

# On how to write Structural Induction proofs dSem (Q1,'05)

Jacob Andersen  
University of Aarhus

October 7th, 2005

## Abstract

The purpose of the note is to clarify how formal written proofs using structural induction are constructed and thus assisting the reader in constructing such proofs for some semantic property using structural induction. This note is based on an exercise used in the course (week 4, exercise 2), where a semantics for boolean expressions should be shown to be deterministic.

## 1 Introduction

This note was written in response to an observation at the week-4 exercises, namely that many of the dSem students are uncertain about how to construct formal proofs – particularly on how to formally prove a property by structural induction. This note explains in great detail, how the solution to the exercise in question is obtained. During the course of this proof I try to cover a lot of problems and questions that have come up both at my exercise class, through emails and from talking to the other TAs.

In the course several proofs using structural induction have been presented at the lectures and exercise classes. However, all of the proofs have been presented orally (albeit using PowerPoint shows). In the mini project and the exam you will have to present a written proof using structural induction. When a proof is presented orally, you can under-specify by resorting to informal explanations often accompanied by pointing and waving your hands (letting the audience ask questions if they are not convinced by the arguments). All these points must be put down on paper when making a written proof, hence these proofs need to be more precise.

Although I will be going into great details in the proof below, this should *not* be considered a comprehensive tutorial on proof techniques. To solve the *particular* exercise I will be using a few common proof techniques which I will explain to some degree when I use them. When you construct your solutions to the mini project and exam problems, you might also need techniques beyond what is covered here. Therefore you may also want to familiarise yourself with these proof principles and techniques. Any introductory book on calculus or logic should cover these techniques.

## 2 SOS for boolean expressions

We will look at simple boolean expressions using the following grammar:

$$b ::= \mathbf{False} \mid \mathbf{True} \mid b \text{ or } b \mid \sim b \quad (1)$$

Note that we are using “or” and “ $\sim$ ” as syntactic logical “or” and “not”. We will be reserving the  $\vee$  and  $\neg$  symbols for the semantic “or” and “not”. Note also that I am using my own variant of boolean expressions – slightly different from the L language used in the course (for instance, to simplify things I have not included the  $E=E'$  construct).

We are going to define an SOS for the evaluation of these boolean expressions. Configurations and terminal configurations are chosen as:

$$\begin{aligned} \Gamma &= \{ \langle b, \rho \rangle \mid b \in \text{BExp}, \rho \in \text{Env} \} \\ \mathbf{T} &= \{ \langle \mathbf{True}, \rho \rangle \mid \rho \in \text{Env} \} \cup \{ \langle \mathbf{False}, \rho \rangle \mid \rho \in \text{Env} \} \end{aligned}$$

Where BExp is the set of boolean expressions in the above defined grammar, Env is a set of runtime environments. Since we do not need environments – or stores – in this exercise (we have no variables), we could (and probably should) have left the environments out of the configurations. However, since environments or stores are generally used in most exercises in the course (and maybe in an exam questions), they are included here for completeness. We will not define environments here, though, but will just regard Env as another mathematical set, which we don’t care about.

A transition relation will be written as:

$$\rho \vdash b \rightarrow b' = \langle b, \rho \rangle \rightarrow \langle b', \rho \rangle$$

Note that using this notation explicitly states that there are no side-effects, since the environment can never change during a transition.

The set of small-step operational semantics transitions,  $\vdash$  ( $\subseteq \Gamma \times \Gamma$ ), is defined recursively as:

$$\begin{aligned} \text{[or 1]} & \frac{\rho \vdash b_1 \rightarrow b'_1}{\rho \vdash b_1 \text{ or } b_2 \rightarrow b'_1 \text{ or } b_2} \\ \text{[or 2]} & \rho \vdash \mathbf{True} \text{ or } b \rightarrow \mathbf{True} \\ \text{[or 3]} & \rho \vdash \mathbf{False} \text{ or } b \rightarrow b \\ \text{[not 1]} & \frac{\rho \vdash b \rightarrow b'}{\rho \vdash \sim b \rightarrow b'} \\ \text{[not 2]} & \rho \vdash \sim \mathbf{True} \rightarrow \mathbf{False} \\ \text{[not 3]} & \rho \vdash \sim \mathbf{False} \rightarrow \mathbf{True} \end{aligned} \quad (2)$$

## 3 Determinism of a small-step SOS

We want prove that the SOS is “deterministic”, but what exactly does that mean? Well, it means that the SOS has the property that no matter which configuration you pick there will be *at most one* transition, that leads (from this configuration) to another configuration. Notice that we do *not* say that there should be “exactly” or “at least one” transition from any configuration!

