Structural Induction

an institution-independent methodology

Răzvan Diaconescu

Simion Stoilow Institute of Mathematics of the Romanian Academy

3rd RO-JP AlgSpec Workshop, Sinaia 2012

Intention

To develop a method for proving induction properties that does not depend upon a particular logical system.

The method should get a result in most of the situations.

It should have firm, clear and simple mathematical foundations.

It should emerge directly and rigidly from the foundations thus leading to a high degree of automation.

Outline

R. Diaconescu.
Structural induction in institutions.
Information and Computation, 209(9):1197–1222, 2011.

Inductive property ρ

In the non-structured case, given specification (Σ, E),

$$\text{(initial model) } 0_{\Sigma,E} \models \rho .$$

N.B.: $E \models \rho$ implies $0_{\Sigma,E} \models \rho$ but the other way around *not* true!

N.B.: This concept independent upon the underlying logic, may be formulated at a very abstract level.

N.B.: This is a model theoretic concept/approach.

(Counter-)example

```
mod! NAT-MAX
  [ NNat ]
  op 0 : -> NNat
  op s_ : NNat -> NNat
  op max : NNat NNat -> NNat
  vars X Y : NNat
  eq max(0,X) = X .
  eq max(X,0) = X .
  eq max(s X, s Y) = s max(X,Y) .
```

$$0_{\Sigma,E} \models (\forall x,y) max(x, max(x,y)) = max(x,y)$$

$$E \not\models (\forall x,y) max(x, max(x,y)) = max(x,y)$$

Structural induction

Ordinary properties difficult to prove,
inductive properties even much more difficult!

Even in logics enjoying complete proof systems,
induction does not admit a complete proof system.

Structural induction as *sufficient* methodology for proving inductive properties.

Actually, (only) for *universal quantifier elimination* (in the inductive properties)!

Bridge to structural induction

$$0_{\Sigma,E} \models (\forall X)\rho \quad \text{if} \quad E \models \theta(\rho) \ \text{ for all 'substitutions' } \ \theta : X \to 0_\Sigma$$

The actual concept of 'substitution' is of course dependent upon the underlying logical system; however possibility to treat it abstractly.

The problem here is that in general this represents an *infinite* set of proof tasks...

Technical prerequisites

1. pushouts of signatures
2. model amalgamation (also for homomorphisms)
3. axiomatic treatment of substitutions
   - ‘depth’ of substitutions
   - ‘atomic’ substitutions
   - etc.

The method

1. Fix the block X of the variables for induction;
   - induction in 'parallel' over the 'variables' in X;
   - the choice of X is a human decision that determines the whole proof process;
2. Consider all 'atomic' substitutions $Q : \ X \to Z$; concretely $Q : \ (x \in X) \mapsto \sigma(\bar{z}_x)$, with σ operation symbol and $\bar{z}_x$ new variables.
3. For each Q prove

$$E \cup \{\psi(\rho) \mid \psi \sqsubset Q\} \ \models_{\Sigma+Z} \ Q(\rho)$$

Finiteness

The finiteness of the structural induction method may be assured as follows:

1. The number of Q is finite when X and the signature are finite.
2. When $\{\psi \mid \psi \sqsubset Q\}$ is finite; at this moment $\sqsubset$ is merely an axiomatization device which is rather uniformly defined in the concrete situations, however in principle it is a parameter of the method.
   - $\sqsubset$ too small means fewer hypotheses hence proof more difficult,
   - $\sqsubset$ too big may endanger the finiteness.

Constructors

Just a *methodological* device for improving the efficiency of the proof process.

For the mappings Q we may replace the original signature by a smaller 'sub-signature of constructors'.
Consequently fewer cases for Q, less complex proof process (sometimes much less!).

$\iota : \Omega \to \Sigma$ '*sub-signature*' *of constructors* for (Σ, E) when

$$0_\Omega \to \textrm{M}\small{\textrm{OD}}(\iota)(0_{\Sigma,E})$$

is 'surjective'.

Constructors

N.B.: This definition is institution-independent via abstract concepts of ‘surjection’.

In concrete situations equivalent proof theoretic definitions prone to formal verification:

For each non-constructor σ and each $\bar{t}$ built only from constructors there exists t' only from constructors such that

$$E \models \sigma(\bar{t}) = t'.$$

Step 0: constructors

```
mod! NAT-MAX
  [ NNat ]
  op 0 : -> NNat
  op s_ : NNat -> NNat
  op max : NNat NNat -> NNat
  vars X Y : NNat
  eq max(0,X) = X .
  eq max(X,0) = X .
  eq max(s X, s Y) = s max(X,Y) .
```

Then $\{0, \texttt{s\_}\}$ is a sub-signature of constructors for `NAT-MAX`.

This gets a(n easy) formal proof.

Step 1: fixing the variables for induction

$(\forall x, y)\texttt{max}(x, \texttt{max}(x, y)) = \texttt{max}(x, y)$

- $X = \{x, y\}$,
- ρ is $\texttt{max}(x, \texttt{max}(x, y)) = \texttt{max}(x, y)$.

Other choices, i.e. $X = \{x\}$ or $X = \{y\}$ may not work.

Step 2: generating the cases

	Q_x	Q_y
1.	$\texttt{0}$	$\texttt{0}$
2.	$\texttt{0}$	$\texttt{s}(zy)$
3.	$\texttt{s}(zx)$	$\texttt{0}$
4.	$\texttt{s}(zx)$	$\texttt{s}(zy)$

Without constructors we would have 9 (=3^2) instead of 4 (=2^2) cases!

Step 3: proof task 1

```
-- cazul Q_x = 0 si Q_y = 0
open NAT-MAX .
red max(0,max(0,0)) == max(0,0) .
close
```

Step 3: proof task 2

```
-- cazul Q_x = 0 si Q_y = s
open NAT-MAX .
op zy : -> NNat .
red max(0,max(0,s zy)) == max(0,s zy) .
close
```

Step 3: proof task 3

```
-- cazul Q_x = s si Q_y = 0
open NAT-MAX .
op zx : -> NNat .
eq max(X,X) = X .
red max(s zx, max(s zx,0)) == max(s zx, 0) .
close
```

N.B.: This case requires a lemma that is discovered easily from the reduction.

Step 3: proof task 4

```
-- cazul Q_x = s si Q_y = s
open NAT-MAX .
ops zx zy : -> NNat .
eq max(zx, max(zx,zy)) = max(zx,zy) .
red max(s zx, max(s zx,s zy)) == max(s zx,s zy) .
close
```

N.B.: This is the only case in which the premise $\{\psi(\rho) \mid \psi \sqsubset Q\}$ is non-empty.

Conclusions

- Institution-independent methodology for structural induction.
- Directly and rigidly based upon foundations.
- High potential for automation.
- Constructors as pure methodological device, with no reflection in the semantics; consequently
    - semantics kept simple and natural;
    - clear roles for the specification and verification levels.

Future Research

1. Structural induction for structured specifications.
2. Play with $\sqsubset$.
3. Why it (almost?) always works?
4. Develop concrete methodologies for various logics.
5. Given the high automation potential, develop proof assistant (on top of CafeOBJ?).