Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 68 (1993)

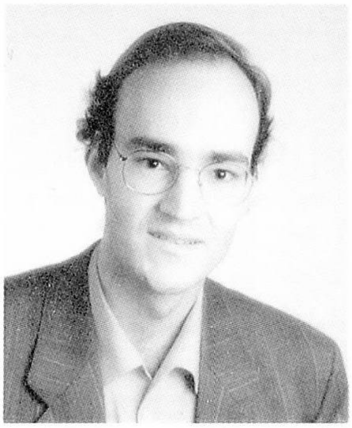
Artikel: Reasoning strategies for engineering problems

Autor: Faltings, Boi

DOI: https://doi.org/10.5169/seals-51837

Reasoning Strategies for Engineering Problems
Stratégies de raisonnement dans les problèmes de génie civil
Strategien des Schliessens bei Ingenieurproblemen

Boi FALTINGS
Dir.AI Lab.
Swiss Fed. Inst. of Technol.
Lausanne, Switzerland



Prof. Boi Faltings received a diploma from the ETH Zurich and a PhD from the Univ. of Illinois, both in Electrical Eng. He founded the Artificial Intelligence Laboratory of EPFL in 1987. His research interests include intelligent design systems.

SUMMARY
Classical expert systems are based on deductive inference. However, most engineering problems require abductive reasoning. This paper discusses the problems caused by simulating abductive reasoning using deductive rules, and how the framework of model-based reasoning allows explicit implementation of abductive inference and thus avoids these problems. The model-based framework also makes it possible to use dependencies for efficient solutions to the problem of constraint relaxation. Model-based reasoning is thus not only useful as an efficient way of formulating knowledge, but also allows more powerful inference strategies.

RÉSUMÉ
Les systèmes experts classiques se basent sur le raisonnement déductif. La plupart des problèmes de génie civil exigent toutefois une manière de procéder plus abstraite, habituellement simulée par de simples règles déductives. Il faut aborder les problèmes qui en découlent par des réflexions se rapportant à des modèles, et qui permettent une mise en application explicite de la transmission par abstraction. Le cadre basé sur un modèle permet d'utiliser des relations pour résoudre le problème de la relaxation de conditions secondaires. Le raisonnement rapporté à un modèle spécifique n'est pas seulement une manière efficace de formuler les connaissances, mais de permettre également des stratégies de transmission plus performantes.

ZUSAMMENFASSUNG
Klassische Expertensysteme basieren auf dem deduktiven Schliessen. Die meisten Ingenieurprobleme erfordern jedoch ein abstrahierendes Vorgehen, das üblicherweise lediglich mit deduktiven Regeln simuliert wird. Es wird gezeigt, wie daraus entstehende Probleme durch auf Modelle bezogene Überlegungen umgangen werden, die eine explizierte Implementierung der Übertragung durch Abstraktion gestattet. Der modellbasierte Rahmen macht es auch möglich, Abhängigkeiten für eine effiziente Lösung des Problems der Lockerung von Nebenbedingungen zu nutzen. Modellbezogenes Schliessen ist daher nicht nur eine wirkungsvolle Art, Wissen zu formulieren, sondern es erlaubt auch leistungsstärkere Strategien der Übertragung.

1 Reasoning Strategies

A very general tool for modeling knowledge and reasoning on computers are *inference rules* taking the form:

```
conditions ⇒ conclusion
```

meaning that whenever `conditions` are given, `conclusion` is also true. The formulation of knowledge as inference rules originated in research on human psychology and was proposed as a formalism for computer programs by Newell and Simon ([5]).

The most natural way to apply inference rules is by *deduction.* A deductive inference engine is a computer program which starts with a set of premises - presumed to be true - and iteratively applies inference rules to add new conclusions to this set of known facts. Rules engines for expert systems often distinguish between *forward* and *backward* chaining, where backward chaining means that inferences are guided to lead to particular *goals.*

Deductive inference has been proven to be *Turing-equivalent* ([4]), meaning that any computation which can be carried out on a digital computer can also be achieved using deductive inference. This may become intuitively clear by seeing that a FORTRAN statement of the form:

```
C = A*A + B*B
```

can be translated into a deductive rule:

```
(∀ x) (∀ y) (A = x) ∧ (B = y) → (C = x*x+y*y)
```

which can be applied as soon as the values of A and B are known.

