Electronic News Journal on Reasoning about Actions and Change 38
Vol. 1: *38–45*, 30.9.1997
http://www.ida.liu.se/ext/etai/actions/nj/9709-1/

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

Wolfgang Bibel:

Let's Plan It Deductively!

The recent IJCAI conference featured an invited talk by Wolfgang Bibel, titled "Let's plan it deductively!" The talked addressed issues within Reasoning about Actions and Change as well as within planning, and is of considerable relevance for our area. Since the format of a conference lecture with a large audience does not allow much discussion, we have invited Wolfgang to participate in an on-line discussion about his article in the present medium, and he has agreed heartily. Questions to Wolfgang can be sent to the Newsletter editor, and will be forwarded to him for answering. Questions and answers will be distributed through Newsletter issues, and will also be accumulated in discussion sessions in the Colloquium. It is intended that this discussion will have the flavor of a conference question period, so it is perfectly fine if the contributions develop into a broader discussion. The following formatting conventions are used. Roman style is used for the statement, question, or answer by the person indicated in the headline just above. *Italic style* is used to indicate sections of text which are customarily indented and marked by > at the beginning of each line in E-mail discussions, that is, quotations from a preceding statement and which the present author is responding to. The **date** at the top of each contribution indicates the date of the Newsletter where that contribution was first presented. This is not necessarily the date when the contribution reached the Newsletter editor; in some cases there has been a delay of one or a few days.

[30.9.1997] Wolfgang Bibel's article is now reported to be available on-line on his web page,
http://aida.intellektik.informatik.th-darmstadt.de/˜bibel/#publications

From: **Marc Friedman** on 22.9.1997.

Dr. Bibel,

I am intrigued by your passion for posing planning as a deduction problem. Since your lecture at IJCAI 97 in Japan, I have been trying to learn all I can about it. But I have some concerns which I think will have to be addressed for your work to be fully appreciated by those of us who are familiar with planning but not really with deduction.

In the process of questioning your work, I will probably do violence to it. I apologize in advance. Please correct me.

0. Linear transition proofs solve the classical planning problem. This is true. Linear backward chainer (LBC), and LIF and LIF+ (Fronhoefer's more recent solutions) are correct. LBC is very clean, too.

1. I think you suggest that deduction has a priveleged place as a basis for classical planning. But planning has other theoretical foundations: the modal truth criterion, and the theory of refinement search. What makes deduction superior to these other bases?

2. You present LBC as an encoding of transition logic (TL) into logic, in particular the languuage of the SETHEO theorem prover. If this were true, AND the implementation compared well with other classical planners, this would be a major step – giving at once a formal AND operational reduction of the problem to deduction!

However, if we look closely at LBC, there is a work-around to make SETHEO into a transition logic engine. TL is not in fact translated to first order logic. Instead, the available propositions are tracked, to prevent two connections from sharing a single proposition. This approach is not truly a reduction. It is an encoding, much like a program that implements a formally sound algorithm, like UCPOP or graphplan, in a formally sound substrate, like PROLOG, or a functional programming language, or as a satisfiable formula. TL loses its priveleged position. Thus it must compete with other approaches on their terms: is it faster or easier to understand, does it do less search, etc.

3. LBC beats UCPOP. But many algorithms have. How does it compare with these?

4. Transition logic solves the frame problem. So does TWEAK. Transiion logic solves the ramification problem. So does UCPOP, via a theorem-proving subroutine. Perhaps TL's ramification solution is a more uniform mechanism, but it is not truly uniform – the linearity restriction is removed. Why prefer one solution to the other?

I think some of the answers are beginning to come to light, and I eagerly await hearing your answers.

From: **Wolfgang Bibel** on 26.9.1997.

Dear Mark Friedman,

Thank you very much for taking your time and listening to my lecture at IJCAI-97, and even giving it further thoughts afterwards.

> *0. Linear transition proofs solve the classical planning problem. This is true. Linear backward chainer (LBC), and LIF and LIF+ (Fronhoefer's more recent solutions) are correct. LBC is very clean, too.*

Thanks for these kind remarks.

> *1. I think you suggest that deduction has a priveleged place as a basis for classical planning. But planning has other theoretical foundataions: the modal truth criterion, and the theory of refinement search. What makes deduction superior to these other bases?*

Let me take this question to start with reminding you of the general gist of my lecture. My lecture ended by generalizing the title to "Let's plan AI deductively!". As we all know the AI endeavor is a very complex one. This complexity led many of us to specialize in small niches such as planning, nonmonotonic reasoning, scheduling, theorem proving, vision, speach, NL, ... you name one of the hundreds of others. In each of them smart (functional - in a broad sense of the word) solutions are developed in a great variety of different, mostly incompatible languages. I do not see how all this could ever converge towards something coherent which deserves the label "artificial intelligence", our common goal. I am not the first to recognize a deplorable splintering of our field. An artificially intelligent agent will have to feature those (and more) smart solutions all at the same time. I do not see how the functional approach could ever achieve this if it does not even get us to the point of hooking a new machine easily to a local network (a problem I faced recently coming here to UBC with my Voyager).

