A Coinductive Formulation of the "Coinduction Theorem" by Michael and Appel

Andreas Abel*

June 26, 2000

1 Inductive Treatment of Safety

In [MA00] Neophytos G. Michael and Andrew W. Appel introduce machine *safety* for a transition relation $\mapsto$ on the set of machine states S as follows:

$$\mathsf{Safe}(s) :\Longleftrightarrow \forall s'.\ s \mapsto^* s' \Rightarrow \exists s''.\ s' \mapsto s''$$

This expresses that a state is safe if after a finite number of transitions starting from this state there is still one more possible transition. Given an invariant $\mathsf{Inv} \subseteq S$, written $\mathsf{Inv}(s)$ for $s \in \mathsf{Inv}$, they define the properties *progress* and *preservation* by:

$$\begin{array}{rcl} \mathsf{Prog}(\mathsf{Inv}) & :\Longleftrightarrow & \forall s.\ \mathsf{Inv}(s) \Rightarrow \exists s'.\ s \mapsto s' \\ \mathsf{Pres}(\mathsf{Inv}) & :\Longleftrightarrow & \forall s, s'.\ \mathsf{Inv}(s) \wedge s \mapsto s' \Rightarrow \mathsf{Inv}(s') \end{array}$$

Prog states that an invariant enables progress if it enables a further transition step. Pres expresses that the invariant is preserved by the step relation. It is clear that a progressive and preserved invariant that holds for the initial state guarantees safety, which they state in the "coinduction theorem":

$$\frac{\mathsf{Inv}(s_0) \qquad \mathsf{Prog}(\mathsf{Inv}) \qquad \mathsf{Pres}(\mathsf{Inv})}{\mathsf{Safe}(s_0)}$$

Assuming progression and preservation of Inv globally, the theorem can be reformulated as

$$\frac{\mathsf{Inv}(s_0) \qquad s_0 \mapsto^* s'}{\exists s''. s' \mapsto s''}$$

This is easily proven by *induction* on $s_0 \mapsto^* s'$:

- Case $s_0 = s'$: Immediately by $\mathsf{Prog}(\mathsf{Inv})$.

*abel@cs.cmu.edu, Carnegie Mellon University, Pittsburgh, PA, USA.

- Case $s_0 \mapsto s \mapsto^* s'$: Preservation yields $\mathsf{Inv}(s)$, on which we apply the induction hypothesis for $s \mapsto^* s'$. $\square$

In the following we show how to define Safe as a coinductive set and how to prove the "coinduction theorem" by coinduction. But first we introduce the concepts of coinduction in duality to these of induction.

2 Induction and Coinduction

Given a complete lattice of sets $(\mathcal{P}(S), \subseteq)$ and a monotone operator $\mathcal{F} : \mathcal{P}(S) \to \mathcal{P}(S)$ we can use the theorem of Knaster and Tarski to construct the least fixpoint μ of $\mathcal{F}$. It is characterized as the least pre-fixpoint of $\mathcal{F}$, which we write down as follows:

$$\begin{array}{rl} \text{(intro)} & \mathcal{F}(\mu) \subseteq \mu \\ \text{(elim)} & \forall P \subseteq S.\ \mathcal{F}(P) \subseteq P \Rightarrow \mu \subseteq P \end{array}$$

This characterization is equivalent to the explicit definition of the least fixpoint via

$$\mu := \bigcap \{P \mid \mathcal{F}(P) \subseteq P\}$$

The first condition stipulates that μ is a pre-fixpoint of $\mathcal{F}$, and the second minimality. The names are not chosen by accident, since these two properties actually show how to introduce and eliminate members of the inductive set μ. This can be seen more clearly in rule notation.

$$\text{(intro)}\ \frac{s \in \mathcal{F}(\mu)}{s \in \mu} \qquad \text{(elim)}\ \frac{s \in \mu \qquad \forall s' \in \mathcal{F}(P).\ s' \in P}{s \in P}$$

A simple example for an inductive set are the natural numbers $\mathbb{N}$ which are generated by the operator $\mathcal{F}(Q) := \{0, 1+n \mid n \in Q\}$. The rule (elim) simply becomes the natural numbers induction scheme for predicate P.