However, deduction is not the only form of logical inference. Consider the following propositions and rule:

```
a) bird(Tweety)
b) flies(Tweety)
c) (∀ x) bird(x) ⇒ flies(x)
```

Three types of inference are possible between these elements, depending on which of them is desired as a conclusion:

- **deduction:** a), c) → b)
  the conditions and the rule justify the conclusion.
- **induction:** a), b) → c)
  the rule is inferred from observing the example of a bird that flies.
- **abduction:** b), c) → a)
  the condition of the rule is inferred to explain the conclusion.

Now consider the typical engineering activities:

- *analysis* =
  find the performance of a given structure: **deduction**
- *diagnosis* =
  find causes that explain given symptoms: **abduction**
- *design* =
  find a structure that satisfies given specifications: **abduction**
- *learning* =
  find a rule that summarizes given observations: **induction**

The surprising conclusion is that many of the activities in which engineers hope to use knowledge-based systems in fact require not deductive, but abductive and inductive reasoning! It is therefore worthwhile to examine the properties of these other kinds of inference.

2 Abductive and Inductive Inference

Abduction and induction are distinguished from deduction by the fact that they usually produce *ambiguous* answers. For example, given the rules:

a) `poor-drainage` ⇒ `excessive-staining`
b) `low-quality-concrete` ⇒ `excessive-staining`
c) `insufficient-covering-of-reinforcement` ⇒ `excessive-staining`

abduction gives three different explanations for the premise `excessive-staining`, corresponding to the rules `a)`, `b)` and `c)`. Different explanations are distinguished only through *corroboration* with additional information, possibly also obtained by abduction. For example, another abductive inference:

d) `poor-drainage` ⇒ `wet-pavement`
e) `humid-climate` ⇒ `wet-pavement`
+ assertion: `wet-pavement`

results in two solutions of which one, `poor-drainage`, is in agreement with one of the choices for the first abductive inference, and gives reason to select it over the other candidates.

Similarly, ambiguities arise in induction because there are usually many rules which fit a given set of observations. Thus, the examples:

`Bridge-27:` {`poor-drainage,freeway,excessive-staining`}
`Bridge-34:` {`poor-drainage,multiple-simple-spans,excessive-staining`}
`Bridge-53:` {`poor-drainage,multiple-simple-spans,freeway,excessive-staining`}

could justify any combination of the following rules:

a) `poor-drainage` ⇒ `excessive-staining`
b) `freeway` ⇒ `excessive-staining`
c) `multiple-simple-spans` ⇒ `excessive-staining`

The ambiguities must be resolved by *refutation*: observing counterexamples to the hypothesized rule.

In fact, the occurrence of ambiguities is the main motivation for using *symbolic* or *qualitative* models for abductive or inductive inference: numerical models would often result in infinite sets of choices which cannot be dealt with in a computer algorithm. It is thus not surprising that knowledge-based systems are an attractive technology for activities which require inductive or abductive inference: learning, diagnosis and design.

3 Implementing Abductive Inference

Although abduction is one of the main motivations for applying knowledge-based systems in engineering, classical expert systems are based on *deductive* inference only, since deduction is most straightforward to implement in an algorithm. Using a deductive system for abductive tasks such as diagnosis means that abduction must be *simulated* using deductive rules. This is carried out most easily by inverting rules defining the knowledge:

`poor-drainage` ⇒ `excessive-staining` is transformed into:
`excessive-staining` ⇒ `poor-drainage`

However, this conversion cannot express the ambiguity which arises when several rules could explain the same observation. To distinguish different possibilities, many expert systems use *certainty factors* or similar measures which estimate the likelihoods of candidates.

