POPL2 class (2020-05-04)

Logic and expert systems

Expert system

- an expert system is a program that guides the user in the solution of some problem which normally requires intervention of a human expert in the field.
    - *diagnosis*, *control* and *planning*. Diagnosis means trying to find the cause of some malfunction, e.g. the cause of an illness.
    - In control-applications the aim is to prevent a system, such as an industrial process, from entering abnormal states.
    - Planning, finally, means trying to find a sequence of state transitions ending in a specified final state via a sequence of intermediate states given an initial one.

Expert system

- It is divided into an *inference engine* and a *knowledge-base*. The knowledge-base contains rules which describe general knowledge about some problem domain. The inference engine is used to infer knowledge from the knowledge-base.
- capability not only to infer new knowledge from existing knowledge but also to explain how/why some conclusion was reached.
- support for incremental knowledge acquisition.
- It may contain rules which are subject to some *uncertainty*.

Expert system

IF the stain of the organism is gram-positive
AND the morphology of the organism is coccus
AND the growth conformation of the organism is clumps
THEN the identity of the organism is staphylococcus (0.7)

Uncertainty of the rule

$$
\begin{aligned}
&identity\_of\_organism(staphylococcus) \leftarrow \\
&\qquad stain\_of\_organism(gram\_positive), \\
&\qquad morphology\_of\_organism(coccus), \\
&\qquad growth\_conformation\_of\_organism(clumps).
\end{aligned}
$$

Diagnosing starting problem in car

- if *Y* is a necessary component for *X* and *Y* is malfunctioning then *X* is also malfunctioning;
- if *X* exhibits a fault-symptom *Z* then either *X* is malfunctioning or there exists another malfunctioning component which is necessary for *X*.

$$\forall X(\exists Y(needs(X,Y) \wedge malfunctions(Y)) \supset malfunctions(X))$$
$$\forall X,Z(symptom(Z,X) \supset (malfunctions(X) \vee \exists Y(needs(X,Y) \wedge malfunctions(Y))))$$

$$malfunctions(X) \leftarrow needs(X,Y), malfunctions(Y).$$

Diagnosing starting problem in car

if *X* exhibits a fault-symptom *Z* then either *X* is malfunctioning or there exists another malfunctioning component which is necessary for *X*.

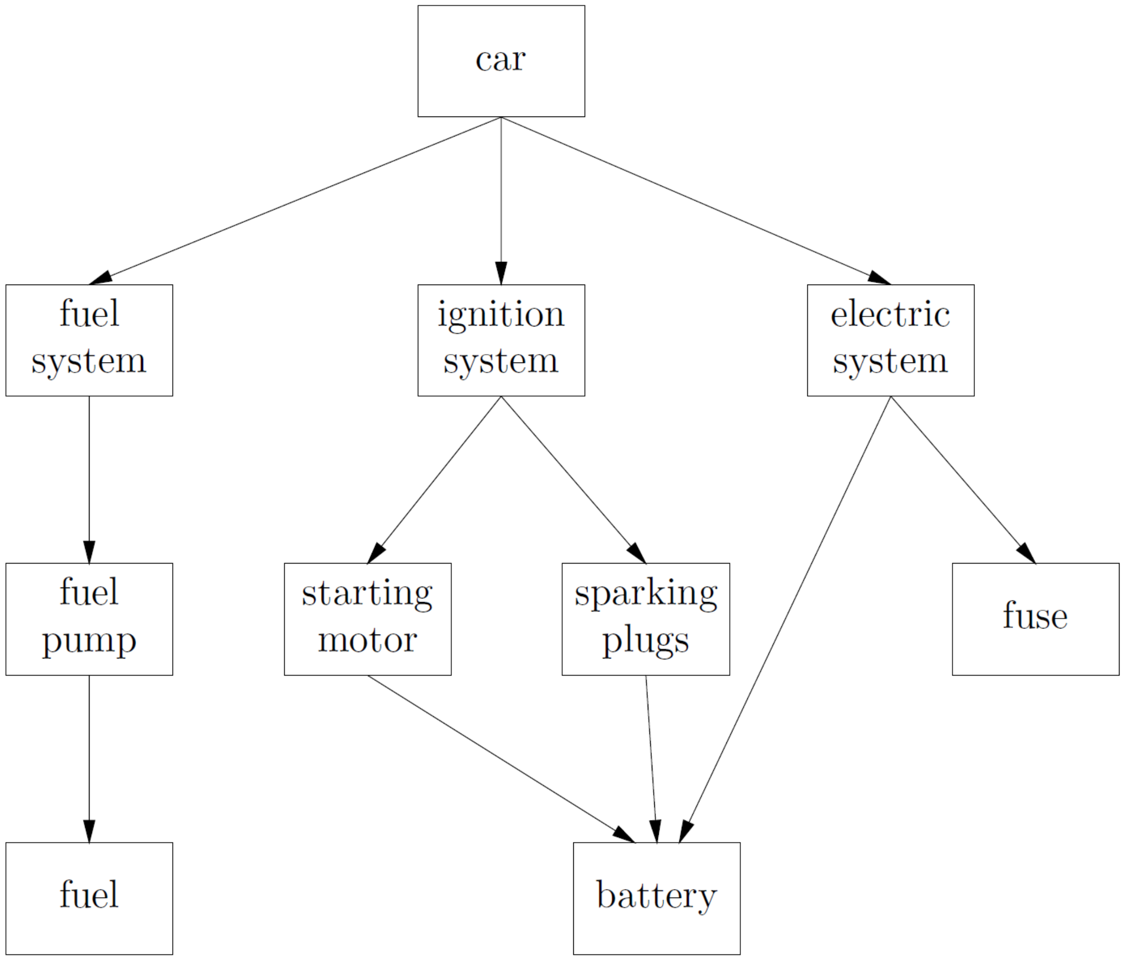
- *X* has an indirect fault if there exists a component which is necessary for *X* and

