Vol. 1: *27-30*, 31.7.1997
http://www.ida.liu.se/ext/etai/received/actions/001/

Summary of on-line discussion during July, 1997 about the following research article:

Paolo Liberatore:

The Complexity of the Language A

Q1. **Erik Sandewall** (11.7)

Paolo,
What is the relationship (or is there any) between your results and the results by Nebel and Bäckström on the complexity of plan validation and temporal projection? Since basic cal-A characterizes updates to the current state, it is at least related to the STRIPS-based framework of classical planning, and deciding the consistency of a set of statements in cal-A would seem to be related to plan validation? For a reference to Nebel/Bäckström, see

Bernhard Nebel and Christer Bäckström:
On the Computational Complexity of Temporal Projection, Planning and Plan Validation.
j-aij-66-125([1]): Artificial Intelligence 66(1):125-160, 1994.

A1. **Paolo Liberatore** (14.7)

This is an interesting question. In general, entailment in reasoning about actions languages such $\mathcal{A}$ is strictly related to (deductive) plan validation. Indeed, a plan is a sequence of actions, and it achieves a goal G iff the goal is true after the execution of the sequence (regardless of the initial state).

As for the specific problems of consistency (or entailment) in $\mathcal{A}$ and plan validation as defined by Bäckström and Nebel, I think there are substantial differences. Let me explain in short what is the source of intractability of these problems. In $\mathcal{A}$ the NP-hardness is due to the

[1] All info:
http://www.ida.liu.se/%7Echrba/publications/article-full.html
Postscript:
ftp://ftp.ida.liu.se/pub/labs/tosca/people/chrba/aij94-v66.ps.gz

incomplete specification of the initial state: one has in general to consider an exponential number of possible initial states. The problems of temporal projection and plan validation analyzed by Bäckström and Nebel are intractable because the possible sequences of actions can be exponentially many. The initial state is fixed (and fully specified), but, in order to verify if a fact is true after an event (or to verify whether a goal is achieved), one has to consider all the possible sequences of actions that respect a given ordering.

Of course both problems (entailment and plan validation) are coNP complete, so one can be reduced to each other, but I do not think there is a "simple" and intuitive way to do that for entailment in $\mathcal{A}$ and plan validation as defined by Bäckström and Nebel.

Q2. **Michael Thielscher** (22.7)

Paolo, I have two questions/comments. First, Your proof of intractability of general $\mathcal{A}$ relies on the construction of effect propositions all of whose conditions are unspecified in the initial state. It therefore seems that even if the initial state is incomplete, if only the number of (relevant) unspecified fluents is small then both consistency checking and entailment is still polynomial, am I right?

A2. **Paolo Liberatore** (24.7)

Yes. Indeed, NP completeness is due to the fact that the possible initial states are exponentially many. If there is only a constant (small) number of unspecified fluents in the initial state, then the number of initial states is constant (in complexity theory, if c is a constant, then 2^c is also a constant). The same holds also when there is a non-constant number of unspecified fluents in the initial state, but no effect proposition have them as preconditions.

Q3. **Michael Thielscher** (22.7)

My other question is, Did you have a look at extensions of $\mathcal{A}$ which support nondeterministic actions? Does tractablity in case of complete initial states still hold?

A3. **Paolo Liberatore** (24.7)

I have not studied the complexity of the extensions of $\mathcal{A}$ in details. At a first look, it seems easy to prove that the dialects that include concurrent actions (the proposals by Baral&Gelfond and Li&Pereira) are more complex than basic $\mathcal{A}$, namely Σ_2 for consistency and Π_2 for entailment. The complete knowledge of the initial state should reduce the complexity to NP and coNP, respectively. The proposal by Bornscheuer&Thielscher should be as easy as basic $\mathcal{A}$ (NP and coNP complete). About non-deterministic actions: I had a look to $\mathcal{AR}_0$ (proposed by Kartha&Lifschitz), and it seems to me that it should be Σ_2 also. A complete knowledge of the inital state does not help, since the domain description can always contain propositions such `A releases F`. There are other formalizations of non-deterministic actions, but I have not analyzed them yet.

Q4. **Erik Sandewall** (28.7)

Paolo,
Your construction in section 4 seems to be similar but not identical to PMON; let me ask you about what the relation may be. In your case, if the action A_k goes from node i to node j, you introduce for each fluent F_o the clauses

$$x_o^j \leftarrow c_o^j, \neg x_o^i$$

$$x_o^j \leftarrow \neg c_o^j, x_o^i$$

Furthermore, each effect proposition, for example

$$A_k \text{ causes } F_0 \text{ if } F_1, F_2$$

is mapped into a clause

$$c_0^j \leftarrow x_1^i, x_2^i, \neg x_0^i$$

The last literal is needed because c_0^j expresses that F_0 has to change from node i to node j, whereas the effect proposition merely assigns its new value, regardless of whether it had it before or not. (I hope I got this right).

Now, this is similar to how occlusion is used in PMON, except that occlusion only states a "permission" to change. (Occlusion is the same as the "release" operator later adopted by Lifschitz). Therefore, the counterpart of the first two clauses above are the "nochange" axioms, which in your notation would be

$$c_o^j \leftarrow x_o^i, \neg x_o^j$$

$$c_o^j \leftarrow \neg x_o^i, x_o^j$$

which together say "if $x_o^i \neq x_o^j$ then c_o^j". For things to come out right, in PMON one has to minimize occlusion before applying the nochange axioms. In your case there does not seem to be any corresponding partitioning of the axioms for the purpose of completion.

My question is whether the difference between your clause pair and the nochange axioms is crucial for the construction, and in particular, whether that is the reason why partitioning of the axioms does not seem to be necessary.

PMON was introduced in "Features and Fluents" (my book at Oxford University Press, 1994) and in Doherty's article at ETAI 1994.

A4. **Paolo Liberatore** (30.7)

I will answer to the last question first. From a computational point of view, entailment under completion is "only" coNP compete, since the completion of a set of clauses is a propositional formula of polynomial size. On the other hand, circumscriptive entailment is Π_2 complete, thus harder.

Since PMON requires a minimization of the occlusion predicate, it should be harder than the completion formulation of $\mathcal{A}$. On the other hand, PMON allows a correct formulation of non-deterministic actions, while completion does not. Right now, I do not see an easy way of extending the translation of section 4 without introducing an explicit minimization, thus increasing the complexity. Of course, this increase of complexity should not be considered bad, since languages with non-deterministic actions are surely harder (in the general case) than simple languages. I agree with you that extending the translation in the sense you pointed out should allow the identification of subclasses of the problem for which the computational complexity is only coNP, since it should be easier to find classes for which the minimazation policy can be replaced by a simpler completion.