Such certainty factors could be computed on the basis of the absolute probabilities that candidates are in fact present. More precisely, given a set of rules:

a ⇒ x
b ⇒ x
c ⇒ x

a set of a priori probabilities p(a), p(b) and p(c) that a, b or c are the correct candidates, and the assumptions that:

- the propositions a, b and c are mutually exclusive.
- there are no other possible explanations for x (closed-world assumption).

one can follow the principle of Bayes and construct a *probabilistic* set of inference rules where the conclusions are asserted to be true with certain probabilities:

x ⇒ a, $p = \frac{p(a)}{p(a)+p(b)+p(c)}$
x ⇒ b, $p = \frac{p(b)}{p(a)+p(b)+p(c)}$
x ⇒ c, $p = \frac{p(c)}{p(a)+p(b)+p(c)}$

Even though many expert systems do not explicitly follow such a construction, the heuristic certainty factors present in systems such as MYCIN ([1]) are an attempt to approximate such inference and thus they are subject to the same limitations[1]. When an assertion is corroborated - asserted through a different inference - its certainty factor is increased correspondingly to reflect this added degree of confidence.

Thus, assuming the probabilities:

```
P(poor-drainage) = 0.1
P(low-quality-concrete) = 0.15
P(insufficient-covering-of-reinforcement) = 0.25
```

the knowledge about excessive-staining can be transformed into the following deductive rules:

```
excessive-staining ⇒ poor-drainage (CF = 0.1/0.5 = 0.2)
excessive-staining ⇒ low-quality-concrete (CF = 0.15/0.5 = 0.3)
excessive-staining ⇒ insufficient-covering-of-reinforcement (CF=0.25/0.5=0.5)
```

The assumptions underlying the simulation of abduction through deduction, however, lead to significant difficulties. First, there is no correct method for combining certainty factors which can take into account interdependence between inference rules. Consequently, it is not possible to guarantee that the results of the inference are always correct. Second, the different possibilities are usually not mutually exclusive. For example, there may well be several causes for one and the same problem. The deductive framework provides no reliable way for dealing with multiple solutions.

Third, the closed-world assumption underlying the construction of the rules is put into question as soon as new knowledge is discovered and has to be added to the system. For example, imagine that it is newly discovered that overstressing causes excessive cracking which in turn causes excessive staining. This could be expressed as a rule:

```
overstressing ⇒ excessive-staining
```

But this means that *all* certainty factors involving excessive-staining have to be revised. Assuming that the probability of overstressing is P(overstressing) = 0.1, the revised rules would now read:

```
excessive-staining ⇒ poor-drainage (CF = 0.1/0.6 = 1/6)
excessive-staining ⇒ low-quality-concrete (CF = 0.15/0.6 = 0.25)
excessive-staining ⇒ insufficient-covering-of-reinforcement (CF=0.25/0.6=5/12)
excessive-staining ⇒ overstressing (CF = 0.1/0.6 = 1/6)
```

[1]The construction given here should not be confused with the technique of *Bayesian networks*, which perform abduction with probabilistic knowledge.

Especially when certainty factors have been obtained through heuristic estimates and tuned so that the system gives the correct answers, such a revision can be a very expensive, if not impossible, task. The limitations become completely inacceptable when rule sets are incomplete and have to be modified while the system is being used.

It is therefore desirable to look for other ways of implementing abductive inference that do not give rise to such problems. This is a primary motivation for *model-based reasoning.*

4 Abductive Inference in Model-based Reasoning

Knowledge about physical systems is generally available in the form of *models.* A model of a device or a part thereof is expressed by a simulation rule of the form:

`cause` $\Rightarrow$ `effect`

When knowledge is formulated as models, tasks such as diagnosis and design require abductive inference. In fact, during the previous discussion in this paper, it was tacitly assumed that knowledge was given in the form of models.

