A Formal Theory of Knowledge and Action

Non-classical Logics and Applications

Seminar - Winter 2008

The Topics of the Lecture:

- The Interplay of Knowledge and Action
- Formal Theories of Knowledge
  - A Modal Logic of Knowledge
  - A Possible-Worlds Analysis of Knowledge
  - Knowledge ,Equality and Quantification
- A Possible-Worlds Analysis of Action
- An Integrated Theory of Knowledge and Action
  - The Dependence of Action on Knowledge
  - The Effects of Action on Knowledge

The Interplay of Knowledge and Action

- When the agent entertains a plan for achieving some goal, he must consider not only whether the physical prerequisites of the plan have been satisfied, but also whether he has all the information necessary to carry out the plan.
- Agent must be able to reason about what he can do to obtain necessary information that he lacks.

AI Planning

- Artificial intelligence planning systems are usually based on the assumption that, if there is an action an agent is physically able to perform, and carrying out that action would result in achievement of a goal P, then the agent can achieve P.

The Test Notion

- The essence of a test is that it is an action with a directly observable result that depends conditionally on an unobservable precondition.

The Instance of Test

- In the use of litmus paper to test the pH of a solution, the observable result is whether the paper has turned red or blue, and the unobservable precondition is whether the solution is acid or alkaline.

Bad Attempt of the Test Formalization

- If we tried to formalize the result of such a test by making simple assertion about what the agent knows subsequent to the action, we would have to include the result that the agent knows whether the solution is acid or alkaline as a separate assertion from the result that he knows the color of the paper.
- If we did this, however, we would completely miss the point that knowledge of the pH of the solution is inferred from other knowledge, rather than being a direct observation.
- In effect, we would be stipulating what actions can be used as tests, rather than creating a formalism within which we can infer what actions can be used as tests.

The Interplay of Knowledge and Action

- Finally ,we must be able to infer that, if the agent knows

  - That the test took place
  - The observable result of the test
  - How the observable result depends on the unobservable precondition

Then we will know the unobservable precondition.

The Conclusion of the Interplay of Knowledge and Action

- From the preceding discussion we can conclude that any formalism that enables us to draw inferences about tests at this level of detail must be able to represent the following types of assertions:

1. After A performs Act, he knows whether Q is true.
2. After A performs Act, he knows that he has just performed Act.
3. A knows that Q will be true after he performs Act if and only if P is true now.
4. If (1),(2) and (3) are true, then, after performing Act, A will know whether P was true before he performed Act.

Formal Theories of Knowledge

A Modal Logic of Knowledge

- The first step in devising a formalism for reasoning about knowledge is to decide what general properties of knowledge we want that formalism to capture.

The properties of knowledge:

- Anything that is known by someone must be true. If *P* is False, we would not want to say that anyone knows *P*.
- If someone knows something, he knows that he knows it.

A Modal Logic of Knowledge

- Probably the most important fact about knowledge that we will want to capture is that agent can reason on the basis of their knowledge.
- All our examples depends on the assumption that, if an agent trying to solve a problem has all the relevant information, he will apply his knowledge to produce a solution.

A Modal Logic of Knowledge

- The principle people normally use in reasoning about what other people know seems to be something like this:

  If we can infer that something is a consequence of what someone knows, then , lacking information to the contrary, we will assume that the other person can draw the same inference.

- We will make the simplifying assumption that agents actually do draw all logically valid inferences from their knowledge.

Common Knowledge

- Finally, we will need to include the fact that these basic properties of knowledge are themselves common knowledge.
- By this we mean that everyone knows these, and everyone knows that everyone knows them, ad infinitum.

A Modal Logic of Knowledge

- The language we will use initially is that of propositional logic, augmented by an operator Know and terms denoted agents.
- The formula Know(A, P) is interpreted to mean that the agent denoted by the term A knows the proposition expressed by the formula P.
- So, if John denoted John and Likes(Bill, Marry) means that Bill likes Mary,

  Know(John, Likes(Bill, Marry)) means that John knows that Bill likes Mary.

The axioms of the logic are inductively defined as all instances of the following schemata:

$M1.$ P, such that P is an axiom of the ordinary propositional logic.

$$M2.\ Know(A,P) \to P$$

$$M3.\ Know(A,P) \to Know(A,(Know(A,P)))$$

$$M4.\ Know(A,(P \to Q)) \to (Know(A,P) \to Know(A,Q))$$

Closed under the principle that:

M5. If P is an axiom, then $Know(A,P)$ is an axiom.

The Theorems of the Logic:

- The closure of the axioms under the inference rule modus ponens (from $(P \rightarrow Q)$ and P, infer Q) defines the theorems of the system.
- This system is very similar to those studied in modal logic. In fact, if A is held Fixed, the resulting system is isomorphic to the modal logic S4. We will refer to this system as the modal logic of knowledge.

A Possible-Worlds Analysis of Knowledge

- Kripke (1963) introduced the idea that a world should be regarded as possible, not absolutely, but only relative to other worlds.
- The relation of one world’s being a possible alternative to another is called the accessibility relation.