Dually to inductive sets, a coinductive set ν is defined as the greatest fixpoint of a monotone operator ν. Post-fixpoint condition and maximality read

$$\begin{array}{rl} \text{(coIntro)} & \nu \subseteq \mathcal{F}(\nu) \\ \text{(coElim)} & \forall P \subseteq S.\ P \subseteq \mathcal{F}(P) \Rightarrow P \subseteq \nu \end{array}$$

This equals the definition

$$\nu := \bigcup \{P \mid P \subseteq \mathcal{F}(P)\}$$

The condition (coIntro) is dual to (intro), thus it is actually an elimination rule. Likewise (coElim) is an introduction rule.

$$\text{(coIntro)}\ \frac{s \in \nu}{s \in \mathcal{F}(\nu)} \qquad \text{(coElim)}\ \frac{s \in P \qquad \forall s' \in P.\ s' \in \mathcal{F}(P)}{s \in \nu}$$

As an example consider the set $\mathbb{N}^\omega$ of infinite streams of natural numbers. They are formed by the operator $\mathcal{F}(Q) := \{a \mid \mathsf{head}\, a \in \mathbb{N},\ \mathsf{tail}\, a \in Q\}$ and obey the following instances of (coIntro) and (coElim)

$$(\mathbb{N}^\omega\text{Elim}) \qquad \frac{a \in \mathbb{N}^\omega}{\mathsf{head}\, a \in \mathbb{N} \qquad \mathsf{tail}\, a \in \mathbb{N}^\omega}$$

$$(\mathbb{N}^\omega\text{Intro}) \qquad \frac{a \in P \qquad \forall a' \in P.\ \mathsf{head}\, a' \in \mathbb{N} \wedge \mathsf{tail}\, a' \in P}{a \in \mathbb{N}^\omega}$$

To show that the stream of zeros $0^\omega = (0, 0, \ldots)$ is a stream of natural numbers, we use ($\mathbb{N}^\omega$Intro) on the set $P = \{0^\omega\}$.[1]

Now back to our problem!

3 Coinductive Treatment of Safety

A state s is safe if from it a state s' can never be reached s.th. no further step is possible. We also could say that every transition sequence starting with s is infinite. Hence safety is a candidate for coinduction. We can define Safe by the rule

$$(\mathsf{SafeE}) \; \frac{\mathsf{Safe}(s)}{s \mapsto s' \qquad \mathsf{Safe}(s')}$$

This reads "from a safe state s there exists a transition to another safe state s'" which is a characterization of the safe states. Interpreting this rule in the coinductive manner, we obtain the operator $\mathcal{F}(Q) := \{s \in Q \mid \exists s' \in S.\ s \mapsto s' \wedge s' \in Q\}$ on sets of states $Q \subseteq S$. The co-elimination rule for Safe is

$$(\mathsf{SafeI}) \; \frac{P(s) \qquad \forall s'.\ P(s') \Rightarrow \exists s''.\ s' \mapsto s'' \wedge P(s'')}{\mathsf{Safe}(s)}$$

Assuming $\mathsf{Prog}(\mathsf{Inv})$ and $\mathsf{Pres}(\mathsf{Inv})$ globally as before, we can restate the safety theorem as

$$\frac{\mathsf{Inv}(s_0)}{\mathsf{Safe}(s_0)}$$

We prove it by coinduction on $\mathsf{Safe}(s_0)$, i.e., we apply the rule (SafeI). Therefore we need an instantiation for P with the properties:

1. $P(s_0)$
2. $P(s') \Rightarrow \exists s''.\ s' \mapsto s'' \wedge P(s'')$

We choose $P = \mathsf{Inv}$. The first condition is the assumption, the second follows immediately from $\mathsf{Prog}(\mathsf{Inv})$ and $\mathsf{Pres}(\mathsf{Inv})$. $\square$

[1] This may sound a little strange and one could as oneself "How do I actually get a stream?". The point is that inductive data is constructed (synthesized), whereas coinductive *codata* like streams etc. is *consumed* (analyzed).

4 Conclusion

We have demonstrated that the definition of safety can be simplified by means of coinduction. The original formulation "a state is safe if every transition sequence starting in it can be extended" has been changed into "a state is safe if the next state is still safe". Instead of defining the reflexive-transitive closure $\mapsto^*$ of the step relation as a means to define safety, we can define safety directly. The coinductive proof is very natural and makes clear that a progressive and preserved invariant implies safety immediately.

References

[JR97] B. Jacobs and J.J.M.M. Rutten. A tutorial on (co)algebras and (co)induction. In *Bulletin of EATCS*, volume 62, pages 222–259, 1997.

[MA00] Neophytos G. Michael and Andrew W. Appel. Machine Instruction Syntax and Semantics in Higher Order Logic. In David McAllester, editor, *Automated Deduction – CADE-17*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 7–24. Springer, June 2000.