I am therefore one of those who strongly believe that only through a rather uniform approach to any of these different facets can we ever hope to accomplish systems that are able to do more than "just" playing chess on a worldmaster level (but nothing else) or prove open problems like Robbins' one as done recently (but again nothing else) etc. It is an illusion to think we could just combine all these niche systems to get out something like a general intelligence. Rather the entire approach must be a more universal one from the very beginning. And if so only through a uniform approach could the enormous complexity be overcome.

If you buy these arguments the next question will be "what uniform approach". There are not that many available. In fact I believe that the logical approach triggered by John McCarthy has no real competitor satisfying all the requirements coming up for such a universal task. Of course this is a rather vague statement since it leaves open what we in detail mean by "logical approach". For the time being many believe that core first-order logic would be part of it, but that there might be variations of it not yet found (like the transitions in TL, second-order predicates etc). Another concern is the lack of efficiency of existing deductive methods a point I come back to shortly.

After these, as I think, important remarks in view of what we are up to I come now back to the details of your question. The modal truth criterion or the theory of refinement search could actually be seen as logical theories. But they are very specialized theories customized to planning and nothing else than planning. While I could well imagine a logical planning environment where they ALSO might play some role (as meta-knowledge) by itself they are by far not sufficient to allow even the simplest extension of the planning task to include for instance a bit of classical reasoning of the planning agent. By the way deduction provides a generic problem solving mechanism in a uniform and rather universal logical language and as such plays a different role in comparison with the examples you mention. For instance the theory of refinement search applies to general deductive search as it does to specific search in the space of partial plans.

> *2. You present LBC as an encoding of transition logic (TL) into logic, in particular the languuage of the SETHEO theorem prover. If this were true, AND the implementation compared well with other classical planners, this would be a major step – giving at once a formal AND operational reduction of the problem to deduction!*
>
> *However, if we look closely at LBC, there is a work-around to make SETHEO into a transition logic engine. TL is not in fact translated to first order logic. Instead, the available propositions are tracked, to prevent two connections from sharing a single proposition. This approach is not truly a reduction. It is an encoding, much like a program that implements a formally sound algorithm, like UCPOP or graphplan, in a formally sound substrate, like PROLOG, or a functional programming language, or as a satisfiable formula. TL loses its priveleged position. Thus it must compete with other approaches on their terms: is it faster or easier to understand, does it do less search, etc.*

I have to start again with a general remark. We (ie my group in Darmstadt, my former group in Munich now represented by people like Fronh"ofer, Letz, Schumann and others, and the entire deduction community for that matter) see our task to provide the best possible generic problem solving mechanism for this universal language (mostly fol). And this challenge keeps all of us busy enough. As an aside I might mention that we have been quite successful in it. For instance in 1996 SETHEO won the world competition among all existing theorem provers. Why then should we above all these efforts do more than offering other specialists (such as those working in planning - but there are many more potential applications) our tools for use in their special field of application? So the little experiment that Fronh"ofer did with SETHEO was indeed only a side-effort done in a few days. It is true that beyond SETHEO we need a TL-SETHEO, ie a theorem prover customized to the logic TL. We did such extensions already for other logics, especially for intuitionistic logic which is relevant for program synthesis (another of our interests) and will, if circumstances permit, eventually do the same for TL. But given that there is so much to be done anyway no promises are given at this point.

As to the privileged position of TL I just point to what I said before: it is the universal logic (ie language and calculus) which gives it the privileged position in comparison with UCPOP (a special algorithm with a narrow range of applications) or graphplan (a coding in propositional logic which does not provide rich enough a logical language to serve the more general purposes - by the way graphplan is a deductive solution to planning as well! but as recent experiments by a Swedish/German group seem to demonstrate rule-based encoding seems to be a more suitable encoding).

> *3. LBC beats UCPOP. But many algorithms have. How does it compare with these?*

Sure. But if I take a general tool from the shelf (such as SETHEO), spend a few hours or days to customize it to a general task like planning and beat with the result a system that was specially developed for the task of planning and only a few years ago was deemed the best of its kind then this is a very strong experimental hint that the deduction technology subsumes that needed for planning and that the efficiency problem already mentioned above is less severe than many might think. Therefore, to my strongest conviction, Dan Weld and others would have contributed more to the advancement of AI had they built those special systems ON TOP of the mature technology reached in deduction at the time of implementation rather than as an independent sideline (which does not at all diminish their remarkable achievements seen in themselves).