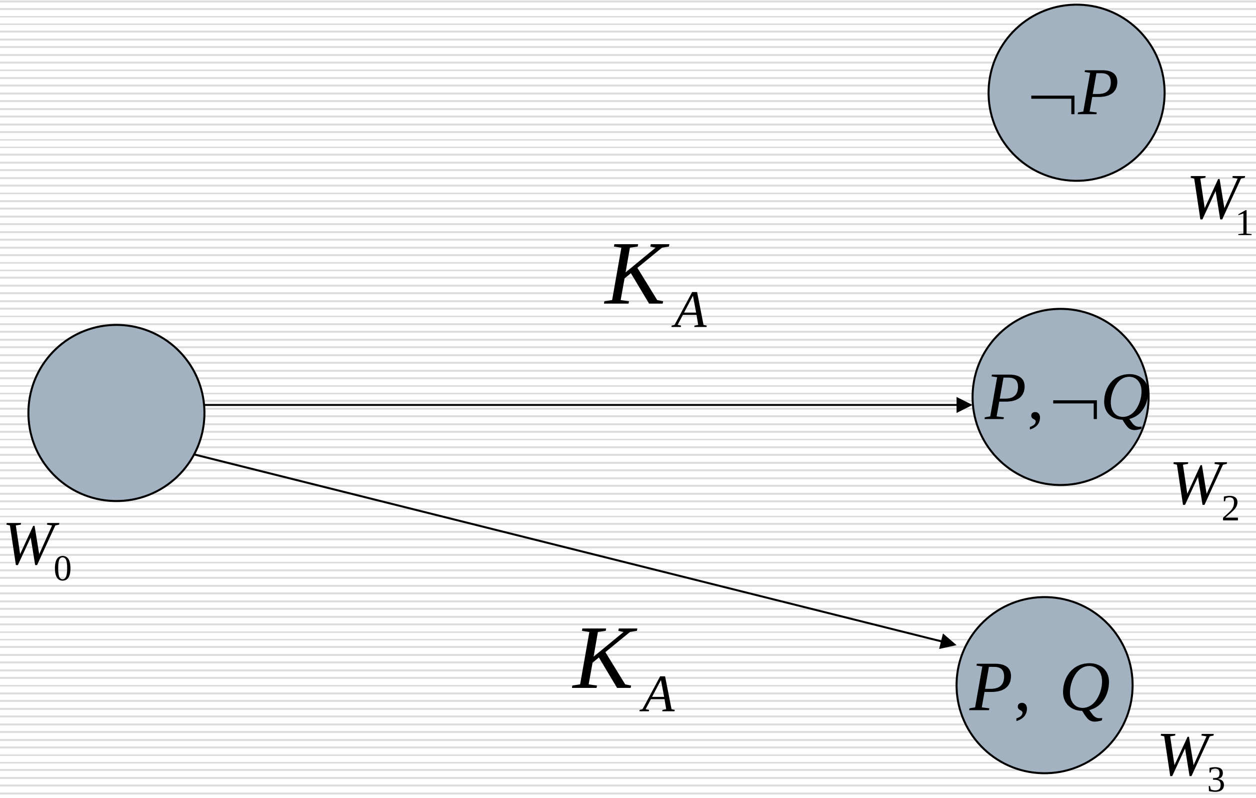
A Possible-Worlds Analysis of Knowledge

- To analyze statements of the form Know(A, P), we will introduce a relation K, such that K(A, W_1, W_2) means that the possible world W_2 is compatible or consistent with what A knows in the possible world W_1.
  In other worlds, for all that A knows in W_1, he might just as well be in W_2. It is the set of worlds $\{w_2 \mid K(A, W_1, w_2)\}$ that we will use to characterize what A knows in W_1.

A Possible-Worlds Analysis of Knowledge

- We will discuss A’s knowledge in W in terms of this set, the set of states of affairs that are consistent with his knowledge in W, rather than in terms of the set of propositions he knows.

A knows that P but A doesn't know whether Q.



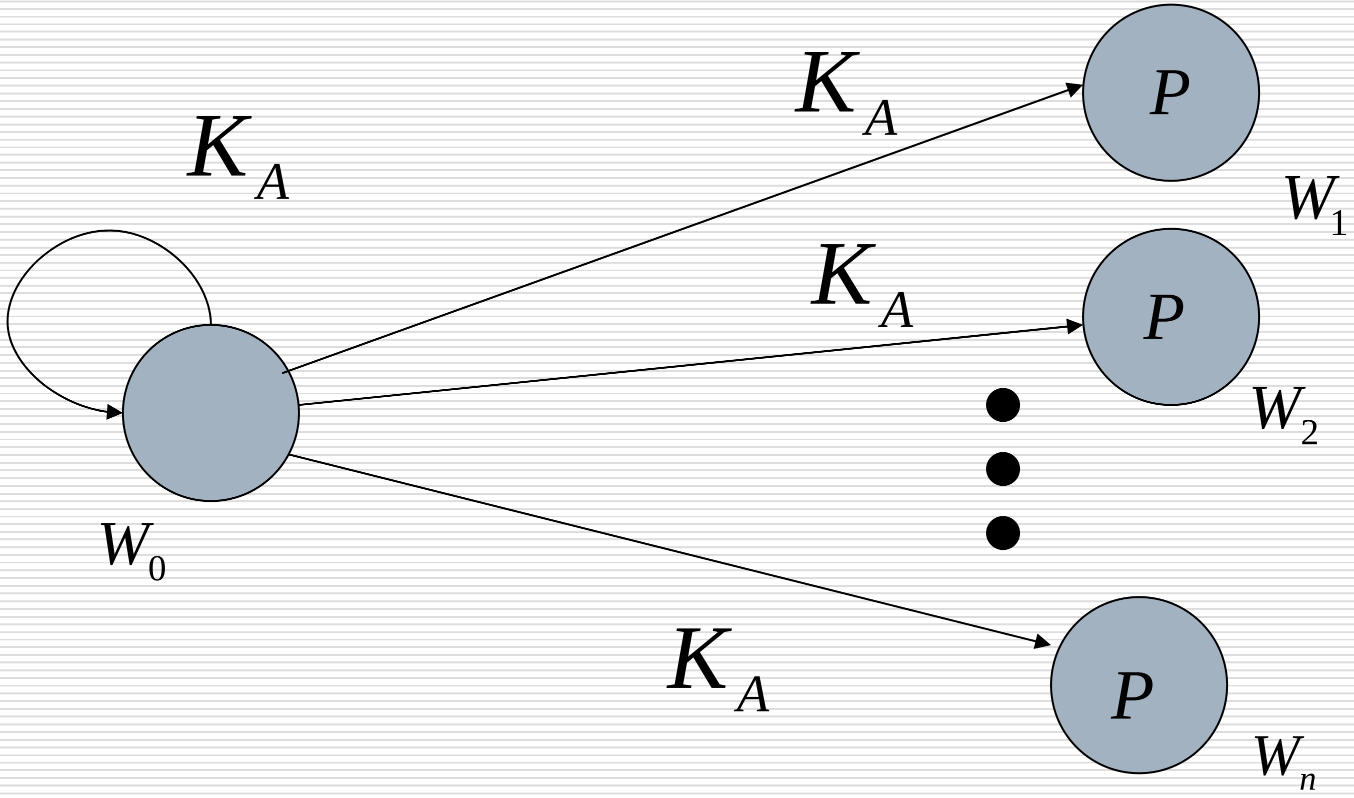
A Possible-Worlds Analysis of Knowledge

- Some of the properties of knowledge can be captured by putting constraints on the accessibility relation K.
- For instance, requiring that the actual world W_0 be compatible with what each knower knows in W_0, i.e., $\forall a_1 (K(a_1, W_0, W_0))$, is equivalent to saying that anything that is known is true.

