Computational Complexity and KR&R:
Is Polynomial Time all that Matters?*

Position Paper for the
Workshop on Tractable Reasoning

Bernhard Nebel
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
W-6600 Saarbrücken, Germany
nebel@dfki.uni-sb.de

Introduction

Tractability, i.e., worst-case solvability in polynomial
time, is considered as a necessary ingredient for any
computational process modeling intelligent behavior
from a cognitive as well as technological point of
view [Levesque and Brachman, 1987; Levesque, 1988;
Bylander, 1991]. Although I agree with this point of
view in principle, I will argue in the following that it is
often appropriate to make finer distinctions than be-
tween worst-case tractable and intractable problems:

• Classifying a problem to be complete for a com-
plexity class provides us with more information
about the structure of the problem and the poten-
tial sources of complexity than just classifying it as
NP-hard.

• After having shown a problem to be NP-hard,
it is interesting to find the precise border be-
tween tractable and intractable subproblems. Of-
ten, it turns out that “natural” restrictions lead to
tractability, i.e., it is not necessary to develop alter-
native semantics or to restrict the expressiveness in
a drastic way. Alternatively, such an analysis may
show that the original definition of the problem is
incorrect.

• Having shown polynomial time solvability does not
mean that the problem can actually be solved in
practice. For this purpose, efficient algorithms have
to be developed. Sometimes, however, only empiri-
cal investigations can tell us which algorithm is effi-
cient in practical cases.

In the next sections, I will justify these arguments us-
ing different examples from the literature and my own
research.

Completeness Results

Often it is considered to be sufficient to prove a prob-
lem to be NP-hard and it is argued that a complete-
ness result w.r.t. some level of the polynomial hierar-
chy, PSPACE, EXPTIME etc. is not relevant from a
“practical” point of view. Of course, this argument is
correct as far as solvability in polynomial time is con-
cerned. However, most of the time we are interested
in identifying subproblems that are tractable. For this
purpose, it is helpful to find the complexity class the
general problem under consideration is complete for.

Such results usually tell a lot about the structure
of the problem and the sources of complexity. For in-
fstance, if a problem can be proven to be PSPACE-
complete, such as subsumption checking in the termi-
нологical logic $\mathcal{ALC}$ [Schmidt-Schauß and Smolka,
1991], removing only one source of complexity (such
as disjunction and negation or the nesting of $\forall$ and $\exists$
quantifiers) will most probably not result in tractabil-
ity [Donini et al., 1991]. Similar arguments hold for
problems that are complete for the second level of the
polynomial hierarchy such as syntax-based belief re-
vision or default reasoning [Nebel, 1991]. Addition-
ally, only if the potential sources of complexity of a
problem have been identified, a coherent presentation
of tractability results for subproblems is possible (see,
e.g., [Eiter and Gottlob, 1991]), I believe.

Another interesting application of completeness re-
results is the possibility to derive (positive or negative)
results concerning the “equivalence of expressiveness.”
For instance, in the area of nonmonotonic logics dif-

* This work has been supported by the German Min-
istry for Research and Technology (BMFT) under contract
ITW 8901 8.
ferent systems can be related by “translations” from
one logic into another one [Kolonige, 1988]. Since all
propositional nonmonotonic logics (AEL, DL, circumscrip-
sion) turn out to be $\Pi_2^P$-complete w.r.t. skeptical
reasoning [Gottlob, 1991], it follows straightforwardly
that they can be translated to each other using a poly-
nomial transformation.

Another example concerns the relationship between
abductive and consistency based diagnosis. In a re-
cent paper, Kolonige [1992, p. 257] conjectures that
it is not possible to transform a general propositional
causal theory in a “local way” into a form such that
consistency based diagnosis gives the same results on
the transformed theory as abduction on the original
theory. Considering the fact that consistency based di-
agnosis (as described by Reiter [1987]) is NP-equivalent
and abduction is $\Sigma_2^P$-equivalent, this conjecture can be
confirmed (provided $\Delta_2^P \neq \Sigma_2^P$).

NP-Hardness and
Tractability of Subproblems

The ultimate goal of the analysis of cognitive tasks
from a computational complexity point of view is, of
course, the identification of tractable methods. If a
particular problem turns out to be NP-hard but the
problem is apparently solved effortlessly by human be-
ings, it is an indication that the problem formulation
is too general (or simply wrong). Levesque and Brach-
man [1987] suggest to cope with this problem by re-
stricting the expressiveness or by using a semantics
that is weaker than the standard two-valued seman-
tics in order to achieve tractability.

Although these are possible solutions, often “nat-
urally occurring restrictions” are enough to make a
problem tractable. Examples are the continuous end-
point algebra [Vilain et al., 1989; Nökel, 1989], the re-
striction of sketch maps to maps where no river has a
source on a street [Selman, 1991], and the restriction of
the role-chain depth in terminologies [Nebel, 1990].