While classical knowledge-based systems *compiled* models into deductive rules, *model-based reasoning* allows using models directly without the need for such compilation. Model-based reasoning systems have many advantages over heuristic expert systems:

- knowledge is straightforward to formulate and maintain.
- the results can be guaranteed to be correct whenever the underlying models are correct.
- combinations of multiple solutions can be found.

Among users of the technology, the prime motivation for model-based reasoning has been the ease of formulating knowledge. However, as we shall now see, the explicit implementation of abduction in model-based reasoning systems also offers significant advantages from the computational point of view, namely guarantees of correctness and the ability to generate combinations of solutions. Model-based reasoning (MBR) has therefore become increasingly successful in recent years.

The general problem of abductive inference in the context of a MBR system can be stated as follows:

> Find all sets of combinations of causes $\{C_1, C_2, ..., C_k\}_j$ which logically entail *all* of the observed effects:
> $\{C_1, C_2, ..., C_k\}_j \vdash \{E_1, E_2, ..., E_n\}$.

This problem can be solved by inverting all rules which allow the inference of an effect E_i to generate the set of individual causes $\{C_j, C_l, ...\}$ which entail E_i. The set of potential solutions is then the set of all combinations of causes such that at least one possible cause for each E_i is contained in the combination. However, this set will contain enormously many solutions, making the problem intractable for practical problems. This is in fact one important reason for constructing heuristic expert systems.

As an example, consider the problem of diagnosing failures of an overhead projector using the models of the device shown in Figure 1. Given the problems

¬`image-lit`: the image is not lit, and
¬hum: there is no hum

abduction would first consider the 5 candidate combinations:

a: { ¬`proj-power` }
b: { ¬`proj-power` ∧ `bulb-broken` }
c: { ¬`proj-power` ∧ `fan-broken` }
d: { `bulb-broken` ∧ `fan-broken` }
e: { ¬`proj-power` ∧ `bulb-broken` ∧ `fan-broken` }