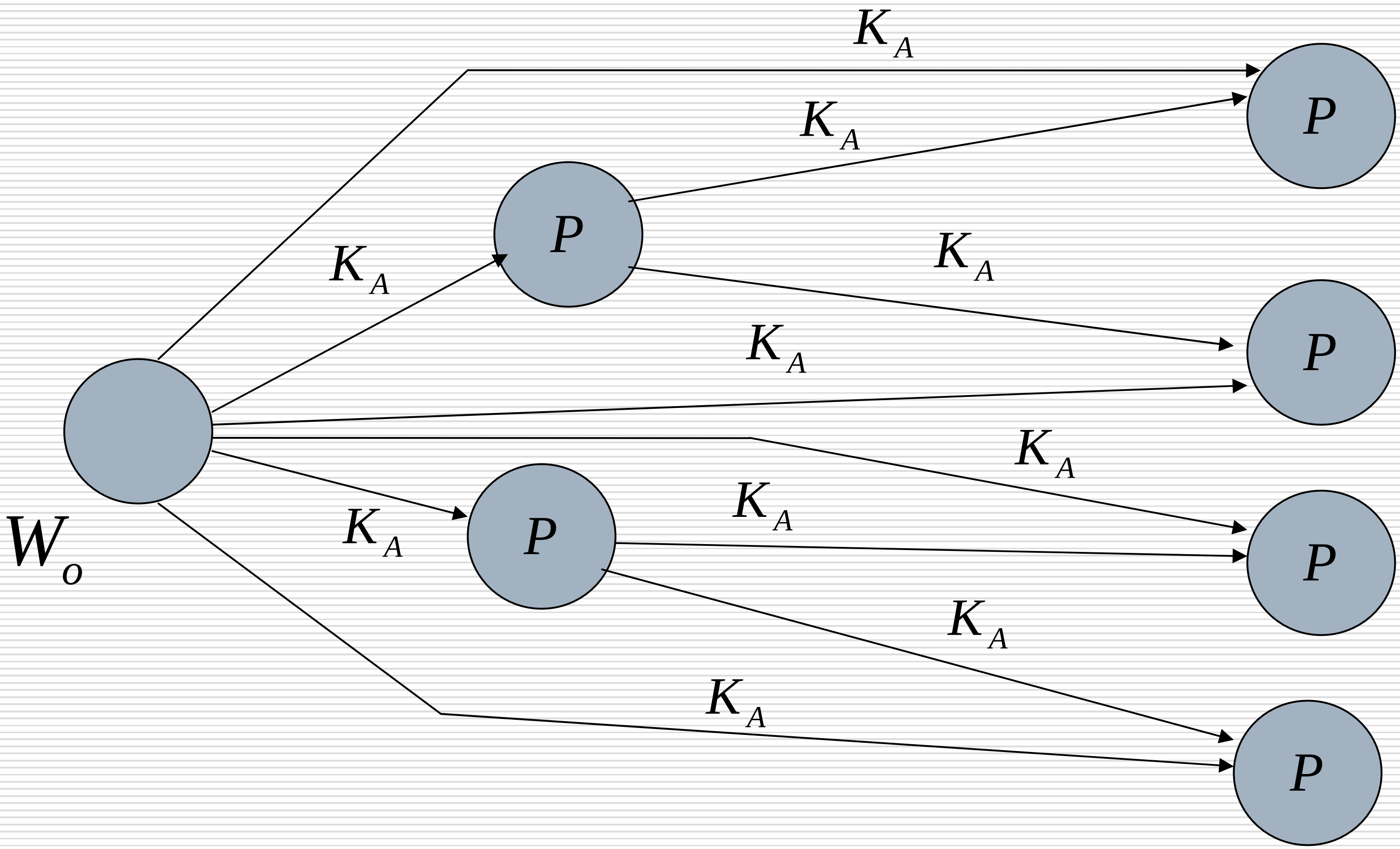
A Possible-Worlds Analysis of Knowledge

- The definition of K implies that , if A knows that P in W_0 , then P must be true in every world W_1 such that $K(A, W_0, W_1)$.
- To capture the fact that agents can reason with their knowledge, we will assume the converse is also true.

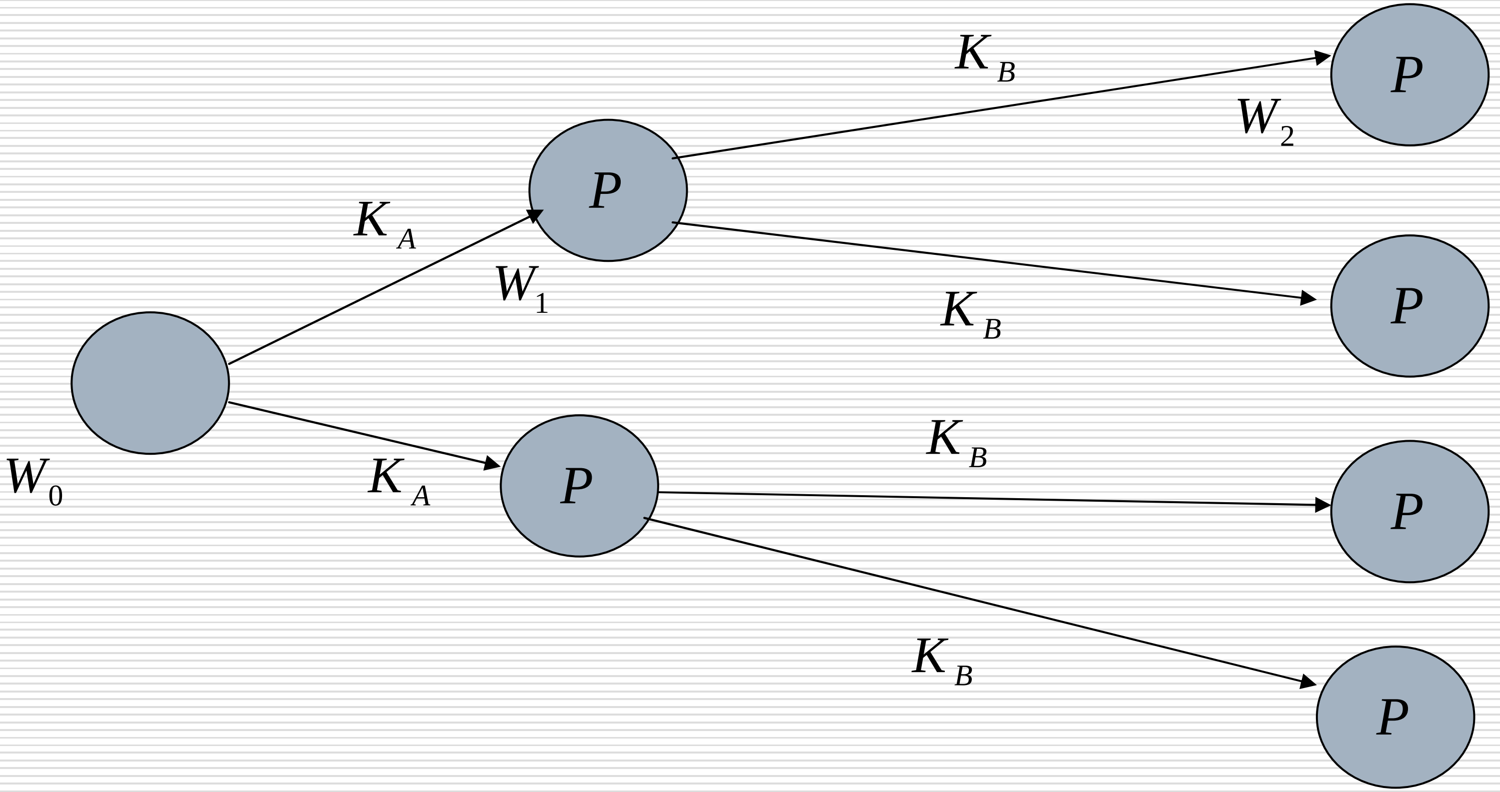
P is true in every world that is compatible with what A knows.



If A knows that P, then he knows that he knows that P



A knows that B knows that P



A Possible-Worlds Analysis of Knowledge

- Given these constraints and assumptions, whenever we want to assert or deduce something that would de expressed in the modal logic of knowledge by Know(A,P), we can instead assert or deduce that P is true in every world that is compatible with what A knows.
- We can express this in ordinary first-order logic, by treating possible worlds as individuals, so that K is just an ordinary relation.

