Another Glance at the Alpern-Schneider Characterization of Safety and Liveness in Concurrent Executions

H.Peter Gumm

Abstract

In order to derive a result such as the Alpern-Schneider theorem characterizing safety and liveness properties of concurrent program executions, it is shown that all that is needed is a $\vee$-preserving map φ between complete Boolean algebras. Every property becomes a conjunction of a safety and a liveness property and safety properties can be characterized by sets of configurations that are to be "avoided".

Aside from the original result of B. Alpern and F.B. Schneider we also provide a new application by considering transition systems with a UNITY-style logic. Safety properties are characterized by a set of forbidden pairs of successive states and progress properties are those allowing all possible state-successor pairs. Every property of a transition system is shown to be a conjunction of a safety and a progress property.

Keywords : Safety, Liveness, Progress, Concurrency, Semantics, UNITY.

1 Introduction

It is well known [4] that temporal properties of concurrent programs in general display two aspects: A *safety* aspect expressing that "nothing will go wrong" and a *liveness* aspect asserting that "something desired will eventually happen". This imprecise observation was first made precise in the work of B. Alpern and F.B. Schneider [1] where convincing formal definitions of *liveness* and *safety* were given and it was shown that every property is a conjunction of a safety and a liveness property.

M. Rem in [5] shed new light on the above mentioned theorem, mainly by defining an equivalence relation on properties so that safety properties are in 1-1 correspondence with the equivalence classes and progress properties are precisely the members of the equivalence class to which *True* belongs. The topological closure operator (the closure of a is written $\overline{a}$) used in [1] was given a concrete definition, so safety properties could be understood as those properties P with $\overline{P} = P$.

In this article we take up the the theme again, initially placing it in a more abstract setting. The theorem becomes a result about a $\vee$-preserving map between two $\vee$-complete Boolean algebras. The application of the theorem to the Alpern-Schneider case is immediate, but we also present a second application which yields a new Alpern-Schneider type result suitable for transition systems whose behaviours are described in terms of properties of the (one-step) transitions : Replacing "liveness" by "progress", we obtain that every property is a conjunction of a progress property and a safety property.

In our second theorem we show that safety properties are precisely the properties definable by a set of "configurations" that are to be avoided. Safety properties are closed under finite conjunction.

In our proofs we get by without needing to show $\overline{P} \vee \overline{Q} = \overline{P \vee Q}$ as is done in [5]. This property is in fact false in our second application.

2 Main Results

Let B_1 and B_2 be $\vee$-complete[1] Boolean algebras and $\varphi : B_1 \longrightarrow B_2$ a $\vee$-preserving map. Let $\sim$ be the *kernel* of φ, i.e.

$$a \sim b \Leftrightarrow \varphi(a) = \varphi(b).$$

Define the *closure* $\overline{a}$ of an element $a \in B_1$ as

$$\overline{a} = \bigvee \{x \in B_1 \mid x \sim a\}.$$

For elements $e \in B_1$ define :
e is a *safety element* $\Leftrightarrow \overline{e} = e$
e is a *liveness element* $\Leftrightarrow \overline{e} = 1.$

Theorem 1 *Every element $e \in B_1$ is a conjunction of a safety and a liveness element.*

Theorem 2 *An element $e \in B_1$ is a safety element if and only if there exists $u \in B_2$ so that*

$$e = \bigvee \{x \in B_1 \mid \varphi(x) \wedge u = 0\} \quad ;$$

e is actually the largest element whose φ-image avoids u.

A careful analysis of the proofs shows that we only require that B_1 is a $\vee$-complete and complemented modular lattice. B_2 must be a $\vee$-semilattice for theorem 1 and a complemented lattice for theorem 2.

Our original proof of theorem 2 assumed that φ was onto. M. Müller noticed that this assumption was not necessary. He also contributed the following corollary :

Corollary 1 *Safety elements are closed under finite conjunctions.*

[1] infinite suprema exist

3 Applications

Application 1 *For a set A (of states) let $\preceq$ denote the prefix relation between elements of A^* and elements of A^ω. For $P \subseteq A^\omega$ let*

$$\varphi(P) = \{x \in A^* \mid \exists \alpha \in P : x \preceq \alpha\} .$$

In this setup theorem 1 yields the Alpern-Schneider theorem. A is a set of possible states of a computation. A^ω is the set of all possible executions and A^* the set of all finite (or partial) executions. B_1 and B_2 are the powersets of A^ω and A^*. The fact that φ is $\vee$-preserving is trivial to check.

Elements of B_1 are called *properties*. A *liveness property* then is a property that admits every possible partial execution, and a *safety property* is a property that is maximal with respect to a given set of allowed partial executions.

Theorem 2 states in this connection that safety properties E are those properties that can be given by a set U of *disallowed* partial executions.

Application 2 *For a set A (of states) let $\asymp$ denote the following relation between A^2 and A^ω:*

$$(x, y) \asymp \alpha \Leftrightarrow \exists i \in \omega : (x, y) = (\alpha_i, \alpha_{i+1}) .$$

For $P \subseteq A^\omega$ let

$$\varphi(P) = \left\{(x, y) \in A^2 \mid \exists \alpha \in P : (x, y) \asymp \alpha\right\} .$$

Note that properties allowing all possible one-step transitions are usually called *progress* properties. In our notation : $P \subseteq A^\omega$ is a *progress property* iff $\forall x, y \in A \; \exists \tau \in P : (x, y) \asymp \tau$.

