conditional statements using knowledge of informational relations represented in a hierarchy of situations. This AI approach contrasts with that of Ginsberg [1986], who does not have a truth-theoretic account of conditionals. Our theory explains the standard 'paradoxes' of conditionals in a natural way as arising from the properties of hierarchies of situations. We have formalized our theory using a conditional logic SC based on a version of situation semantics. We have implemented (in Prolog) a system that evaluates atomic SC formulae by constructing most general subtypes of situation types - in the case of planning schemas, this system is essentially carrying out a subtask of planning. We have not considered ways of efficiently implementing a system that is capable of dealing with arbitrary SC formulae. In future work, we intend extending our formalism to include an account of knowledge and belief.

Acknowledgements

This work was carried out as Ph.D. research in the Department of Computer Science at the University of Essex in the U.K. I gratefully acknowledge valuable discussions had with my supervisor, Prof. Raymond Turner, my thesis examiners Dr Sam Steel and Prof. Allan Ramsay, and my colleague at Essex, Dr John Bell. I would also like to acknowledge the financial support of the Commonwealth Scholarship Commission in the U.K.

References

Barwise, J. [1985] 'The Situation in Logic-II: Conditionals and Conditional Information.' Technical Report CSLI-85-21, Center for the Study of Language and Information, Stanford University.

Barwise, J. & Perry, J. [1983] *Situations and Attitudes.* MIT Press, Cambridge, MA.

Ginsberg, M.L. [1986] 'Counterfactuals.' *Artificial Intelligence,* 30, 35-79.

Hanson, N.R. [1961] 'Causality.' in Beauchamp, T.L. (Ed.) [1974] *Philosophical Problems of Causation.* Dickenson, Encino, CA.

Lewis, D.K. [1973] *Counterfactuals.* Blackwell, Oxford.

Mackie, J.L. [1965] 'Causes and Conditions.' in Sosa, E. (Ed.) [1975] *Causation and Conditionals.* Oxford University Press, Oxford.

Schank, R.C. & Abelson, R.P. [1977] *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Stalnaker, R.C. [1968] 'A Theory of Conditionals.' in Sosa, E. (Ed.) [1975] *Causation and Conditionals.* Oxford University Press, Oxford.

Stalnaker, R.C. & Thomason, R.H. [1970] 'A Semantic Analysis of Conditional Logic* *Theoria,* 36, 23-42.

van Fraassen, B.C. (1966) 'Singular Terms, Truth-value Gaps, and Free Logic' *Journal of Philosophy,* 63, 481-495.

Wobcke, W.R. [1988a] 'A Theory of Conditionals in Situation Semantics.' Technical Report CSCM-32, Centre for Cognitive Studies, University of Essex.

Wobcke, W.R. [1988b] 'A Logical Theory of Plan Recognition.' *Proceedings AAAI-88 Workshop on Plan Recognition,* (extended version appears as Technical Report CSCM-31, Centre for Cognitive Studies, University of Essex).

Wobcke, W.R. [1988c] *A Global Theory of Inheritance.' *Proceedings ECAI-88,* 214-219. (extended version appears as Technical Report CSCM-33, Centre for Cognitive Studies, University of Essex).