A Possible-Worlds Analysis of Knowledge

- We will therefore introduce an operator T such that T(W,P) means that the formula P is true in the possible world W.
- If we let W denote the actual world, we can convert the assertion Know(A,P) into

$$\forall w_1 (K(A, W, w_1) \rightarrow T(w_1, P))$$

A Possible-Worlds Analysis of Knowledge

- It may seem that we have not made any real progress, although we have gotten rid of one nonstandard operator, Know, we have introduced another one, T.
- However, T has an important property that Know does not. Namely, T "distributes" over ordinary logical operators. So we can transform any formula so that T is applied only to atomic formulas.
- We can then turn T into an ordinary first-order relation by treating all the atomic formulas as names of atomic propositions, or we can get rid of T by replacing the atomic formulas with predicates on possible worlds.

Knowledge, Equality and Quantification

- The formalization of knowledge presented so far is purely propositional; a number of additional problems arise when we attempt to extend the theory to handle equality and quantification.
- For instance, we are not entitled to infer Know(A,P(C)) from B=C and Know(A,P(B)) because A might not know that the identity holds.

The Equality

- The possible-world analysis of knowledge provides a very neat solution to this problem, once we realize that a term can denote different objects in different possible worlds.
- For instance, if B is the expression “the number of planets” and C is “nine”, then, although B=C is true in the actual world, it would be false in a world in which there was a tenth planet.
- Thus, we will say that an equality statement such as B=C is true in a possible world W just in case the denotation of the term B in W is the same as the denotation of the term C in W.

The Equality

- Given this interpretation, the inference of Know(A,P(C)) from B=C and Know(A,P(B)) will be blocked (as it should be).
- To infer Know(A,P(C)) from Know(A,P(B)) by identity substitution, we would have to know that B and C denote the same object in every world compatible with what A knows, but the truth of B=C guarantees only that they denote the same object in the actual world.

The Equality

- On the other hand, if Know(A,P(B)) and Know(A,(B=C)) are both true, then in all worlds that are compatible with what A knows, the denotation of B is in the extension of P and is the same as the denotation of C; hence, the denotation of C is in the extension of P. From this we can infer that Know(A,P(C)) is true.

The Quantification

- The introduction of quantifiers also causes problems.
- For example, consider the sentence “Ralph knows that someone is a spy”.
- This sentence has at least two interpretations:
  - One is that Ralph knows that there is at least one person who is a spy, although he may have no idea who that person is.
  - The other interpretation is that there is a particular person whom Ralph knows to be a spy.

The Quantification

- This ambiguity can be explain as a difference of scope.
- The idea is that indefinite noun phrases such as “someone” can be analyzed in context by paraphrasing sentences of the form P(“someone”) as “There exists a person x such that P(x)”, or, more formally, $\exists x(Person(x) \wedge P(x))$

The Quantification

- So in the sentence “A knows that someone is a P” we can eliminate “someone” by applying the rule to either the whole sentence or only the subordinate clause, “someone is a P”.

The Quantification

- So we have two representation for the sentence "Ralph knows that someone is a spy".

$$(1)\ Know(Ralph, \exists x(Person(x) \wedge Spy(x)))$$

$$(2)\ \exists x(Person(x) \wedge Know(Palph, Spy(x)))$$

The Quantification

- Another option to write formula similar to (2) is to point out that a sentence of the form “A knows who (or what) B is” intuitively seems to be equivalent to “there is someone (or something) that A knows to be B”. But this can be represented formally as $\exists x(\mathrm{Know}(A,(x=B))$
- To take a specific example, ”John knows who the President is” can be paraphrased as “There is someone whom John knows to be the President”, in formal representation:

$$(3)\ \exists x(Know(John,(x=\mathrm{President})))$$

The Quantification

- The possible-world analysis, however, provides us with a very natural interpretation that $\exists x(P(x))$ is true just in case there is some value x satisfies P.
- If P is Know(A,Q(x)), then a value for x satisfies P(x) just in case that value satisfies Q(x) in every world that is compatible with what A knows. So (2) is satisfied if there is a particular person who is a spy in every world that is compatible with what A knows. That is, in every such world the same person is a spy.
- On the other hand, (1) is satisfied if, in every world compatible with that A knows, there is some person who is a spy, but it does not have to be the same one in each case.

The Quantification

- For instance, the proposition that John knows that 321-1234 is Bill's telephone number might be represented as

  (4) $Know(John, (321\text{-}1234 = Phone\text{-}num(Bill)))$

  which does not involve quantifying in.
- We would want to able to infer from this, however, that John knows what Bill's telephone number is, which would be represented as

  (5) $\exists x(Know(John, (x = Phone\text{-}num(Bill))))$

The Quantification

- It might seem that (5) can be derived from (4) simply by the logical principle of existential generalization, but that principle is not always valid in knowledge contexts.
- Suppose that (4) were not true, but that instead John simply knew that Mary and Bill had the same telephone number. We could represent this as

(6) Know(John,(Phone-num(Mary)=Phone-num(Bill)))

The Quantification

- It is clear that we would not want to infer from (6) that John knows what Bill’s telephone number is – yet, if existential generalization were universally valid in knowledge contexts, this inference would be valid.

Standard Identifiers

- It therefore seems that, in knowledge contexts, existential generalization can applied to some referring expressions (“321-1234”), but not to others (“Mary’s telephone number”).
- We will call the expressions to which existential generalization can applied standard identifiers, since they seem to be the ones an agent would use to identify an object for another agent.

Rigid Designators

- Standard identifiers are simply terms that have the same denotations in every possible world.
- Following Kripke (1972) we will call terms that have the same denotation in every possible world rigid designators.
- The conclusion that standard identifiers are rigid designators seems inescapable.

A Possible-Worlds Analysis of Action

A Possible-Worlds Analysis of Action

- In the precedent section, we have present a framework for describing what someone knows in terms of possible world.
- To characterize the relation of knowledge to action, we need a theory of action in these same terms.
- Most AI programs that reason about action are based on a view of the world as a set of possible states of affairs, with each action determining a binary relation between states of affairs – one being the outcome of performing the action in the other.

A Possible-Worlds Analysis of Action

- Knowledge about the past and future can be handled by modal tense operators, with corresponding accessibility relations between possible worlds.
  We could have a tense oparator Future such that Future(P) means that P will be true at some time to come. If we let F be an accessibility relation such that $F(W_1, W_2)$ means that the world W_2 lies in the future of the world W_1, then we can define Future(P) to be true in W_1 just in case there is some W_2 such that $F(W_1, W_2)$ holds and P is true in W_2.

The Situation Calculus

- The situation calculus is a first-order language in which predicates that can vary in truth-value over time are given an extra argument to indicate what situations they hold in, with a function Result that maps an agent, an action, and a situation into the situation that result from the agent's performance of the action in the first situation.
- Statements about the effects of actions are then expressed by formulas like P(Result(A,Act,S)), which means that P is true in the situation that results from A's performing Act in situation S.

A Possible-Worlds Analysis of Action

- To integrate these ideas into our logic of knowledge, we will reconstruct the situation the situation calculus as a modal logic.
- In parallel to the operator Know for talking about knowledge, we introduce an object language operator Res for talking about the results of events.