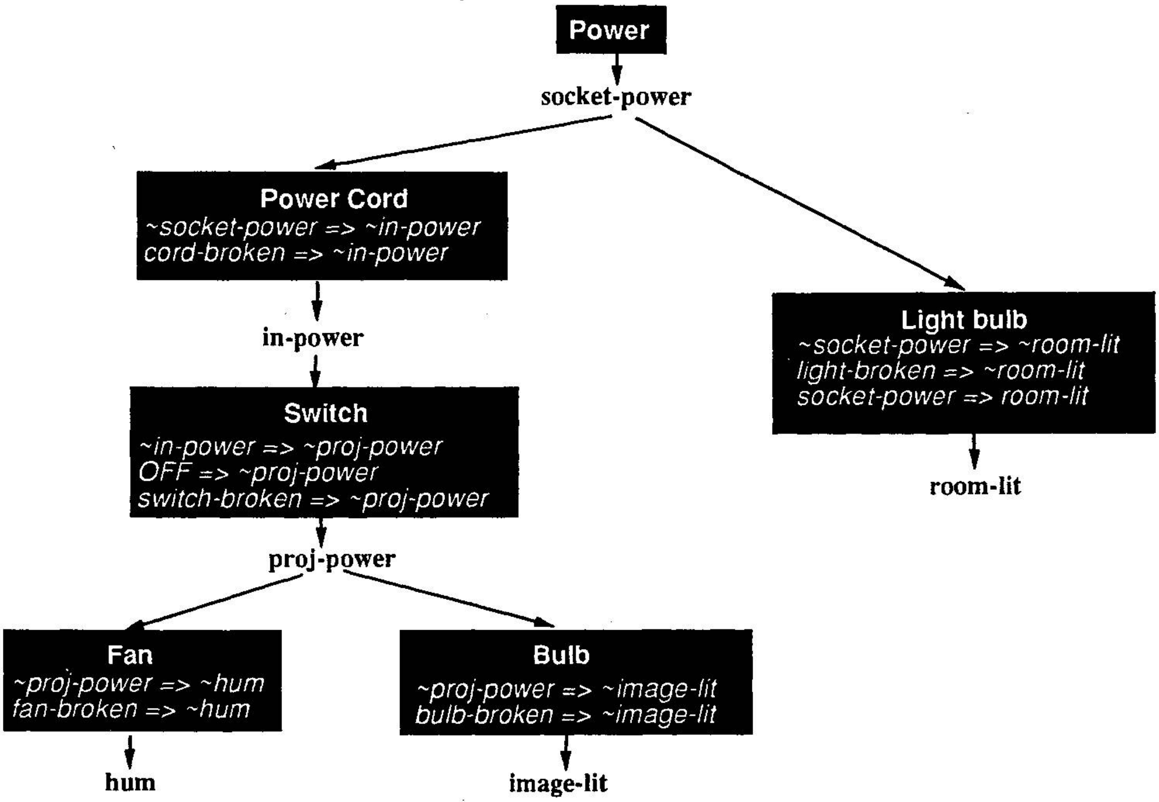


Figure 1: *Models used for diagnosing a projector.*

where candidate a) is of course the most likely one. By recursive abduction, ¬`proj-power` can in turn be explained by any of the $2^4 = 16$ combinations of

`switch-broken`, `OFF`, `cord-broken`, ¬`socket-power`

This means that there are altogehter $5 \cdot 16 = 80$ candidate combinations of causes to be searched. In an example of practical size, there may be thousands of candidates, resulting in an unmanageable complexity.

DeKleer observed ([2]) that the set of solutions could be described by specifying only *minimal* combinations which are required to entail the given conclusions. Any solution to abductive inference is in fact a superset of one or several such minimal combinations. This observation and its realization in an associated reasoning engine, the assumption-based truth maintenance system (ATMS), have been the basis for the practical success of model-based reasoning.

For the example of diagnosis, the intuition behind DeKleer's observation can be explained as follows. Assume that causes for failure are modelled by giving the faulty component, and that the set $\{C_1, C_2, C_3\}$ of faulty components entails all observed failures and is thus a solution to the diagnostic problem. Imagine now a fourth component C_4 which is really faulty, but its fault is masked by the faults of C_1, C_2 and C_3. Obviously, $\{C_1, C_2, C_3, C_4\}$ is also a solution to the diagnostic problem. In fact, since any component could potentially play the role of C_4, *any* superset of $\{C_1, C_2, C_3\}$ is a diagnosis. The very large space of potential diagnoses can be represented by the minimal candidates only, often an extreme economy. In the example of the projector failure, the space of 80 candidate combinations obtained from the symptoms ¬`image-lit` and ¬`hum` can thus be represented by the minimal candidates:

{ `bulb-broken` ∧ `fan-broken` }, { `switch-broken` }, { `OFF` }, { `cord-broken` }, { ¬`socket-power` }

Contrary to systems based on deductive rules which map symptoms directly into faults, it is now straightforward to reason about combinations of multiple faults. Furthermore, it is possible to bring

