Computational Logic, Human Thinking and Action

Robert Kowalski
Imperial College London
Conclusions

The task of an intelligent agent is to make its goals true in the world as seen through its beliefs.
Conclusions

The task of an intelligent agent is to make its goals true in the world as seen through its beliefs.

• Goals
  o are more fundamental than beliefs
  o resemble production system rules

• Beliefs
  o have the form of logic programs
Conclusions

The task of an intelligent agent is to make its goals true in the world as seen through its beliefs.

• Goals
  o are more fundamental than beliefs
  o resemble production system rules

• Beliefs
  o have the form of logic programs

• Computational Logic (CL)
  o combines goal and beliefs
  o embeds abductive logic programming (ALP) in an agent cycle
Outline of the talk

Overview

Logic programs represent beliefs

Production systems represent goals (but have no logic)

Computational Logic combines goals and beliefs embedded in an observe-think-decide-act agent cycle
An agent’s task in life is to perform actions to make its goals and observations true.
Goal: $\text{fire}(T) \rightarrow \text{eliminate}(T+1) \lor \text{escape}(T+2)$
Observation: $\text{fire}(1)$
Action: $\text{escape}(3)$

Goal and observation are true in the model of the world described by

$\{\text{fire}(1), \text{escape}(3)\}$
Goal: \( \text{threat}(T) \rightarrow \text{eliminate}(T+1) \lor \text{escape}(T+2) \)  
Belief: \( \text{threat}(T) \leftarrow \text{fire}(T) \)  
Observation: \( \text{fire}(1) \)  
Action: \( \text{escape}(3) \)  

Goal and observation are true in the model of the world described by \( \{ \text{fire}(1), \text{threat}(1), \text{escape}(3) \} \).
The distinction between goals and beliefs is the foundation of SBVR

SBVR – From Wikipedia:

“The Semantics of Business Vocabulary and Business Rules (SBVR) is an adopted standard of the Object Management Group (OMG) intended to be the basis for formal and detailed natural language declarative description of a complex entity, such as a business.”
The distinction between goals and beliefs is the foundation of SBVR

SBVR – From Wikipedia:

“The Semantics of Business Vocabulary and Business Rules (SBVR) is an adopted standard of the Object Management Group (OMG) intended to be the basis for formal and detailed natural language declarative description of a complex entity, such as a business.”

From Baisley, Hall and Chapin:

“Distinguishing between guidance (rules that people break) and structural rules (rules about meaning) is very important in understanding business rules.”
It is obligatory that each person on a bus has a ticket.

A person on a bus either has a ticket or is breaking the rule.

It is logically necessary that each person on a bus has a ticket. 

Being on a bus implies that there is a ticket.

These modalities are not nested as in normal modal logic.
It is obligatory that each person on a bus has a ticket.

A person on a bus either has a ticket or is breaking the rule.

It is logically necessary that each person on a bus has a ticket.

Being on a bus implies that there is a ticket.

These modalities are not nested as in normal modal logic.

Goal: \( a \text{ person is on a bus } \rightarrow \text{ the person has a ticket.} \)

Belief: \( a \text{ person is on a bus } \rightarrow \text{ the person has a ticket.} \)
The distinction between goals and beliefs is fundamental in database systems

Datalog rules = beliefs
\[ \text{manager}(X) \rightarrow \text{employee}(X) \]

Integrity constraints = goals
\[ \text{manager}(X) \rightarrow \exists Y \supset \text{supervises}(X, Y) \]
\[ \text{employee}(X), \text{employer}(X) \rightarrow \text{false} \]
\[ \text{supervises}(X, Y), \text{supervises}(X', Y) \rightarrow X = X' \]

From Datalog± (including ontologies and integrity constraints; Cali, Gottlob, Lukasziewicz; 2009)
Abductive logic programming (ALP) combines goals (integrity constraints) and beliefs (logic programs).

Beliefs:

\[
\text{conclusion if condition}_1 \text{ and } \ldots \text{ and condition}_n
\]

or:

\[
\forall X \ [\text{condition}_1 \land \ldots \land \text{condition}_n \rightarrow \text{conclusion}]
\]
Abductive logic programming (ALP) combines goals (integrity constraints) and beliefs (logic programs).

Beliefs:

\text{conclusion} \text{if condition}_1 \text{ and } \ldots \text{ and condition}_n \\
\text{or: } \forall X [\text{condition}_1 \land \ldots \land \text{condition}_n \rightarrow \text{conclusion}]

Maintenance goals:

\text{If } \text{condition}_1 \text{ and } \ldots \text{ and condition}_n \\
\text{then conclusion}_1 \text{ or } \ldots \text{ or conclusion}_m \\
\text{or: } \forall X [\text{condition}_1 \land \ldots \land \text{condition}_n \\
\rightarrow \exists Y [\text{conclusion}_1 \text{ or } \ldots \text{ or conclusion}_m]
Abductive logic programming (ALP) combines goals (integrity constraints) and beliefs (logic programs)

Beliefs:
\[ \text{conclusion if condition}_1 \text{ and .... and condition}_n \]
or:
\[ \forall X \left[ \text{condition}_1 \land \ldots \land \text{condition}_n \rightarrow \text{conclusion} \right] \]

Maintenance goals:
\[ \text{If condition}_1 \text{ and .... and condition}_n \]
then \[ \text{conclusion}_1 \text{ or .... or conclusion}_m \]
or:
\[ \forall X \left[ \text{condition}_1 \land \ldots \land \text{condition}_n \rightarrow \exists Y \left[ \text{conclusion}_1 \text{ or .... or conclusion}_m \right] \right] \]

It can be hard to tell the difference.
ALP agents combine forward and backward reasoning

- Consequences
- Goals
- Beliefs
- Alternative plans
- Consequences
- Actions
- Observations
- Alternative explanations
Forward and backward reasoning

Forward reasoning

Given \( A, B \) and \( \text{if } A \text{ and } B \text{ then } C \) derive \( C \).

Backward reasoning

Given goal \( C \) and \( C \text{ if } A \text{ and } B \) derive subgoals \( A \text{ and } B \)?
Forward reasoning derives consequences from assumption.

But also derives achievement goals from maintenance goals:

Given beliefs $A, B$
and maintenance goal if $A$ and $B$ then $C$!
derive achievement goal $C$!
Thinking about actions, beliefs and personal goals can all be described in terms of a common framework, which asserts that thinking consists of search and inference.