In fact, this would be a very bad property (if you don't see why immediately, this is a good time to take a break to think about it). We have seen several examples of non-deterministic constructs in this course; remember the **flip** and **maybe** constructs from the exercises in week 3?

So how do we formulate determinism in the language of mathematics? Remember that we are dealing with sets. We have the set,  $\Gamma$ , of configurations and the set,  $\vdash \subseteq \Gamma \times \Gamma$ , of transitions. So when we say that there is at most one transition *from* each configuration, it means that when you pick an element,  $\gamma \in \Gamma$ , you will find at most one element (pair) in  $\vdash$  with  $\gamma$  as the first part of the pair. Another way to say this is: Whenever you pick two elements from  $\vdash$  both with the same configuration,  $\gamma$ , in the first position, the second position configurations will be identical, or with symbols:

$$\forall \gamma, \gamma', \gamma'' \in \Gamma : ((\gamma, \gamma') \in \vdash \wedge (\gamma, \gamma'') \in \vdash) \Rightarrow \gamma' = \gamma''$$

Since the transition relation is explicitly side-effect free, by pushing and substituting symbols, this claim can also be written as:

$$\forall \rho \in \text{Env} : \forall b, b', b'' \in \text{BExp} : ((\rho \vdash b \rightarrow b') \wedge (\rho \vdash b \rightarrow b'')) \Rightarrow b' = b'' \quad (3)$$

Remember that the notation  $\rho \vdash b \rightarrow b'$  is a “shorthand” for  $(\langle b, \rho \rangle, \langle b', \rho \rangle) \in \vdash$ .

## 4 Preparing for a SI proof

We now have a property, determinism, defined in (3) and we hope to be able to prove that this property holds for the set,  $\vdash$ , of transitions defined in (2). Since the set of transitions is defined recursively on the *structure* of the language, the natural way to construct a proof of such a property is to use structural induction. We can trivially rewrite (3) as:

$$\forall b \in \text{BExp} : P(b) \quad (4)$$

Where:

$$P(b) = \forall \rho \in \text{Env} : \forall b', b'' \in \text{BExp} : ((\rho \vdash b \rightarrow b') \wedge (\rho \vdash b \rightarrow b'')) \Rightarrow b' = b'' \quad (5)$$

So we need to show  $P(b)$  for all possible  $bs$ . But all the  $bs$  are defined recursively in (1). So if we can prove that the property holds for each of the expression constructs in (1) *if* the property holds for any subexpressions, then we are done. So we want to prove the following claims:

$$P(\mathbf{False}) \quad (6)$$

$$P(\mathbf{True}) \quad (7)$$

$$P(b_1) \wedge P(b_2) \Rightarrow P(b_1 \text{ or } b_2) \quad (8)$$

$$P(b) \Rightarrow P(\sim b) \quad (9)$$

And if that succeeds we are finished, since  $((6) \wedge (7) \wedge (8) \wedge (9)) \Rightarrow (4) \Leftrightarrow (3)$ .

## 5 Proving $P(\mathbf{False})$ and $P(\mathbf{True})$

We will just show  $P(\mathbf{False})$  since the proof of  $P(\mathbf{True})$  is similar.