An even more drastic case is the two-level morphol-
y. Parsing is NP-complete (in the size of the rule set)
[Barton, 1986]. However, the combinatorical explosion
never shows up since the number of rules responsible
for the explosion is very limited (at most 2) for each
(known) natural language [Koskenniemi and Church,
1988].

Of course, it is not always possible to identify such
“natural restrictions” and the only way to achieve
tractability is to come up with some form of approx-
imation or with an incomplete method. While it is
not obvious how to measure the “quality” of incom-
plete methods, I claim that a general-purpose approx-
imation method should at least give accurate results
if it is applied to important special cases that can be
solved in polynomial time. For instance, Allen’s con-
straint propagation method to compute minimal labels
for time interval networks is incomplete in general but
complete for the important special case of the con-
tinuous endpoint algebra.

A negative example is Dean and Boddy’s incomplete
method for temporal projection [Dean and Boddy, 1987;
Dean and Boddy, 1988]. Although the method is com-
plete for the tractable case of totally ordered event
sets, it turns out to be incomplete for another im-
portant special case that can be solved in polynomial
time, namely, unconditional non-linear plans [Nebel
and Bäckström, 1991]. In fact, the definition of tempo-
ral projection itself seems to be wrong since for some
special cases planning appears to be easier than tem-
poral projection. Further, the claim that temporal
projection is the problem underlying plan validation
seems not to be justified because the latter problem
is tractable for unconditional, non-linear plans while
the former problem is NP-hard in this case [Nebel
and Bäckström, 1991; Nebel and Bäckström, 1992]. Sum-
marizing, this seems to be an example where a com-
plexity analysis is helpful in judging the validity of the
definition of a problem.

Tractability versus Efficiency and the
Utility of Empirical Investigations

Although proving a problem to be tractable is desir-
able, it does not mean that the problem can now be
solved in practice. Even if a low-order polynomial al-
gorithm has been identified, this still means that the
runtime can grow faster than desired. For instance,
considering terminological representation systems, the
best known classification algorithms (for constructing
the presentation of a partial order) all seem too have a
worst-case and “practical case” complexity that is not
better than $O(n^2)$ ($n$ being the number of concepts)
[Heinsohn et al., 1992].

From a theoretical point of view, quadratic com-
plexity is quite good. From a practical point of view, it
means, however, that knowledge bases with 1000 con-
cepts are easily dealt with while KBs that are ten times
larger require a couple of hours. In order to give an im-
pression what that means in practice, Figures 1 and 2
(taken from [Heinsohn et al., 1992]) show the runtime
plotted against the knowledge base size for randomly
generated knowledge bases for six different terminolog-
cal knowledge representation systems.

Assuming that very large knowledge bases (i.e., KBs
larger than 10,000 concepts) will become relevant in
the near to mid-term future, there exists clearly a
need to make classification of terminological knowledge
Constructing the presentation of a partial order from an underlying partial order needs $O(n^2)$ comparisons in the worst case when all elements are incomparable. So, there is no chance of finding an algorithm with a better worst case complexity. Nevertheless, knowledge bases constructed in practice seem to be better behaved. However, it is very unlikely that we will be able to find algorithms that have a provably better complexity in the average case since we neither know how many partial orders exist for a given cardinality [Aigner, 1988] nor do we have an idea how to describe the structure of knowledge bases occurring in practice.

The only way to proceed at this point seems to be to use empirical methods to compare different classification algorithms. In an experiment, we tested a simple classification method, an enhanced version of this method that exploits intermediate results, and a classification method using a binary insertion strategy in a chain covering [Baader et al., 1992]. Figures 2 and 3 show the relative number of subsumption tests that were used in the simple and the enhanced method compared with the number of subsumption tests performed by the brute force method ($n \times (n - 1)$ tests) using random KBs as test data.

Interestingly, it turned out that the algorithm we conjectured to be the fastest one—the algorithm using binary insertion—is indeed very efficient on randomly generated partial orders but slower than the enhanced method on knowledge bases occurring in practice. This result shows that empirical investigations are indeed necessary. Another interesting result of our investigation was that the optimizations in the second method...
are effective only for some knowledge bases, namely, those that were designed for linguistic purposes. We do not have an explanation for this somehow funny behavior but conjecture that the reason for this behavior has to do with the fact that the linguistic knowledge bases contain more implicit relations between concepts than the other KBs.

References


Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In Allen et al. [1991], pages 151–162.


Klaus Nökel. Convex relations between time intervals. In J. Rettie and K. Leidlmair, editors, Proceedings der 5. Österreichischen Artificial Intelligence-Tagung,