We search for certain objects and then make inferences from and about the objects we have found.” (page 6)
Baron’s view of thinking and deciding

- Achievement goal
- Search
- Alternative solutions
- Inference
- Consequences
- Decisions
- Actions
ALP agents need to make decisions

- Consequences
- Observations
- Alternative explanations
- Goals
- Decisions
- Consequences
- Alternative plans
- Beliefs
- Consequences
- Decisions
- Actions
The dual process model combines two systems of thinking

System 1 operates automatically and quickly, with little or no effort and no sense of voluntary control.

System 2 allocates attention to the effortful mental activities that demand it, including complex computations.
The dual process model

System 1 “quickly proposes intuitive answers to judgement problems as they arise”,

System 2 “monitors the quality of these proposals, which it may endorse, correct, or override”.

System 2 is activated when an event is detected that violates the model of the world that system 1 maintains.
The Dual Process Model

The world

System 1: Heuristic short cuts

System 2

Observations

Actions
ALP agents (CL) as a unifying framework

System 1: Heuristic short cuts

System 2

Alternative explanations

Consequences

Decisions

Goals

Beliefs

Consequences

Actions

Observations

Alternative plans
Outline of the talk

Overview

Logic programs represent beliefs

Production systems represent goals (but have no logic)

Computational Logic combines goals and beliefs embedded in an observe-think-decide-act agent cycle
The London underground emergency notice as a logic program

**Emergencies**

Press the alarm signal button **to** alert the driver.

The driver will stop
if any part of the train is in a station.

If **not**, the train will continue to the next station, where help can more easily be given.

There is a 50 pound penalty **for** improper use.
The hidden logic (+ control) of the Emergency Notice

Reason backwards to reduce goals to subgoals:

*the driver is alerted
*if you press the alarm signal button.*
The hidden logic (+ control) of the Emergency Notice

Reason forwards to derive possible consequences of actions:

the driver will stop the train in a station
if the driver is alerted
and any part of the train is in the station.

the driver will stop the train in the next station
if the driver is alerted
and not any part of the train is in a station.

help can more easily be given in an emergency
if the train is in a station.

You may be liable to a £50 penalty
if you use the alarm signal button improperly
Backward reasoning as a guide for clear thinking, writing and problem solving
Backward reasoning generates a pyramid (or triangle or and-or tree)
As Sherlock Holmes explained to Dr. Watson, in *A Study in Scarlet*:

“In solving a problem of this sort, the grand thing is to be able to reason backward. That is a very useful accomplishment, and a very easy one, but people do not practise it much.

In the everyday affairs of life, it is more useful to reason forward, and so the other comes to be neglected. There are fifty who can reason synthetically for one who can reason analytically.”
Sherlock Holmes used backward reasoning to generate hypotheses to explain observations. This is called abduction. He called it “deduction”.
Goals in logic programming are restricted to achievement goals

\[ \text{condition}_1 \land \text{condition}_2 \; \ldots \; \land \text{condition}_n ? \]

where \( \text{condition}_1 \) and \( \text{condition}_2 \; \ldots \; \text{condition}_n \) are atomic formulas or negations of atomic formulas.

Variables \( X \) represent values that need to be found.

\[ \exists \; X \left[ \text{condition}_1 \land \text{condition}_2 \; \ldots \; \land \text{condition}_n \right] ? \]
The logic programming view of thinking

Achievement goal

Alternative solutions

Backward or forward reasoning
Outline of the talk

Overview

Logic programs represent beliefs

Production systems represent goals (but have no logic)

Computational Logic combines goals and beliefs embedded in an observe-think-decide-act agent cycle
Production Systems  —Herbert A. Simon

Production systems are computer languages that are widely employed for representing the processes that operate in models of cognitive systems (Newell and Simon 1972).

In a production system, all of the instructions (called productions) take the form:

\[
\text{IF} \langle \text{conditions} \rangle, \text{THEN} \langle \text{actions} \rangle,
\]

That is to say, “if certain conditions are satisfied, then take the specified actions” (abbreviated C → A). Production system languages have great generality: they can possess the full power and generality of a Turing machine (see Turing). They have an obvious affinity to the classical stimulus-response (S → R) connections in psychology, but greater complexity and flexibility, for, in production systems, both
Production rules implement reactive rules, which are a kind of maintenance goals.

\[
\text{threat} \rightarrow \text{eliminate} \\
\text{threat} \rightarrow \text{escape}
\]

Production systems use “conflict resolution” to decide between conflicting actions. (They confuse “and” and “or”.)

In production rules, \(\rightarrow\) does not mean logical if-then. Change of state is implicit.
The Production System view of thinking

- Observations
- Reactive rules
- Check conditions
- Conflict resolution
- Actions

The world
Three kinds of production rules

• Reactive rules (or maintenance goals):

  \( \text{threat} \rightarrow \text{eliminate} \)
  \( \text{threat} \rightarrow \text{escape} \)
Three kinds of production rules

- Reactive rules (or maintenance goals):
  \[ \text{threat} \rightarrow \text{eliminate} \]
  \[ \text{threat} \rightarrow \text{escape} \]

- Logic programs (or beliefs) executed forward:
  \[ \text{fire} \rightarrow \text{threat} \]
Three kinds of production rules

• Reactive rules (or maintenance goals):
  \[ \text{threat} \to \text{eliminate} \]
  \[ \text{threat} \to \text{escape} \]

• Logic programs (or beliefs) executed forward:
  \[ \text{fire} \to \text{threat} \]

• Logic programs (or beliefs) executed backwards, simulated by forward chaining:
  \[ \text{eliminate} \to \text{get help} \]
  \[ \text{get help} \to \text{press the alarm} \]
Many authors are confused about the relationship between logic and production systems.

“Unlike logic, rule-based systems can easily represent strategic information about what to do”:

IF you want to go home
AND you have the bus fare,
THEN you can catch a bus.
Many authors are confused about the relationship between logic and production systems.

“Unlike logic, rule-based systems can easily represent strategic information about what to do”:

IF you want to go home
AND you have the bus fare,
THEN you can catch a bus.

Logic program:  you go home
               if you have the bus fare,
               and you catch a bus.
Many authors are confused about the relationship between deduction and search

“In logic-based systems the fundamental operation of thinking is logical deduction, but from the perspective of rule-based systems the fundamental operation of thinking is search.”
The relationship between deduction and search

IF you drive on highway 1,
THEN you can get from university city to home city.

IF you take the parkway,
THEN you can get from university city to the highway.