$$\forall X(\exists Y(needs(X,Y) \wedge malfunctions(Y)) \supset malfunctions(X))$$
$$\forall X,Z(symptom(Z,X) \supset (malfunctions(X) \vee \exists Y(needs(X,Y) \wedge malfunctions(Y))))$$

$$\forall X,Y(symptom(Y,X) \supset (malfunctions(X) \vee indirect(X)))$$
$$\forall X(\exists Y(malfunctions(Y) \wedge needs(X,Y)) \supset indirect(X))$$

$$malfunctions(X) \leftarrow symptom(Y,X), not\ indirect(X).$$
$$indirect(X) \leftarrow needs(X,Y), malfunctions(Y).$$

Taxonomy of car engine



$needs(car, ignition\_system).$
$needs(car, fuel\_system).$
$needs(car, electric\_system).$
$needs(ignition\_system, starting\_motor).$
$needs(ignition\_system, sparking\_plugs).$
$needs(electric\_system, fuse).$
$needs(electric\_system, battery).$
$needs(fuel\_system, fuel\_pump).$
$needs(sparking\_plugs, battery).$
$needs(starting\_motor, battery).$
$needs(fuel\_pump, fuel).$

Car symptom diagnosis

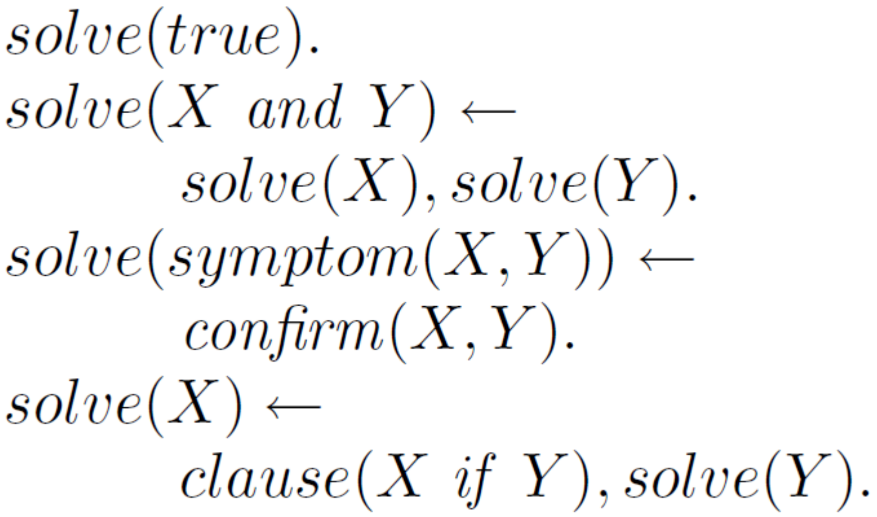
- How to specify symptoms ?
  - symptoms exhibited by a specic car depend on the particular car in a specic moment of time.
  - added to the database when diagnosing the cause of malfunction of that particular car
- The inference engine is used to infer new knowledge from existing knowledge
- *forward*- and *backward-chaining*
  - either start from what is already known and infer new knowledge from this,
  - start from the conclusion to be proved and reason backwards until the conclusion depends on what is already known.