> *4. Transition logic solves the frame problem. So does TWEAK.*

TWEAK is based on STRIPS (sort of) and - as I mentioned explicitly in my lecture - STRIPS is very closely related to TL as far as transitions are concerned. But STRIPS (and TWEAK) is not a logic so lacks the generality needed for the purposes outline above.

> *Transition logic solves the ramification problem. So does UCPOP, via a theorem-proving subroutine. Perhaps TL's ramification solution is a more uniform mechanism, but it*

> *is not truly uniform – the linearity restriction is removed. Why prefer one solution to the other?*

To the best of my knowledge Michael Thielscher (to whose work I referred in my lecture in this context) was the first who gave a solution to the ramification problem which overcomes deficiencies of any previous solution (including UCPOP's one). The lecture as well as the paper (and further references therein) point out these examples where no previous solution would model reality in a correct way. A better solution in this sense must be preferred to a deficient one. In addition there is the uniformity and universality provided by the logic as pointed out now already several times. I do not understand what you mean by the phrase "the linearity restriction is removed".

Wolfgang Bibel

From: **Marc Friedman** on 26.9.1997.

> *...by the logic as pointed out now already several times. I do not understand what you mean by the phrase "the linearity restriction is removed".*

Oh. Maybe I said it wrong. I meant that if there are synchronic rules, and transition rules, represented in your talk by two different kinds of implication arrows, then there are two different mechanisms – one which limits each proposition to use in a single connection, and one which does not.

Thanks, Marc

Answer from: **Wolfgang Bibel** on 29.9.1997.

> *Oh. Maybe I said it wrong. I meant that if there are synchronic rules, and transition rules, represented in your talk by two different kinds of implication arrows, then there are two different mechanisms – one which limits each proposition to use in a single connection, and one which does not.*

But then - what should be the problem with this technical change in the syntactic characterization of valid formulas which reflects our semantic use of (two different) implications?

From: **Patrick Hayes** on 29.9.1997.

I'm surprised to read that any theorem-prover has solved the frame problem, since the FP is a problem in representation, not in theorem-proving, and has nothing particularly to do with how deductions are processed. It is also rather surprising to read that some specialised logic has solved the FP, since to do so its semantics would have to embody all known and future causal laws. Could someone briefly explain how a better deductive search engine, or an exotic logic, can solve a problem in representation?

On what might be a related matter, Bibel claims that ' deduction provides a generic problem solving mechanism ' (response to Friedman, ETAI Newsletter on Actions and Change, 26.9.1997). Taken literally, this is clearly false, since deduction itself provides no mechanism whatever: one only gets a mechanism when one chooses a strategy for performing deductions. For example, unification is not imposed by deduction; other strategies for instantiating universal variables are possible, computationally ridiculous but deductively perfectly valid. So Bibel must be understood as referring not to 'deduction' per se, but to a particular deductive strategy, or class of deductive strategies. Perhaps in his original lecture (which I havn't yet got access to) he tells us which ones they are, but a brief summary would be helpful.

Answer from: **Wolfgang Bibel** on 30.9.1997.

> *I'm surprised to read that any theorem-prover has solved the frame problem,*

Where did you read this phrase? If I said so (which I hope I did not) then I apologize for it. You are clearly right with:

> *since the FP is a problem in representation, not in theorem-proving, and has nothing particularly to do with how deductions are processed.*
>
> *It is also rather surprising to read that some specialised logic has solved the FP, since to do so its semantics would have to embody all known and future causal laws.*

How do you mean this sentence? Does, in analogy, the semantics of first-order logic "have to embody all known and future" declarative knowledge?

> *Could someone briefly explain how a better deductive search engine, or an exotic logic, can solve a problem in representation?*

Why don't you read the respective parts of the paper. Again, at issue is a logic not a search engine (although my paper does also connect the logic with fol for which there is a great variety of search engines available). The logic is equivalent with linear logic which you may have heard of before.

> *On what might be a related matter, Bibel claims that ' deduction provides a generic problem solving mechanism ' (response to Friedman, ETAI 26.9.1997). Taken literally, this is clearly false, since deduction ...*

The context makes it clear that "deduction" refers to an area of research. It is indeed the goal of the deduction community to work out such generic problem solving mechanisms.

> *... itself provides no mechanism whatever: one only gets a mechanism when one chooses a strategy for performing deductions. For example, unification is not imposed by deduction; other strategies for instantiating universal variables are possible, computationally ridiculous but deductively perfectly valid.*

of course, of course

> *So Bibel must be understood as referring not to 'deduction' per se, but to a particular deductive strategy, or class of deductive strategies. Perhaps in his original lecture (which I havnt yet got access to) he tells us which ones they are, but a brief summary would be helpful.*

As I just said "deduction" refers to a field, a community.