IF you take a bus from the bus depot,
THEN you can get from university city to home city.
etc.
The relationship between deduction and search

**IF** you drive on highway 1,
**THEN** you can get from university city to home city.

**IF** you take the parkway,
**THEN** you can get from university city to the highway.

**IF** you take a bus from the bus depot,
**THEN** you can get from university city to home city.

**etc.**

Logic program:

- *you can get from A to B*
  - *if there is a Road from A to B and you drive on the Road*

- *you can get from A to B*
  - *if there is a Bus from A to B and you take the Bus.*

- *there is highway 1 from university city to home city.*

**etc.**
Many authors are confused about the relationship between logic and default reasoning. But the developers of rule-based systems have been happy to use some of the representational rigor of logic-based systems for their increased computational power. One advantage comes from the fact that rules do not have to be interpreted as universally true. The logical generalization (for all $x$) \((\text{student}(x) \rightarrow \text{overworked}(x))\) must be interpreted as saying that every student is overworked. But the rule that \(\text{If } x \text{ is a student } \Rightarrow \text{ then } x \text{ is overworked}\) can be interpreted as a default, that is, as a rough generalization that can admit exceptions. We might have another rule that says that \(\text{if } x \text{ is a student and } x \text{ is taking only easy courses, then } x \text{ is not overworked}\). These two rules might coexist in the same system, but the result...
Logic programs can represent default reasoning.

If $x$ is a student, then $x$ is overworked.

If $x$ is a student and $x$ is taking only easy courses, then $x$ is not overworked.
Logic programs can represent default reasoning.

IF x is a student, THEN x is overworked.

IF x is a student AND x is taking only easy courses, THEN x is not overworked.

\[ X \text{ is overworked if } X \text{ is a student and not } X \text{ is taking only easy courses.} \]

Logic programs can have negative conditions, interpreted as negation as failure.
Many authors are confused about the relationship between deduction and search.

Most of Thagard’s examples of rules are examples of logic programs.
AgentSpeak(L): BDI Agents speak out in a logical computable language

Abstract

Belief-Desire-Intention (BDI) agents have been investigated by many researchers from both a theoretical specification perspective and a practical design perspective. However, there still remains a large gap between theory and practice. The main reason for this has been the complexity of theorem-proving or model-checking in these expressive specification logics. Hence, the implemented BDI systems have tended to use the three major attitudes as data structures, rather than as modal operators. In this paper, we provide an alternative formalization of BDI agents by providing an operational and proof-theoretic semantics of a language AgentSpeak(L). This language can be viewed as an abstraction of one of the implemented BDI systems (i.e., PRS) and allows agent programs to be written and interpreted in a manner similar to that of horn-clause logic programs. We
**AgentSpeak(L):**

**Definition 5** If $e$ is a triggering event, $b_1, \ldots, b_m$ are belief literals, and $h_1, \ldots, h_n$ are goals or actions then $e ; b_1 \land \ldots \land b_m \leftarrow h_1 ; \ldots ; h_n$ is a plan. The expression to the left of the arrow is referred to as the *head* of the plan and the expression to the right of the arrow is referred to as the *body* of the plan. The expression to the right of the colon in the head of a plan is referred to as the *context*. For convenience, we shall rewrite an empty body with the expression *true*.

With this we complete the specification of an agent. In summary, a designer specifies an agent by writing a set of base beliefs and a set of plans. **This is similar to a logic programming** specification of facts and rules. However, some of the major differences between a logic
AgentSpeak(L):

+location(waste,X): location(robot,X) & location(bin,Y) <- pick(waste);
  !location(robot,Y);
  drop(waste).
AgentSpeak(L):

\begin{align*}
+ & \text{location}(\text{waste}, X) : \text{location}(\text{robot}, X) \land \\
& \text{location}(\text{bin}, Y) \\
\leftarrow & \text{pick}(\text{waste}); \\
& !\text{location}(\text{robot}, Y); \\
& \text{drop}(\text{waste}).
\end{align*}

Maintenance goal in logical form with explicit time:

\begin{align*}
\text{location}(\text{waste}, X, T1) \land & \text{location}(\text{robot}, X, T1) \land \\
& \text{location}(\text{bin}, Y, T1) \\
\rightarrow & \text{pick}(\text{waste}, T1 +1) \land \\
& \text{reach}(\text{robot}, Y, T2) \land \\
& \text{drop}(\text{waste}, T2+1)
\end{align*}

Notice that \( \leftarrow \) is opposite to the logical reading.
Two kinds of BDI rules

• Logic programs (or beliefs) executed backwards, simulated by forward chaining:

\[ \text{goal} \leftarrow \text{sub-goals and actions} \]
(i.e. \( \text{goal} \leftarrow \text{sub-goals and actions} \))
Two kinds of BDI rules

• Logic programs (or beliefs) executed backwards, simulated by forward chaining:

\[ \text{goal} \leftarrow \text{sub-goals and actions} \]
(i.e. \( \text{goal} \leftarrow \text{sub-goals and actions} \))

• Reactive rules (or maintenance goals):

\[ \text{event and conditions} \leftarrow \text{goals and actions} \]
(meaning \( \text{event and conditions} \rightarrow \text{goals and actions} \))
Outline of the talk

Overview

Logic programs represent beliefs

Production systems represent goals (but have no logic)

Computational Logic combines goals and beliefs embedded in an observe-think-decide-act agent cycle
CL/ALP combines forward and backward reasoning.

- **Observations**
  - smoke(T) ← fire(T)
  - smoke(1)

- **Actions**
  - press the alarm(T)
  - press the alarm(2)
  - get help(T) ← press the alarm(T)
  - eliminate(T) ← get help(T)

- **Relations**
  - threat(T) → eliminate(T+1) ∨ escape(T+2)
  - eliminate(T) ← get help(T)
  - get help(T) ← press the alarm(T)
  - press the alarm(2)
CL/ALP is compatible with the dual process theory

\[
\begin{align*}
\text{threat}(T) & \rightarrow \text{eliminate}(T+1) \lor \text{escape}(T+2) \\
\text{eliminate}(T) & \leftarrow \text{get help}(T) \\
\text{get help}(T) & \leftarrow \text{press the alarm}(T) \\
\text{press the alarm}(2) & \leftarrow \text{fire}(T) \\
\text{smoke}(1) & \leftarrow \text{fire}(T) \\
\text{smoke}(T) & \leftarrow \text{fire}(T) \\
\text{smoke}(T) & \rightarrow \text{press the alarm}(T+1)
\end{align*}
\]
Abductive Logic Programming (ALP)