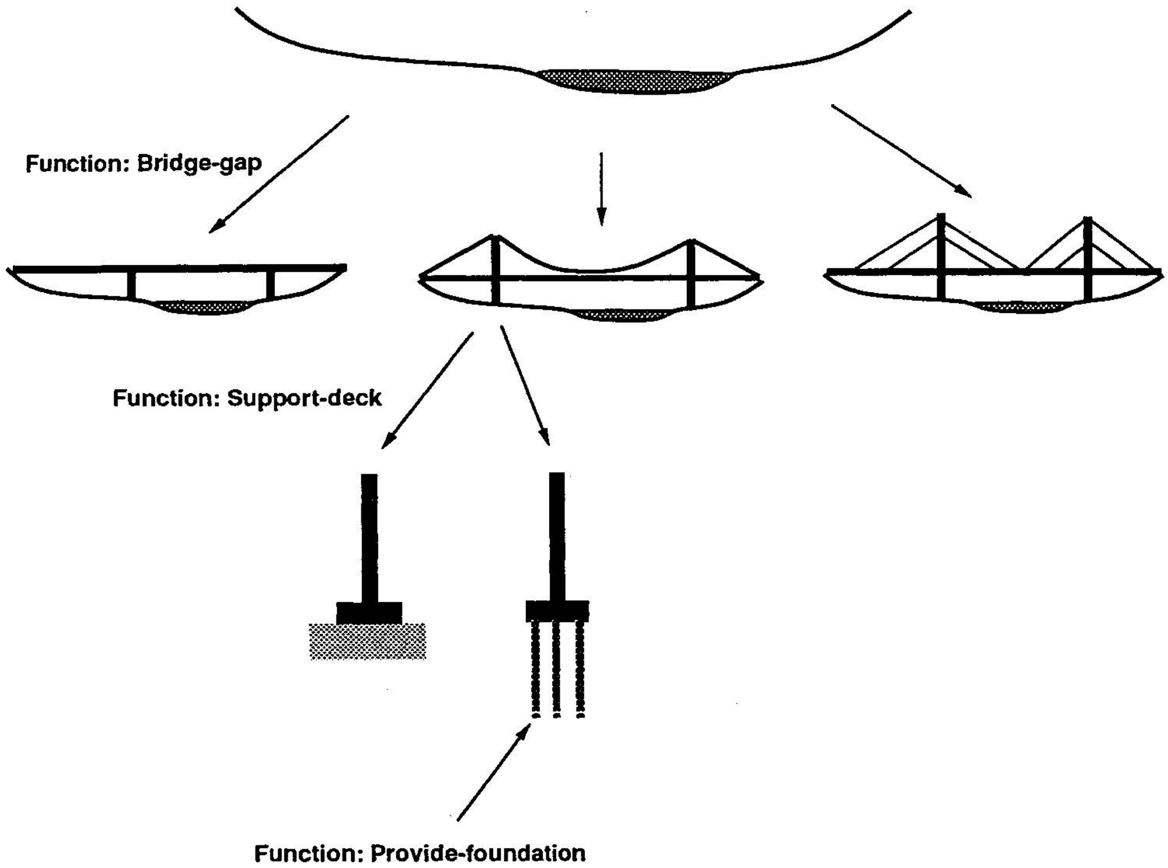


Figure 2: *Designing a bridge by functional decomposition. Shown here are two functions: stable support and providing the deck.*

new elements into consideration without profound changes to the knowledge base. For example, the fact that the room lighting is usually fed by the same electric circuit can be added to the knowledge base as the model of the light bulb, as shown on the right in Figure 1. Once this model has been added, the observation that ¬room-lit can be abduced to ¬socket-power and give this candidate a much higher probability. Conversely, the observation room-lit allows the abduction socket-power which rules out the conflicting candidate ¬socket-power.

The use of minimal candidates has been proposed as the key idea of a program called the General Diagnostic Engine (GDE, [3]) which uses an assumption-based truth maintenance system (ATMS) as a tool which generates the set of all minimal candidates in parallel. Since then, it has been shown that many abductive inference problems in diagnosis and design can also be solved efficiently using a sequential search, but nevertheless maintaining the advantage of computing with minimal combinations only. In general, when abduction can be applied recursively to arbitrary depths, as is common in design, the space of potential solutions is infinite and cannot be obtained using an ATMS, requiring instead sequential search.

Explicit abduction based on models has also been used in design, but with a less systematic approach due to the fact that design not as well-defined as diagnosis. Systems that perform design by functional decomposition, such as VEXED ([6]), perform abduction on rules of the form:

structure $\Rightarrow$ function

An example of how a bridge design might be obtained by such an abductive system is shown in Figure 2. The first goal to be abduced is that of providing a deck, which can be achieved by one of three bridge types. Depending on the solution chosen for the bridge, the second stage of abduction selects possible types of pier along with its foundations. In parallel, the goal of providing stable foundations