So we want to prove the claim:

$$\forall \rho, b', b'' : ((\rho \vdash \mathbf{False} \rightarrow b') \wedge (\rho \vdash \mathbf{False} \rightarrow b'')) \Rightarrow b' = b''$$

To prove this we consult our set of transitions in (2), and realise that no transitions exists. Remember that  $\rho \vdash \mathbf{False} \rightarrow b'$  is a “shorthand” for  $(\langle \mathbf{False}, \rho \rangle, \langle b', \rho \rangle) \in \vdash$  and this expression is obviously false for any choice of  $\rho$  and  $b'$ . Therefore the left-hand side of the implication is false, which means that the implication is trivially true (“holds vacuously”).

Some have expressed discomfort with this argument. To see why it is OK, remind yourself what we are proving here: Determinism. We want to show that whatever happens, only that can happen. In this case nothing ever happens, because no transitions exist from a **False** configuration, and this “lack of action” is of course also deterministic.

## 6 Proving $P(b_1) \wedge P(b_2) \Rightarrow P(b_1 \text{ or } b_2)$

We want to show that an implication is valid for any choice of  $b_1$  and  $b_2$ . To show that some predicate holds for all elements in a set, we imagine a game with an “opponent” (who is allowed to choose any element from the set) choosing one for us. So “let  $b_1$  and  $b_2$  be given” (by the opponent). There are two possibilities: Either the left hand side of the implication is false or it is true. If the left hand side is false then the implication is trivially true, therefore the only case we need to consider is the case where the left hand side is true. Therefore we assume *without loss of generality* (WLOG) that the left hand side is true, and we must show that the right hand side of the implication is true.

Under our assumption (the “induction hypothesis”) we need to show the following claim:

$$\forall \rho, b', b'' : ((\rho \vdash b_1 \text{ or } b_2 \rightarrow b') \wedge (\rho \vdash b_1 \text{ or } b_2 \rightarrow b'')) \Rightarrow b' = b''$$

So let  $\rho$ ,  $b'$  and  $b''$  be given (still by the opponent). We now need to show the implication. Again WLOG we assume the left hand side, i.e.  $b'$  and  $b''$  have been chosen in such a way that  $(\rho \vdash b_1 \text{ or } b_2 \rightarrow b') \wedge (\rho \vdash b_1 \text{ or } b_2 \rightarrow b'')$  and we want to prove the right hand side,  $b' = b''$ .

So we know that the left hand side of the implication is true. In other words, we know that there is an element,  $\xi_1$ , in the set  $\vdash$  such that  $\rho \vdash b_1 \text{ or } b_2 \rightarrow b'$ . We also know that there is an element,  $\xi_2$ , (possibly the same, we don't know that) such that  $\rho \vdash b_1 \text{ or } b_2 \rightarrow b''$ . Since  $\xi_1, \xi_2 \in \vdash$ , they are generated by the set definition in (2). If we take a look at these transition rules, we realise that only three of the rules could have generated the elements  $\xi_1$  and  $\xi_2$  in the set  $\vdash$  – namely the three first rules ([or1], [or2] and [or3]).

Could one of these rules have generated  $\xi_1$  while another rule would generate  $\xi_2$ ? To answer that question we need to examine the three rules. Remember that  $b_1$  is fixed (by our opponent). Could rule [or 2] and [or 3] be used at the same time? The answer is obviously “no” since our fixed  $b_1$  could not be both **True** and **False** at the same time. Could [or 1] and [or 2] be used at the same

time? Again the answer is “no”, since there is no  $\rho \vdash \mathbf{True} \rightarrow \dots$  rules. The same argument goes for the final combination [or 1] and [or 3]. Note that the argument in this paragraph is an essential argument, which will (usually) fail if any non-deterministic construct is introduced in the language – e.g. the `flip` construct, which we have seen earlier in the course.

We only have three cases left to examine now:  $\xi_1$  and  $\xi_2$  could both be generated by one of the three “or” rules. For each of these cases, we need to check that  $b' = b''$ . Remember that  $\rho$ ,  $b_1$  and  $b_2$  are all fixed.