Goal G: \( \text{threat}(T) \rightarrow \text{eliminate}(T+1) \lor \text{escape}(T+2) \)

Beliefs B: \( \text{threat}(T) \leftarrow \text{fire}(T) \)
\( \text{smoke}(T) \leftarrow \text{fire}(T) \)
\( \text{eliminate}(T) \leftarrow \text{get help}(T) \)
\( \text{get help}(T) \leftarrow \text{press the alarm}(T) \)

Observation O: \( \text{smoke}(1) \)
Abductive Logic Programming (ALP)

Goal G: \( \text{threat}(T) \rightarrow \text{eliminate}(T+1) \lor \text{escape}(T+2) \)

Beliefs B: \( \text{threat}(T) \leftarrow \text{fire}(T) \)
\( \text{smoke}(T) \leftarrow \text{fire}(T) \)
\( \text{eliminate}(T) \leftarrow \text{get help}(T) \)
\( \text{get help}(T) \leftarrow \text{press the alarm}(T) \)

Observation O: \( \text{smoke}(1) \)

Assumptions \( \Delta \): \( \text{fire}(1) \)
\( \text{press the alarm}(2) \)

Abduction: \( \text{fire}(1) \) explains \( O \)

Planning: \( \text{press the alarm}(2) \) achieves \( G \)
Abductive Logic Programming (ALP)

Goal G: \[ \text{threat}(T) \rightarrow \text{eliminate}(T+1) \lor \text{escape}(T+2) \]

Beliefs B: \[
\begin{align*}
\text{threat}(T) & \leftarrow \text{fire}(T) \\
\text{smoke}(T) & \leftarrow \text{fire}(T) \\
\text{eliminate}(T) & \leftarrow \text{get help}(T) \\
\text{get help}(T) & \leftarrow \text{press the alarm}(T)
\end{align*}
\]

Observation O: \[ \text{smoke}(1) \]

Assumptions \( \Delta \): \[
\begin{align*}
\text{fire}(1) \\
\text{press the alarm}(2)
\end{align*}
\]

Abduction: \[ \text{fire}(1) \text{ explains } O \]

Planning: \[ \text{press the alarm}(2) \text{ achieves } G \]

\[ G \cup O \text{ is true in the model of the world determined by } B \cup \Delta. \]
ALP - Different $\Delta$ can solve the same task.

The challenge is to find the best $\Delta$ within the computational resources available.

In classical decision theory, actions are evaluated by the expected utility of their consequences.

In philosophy of science, explanations are evaluated by their probability and explanatory power. (The more observations explained the better.)

In ALP, actions and assumptions are combined in $\Delta$, and are treated in the same way, and forward reasoning is used to derive their possible consequences,
Computational Logic as a unifying framework

Maintenance goals

Achievement goals

Consequences

Decisions

System 2

Abductive explanations

System 1: Heuristic short cuts

Observations

The world

Actions
Conclusions

Computational Logic

- combines goal and beliefs
- inspired by models of human thinking and decision making
- provides a foundation for more human-oriented computing
- can help people think and communicate more effectively.
• CL as Generator of Human Action

• **CL as Language of Thought (LOT)**

• CL as a unifying framework
How to obtain evidence about the Nature of Human Thought?

Study natural language texts designed to be easy to understand.

The London Underground Emergency Notice

Study advice about effective natural language communication.

The Pyramid Principle
Joseph Williams: Toward Clarity and Grace
How to obtain evidence about the Language of Thought (LOT)?

Study natural language texts designed to be easy to understand.

The London Underground Emergency Notice

Study advice about effective natural language communication.

The Pyramid Principle
Joseph Williams: Toward Clarity and Grace
Not: The teacher gave the student a good mark. 
She was happy.

But: The teacher was happy. 
Or: The student was happy.

Not: Our lack of knowledge of the topic of the talk prevented us from understanding it. 

Or: Because we did not know the topic of the talk, we could not understand the talk.

I.E. A person cannot understand a talk if the person does not know the topic of the talk. 
We did not know the topic of the talk.
Williams: Two Principles of Coherence

1. Put at the beginning of a sentence those ideas that you have already mentioned, referred to, or implied, or concepts that you can reasonable assume your reader is already familiar with, and will readily recognise.

2. Put at the end of your sentence the newest, the most surprising, the most significant information: information that you want to stress – perhaps the information that you will expand on in your next sentence.
Coherence

Example:  

A.  
If A then B.  
If B then C.  
Therefore C.

Example:  

C?  
C if B.  
B if A.  
A.  
Therefore C.
In CL, goals and beliefs are combined in a connectionist network

Goal: if there is an emergency
then I deal with it myself
or I get help or I escape.

Beliefs:
there is an emergency if there is a fire.
I get help if there is an emergency
and I am on a train
and I alert the driver of the train.
To express yourself coherently, connect new ideas with existing ideas.

Goal: if there is an emergency then I deal with it myself or I get help or I escape.

Beliefs:

- if there is an emergency then I get help if there is an emergency and I am on a train and I alert the driver of the train.
- if I press the alarm button then I alert the driver of the train if I press the alarm button.
- if I alert the driver of the train then the driver will stop in a station if I alert the driver of the train and the train ...
- if I press the alarm button then I may receive a £50 penalty if I press the alarm button and I do so improperly.
• CL as Generator of Human Action

• CL as Language of Thought (LOT)

• CL as a unifying framework
The CL Agent Model as a unifying framework

- Decision theory for choosing between alternative actions
  - Clausal form of FOL for heuristics
  - Logic programs for beliefs
  - Clausal form of FOL for goals
  - Semantics
“Rule-based systems” (production systems) as an alternative model of human thinking

“Unlike logic, rule-based systems can easily represent strategic information about what to do”:

*If you want to go home and you have the bus fare, then you can catch a bus.*

But this misses the real logic of the strategy:

*You go home if you have the bus fare and you catch a bus.*

Backward reasoning with this logic behaves like forward reasoning with the rule.
Smart Choices:
73 customer reviews

5 star – 49
4 star – 17
3 star – 3
1 star - 1
Smart choices – a better decision theory

Classical decision theory assumes that all of the alternative actions are fixed and given in advance.