A Possible-Worlds Analysis of Action

- Res will be a two-place operator whose first argument is a term denoting an event, and whose second argument is a formula.
- Res(E,P) will mean that it is possible for the event E to occur and that, if it did, the formula P would then be true.
- The possible-world semantics for Res will be specified in terms of an accessibility relation R, parallel to K, such that $R(:E, W_1, W_2)$ means that W_2 is the situation that would result from the event :E happening W_1.

A Possible-Worlds Analysis of Action

R1. $\forall w_1, w_2, w_3, e_1((R(e_1, w_1, w_2) \wedge R(e_1, w_1, w_3)) \rightarrow (w_2 = w_3))$

R2. $\forall w_1, t_1, p_1$

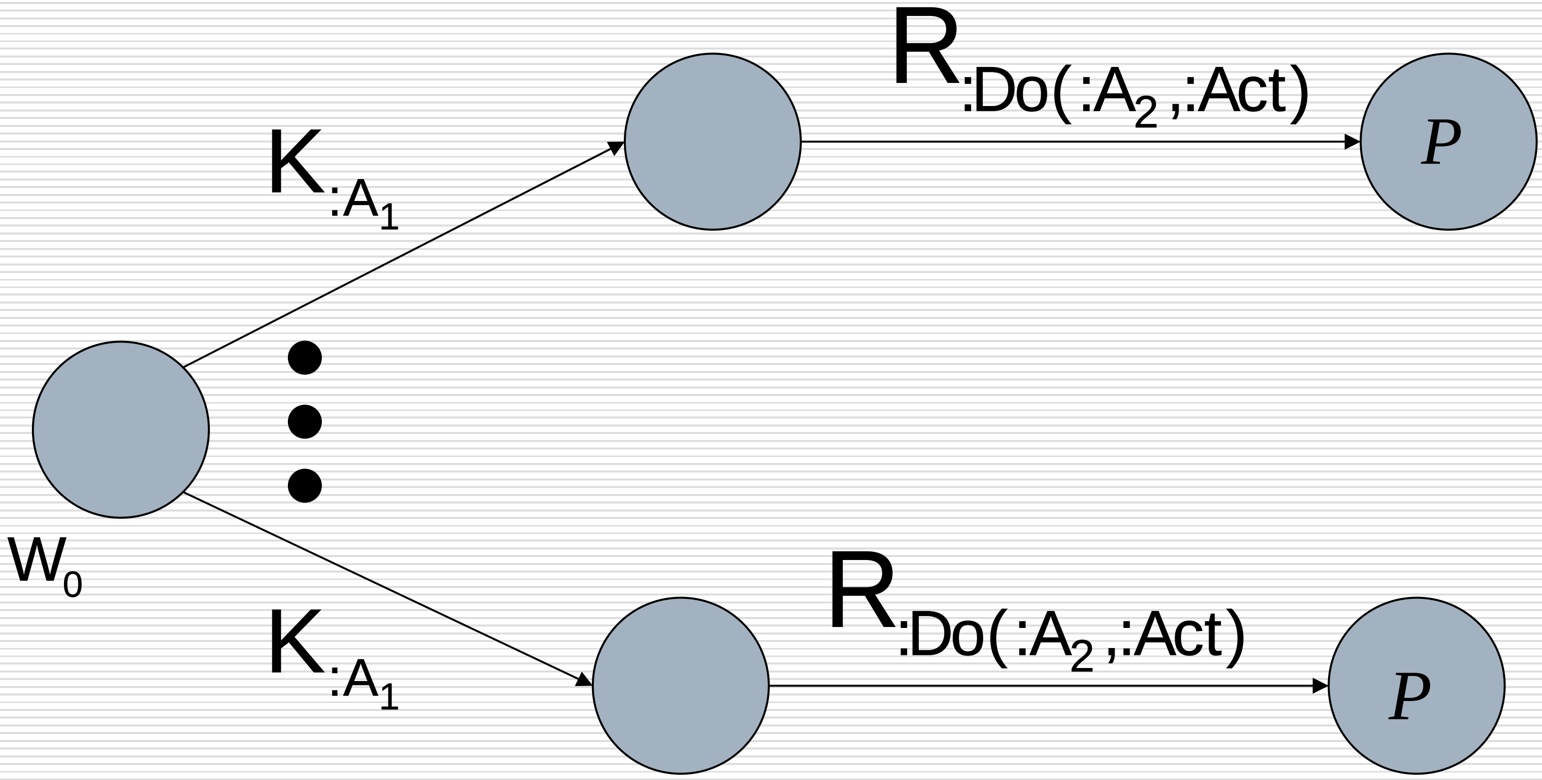
$(T(w_1, Res(t_1, p_1)) \equiv \exists w_2(R(D(w_1, t_1), w_1, w_2) \wedge T(w_2, p_1)))$

A Possible-Worlds Analysis of Action

- The type of event we will normally be concerned with is the performance of an action by an agent.
- We will let Do(A,Act) be a description of the event consisting of the agent denoted by A performing the action denoted by Act.
- We will want Do(A,Act) to be the standard way to referring to the event of A's carrying out the action Act, so Do will be a rigid function.
- Hence, Do(A,Act) will be a rigid designator of an event if A is a rigid designator of an agent and Act a rigid designator of an action.

$$\mathrm{True}(\mathrm{Know}(A_1, \mathrm{Res}(\mathrm{Do}(A_2, \mathrm{Act}), P))) \equiv$$

$$\forall w_1 (K(:A_1, W_0, w_1) \rightarrow \exists w_2 (R(:\mathrm{Do}(:A_2, :\mathrm{Act}), w_1, w_2) \wedge T(w_2, P)))$$



Complex Combinations of Actions

R3. $\forall w_1, t_1, t_2, t_3, p_1$

$((T(w_1, p_1) \rightarrow$

$(D(w_1, Do(t_1, If(p_1, t_2, t_3))) = D(w_1, Do(t_1, t_2)))) \wedge$

$(\neg T(w_1, p_1) \rightarrow$

$(D(w_1, Do(t_1, If(p_1, t_2, t_3))) = D(w_1, Do(t_1, t_3)))))$

R4. $\forall w_1, t_1, t_2, p_1$

$(D(w_1, Do(t_1, While(p_1, t_2))) =$

$D(w_1, Do(t_1, If(p_1, (t_2; While(p_1, t_2)), Nil)))$

Complex Combinations of Actions

- To define the denotation of events that consist or carrying out action sequences, we need some notation for talking about sequences of events.
- First, we will let “;” be a polymorphic operator in the object language, creating descriptions of event sequences in addition to action sequences.

Complex Combinations of Actions

- Speaking informally, if E1 and E2 are event descriptions, then (E1;E2) names the event sequence consisting of E1 following by E2, just as (Act1;Act2) names the action sequence consisting of Act1 followed by Act2.
- In the metalanguage, event sequences will be indicated with angle brackets, so that <:E1,:E2> will mean :E1 followed by :E2.