We look at the first rule (“or 1”):

$$\frac{\rho \vdash b_1 \rightarrow b'_1}{\rho \vdash b_1 \mathbf{or} b_2 \rightarrow b'_1 \mathbf{or} b_2}$$

$\rho$ ,  $b_1$  and  $b_2$  are fixed and applying the induction hypothesis on  $b_1$  (remember, this is one of our assumptions)  $P(b_1)$  gives us a unique  $b'_1$ . Since both  $\xi_1$  and  $\xi_2$  was generated from this rule, we get:

$$b' = b'_1 \mathbf{or} b_2 = b''$$

We now take a look at the second rule:

$$\rho \vdash \mathbf{True} \mathbf{or} b \rightarrow \mathbf{True}$$

Again  $b$  (formerly known as  $b_2$ ) is fixed. Since both  $\xi_1$  and  $\xi_2$  are generated from this rule, we get:

$$b' = \mathbf{True} = b''$$

The third rule is:

$$\rho \vdash \mathbf{False} \mathbf{or} b \rightarrow b$$

And the reader should easily verify that again  $b' = b''$ .

## 7 Proving $P(b) \Rightarrow P(\sim b)$

This proof is left as an exercise to the reader. The proof of this claim and the proof above are almost identical. This concludes the proof.  $\square$

## 8 Proof using less text and more symbols

Some people are under the impression that a proof is more “mathematical” if more symbols and less “human” text is used. In honour of those of you, who have this idea, I present the proof from section 6 once again. This time in the language of pure mathematical symbols. Those of you who prefer human readable proofs can skip this section.

We need the following fact:

$$\begin{aligned} & \forall b_1, b_2 \in \text{BExp} : \forall \gamma \in \Gamma : \forall \rho \in \text{Env} : \\ & (\langle b_1 \mathbf{or} b_2, \rho \rangle, \gamma) \in \vdash \Rightarrow \quad (10) \\ & (\gamma = \langle \mathbf{True}, \rho \rangle \wedge b_1 = \mathbf{True}) \vee (\gamma = \langle b_2, \rho \rangle \wedge b_1 = \mathbf{False}) \vee \\ & (\exists b'_1 \in \text{BExp} : \gamma = \langle b'_1 \mathbf{or} b_2, \rho \rangle \wedge b_1 \notin \{\mathbf{True}, \mathbf{False}\} \wedge (\langle b_1, \rho \rangle, \langle b'_1, \rho \rangle) \in \vdash) \end{aligned}$$

Note that this fact is not easily deduced from (2) even though at a first glance it seems obvious. Why? In (2) we define the set of transitions,  $\vdash$ , by recursively stating which elements are *in* the set on the basis of elements already in the set. Now we need a mechanism to prove that some element is *not in* the set, which is far from trivial.<sup>1</sup> To cut a corner we just take (10) as a given fact since proving it would be far beyond the scope of this note (that topic would belong in an advanced math course).

We want to prove  $P(b_1) \wedge P(b_2) \Rightarrow P(b_1 \text{ or } b_2)$ , and we remind ourselves that

$$P(b_1 \text{ or } b_2) = (\forall \rho \in \text{Env} : \forall \gamma', \gamma'' \in \Gamma : ((\langle b_1 \text{ or } b_2, \rho \rangle, \gamma') \in \vdash \wedge (\langle b_1 \text{ or } b_2, \rho \rangle, \gamma'') \in \vdash) \Rightarrow \gamma' = \gamma'')$$

So let  $\gamma, \gamma', \gamma'' \in \Gamma$  be given such that  $\gamma = \langle b_1 \text{ or } b_2, \rho \rangle$ . We assume (10),  $P(b_1) \wedge P(b_2)$  and the left hand side of the implication we want to prove. We shall show that  $\gamma' = \gamma''$ :