To make smarter decisions:
- identify the goals that motivate the alternatives
- identify the beliefs that reduced the goals to actions
- judge whether the beliefs are true
- investigate whether there are any other relevant true beliefs
- investigate whether there are any other relevant goals
- identify events that can trigger motivating goals and prepare for them before they happen.
Conflicting ways of solving different goals can sometimes be resolved by finding alternative solutions.
Conflicting ways of solving different goals can sometimes be resolved by finding alternative solutions.

- Improve enjoyment of life
- Provide for old age
- Save money
- Improve standard of living
- Increase pay
- Work harder
- Go on strike
- Work less hard
Conclusion: Computational Logic as a unifying framework

The world

Observations → Abductive explanations → Consequences → Decide → Maintenance goals → Achievement goals → Candidates → Consequences → Decide → Actions

Heuristic short cuts
Conclusions

- The Computational Logic combines and unifies
  - Logic
  - Connectionism
  - Production Systems
  - Decision Theory

- Computational Logic can help people
  - communicate better
  - make smarter decisions

- Computational Logic can help computer scientists and engineers
  - develop more human-oriented computer languages
  - more intelligent computer applications
Baron’s view of search in relation to thinking and deciding
Conflicting ways of solving different goals can sometimes be resolved by finding alternative solutions e.g.

Achievement goals:
- Improve enjoyment of life
- Provide for old age

Beliefs:
- You improve enjoyment of life if you work less hard.
- You provide for old age, if you save money and work harder.
Conflicting ways of solving different goals can sometimes be resolved by finding alternative solutions

- Improve enjoyment of life
  - or
  - Provide for old age standard of living and
    - Save money
      - Increase pay
        - or
        - Work harder
          - Go on strike
            - Work harder
        - or
        - Work less hard
smart choices

given alternative choices,
analise goals affected by the choices
and other ways of solving the goals
choose a solution(s) that maximises all the goals,
possibly generating an alternative to the original choice.
Complex decisions can often be replaced by heuristic rules

Instead of the high-level maintenance goals:

\[
\text{If a person attacks me,} \\
\text{then I attack the person or I get help or I try to escape.}
\]

and complex decision between the actions:

\[
\text{I attack the person or} \\
\text{I get help or} \\
\text{I try to escape}
\]

we can employ simpler, lower-level heuristic maintenance goals in logical form:

\[
\text{If a person attacks me and I am stronger than the person,} \\
\text{then I attack the person}
\]

\[
\text{If a person attacks me and I am weaker than the person,} \\
\text{then I get help}
\]

\[
\text{If a person attacks me and I and my helpers are weaker than the person,} \\
\text{then I try to escape}
\]
The network of goals and beliefs can use information about previously useful connections

- Links can have forward or backward directions.

- Links can be weighted by statistics about how often they have been used successfully in the past.

- Input observations and goals can be assigned different strengths (or utilities).

- The strength of observations and goals can be propagated through the graph in proportion to the weights on the links.

- Activating links with the highest weighted strengths is like the activation networks of Patie Maes.
Feed-forward neural networks can be represented as logic programs (from Computational Intelligence, Poole, Mackworth, Goebel, 1998)

inputs  hidden units  output

known
new
short
home

reads

reads with strength $W$
if arguably reads with strength $W_1$
and arguably doesn’t read with strength $W_2$
and $W = f(2.98 + 6.88W_1 - 2.1W_2)$
arguably reads with strength $W_1$
if known with strength $W_4$
and new with strength $W_5$
and short with strength $W_6$
and home with strength $W_7$
and $W_1 = f(\text{–5.25} + 1.98W_4 + 1.86W_5 + 4.71W_6 \text{–0.389}W_7)$

arguably doesn’t read with strength $W_2$
if known with strength $W_4$
and new with strength $W_5$
and short with strength $W_6$
and home with strength $W_7$
and $W_2 = f(0.493 - 1.03W_4 - 1.06W_5 - 0.749W_6 + 0.126W_7)$
In English

A person will read a paper
if there is strong reason to read the paper and
there is no sufficiently strong reason not to read the paper.

There is a reason to read the paper
if the author is known to the person, the topic is new,
the paper is short and the person is at home.

There is a reason not to read the paper
if the author is not known to the person, the topic is old,
the paper is long and the person is not at home.
Discovery. Sometimes writers put their main POINT sentences last because they want their readers to work through an argument or a body of data to experience a sense of discovery. They believe that the development of the POINT is as important as the POINT itself. In fact, that kind of organization characterizes parts of this book: we have frequently begun with some contrasting passages to develop a small-p point, in the hope that you would grasp it a moment before you read the POINT sentence. As we have emphasized, though" most readers in most professional contexts prefer documents with main POINT early. Articles in many sciences hard or soft begin with abstracts that typically contain the POINT of the article. Readers in those areas also know that, after reading the abstract, they can go directly to the conclusion if they want to see the main POINT expressed in more detail. These readers employ a reading strategy that creates a POINT-first form: if they don't find the POINT on the first page, they flip to the conclusion, where they expect to find it.
Clausal logic is a simplified form of first-order logic (FOL)

In clausal logic, sentences have a simplified form, e.g.:

\[\text{has-feathers}(X) \leftarrow \text{bird}(X).\]
\[\text{bird}(john).\]

In standard FOL, the same beliefs can be expressed in infinitely many, equivalent ways, including:

\[\neg (\exists X ((\neg \text{has-feathers } (X) \land \text{bird}(X)) \lor \neg \text{bird}(john)))\]
\[\neg (\exists X ((\neg \text{has-feathers } (X) \lor \neg \text{bird}(john)) \land (\text{bird}(X) \lor \neg \text{bird}(john))))\]

In clausal logic, reasoning is simpler than in standard FOL and can be reduced to forward or backward reasoning.
It can be difficult or impossible to put thoughts into words.
Ideas in writing should always form a pyramid

Only one answer on top level
Ideas: relate horizontally (grouping or argument)
Each grouping: same kind of idea
Ideas: must be MECE

Groupings must be in logical order
Divide a whole into its parts
Determine the causes of an effect
Classify like things

The order dictated by the grouping
XYZ Company
Division A
Division B
Division C
Structural order

Effect
Cause 1
Cause 2
Cause 3
Time order

Top three
Tokyo
Dublin
London
Degree order

Ideas: summary of ideas grouped below
Ideas: Generate questions in readers mind
Ideas: relate vertically
A good argument forces reader into dialogue
Pyramid logic: improves structure
Clausal logic as a theory of the LOT can help people to communicate more effectively

By expressing communications:

Clearly So that their meaning is unambiguous.