Complex Combinations of Actions

R5. $\forall w_1, t_1, t_2, t_3$

$$(D(w_1, Do(t_1,(t_2;t_3))) =$$
$$D(w_1, Do(t_1,t_2);Do(@(D(w_1,t_1)),t_3))))$$

R6. $\forall w_1, w_2, t_1, t_2$

$$(R(D(w_1,t_1), w_1, w_2) \rightarrow$$
$$(D(w_1,(t_1;t_2)) = \langle D(w_1,t_1), D(w_2,t_2) \rangle))$$

Complex Combinations of Actions

- Finally, we need to define the accessibility relation R for event sequences and for events in which the null action is carried out.

$$\text{R7. } \forall w_1, w_2, e_1, e_2 \; (R(\langle e_1, e_2 \rangle w_1, w_2) \equiv \exists w_3 (R(e_1, w_1, w_3) \wedge R(e_2, w_3, w_2)))$$

$$\text{R8. } \forall w_1, a_1 (R(:Do(a_1, :Nil), w_1, w_1))$$

An Integrated Theory of knowledge and Action

The Dependence of Action on Knowledge

- As we point out in the introduction, knowledge and action interact in two principal ways:
  - Knowledge is often required prior to taking action
  - Actions can change what is known.

The Dependence of Action on Knowledge

- Our main thesis is that the knowledge prerequisites for an action can be analyzed as a matter of knowledge what action to take.

The Dependence of Action on Knowledge

- Recall the example of trying to open a locked safe.
- Why is it that, for an agent to achieve this goal by using the plan “Dial the combination of the safe”, he must to know the combination?
- The reason is that an agent could know that dialing the combination of the safe would result in the safe’s being open, but still not know what to do because he does not know what the combination of safe is.

The Dependence of Action on Knowledge

- In our possible-world semantics of knowledge , the usual way of knowing what entity is referred to by a description B is by having some description C that is a rigid designator, and by knowing that B=C.
- In particular, knowing what action is referred to by an action description means having a rigid designator for the action described.
- But, if this is all the knowledge that is required for carrying out the action, then a rigid designator for an action must be an executable description of the action-in the same sense that a computer program is an executable description of a computation to an interpreter for the language in which the program is written.

The Dependence of Action on Knowledge

- In many of these cases, if an agent knows the general procedure and what objects the procedure is to be applied to, then he knows everything that is relevant to the task.
- That is, we assume that anyone who knows what combination he is to dial and what safe he is to dial it on thereby knows what action he is to perform.

The Dependence of Action on Knowledge

- There are other procedures we might also wish to assume that any one could perform, but that cannot be represented as rigid functions.
- Suppose that, in the blocks world, we let Puton(B,C) denote the action of putting B on C. Even though we would not want to question anyone's ability to perform Puton in general, knowing what objects B and C are will not de sufficient to perform Puton(B,C); knowing where they are is also necessary.

The Dependence of Action on Knowledge

- We could have a special axiom stating that knowing what action Puton(B,C) is requires knowing where B and C are, but this will be superfluous if we simply assume that everyone knows the definition of Puton in terms of more primitive actions.
- If we define Puton(X,Y) as something like:

(Movehand(Location(X));
Grasp;
Movehand(Location(Top(Y)));
Ungrasp;)

The Dependence of Action on Knowledge

- To formalize this theory, we will introduce a new object language operator Can.
- Can(A,Act,P) will mean that A can achieve P by performing Act.
- We will not give a possible-world semantics for Can directly; instead we will give a definition of Can in terms of Know and Res.

$$\forall a(\exists x(Know(a,((x = Act) \wedge Res(Do(a,Act),P)))) \rightarrow Can(a,Act,P))$$

Complex Action

- For an agent to be able to achieve a goal by performing a complex action, all that is really necessary is that he know what to do first, and that he know that he will know what to do at each subsequent step.

$$\forall a(\exists x(Know(a,((Do(a,(x;Act_1))=Do(a,Act)\wedge$$
$$Res(Do(a,x),Can(a,Act_1,P)))))$$
$$\rightarrow Can(a,Act,P))$$

The Dependence of Action on Knowledge

- Finally, with the following metalanguage axiom we can state that these are only two conditions under which an agent can use a particular action to achieve a goal:

C1. $\forall w_1, t_1, t_2, t_3, p_1$

$$((t_2 = @(D(w_1, t_1))) \rightarrow$$
$$(T(w_1, Can(t_1, t_3, p_1)) \equiv$$
$$(T(w_1, Exist(X, Know(t_1, And(Eq(X, t_3), Res(Do(t_2, t_3), p_1))))) \vee$$
$$\exists t_4 (T(w_1, Exist(X, Know(t_1, And(Eq(Do(t_2, (X; t_4)), Do(t_2, t_3)),$$
$$Res(Do(t_2, X),$$
$$Can(t_2, t_4, p_1))))))))))$$

The Dependence of Action on Knowledge

Letting t_1 = A, $t_2 = A_1$ and t_3 = Act, C1 says that, for any formula P, if A_1 is the standard identifier of the agent denoted by A, then A can achieve P by doing Act if and only if :

- A knows what action Act is and knows that P would be true as a result of his doing Act
- Or there is an action description $t_4 = Act_1$ such that, for some action X, A knows that his doing X followed by Act_1 is the same event as his Act and knows that his doing X would result his being able to achieve P by doing Act_1 .

An Integrated Theory of knowledge and Action

- As a simple illustration of these concept, we will show how to derive the fact that an agent can open a safe, given the premise that he knows the combination.
- To do this, the only additional fact we need is that, if an agent does dial the correct combination of safe, the safe will the be open:

D1. $\forall w_1, a_1, x_1$

$(:\mathrm{Safe}(x_1) \rightarrow$

$\exists w_2 (R(:\mathrm{Do}(a_1, :\mathrm{Dial}(:\mathrm{Comb}(w_1, x_1), x_1)), w_1, w_2) \wedge$

$:\mathrm{Open}(w_2, x_1)))$

Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X))))))

Can(A,Dial(Comb(X),X),Open(X))))

1. $T(W_0,$	ASS
$\quad And(Safe(@(x_1)),$	
$\quad\quad Exist(Y,Know(A,Eq(Y,Comb(@(x_1))))))))$	
2. $:Safe(x_1)$	1, L2, L9
3. $\forall w_1(K(:A(W_0),W_0,w_1) \rightarrow$	1, L2, L7, K1, L11,
$\quad (:C=:Comb(w_1,x_1)))$	L13, L10, L12
4. $K(:A(W_0),W_0,w_1)$	ASS
5. $:C=:Comb(w_1,x_1)$	3, 4
6. $:Dial(:C,x_1)=:Dial(:Comb(w_1,x_1),x_1)$	5
7. $T(w_1,$	L10, L12, L12a,
$\quad Eq(@(:Dial(:C,x_1)),$	L13
$\quad\quad Dial(Comb(@(x_1)),@(x_1))))$	

Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X)))))) Can(A,Dial(Comb(X),X),Open(X))))

8. $\exists w_2(R(:Do(:A(W_0),$ $:Dial(:Comb(w_1,x_1),x_1)),$ $w_1,w_2) \wedge$ $:Open(w_2,x_1)))$ — 2, D1

9. $T(w_1,$ $Res(Do(@(D(W_0,A)),$ $Dial(Comb(@(x_1)),@(x_1))),$ $Open(@(x_1))))$ — L11, L10, L12a, L9, L2

Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X)))))

Can(A,Dial(Comb(X),X),Open(X))))