$$\begin{aligned} & [\forall \tilde{b}_1, \tilde{b}_2 \in \text{BExp} : \forall \tilde{\gamma} \in \Gamma : \forall \tilde{\rho} \in \text{Env} : (\langle \tilde{b}_1 \text{ or } \tilde{b}_2, \tilde{\rho} \rangle, \tilde{\gamma}) \in \vdash \Rightarrow \\ & \left( (\tilde{\gamma} = \langle \mathbf{True}, \tilde{\rho} \rangle \wedge \tilde{b}_1 = \mathbf{True}) \vee (\tilde{\gamma} = \langle \tilde{b}_2, \tilde{\rho} \rangle \wedge \tilde{b}_1 = \mathbf{False}) \vee \right. \\ & \left. (\exists \tilde{b}'_1 \in \text{BExp} : \tilde{\gamma} = \langle \tilde{b}'_1 \text{ or } \tilde{b}_2, \tilde{\rho} \rangle \wedge \tilde{b}_1 \notin \{\mathbf{True}, \mathbf{False}\} \wedge (\langle \tilde{b}_1, \tilde{\rho} \rangle, \langle \tilde{b}'_1, \tilde{\rho} \rangle) \in \vdash) \right)] \\ & \wedge P(b_1) \wedge P(b_2) \wedge [(\langle b_1 \text{ or } b_2, \rho \rangle, \gamma') \in \vdash \wedge (\langle b_1 \text{ or } b_2, \rho \rangle, \gamma'') \in \vdash] \\ & \Downarrow \\ & [\forall \tilde{b}_1, \tilde{b}_2 \in \text{BExp} : \forall \tilde{\gamma} \in \Gamma : \forall \tilde{\rho} \in \text{Env} : (\langle \tilde{b}_1 \text{ or } \tilde{b}_2, \tilde{\rho} \rangle, \tilde{\gamma}) \in \vdash \Rightarrow \\ & \left( (\tilde{\gamma} = \langle \mathbf{True}, \tilde{\rho} \rangle \wedge \tilde{b}_1 = \mathbf{True}) \vee (\tilde{\gamma} = \langle \tilde{b}_2, \tilde{\rho} \rangle \wedge \tilde{b}_1 = \mathbf{False}) \vee \right. \\ & \left. (\exists \tilde{b}'_1 \in \text{BExp} : \tilde{\gamma} = \langle \tilde{b}'_1 \text{ or } \tilde{b}_2, \tilde{\rho} \rangle \wedge \tilde{b}_1 \notin \{\mathbf{True}, \mathbf{False}\} \wedge (\langle \tilde{b}_1, \tilde{\rho} \rangle, \langle \tilde{b}'_1, \tilde{\rho} \rangle) \in \vdash) \right)] \\ & \wedge [(\langle b_1 \text{ or } b_2, \rho \rangle, \gamma') \in \vdash \wedge (\langle b_1 \text{ or } b_2, \rho \rangle, \gamma'') \in \vdash] \wedge \\ & [\forall \tilde{b}'_1, \tilde{b}''_1 \in \text{BExp} : ((\langle b_1, \rho \rangle, \langle \tilde{b}'_1, \rho \rangle) \in \vdash \wedge (\langle b_1, \rho \rangle, \langle \tilde{b}''_1, \rho \rangle) \in \vdash) \\ & \Rightarrow \tilde{b}'_1 = \tilde{b}''_1] \\ & \Downarrow \end{aligned}$$

---

<sup>1</sup>The natural numbers,  $\mathbb{N}$  is defined as the smallest set (least fix-point) such that (i)  $1 \in \mathbb{N}$  and (ii)  $n \in \mathbb{N} \Rightarrow n + 1 \in \mathbb{N}$ . So how do you prove that  $\pi$  or  $-42$  does not belong to  $\mathbb{N}$ ? Both of the numbers are real, and the set  $\mathbb{R}$  also satisfies (i) and (ii) (but is not the least fix-point).