Expert systems and logic

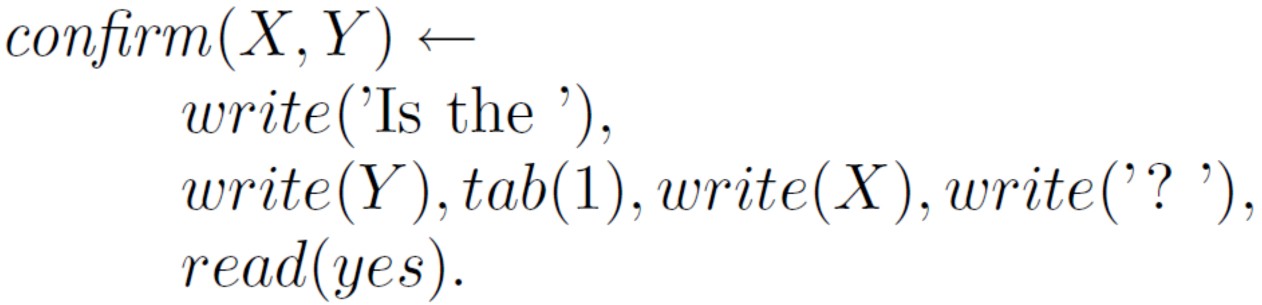
- probabilities of rules and the user interface,
- knowledge-base is usually *incomplete*
- knowledge-base only contains general knowledge concerning different faults and symptoms
- information has to be added to its knowledge-base whilst diagnosing the individual.
- *abduction : General knowledge P* of the world and a symptom or an observation *F* one can say that the aim of the expert system is to find a cause C such that *P U C -> F*
- expert systems have the capability to *explain* their conclusions.

Extending logic program to expert system

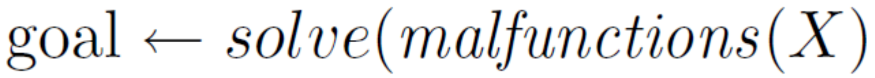
- diagnosing an illness, the symptoms of a specific patient have to be collected during “run-time".
- *query the user* for information.



```
solve(true).
solve(X and Y) ←
        solve(X), solve(Y).
solve(symptom(X, Y)) ←
        confirm(X, Y).
solve(X) ←
        clause(X if Y), solve(Y).
```