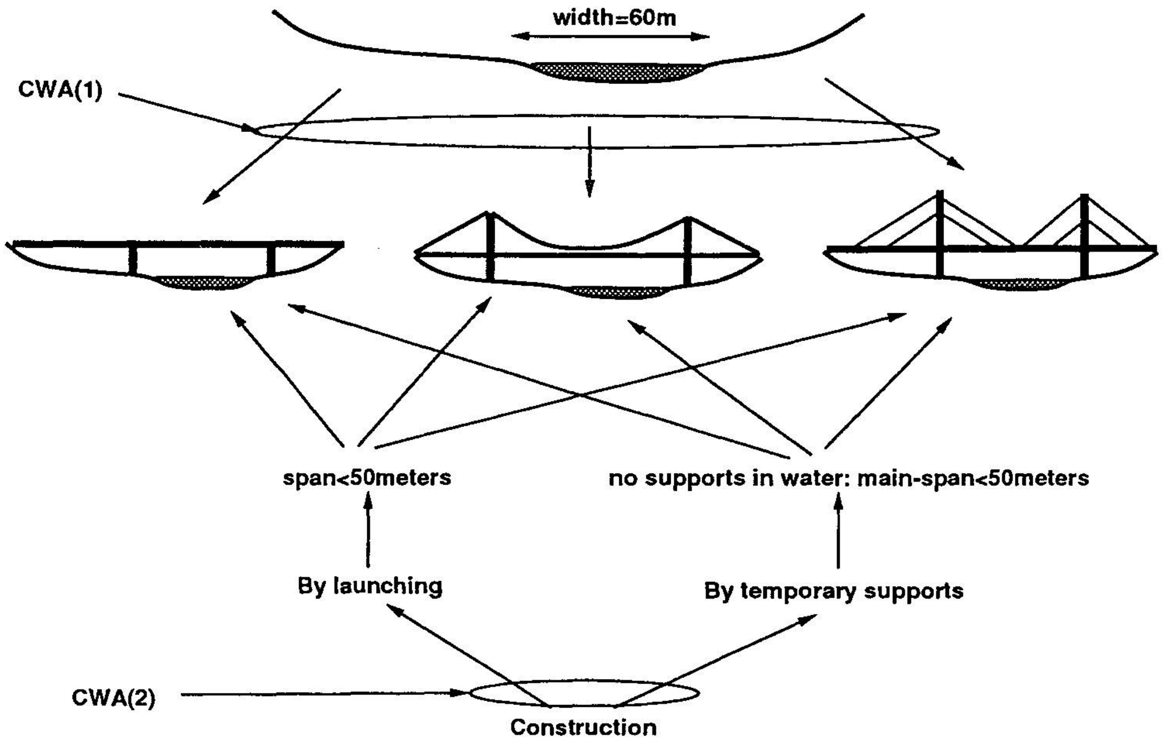


Figure 3: *Example of a conflict between conflicting requirements in a bridge design. The source of the conflict can be traced to one of two closed-world assumptions.*

leads to abduction of another set of choices on pier foundations. Since only one of them can be chosen, possible pier foundations are given by the intersection of the two sets.

Applying abduction in design is subject to complexity problems which are much worse than those observed in diagnosis. This is because the space of possible structures is usually very large, if not infinite, making the space of potential solutions impossible to search completely. Furthermore, because of compatibility constraints between components, the space is not *monotonic*: while a combination of structures $\{S_1, S_2\}$ can have function F, the superset $\{S_1, S_2, S_3\}$ may not have it due to interference of component S_3. The idea of minimal combinations is therefore far less useful in design than it is in diagnosis.

5 Using Explicit Closed-World Assumptions

The validity of an abductive inference depends crucially on a *closed-world assumption* (CWA) that there exists no other rule, unknown to the system, by which the observation could be obtained. The fact that a closed-world assumption is violated becomes obvious when the system does not find a solution, or when the solution proposed is wrong. In classical expert systems, it would be an extremely difficult problem to determine *why* the system did not find a better solution. In model-based reasoning, however, this can be solved more easily by explicitly representing the closed-world assumptions underlying the reasoning.