$$\begin{aligned}
&\Downarrow \\
&[(b_1 = \mathbf{True} \wedge \gamma' = \langle \mathbf{True}, \rho \rangle \wedge \gamma'' = \langle \mathbf{True}, \rho \rangle) \\
&\vee (b_1 = \mathbf{False} \wedge \gamma' = \langle b_2, \rho \rangle \wedge \gamma'' = \langle b_2, \rho \rangle) \\
&\vee (b_1 \notin \{\mathbf{True}, \mathbf{False}\} \wedge \exists b'_1, b''_1 \in \text{BExp} : \\
&\quad (\gamma' = \langle b'_1 \text{ or } b_2, \rho \rangle \wedge (\langle b_1, \rho \rangle, \langle b'_1, \rho \rangle) \in \vdash \\
&\quad \wedge \gamma'' = \langle b'_1 \text{ or } b_2, \rho \rangle \wedge (\langle b_1, \rho \rangle, \langle b''_1, \rho \rangle) \in \vdash)] \\
&\wedge [\forall \tilde{b}'_1, \tilde{b}''_1 \in \text{BExp} : ((\langle b_1, \rho \rangle, \langle \tilde{b}'_1, \rho \rangle) \in \vdash \wedge (\langle b_1, \rho \rangle, \langle \tilde{b}''_1, \rho \rangle) \in \vdash) \\
&\quad \Rightarrow \tilde{b}'_1 = \tilde{b}''_1] \\
&\Downarrow \\
&(b_1 = \mathbf{True} \wedge \gamma' = \langle \mathbf{True}, \rho \rangle = \gamma'') \\
&\vee (b_1 = \mathbf{False} \wedge \gamma' = \langle b_2, \rho \rangle = \gamma'') \\
&\vee (b_1 \notin \{\mathbf{True}, \mathbf{False}\} \wedge \exists b'_1, b''_1 \in \text{BExp} : \\
&\quad (\gamma' = \langle b'_1 \text{ or } b_2, \rho \rangle \wedge \gamma'' = \langle b''_1 \text{ or } b_2, \rho \rangle \wedge b'_1 = b''_1)) \\
&\Downarrow \\
&\gamma' = \gamma''
\end{aligned}$$

□

I hope that those of you who are still with me so far, realise that this kind of proofs are “write-only” (maybe unless the reader is autistic). However this kind of proofs *can* be relevant if you want a theorem verifier to verify the correctness of your proof.

Making a proof is a matter of conveying ideas from the writer to the reader in a way which convinces the reader of its correctness. If the reader is a human being this is best done in a “humane” way – like it is shown in sections 4 to 7. If you study the proof in this section carefully and compare it to the former proof, you will see that they are identical. One is just as formal as the other and they convey the same ideas but the former proof does a much better job of “convincing the reader” than the latter (provided the reader is a human). It is no secret that we prefer the former kind of proofs in the mini project and exam solutions.

## 9 Thoughts about written proofs and exams

The proof above is presented in a very detailed and pedantic way. Usually a lot of details are omitted and the reader is expected to “fill out the blanks” in the back of his/her head while reading the proof. The question about how detailed a proof should be often pops up – especially when a written exam is approaching.

A good answer to this question was given by associate professor Klaus Thomsen (IMF) to one of my classmates when I took the calculus introductory course (year 2000). The basic ideas in his answer (subject to my interpretation and passing time) were as follows:

You have to convince the reader about the validity of the proof but since the exam is also testing your ability to construct a proof, you also have to convince the reader that you are capable of doing so. This means that as a starting point

you will have to show every detail – in contrast to textbook writers who will typically skip all the uninteresting details. If you show the reader that you *can* do that, you can skip the details when they are trivial – e.g. identical to an earlier part of the proof. In other words it is a matter of building up trust. You will want the reader to trust that you *can* do the details but that you have thought the proof through (on a scratch paper) and come to the conclusion that it is so trivial, that it is not necessary to show the details.

However, if you don't show the details because you are lazy or because you can't figure it out, the reader will recognise it immediately – it is not hard to tell the difference, because the intelligent student will select the interesting cases and show them (ignoring the trivial or identical – like I did with the **True** case in section 5), while the lazy student will select either the easy cases or a few random cases (often ignoring the interesting cases because he never went there) and also avoid the hairy details. If you are still in doubt after reading this, then always prove all details – then you are sure that you never lose any points.

## 10 An extra exercise

As an extra exercise for the reader I propose the following problem:

Redefine the “or” semantics to be a parallel evaluation (cf. pp. 30-31 in Plotkin) and re-do the proof of determinism. *Hint: The strict notion of “determinism” used above cannot be used in parallel evaluation, so you will need to re-think the problem from the beginning and define a suitable notion of “determinism” for this problem.*