10. $T(w_1, And(Eq(@(:Dial(:C, x_1)),$ 7, 9, L2

$Dial(Comb(@(x_1)), @(x_1))),$

$Res(Do(@(D(W_0, A)),$

$Dial(Comb(@(x_1)), @(x_1))),$

$Open(@(x_1)))))$

11. $K(:A(W_0), W_0, w_1) \rightarrow$ DIS(4,10)

$T(w_1, And(Eq(@(:Dial(:C, x_1)),$

$Dial(Comb(@(x_1)), @(x_1))),$

$Res(Do(@(D(W_0, A)),$

$Dial(Comb(@(x_1)), @(x_1))),$

$Open(@(x_1)))))$

Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X))))))
Can(A,Dial(Comb(X),X),Open(X))))

12. T(W_0, 11, L11, K1
Know(A,
And(Eq(@(:Dial(:C,x_1)),
Dial(Comb(@(x_1)),@(x_1))),
Res(Do(@(D(W_0,A)),
Dial(Comb(@(x_1)),@(x_1))),
Open(@(x_1))))))

Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X)))))) Can(A,Dial(Comb(X),X),Open(X))))

13. $T(W_0$,Exist(X, Know(A, And(Eq(X, Dial(Comb(@(x_1)), @(x_1))), Res(Do(@(D(W_0,A)), Dial(Comb(@(x_1)), @(x_1))), Open(@(x_1))))) 12, L7

Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X)))))) Can(A,Dial(Comb(X),X),Open(X))))

14. $T(W_0,$ 13,C1

$Can(A,$

$Dial(Comb(@(x_1)),@(x_1)),$

$Open(@(x_1))))$

15. $T(W_0,$ DIS(1,14)

$And(Safe(@(x_1)),$

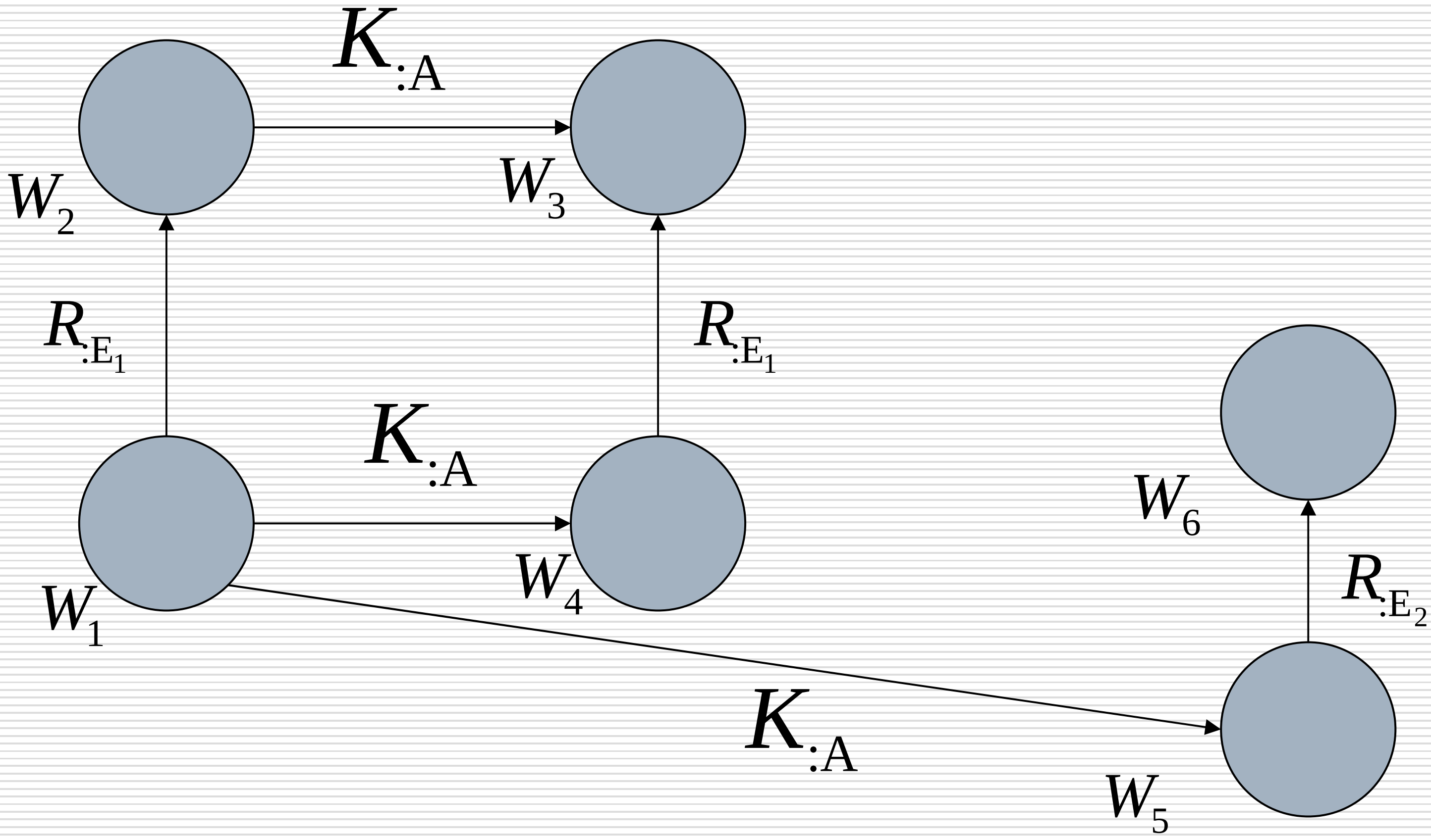
$Exist(Y,Know(A,Eq(Y,Comb(@(x_1)))))))) \rightarrow$