Simply So that their meaning is close to their canonical form.

Coherently So that it is easy to link new information to old information.
The syntax of logic programs

*Clauses* have the form:

\[
\text{conclusion} \leftarrow \text{condition}_1 \land \text{condition}_2 \ldots \land \text{condition}_n
\]

or

\[
\forall X [ \text{condition}_1 \land \text{condition}_2 \ldots \land \text{condition}_n \rightarrow \text{conclusion} ]
\]

i.e. for all \(X\), conclusion if \(\text{condition}_1\) and \(\text{condition}_2\) \ldots and \(\text{condition}_n\)

where \(\text{conclusion}\) is an atomic formula

and \(\text{condition}_i\) are atomic formulas or negations of atomic formulas.

If \(n = 0\), then the clause is a “fact”

\[
\text{conclusion} \text{ if true}
\]

i.e.

\[
\text{conclusion}
\]

If \(\text{conclusion}\) and all \(\text{condition}_i\) are atomic formulas, then the clause is a Horn clause.
The syntax of maintenance goals = a variant of the clausal form of first order logic

**Goals:** clauses of the form:

$$\forall X [condition_1 \land condition_2 \ldots \land condition_n \rightarrow \exists Y [conclusion_1 \lor conclusion_2 \ldots \lor conclusion_m]]$$

where $X$ is the set of all variables that occur in the $condition_i$ and $Y$ is the set of all variables that occur only in the $conclusion_j$

If $m = 0$, then the goal is equivalent to a denial (or constraint):

$$condition_1 \land condition_2 \ldots \land condition_n \rightarrow false$$

i.e. $$\neg [condition_1 \land condition_2 \ldots \land condition_n]$$

It can sometimes be hard to tell the difference between a goal and a belief.
British Nationality
Act 1981
1981 CHAPTER 61

An Act to make fresh provision about citizenship and nationality, and to amend the Immigration Act 1971 as regards the right of abode in the United Kingdom.
[30th October 1981]

BE IT ENACTED by the Queen's most Excellent Majesty, by and with the advice and consent of the Lords Spiritual and Temporal, and Commons, in this present Parliament assembled, and by the authority of the same, as follows:—

PART I

BRITISH CITIZENSHIP

Acquisition after commencement

1.—(1) A person born in the United Kingdom after commencement shall be a British citizen if at the time of the birth by birth or adoption,

(a) a British citizen; or

(b) settled in the United Kingdom.

(2) A new-born infant who, after commencement, is found abandoned in the United Kingdom shall, unless the contrary is shown, be deemed for the purposes of subsection (1)—

(a) to have been born in the United Kingdom after commencement; and

(b) to have been born to a parent who at the time of the birth was a British citizen or settled in the United Kingdom.
1.-(1) A person born in the United Kingdom after commencement shall be a British citizen if at the time of the birth his father or mother is –
   (a) a British citizen; or
   (b) settled in the United Kingdom.

The meaning of subsection 1.-(1)

A person shall be a British citizen by 1.-(1)
if the person was born in the United Kingdom
and the person was born after commencement
and a parent of the person was a British citizen
at the time of the person’s birth or
a parent of the person was settled in the United Kingdom at the time of the person’s birth.
Heuristics are often represented as *condition-action* rules in production systems

Declarative “working memory” consisting of atomic sentences, and Procedures consisting of condition-action rules:

*If conditions C, then do actions A.*

Procedures look like logical conditionals, but do not have a logical semantics.

Production system cycle:
- observe a current input
- use *forward chaining* to match the input with a condition in C
- use *backward chaining* to verify the remaining conditions of C
- perform conflict-resolution to choose a single rule if the conditions C of more than one rule are satisfied, and
- execute the associated actions A.
Maintenance goals generate actions

*If a person attacks me, then I fight back or I get help or I try to escape.*

Given an observation or consequence of an observation:

*john attacks me*

Reason forwards to derive the achievement goal:

*I fight back or I get help or I try to escape*

Decide between the different actions:

*I attack the person or I get help or I try to escape*
How are thinking and logic related?

In the philosophy of language, there are three main theories:

Human thinking does not have a language-like structure at all. So communicating thoughts from writer to reader is almost a miracle.

The LOT is a form of the public, natural language that we speak. So communicating thoughts from writer to reader is trivial. Just say what you think.

The LOT is a private language-like representation, which does not depend on the natural language that we speak. So communications can be improved by expressing them in a form that is close to the language of thought, because this will reduce the amount of effort the reader needs to translate communications into thoughts.

I will argue that the LOT is a private, language-like representation that has a simplified logical form, which has a connectionist structure.
How to investigate the LOT? Part 1 of 2

According to relevance theory [Sperber and Wilson, 1986], people understand natural language by attempting to extract the most information for the least effort.

It follows that:
If you want to find out whether there is a LOT, and what it is like, then study natural language texts that communicate useful information and are easy to understand.

Understanding the LOT can help us:

• communicate more effectively with other people
• develop better computer languages
How to investigate the LOT? Part 2 of 2

According to relevance theory, people understand natural language by attempting to extract the most information for the least effort.

It follows that:

If you want to find out whether there is a LOT, and what it is like, then study advice about effective natural language communication.

Understanding the LOT can help us:

• to communicate more effectively with other people
• to develop better computer languages
To express yourself effectively in natural language

1. Avoid ambiguity. e.g.
   Not: The teacher gave the student a good mark.
       She was happy.
   Better: The teacher was happy with the student’s work.
   Or:    The student was happy with the good mark.

2. Avoid unnecessary complexity. e.g.
   Not: Our lack of knowledge of the topic of the talk prevented us from understanding it.
   Better: Because we did not know the topic of the talk, we could not understand the talk.
   Or:    A person cannot understand a talk if the person does not know the topic of the talk.
           We did not know the topic of the talk.

3. Connect related ideas together.
The production system cycle

Condition-action rules:
If conditions, then do actions.

Consult working memory, to check other conditions

Candidate actions

Conflict resolution

Observe

The World

Act
Conflict resolution

Several conflicting actions can be derived at the same time.

For example:

- If someone attacks me, then attack them back.
- If someone attacks me, then get help.
- If someone attacks me, then try to escape.

The agent needs to use “conflict resolution” to decide what to do.

Production systems do not have a logical semantics.
Computational Logic and Human Thinking

• CL as the Language of Thought (LOT)

• CL as a connectionist model of the mind

• Production systems as an alternative model of the Mind

• CL as a unifying framework
Thinking about actions, beliefs and personal goals can all be described in terms of a common framework, which asserts that thinking consists of search and inference.

