Another proof of the Harmony Theorem

The following cases are straightforward:

Proof of Theorem 2

Induction on the complexity of the Program

So now we must look into the simultaneous inductive

Proof of Theorem 2

Induction on the complexity of the Formula φ

Moreover, the substitutions to the right are simultaneous for all A.

For all A, a, and programs, there exists an

Auxiliary Strengthened Reduction

For all A, q, and programs, with action a,

Main Reduction

Theorem 2, Theorem 1

Dynamic-epistemic Languages. We then prove the latter is no richer than its static base.

Our static language is E-PDL; we add modal operators and update models. A, V, a, n, V, a, n, k.

Another proof of the Harmony Theorem

DRAFT, Johan, 6 Feb 2005

A FIXED-POINT LOOK AT UPDATE LOGIC
Note the role of the initial \("d\) this is not one of the recursion variables.

\[(\#)\]
\[d^* \Phi \Leftrightarrow (d^* \Phi [d/\Lambda]) \land (\bar{d}^* \Phi)\]

\[(\#)\]
\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]

\[(\#)\]
Proof of Lemma 1
These meanings obviously describe a simultaneous fixed-point. That they are also the smallest follows by a simple induction, showing that the usual meanings are contained in any solution for the simultaneous fixed-point equation.

The following Lemma tells us something interesting about the \( \pi \)-calculus.

Simultaneous equations of a special shape can be solved one by one. Moreover, the disjunctive special form of these equations guarantees solutions inside dynamic logic.

Lemma 2
Solving Simultaneous Equations
Any system of simultaneous fixed-point equations of the shape (\( ^* \)) with its special forms as above has an explicit solution for each \( q \) in \( E-PDL \).

Proof of Lemma 2
The inductive procedure producing explicit \( E-PDL \) solutions works line by line – much like Gaussian Elimination in a system of linear equations.

Case 1
There is only one \( q \)-variable, as with public announcements.

The line reads \( q \) \( p \) \( \varphi \). The explicit solution, in the right format:

\[ q \varphi = \varphi \]

As required, no variable of the original sort occurs in the program.

Case 2
There are \( n \) lines in the recursion schema, with \( n > 1 \).

We first solve for the variable \( q_1 \) as in Case 1 – obtaining an explicit \( E-PDL \) formula \( \varphi_1 \) which may still contain the other recursion variables. We then substitute this solution in the remaining \( n-1 \) equations, and solve these inductively.

Finally, the solutions thus obtained for the \( \varphi \)'s \( \varphi_2, \ldots, \varphi_n \)'s are substituted in the simultaneous solution in the remaining equations, and solve these induction.

We first solve for the variable \( \varphi \)'s as in Case 1 – obtaining an explicit \( E-PDL \) formula.

This concludes the proofs of Theorems 1 and 2.


Illustration

Compute the solutions for the update model

PRE

\[ a = p \]

\[ b = T \]

Writing out the fixed point equation for, e.g., \(<A, a>\) in this specific case will really show how the above fixed-point procedure works.

Extensions to the Full Language of the \(\&\)-Calculus

The preceding suggests that we can have the Harmony Theorem just as well for the full language of the \(\&\)-calculus. The proof might even be simpler, by merely pushing the operators \(<A, a>\) through \(\&\)-operators in successive approximations for the fixed-point. One normally plugs in successive approximations for the fixed-point, but the problem is that the usual Substitution Lemma does not hold in a setting where models change. E.g., \(M, s, p := <A!>)\) need not be equivalent to \(M, s, \lceil <A!>) / p \rceil \) as the latter formula may have changed.

I solve this by defining a special substitution operator where \(\psi < d' \triangledown \triangledown > \) and after their model shift, which values of formulas \(<B>\) equivalent to \(\psi < d' \triangledown \triangledown >\) are the latter formula may have changed. Theorem 3

\(\pi_{PAL} = \pi_{EL} \)

Proof of Theorem 3

(Sketch. I should double-check–but the idea seems obvious.)

Extensions to the Full Language of the \(\&\)-Calculus

In this specific case, the fixed point equation for \(\psi < d' \triangledown \triangledown >\) works.

Writing the fixed point equation for \(\psi \)

Completing the solutions for the update model
Now I prove the following fact about approximations by induction:

If some calculation holds, then prove the following fact about approximations by induction:

Now I prove the following fact about approximations by induction:

Further Questions

3

A tangential interest for the $\mathcal{E}$-calculus.

Cf. my paper on Fixed-Point Logic for Studia Logica for the general motivation. The

A tangential interest for the $\mathcal{E}$-calculus.

Cf. my paper on Fixed-Point Logic for Studia Logica for the general motivation. The

A tangential interest for the $\mathcal{E}$-calculus.

Cf. my paper on Fixed-Point Logic for Studia Logica for the general motivation. The

A tangential interest for the $\mathcal{E}$-calculus.

Cf. my paper on Fixed-Point Logic for Studia Logica for the general motivation. The

A tangential interest for the $\mathcal{E}$-calculus.

Cf. my paper on Fixed-Point Logic for Studia Logica for the general motivation. The
We know it for E-PDL, and can now go in two directions.

Weaker languages: some version of EL-RC (see previous email)?

Stronger languages: the A-calculus itself? (Probably, with same tricks as above.)

Fragments in between that and E-PDL: what about the A-B-calculus?

3.2 Reduction axioms for other notions of group knowledge.

Common knowledge/belief are just one notion of group knowledge that is relevant to the epistemic actions that we consider. Test case for the reduction axiom approach: does it work for other notions, such as implicit group knowledge? This refers to the intersection of all agent accessibility relations: $\bigcap_{i \in I} R_i$.

I think it may. E.g., here is a valid reduction law for public announcement:

$$\phi \iff \phi \land \Box_{\forall}$$

3.3 Other deductive formats

Reduction axioms work with a typical atomic case which is not valid schematically.

But schematic validities are also attractive, including some that would not be so immediate. Reduction axioms work with a typical atomic case which is not valid schematically:

$$\phi \iff \forall \varphi.D \Rightarrow \Box_{\forall} \varphi$$

I think it may. E.g., here is a valid reduction law for public announcement:

The intersection of all agent accessibility relations: (No longer bisimulation-invariant.)

Does it work for other notions, such as implicit group knowledge? This refers to the intersection of all agent accessibility relations: $\bigcap_{i \in I} R_i$.

I think it may. E.g., here is a valid reduction law for public announcement:

We now know for E-PDL and can now go in two directions.