$T(W_0,Can(A,Dial(Comb(@(x_1))@(x_1)),Open(@(x_1))))$

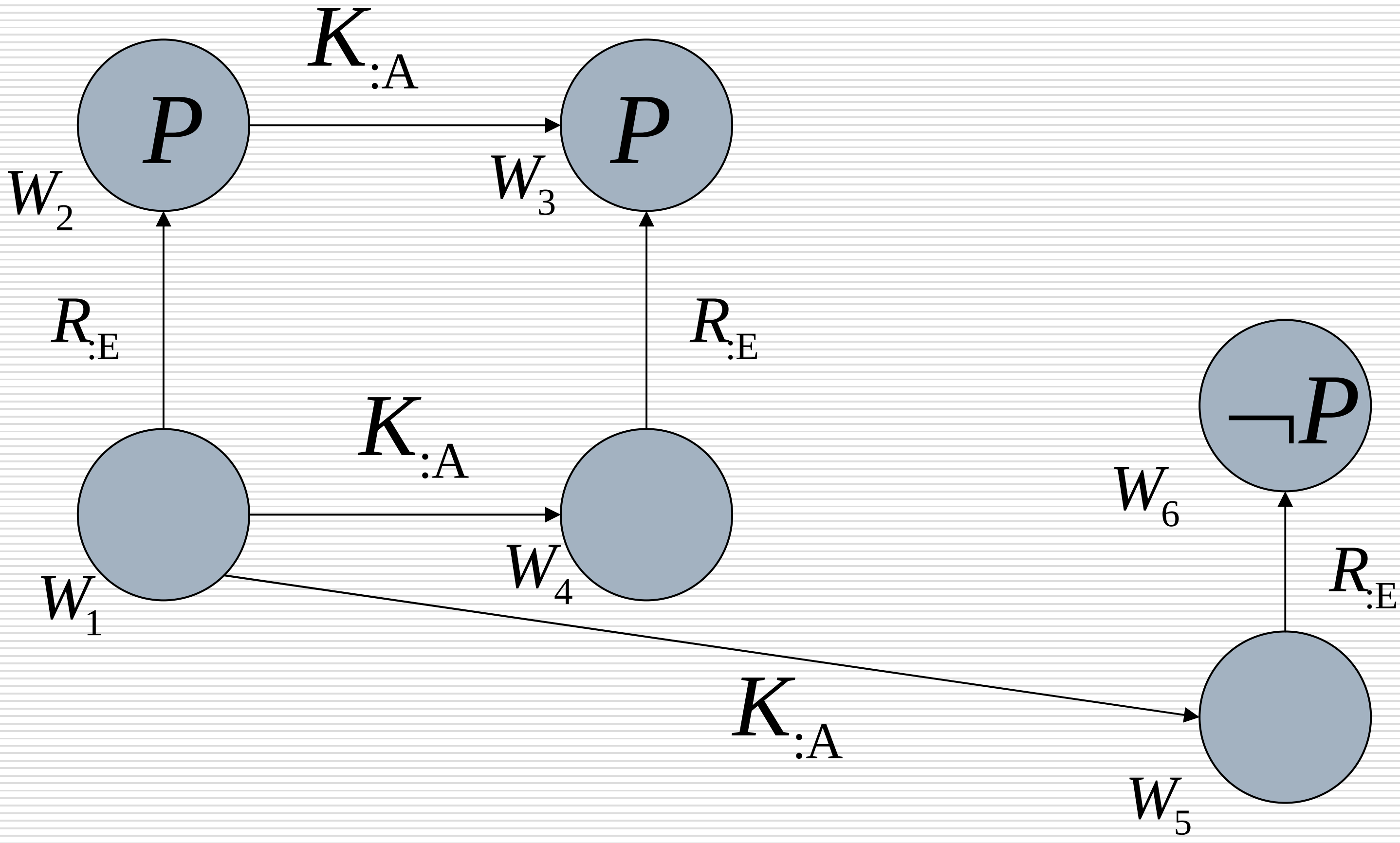
Prove: True(All(X,Imp(And(Safe(X),Exist(Y,Know(A,Eq(Y,Comb(X))))))
Can(A,Dial(Comb(X),X),Open(X))))

16. True(All(X, 15, L4, L8, L1
Imp(And(Safe(X),
Exist(Y,
Know(A,
Eq(Y,Comb(X)))))),
Can(A,Dial(Comb(X),X),Open(X))))

The effect of a noninformative action on the agent's knowledge



The effect of a informative action on the agent's knowledge



The Effect of Action on Knowledge

$$\text{T1. } \forall w_1, w_2, a_1$$
$$(R(Do(a_1, :Test), w_1, w_2) \rightarrow$$
$$\forall w_3 (K(a_1, w_2, w_3) \equiv$$
$$(\exists w_4 (K(a_1, w_1, w_4) \wedge R(:Do(a_1, :Test), w_4, w_3)) \wedge$$
$$(:Red(w_2) \equiv :Red(w_3)))))$$

Given: True(Know(A, Imp(Acid,Res(Do(A, Test), And(Acid,Red)))))

True(Know(A, Imp(Not(Acid),Res(Do(A, Test),

And(Not(Acid),Not(Red))))))

True(Acid)

Prove: True(Res(Do(A, Test),Know(A, Acid)))

1. $\forall w_1 (K(:A, W_0, w_1) \rightarrow$ Given, L1, L4, R2,

$(:Acid(w_1) \rightarrow$ L2, L9, L12, L11a

$\exists w_2 (R(:Do(:A, :Test), w_1, w_2) \wedge$

$:Acid(w_2) \wedge :Red(w_2))))$

2. $\forall w_1 (K(:A, W_0, w_1) \rightarrow$ Given, L1, L4, R2, L2,

$(\neg :Acid(w_1) \rightarrow$ L6, L9, L12, L11a

$\exists w_2 (R(:Do(:A, :Test), w_1, w_2) \wedge$

$\neg :Acid(w_2) \wedge \neg :Red(w_2))))$

$$3.\ :Acid(W_0) \qquad\qquad \text{L1, L9}$$

$$4.\ :Acid(W_0) \rightarrow$$

$$\exists w_2 (R(:Do(:A,:Test), w_1, w_2) \wedge \qquad\qquad \text{1, K2}$$

$$:Acid(w_2) \wedge :Red(w_2))$$

$$5.\ R(:Do(:A,:Test), W_0, W_1) \qquad\qquad \text{3, 4}$$

$$6.\ :Red(W_1) \qquad\qquad \text{3, 4}$$

$$7.\ \forall w_2 (K(:A, W_0, w_2) \equiv \qquad\qquad \text{5, T1}$$

$$(\exists w_3 (K(:A, W_0, w_3) \wedge$$

$$R(:Do(:A,:Test), w_1, w_2) \wedge$$

$$(:Red(W_1) \equiv :Red(w_2))))$$

$8.\ K(:A, W_1, w_2)$ ASS

$9.\ K(:A, W_0, W_3)$ 7, 8

$10.\ R(:Do(:A,:test), W_3, w_2)$ 7, 8

$11.\ :Red(W_1) \equiv :Red(w_2)$ 7, 8

$12.\ :Red(w_2)$ 6, 11

$13.\ \neg :Acid(W_3) \rightarrow$ 2, 9

$$\exists w_4 (R(:Do(:A,:Test), W_3, w_4) \wedge$$

$$\neg :Acid(w_4) \wedge \neg :Red(w_4))$$

14. $\neg :Acid(W_3)$ ASS

15. $R(:Do(:A,:Test),W_3,W_4)$ 13, 14

16. $\neg :Red(W_4)$ 13, 14

17. $w_2 = W_4$ 15, R1

18. $\neg :Red(w_2)$ 16, 17

19. False 12, 18

20. $:Acid(W_3)$ DIS(14,19)

21. $:Acid(W_3) \rightarrow$ 1, 9

$$\exists w_4 (R(:Do(:A,:Test),W_3,w_4) \wedge$$

$$:Acid(w_4) \wedge :Red(w_4))$$

22. $R(:Do(:A,:Test),W_3,W_4)$	20, 21
23. $:Acid(W_4)$	20, 21
24. $w_2 = W_4$	15, 22
25. $:Acid(w_2)$	23, 24
26. $K(:A,W_1,w_2) \rightarrow :Acid(w_2)$	DIS(8, 25)
27. $R(:Do(:A,:Test),W_0,W_1) \wedge \forall w_2(K(:A,W_1,w_2) \rightarrow :Acid(w_2))$	5, 26
28. True(Res(Do(A,Test),Know(A,Acid)))	27, L9, L11a,L12, K2,R2,L1

The effect of the test on the agent's knowledge