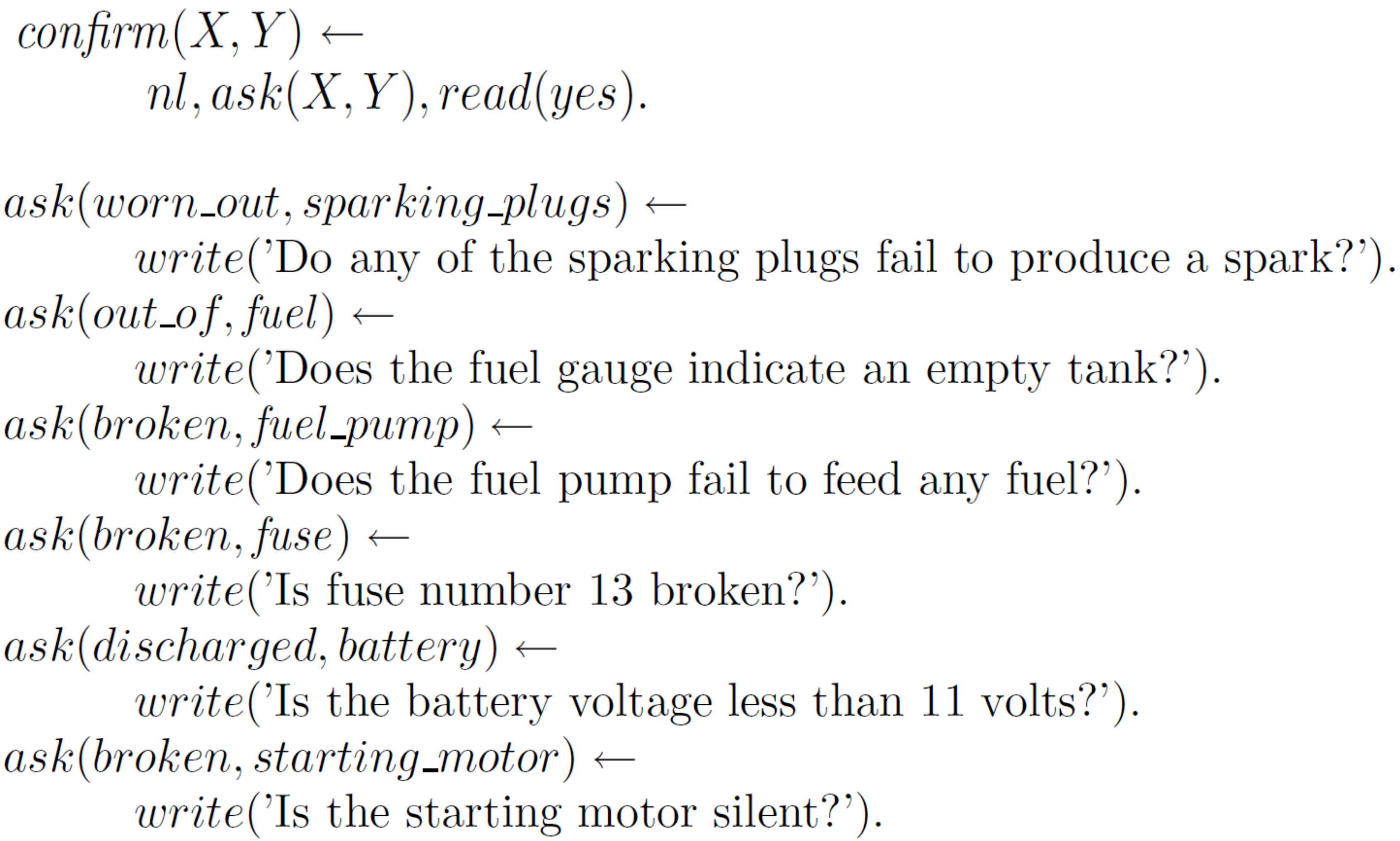
```
confirm(X, Y) ←
        write('Is the '),
        write(Y), tab(1), write(X), write('? '),
        read(yes).
```



```
goal ← solve(malfunctions(X))
```

“Is the tyre flat?".

Car diagnostics



$$
\begin{aligned}
&\mathit{confirm}(X,Y) \leftarrow\\
&\qquad \mathit{nl}, \mathit{ask}(X,Y), \mathit{read}(\mathit{yes}).\\
\\
&\mathit{ask}(\mathit{worn\_out}, \mathit{sparking\_plugs}) \leftarrow\\
&\qquad \mathit{write}(\text{'Do any of the sparking plugs fail to produce a spark?'}).\\
&\mathit{ask}(\mathit{out\_of}, \mathit{fuel}) \leftarrow\\
&\qquad \mathit{write}(\text{'Does the fuel gauge indicate an empty tank?'}).\\
&\mathit{ask}(\mathit{broken}, \mathit{fuel\_pump}) \leftarrow\\
&\qquad \mathit{write}(\text{'Does the fuel pump fail to feed any fuel?'}).\\
&\mathit{ask}(\mathit{broken}, \mathit{fuse}) \leftarrow\\
&\qquad \mathit{write}(\text{'Is fuse number 13 broken?'}).\\
&\mathit{ask}(\mathit{discharged}, \mathit{battery}) \leftarrow\\
&\qquad \mathit{write}(\text{'Is the battery voltage less than 11 volts?'}).\\
&\mathit{ask}(\mathit{broken}, \mathit{starting\_motor}) \leftarrow\\
&\qquad \mathit{write}(\text{'Is the starting motor silent?'}).
\end{aligned}
$$