We search for certain objects and then make inferences from and about the objects we have found.” (page 6)
As Sherlock Holmes explained to Dr. Watson, in *A Study in Scarlet*:

“In solving a problem of this sort, the grand thing is to be able to reason backward. That is a very useful accomplishment, and a very easy one, but people do not practise it much. In the everyday affairs of life it is more useful to reason forward, and so the other comes to be neglected. There are fifty who can reason synthetically for one who can reason analytically.”

…….

“Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds, and argue from them that something will come to pass. There are few people, however, who, if you told them a result, would be able to evolve from their own inner consciousness what the steps were which led up to that result. This power is what I mean when I talk of reasoning backward, or analytically.”
Joseph M. Williams

*Style*

Toward Clarity and Grace

*With two chapters coauthored by*  
Gregory G. Colomb

Contents

Preface ix
1 Causes 1
2 Clarity 17
3 Cohesion 45
4 Emphasis 67
5 Coherence I 81  
*With Gregory G. Colomb*
6 Coherence II 97  
*With Gregory G. Colomb*
7 Concision 115
8 Length 135
9 Elegance 153
10 Usage 169
Notes 199
Acknowledgments 201
Index 203

The University of Chicago Press  
Chicago and London
Clarity

Not: The teacher gave the student a good mark.
She was happy.

But: The teacher was happy.
Or: The student was happy.

?: Our lack of knowledge of the topic of the talk prevented us from understanding it.
Or: Because we did not know the topic of the talk, we could not understand the talk.
Or: A person cannot understand a talk if the person does not know the topic of the talk. We did not know the topic of the talk.
Thinking generates actions, to help an agent survive and prosper.
Goal: I deal with the emergency myself
or I get help
or I escape.

Observe

Beliefs:
there is an emergency
there is a fire.

Think

Decide

I alert the driver of the train.
I press the alarm.

Act

The World
The Logic of the agent cycle

Goal: if there is an emergency then I deal with it myself or I get help or I escape.

Beliefs:
- I get help if there is an emergency and I am on a train and I alert the driver of the train.
- there is an emergency if there is a fire.
- I alert the driver of the train if I press the alarm button.
Baron’s view of an intelligent agent

The world

Achievement goals

Search

Inference

Consequences

Decide

Actions
Computational Logic as the Language of Thought of an intelligent agent

- Observations as inputs
- Actions as outputs
- Beliefs as a model of the world
- Forward reasoning using beliefs
- Backward reasoning using beliefs as problem-solving procedures
- Maintenance goals triggered by forward reasoning

The world
Abductive Logic Programming

Goal: if there is an emergency then I deal with it myself or I get help or I escape.

Beliefs:
- I get help if there is an emergency and I am on a train and I alert the driver of the train.
- I alert the driver of the train if I press the alarm button.
- if there is a fire.

Observe
The Heart of the Problem -
What is the meaning of if A then B?

Classical Logic: If A is true then B is true. e.g.

i.e. If X is a bird, then X can fly.
equivalently bird(X) → fly(X).

If X is mother of Y, then X is parent of Y.
i.e. mother(X, Y) → parent(X, Y)
equivalently parent(X, Y) ← mother(X, Y)
The Heart of the Problem -
What is the meaning of if A then B?

Change of state. e.g. If A happens then do B.

If it is raining,
then cover yourself with an umbrella.

If you are hungry,
then buy food, cook the food, and eat the food.

If you want to go home for the weekend, and you have the bus fare,
then take the bus.

If you increase an employee's salary,
then increase the employee's manager's salary.
What is the meaning of if A then B?
A proposed solution in classical logic.

Add explicit time:
If A happens at time T, then you do B at time T+n.

If it is raining at time T,
then you cover yourself with an umbrella at time T+1.

If you are hungry at time T
then you buy food at time T+1, you cook the food at time T+2,
and you eat the food at time T+3.

If you increase an employee's salary at time T1,
then you increase the employee's manager's salary at time T2
and T1 < T2 < T1 + 10.
The dual process model combines two systems of thinking.

System 1 operates automatically and quickly, with little or no effort and no sense of voluntary control.

System 2 allocates attention to the effortful mental activities that demand it, including complex computations.

The operations of system 2 are often associated with the subjective experience of agency, choice, and concentration.
The dual process model of thinking

System 1 “quickly proposes intuitive answers to judgement problems as they arise”,

System 2 “monitors the quality of these proposals, which it may endorse, correct, or override”.

When system 1 runs into difficulty, it calls on system 2.

System 1 continuously generates suggestions for system 2.

System 2 is activated when an event is detected that violates the model of the world that system 1 maintains.
The Dual Process Model of thinking

The world

System 1

System 2

Observations

Actions
The Dual Process Model viewed in logical terms

The World

threat(T) → eliminate(T+1) ∨ escape(T+2)

fire(T) → eliminate(T+1)
fire(T) ∧ ¬ eliminate(T+1) → escape(T+2)

threat(T) ←
fire(T)

Decide
Computational Logic represents goals and beliefs in logical form.

The task of an intelligent agent is to perform **actions**, to make its **goals and observations true** in the model of the world determined by its **beliefs**.
Some authors confuse production rules and logic programs

Rules can be used to reason either forward or backward. Reasoning backward, a student might think that “To get home, I can take the highway, which requires taking the parkway, which requires taking Main Street, which requires getting a car.” The goal is to get home, but the plan is constructed by considering a series of subgoals such as getting to the highway. Reasoning forward, the student might use inference akin to modus ponens to see that “Main Street gets me to the parkway, which gets me to the highway.” Forward and backward reasoning both try to find a series of rules that can be used to get from the starting point to the goal, but they differ in the search strategy employed.
Backward reasoning interprets logic programs as goal-reduction procedures

Backward reasoning interprets $C$ if $A$ and $B$ as a procedure for solving $C$ by solving the subgoals $A$ and $B$.