As an example, consider the design of a bridge across a river which is 60 meters wide (Figure 3). Assume that abduction to provide the main function - a deck spanning the river - results in three different bridge solutions shown in the figure, and a closed-world assumption CWA1 denotes the assumption that there are no other bridge types. On the other hand, designing the construction methods may leave only construction by launching and by using intermediate supports. The closed-world assumption CWA2 denotes the assumption that there are no other construction methods which apply to this problem. When no intermediate supports may be placed in the water, there is no construction method which is compatible with any of the proposed bridge types. This could mean that there is in fact no solution to the problem. However, it is more likely that the designer should look for other bridge designs or construction methods which do not have this conflict. This amounts to questioning one of the two closed-world assumptions, CWA1 and CWA2, which underly the contradiction.

A model-based reasoning system can use explicit closed-world assumptions in order to pinpoint the sources of conflict in reasoning. For design, this effect could be obtained by adding to every abductive inference an additional possibility `CWAi-violated` which is never in conflict with any other part of the solution. Solutions which contain such possibilities indicate solutions that would exist given additional possibilities. The designer can thus *choose* whether and where to look for innovative solutions in order to improve the design.

In model-based diagnosis, the use of explicit closed-world assumptions has been introduced by the work on GDE+ ([7]). The system starts its diagnosis by considering only a limited set of the most common faults. When no solution can be found at this level, potential violations of closed-world assumptions guide the system to extend the set of faults under consideration to new candidates which could result in extending the set of diagnoses. In the example of the projector, one might first start by only considering faults of the bulb the fan and the switch. However, if all of them have been inspected and found to be working, the system might extend its search to also suspect the power cord and the electricity supply. In this way, a very large space of potential diagnoses can be considered while still maintaining the efficiency of the system.

6 Conclusions

Most knowledge of physical systems is formulated in the form of *models*, mapping characteristics of devices and structures into behaviors. Consequently, many engineering tasks require abductive or inductive inference. In contrast to deduction, which always provides sound solutions, abduction and induction often produce ambiguous results. These ambiguities are one of the main motivations for the use of knowledge-based systems.

However, classical deductive expert systems provide poor support for such inference strategies. Simulating abduction in a deductive framework requires the use of certainty factors or other probabilistic mechanisms in order to discriminate between potential solutions. These require several unrealistic assumptions, and make it difficult, if not impossible, to extend an expert system with new knowledge.

The framework of model-based reasoning is based on explicit abduction on models. It allows formulating knowledge in a modular way as models, and performing abductive inference in a sound and potentially efficient way. Furthermore, explicit formulation of closed-world assumptions makes it possible to detect missing knowledge which precludes reasoning from providing useful results. Such advantages mean that model-based reasoning should be considered for every knowledge-based system in civil engineering.

Acknowledgements

I would like to thank Dr. Ian Smith, LIA-EPFL, for discussions about knowledge-based systems in civil engineering.

References

[1] **Randall Davis, Bruce Buchanan, Edward Shortliffe**: "Production Rules as a Representation for a Knowledge-based Consultation Program," *Artificial Intelligence* **8**, 1977

[2] **Johan deKleer**: "An Assumption-based Truth Maintenance System," *Artificial Intelligence* **28**, 1986

[3] **Johan deKleer, Brian Williams**: "Diagnosing Multiple Faults," *Artificial Intelligence* **32**, 1987

[4] **R.A. Kowalski**: "Logic for Problem Solving," Elsevier, 1979

[5] **Allan Newell, Herbert Simon**: "GPS - A Program that Simulates Human Thought," in E. Feigenbaum, J. Feldman (eds.): *Computers and Thought*, McGraw-Hill, New York 1963

[6] **Louis Steinberg**: "Design as Top-Down Refinement plus Constraint Propagation," in: C. Tong and D. Sriram (eds): *Artificial Intelligence in Design* **1**, 1992

[7] **Peter Struss, Oskar Dressler**: "Physical Negation: Integrating Fault Models into the General Diagnostic Engine," *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Morgan-Kaufmann, 1989