Backward reasoning interprets the logic program
You go home
if you have the bus fare and you catch a bus

as the procedure
If you want to go home
and you have the bus fare,
then you can catch a bus.
CL = abductive logic programming (ALP) embedded in an observe-think-decide-act agent cycle.
Abductive Logic Programming (ALP) combines goals and beliefs

Logic programs = beliefs B
\[ X = X \]
\[ \text{country(usa)} \quad \text{country(canada)} \]
\[ \text{adjacent(usa, canada)} \quad \text{etc.} \]

Integrity constraints = goals G
\[ \text{country}(X) \rightarrow \text{colour}(X, \text{red}) \vee \text{colour}(X, \text{blue}) \vee \text{colour}(X, \text{yellow}) \]
\[ \text{colour}(X, C), \text{colour}(X, D) \rightarrow C = D \]
\[ \text{colour}(X, C), \text{colour}(Y, C), \text{adjacent}(X, Y) \rightarrow \text{false} \]

Abducible predicates = candidate hypotheses A
\[ \text{colour}(\text{usa}, \text{red}), \text{colour}(\text{usa}, \text{blue}), \text{etc.} \]
Abductive Logic Programming (ALP)

Logic programs = beliefs $B$

$x = x$

\textit{country(usa)} \quad \textit{country(canada)}

\textit{adjacent(usa, canada)} \quad \text{etc.}

Integrity constraints = goals $G$

$\text{country}(X) \rightarrow \text{colour}(X, \text{red}) \lor \text{colour}(X, \text{blue}) \lor \text{colour}(X, \text{yellow})$

$\text{colour}(X, C), \text{colour}(X, D) \rightarrow C = D$

$\text{colour}(X, C), \text{colour}(Y, C), \text{adjacent}(X, Y) \rightarrow \text{false}$

Abducible predicates = candidate hypotheses $A$

$\text{colour}(\text{usa}, \text{red}), \text{colour}(\text{usa}, \text{blue}), \text{etc.}$

The task is to generate hypotheses $\Delta \subseteq A$ such that $G$ is true in the minimal model of $B \cup \Delta$. 
ALP Proof procedures combine forward and backward reasoning.
An agent’s task in life is to perform **actions** to make its **goals** and **observations** true in the **model of the world** determined by its **beliefs**.
Backward reasoning interprets logic programs as goal-reduction procedures

Backward reasoning interprets the logic program

*the driver is alerted if you press the alarm signal button.*

as the procedure

*If you want to alert the driver then you press the alarm signal button.*
Some authors are confused about the relationship between deduction and search

“In logic-based systems the fundamental operation of thinking is logical deduction, but from the perspective of rule-based systems the fundamental operation of thinking is search.”

**IF** you drive on highway 1, **THEN** you can get from university city to home city.

**IF** you take the parkway, **THEN** you can get from university city to the highway.

**IF** you take a bus from the bus depot, **THEN** you can get from university city to home city. etc.
Logic program with explicit time:

\[
\begin{align*}
reach(\text{robot}, X, T) & \leftarrow location(\text{robot}, X, T) \\
reach(\text{robot}, X, T2) & \leftarrow location(\text{robot}, Y, T1) \land \\
& \quad \neg X = Y \land \\
& \quad adjacent(Y, Z) \land \\
& \quad \neg location(\text{car}, Z, T1) \land \\
& \quad move(\text{robot}, Z, T1) \land \\
& \quad reach(\text{robot}, X, T2) \land T1 < T2
\end{align*}
\]

Here \textit{move(\text{robot}, Z, T1)} is a primitive (atomic) action. If it succeeds, it initiates \textit{location(\text{robot}, Z, T1)} and it terminates \textit{location(\text{robot}, X, T1)}.

The program is teleo-reactive.
Abductive Logic Programming

Beliefs B:  
there is smoke if there is a fire
there is an emergency if there is a fire
I get help if I press the alarm button

Goal G:  
if there is an emergency
then I deal with it myself or I get help or I escape

Observation O: there is smoke
CL/ALP combines forward and **backward** reasoning

- If there is an emergency, then I deal with it myself or I get help or I escape.
- If there is smoke, there is an emergency.
- If there is a fire, I get help if I press the alarm button.
- If I press the alarm button, there is a fire.
- There is a fire.

Observations: there is smoke, if there is a fire.

Actions: I press the alarm button, I get help.
Abductive Logic Programming (ALP)

Beliefs B:  
- there is smoke if there is a fire
- there is an emergency if there is a fire
- I get help if I press the alarm button

Goal G:  
- if there is an emergency
  then I deal with it myself or I get help or I escape

Observation O:  
- there is smoke

Assumptions \( \Delta \):  
- there is a fire
- I press the alarm button

\( G \cup O \) is true in the model of the world determined by \( B \cup \Delta \).

Abduction:  
- there is a fire explains \( O \)

Planning:  
- I press the alarm button achieves \( G \)
The Dual Process Model viewed in logical terms

- \( \text{threat}(T) \rightarrow \text{eliminate}(T+1) \)
- \( \text{fire}(T) \wedge \neg \text{eliminate}(T+1) \rightarrow \text{escape}(T+2) \)

Decision:

- Decide
if there is an emergency
then I deal with it myself
or I get help

there is smoke
if there is a fire
there is an emergency
if there is a fire

there is smoke
if there is a fire

I get help
if I press the alarm button

I press the alarm button

CL/ALP combines forward and backward reasoning

Observations

Actions
ALP agents as a unifying framework

The world

Observations → Consequences → Decisions → Maintenance goals → Achievement goals → Consequences → System 1: Heuristic short cuts → System 2

Abductive explanations → Decisions
Logic program: 

you go home  
if you have the bus fare,  
and you catch a bus.

More precisely and more generally:

\[
\text{at}(\text{Agent, Destination, T2})  
\leftarrow  
\text{at}(\text{Agent, Location, T1}) \land  
\text{have}(\text{Agent, Money, T1}) \land  
\text{busRoute}(\text{Bus, Location, Destination, Fare}) \land  
\text{Fare} < \text{Money} \land  
\text{take}(\text{Agent, Bus, Location, T1}) \land  
\text{arrives}(\text{Bus, Destination, T2}) \land  \text{T1} < \text{T2}
\]
AgentSpeak(L):

+!location(robot,X):location(robot,X) ← true. (P2)

+!location(robot,X):location(robot,Y) &
  (not (X = Y)) &
  adjacent(Y,Z) &
  (not (location(car, Z)))
  ← move(Y,Z);
  +!location(robot,X). (P3)

Logic program with explicit time:

reach(robot, X, T) ← location(robot, X, T)
reach(robot, X, T2) ← location(robot, Y, T1) ^
  not X = Y ^
  adjacent(Y, Z) ^
  not location(car, Z, T1) ^
  move(robot, Z, T1) ^
  reach(robot, X, T2) ^ T1 < T2