In the present setting, with B_1 and B_2 the powersets of A^ω and A^2, and the above $\vee$-preserving map φ, the liveness elements given by the abstract theory correspond to the progress properties as defined above. The safety elements correspond to those properties that can be defined by a set F of forbidden state-successor pairs. This, however, amounts to forbidding all finite prefixes containing a pair from F, so every safety element is also a safety property in the usual sense. Therefore we get :

Theorem 3 *Every property is a conjunction of a safety property and a progress property.*

This application seems particularly relevant for transition systems whose logical calculus is based on properties of single transitions. In UNITY, [3] for instance, properties are based on Hoare-triples $\{p\}\, s\, \{q\}$ where p and q are first order properties and s is a transition, given by a *multiple conditional assignment.*

Safety properties are derived from the basic temporal property *unless*, which is defined by quantifying over all statements s in a program P:

$$p \textit{ unless } q \Leftrightarrow \forall s \in P : \{p \wedge \neg q\}\, s\, \{p \vee q\} .$$

An execution α therefore satisfies p *unless* q if and only if there is no pair $(x, y) \asymp \alpha$ with $x \models p \wedge \neg q$ and $y \models \neg p \wedge \neg q$.

A similar remark applies to the derived unary safety property *invariant*, given by :

$$invariant\ q \Leftrightarrow q\ unless\ False.$$

4 Proofs

Given a $\vee$-preserving map φ between $\vee$-complete Boolean algebras B_1 and B_2 as requested, together with the definitions

$$a \sim b \Leftrightarrow \varphi(a) = \varphi(b),$$

and

$$\overline{a} = \bigvee \{x \in B_1 \mid x \sim a\},$$

we first show the following properties:

(i) $a \leq \overline{a}$,

(ii) $a \sim \overline{a}$,

(iii) $\overline{a} = \overline{\overline{a}}$,

(iv) $a \sim b \Leftrightarrow \overline{a} = \overline{b}$, and

(v) $a \sim b \Rightarrow a \vee c \sim b \vee c$.

From the definition of $\overline{a}$ and the reflexivity of $\sim$, (i) is immediate. For (ii) we calculate :

$$\begin{aligned} \varphi(\overline{a}) &= \varphi\left(\bigvee \{x \in B_1 \mid x \sim a\}\right) \\ &= \bigvee \{\varphi(x) \mid x \in B_1, x \sim a\} \\ &= \bigvee \{\varphi(x) \mid x \in B_1, \varphi(x) = \varphi(a)\} \\ &= \bigvee \{\varphi(a)\} \\ &= \varphi(a) . \end{aligned}$$

Hence $a \sim \overline{a}$, and

$$\begin{aligned} \overline{\overline{a}} &= \bigvee \{x \in B_1 \mid \varphi(x) = \varphi(\overline{a})\} \\ &= \bigvee \{x \in B_1 \mid \varphi(x) = \varphi(a)\} \\ &= \overline{a}. \end{aligned}$$

If $a \sim b$ then $\overline{a} = \overline{b}$ from the definition of the closure operator. Conversely, if $\overline{a} = \overline{b}$ then $a \sim \overline{a} = \overline{b} \sim b$ using (ii). Property (v) follows from the fact that φ preserves $\vee$: If $\varphi(a) = \varphi(b)$ then

$$\begin{aligned} \varphi(a \vee c) &= \varphi(a) \vee \varphi(c) \\ &= \varphi(b) \vee \varphi(c) \\ &= \varphi(b \vee c) . \end{aligned}$$

To show Theorem 1 we now use the same decomposition as [1] and [5]: Given any $e \in B_1$, then using (i) and the modular law [2] :

$$\begin{aligned} \overline{e} \wedge (e \vee \neg\overline{e}) &= e \vee (\overline{e} \wedge \neg\overline{e}) \\ &= e. \end{aligned}$$

$\overline{e}$ is a safety element from (iii), and $e \sim \overline{e}$ from (ii), so from (v) : $e \vee \neg\overline{e} \sim \overline{e} \vee \neg\overline{e} = 1_{B_1}$, is a progress element.

<u>Proof of theorem 2 :</u> We need two further properties for arbitrary $a \in B_1$ and $u \in B_2$:
(vi) $\overline{a} = \bigvee \{x \in B_1 \mid \varphi(x) \leq \varphi(a)\}$, and
(vii) $a = \bigvee \{x \in B_1 \mid \varphi(x) \leq u\} \Rightarrow a = \overline{a}$.

For (vi) we show that $\bigvee \{x \in B_1 \mid \varphi(x) = \varphi(a)\} = \bigvee \{x \in B_1 \mid \varphi(x) \leq \varphi(a)\}$. Containment of the left hand side in the right hand side is obvious, since less elements are involved in the join. For the inverse containment, let $\varphi(x) \leq \varphi(a)$, then $\varphi(x \vee a) = \varphi(x) \vee \varphi(a) = \varphi(a)$, so $x \leq (x \vee a) \leq \overline{a}$.

Assuming the hypothesis of (vii), we need to conclude $\overline{a} \leq a$. From the description of $\overline{a}$ in (vi), it is enough to derive the implication $\varphi(x) \leq \varphi(a) \Rightarrow x \leq a$. But from the definition of a in the hypothesis of (vii),

$$\begin{aligned} \varphi(x) &\leq \varphi(a) \\ &= \varphi\left(\bigvee \{x \mid \varphi(x) \leq u\}\right) \\ &= \bigvee \{\varphi(x) \mid \varphi(x) \leq u\} \\ &\leq u, \end{aligned}$$

hence $x \leq a$.

From (vi) and the fact that $y \leq z$ implies that $y \wedge \neg z = 0$ we now obtain the first direction of theorem 2, namely

$$\overline{e} = \bigvee \{x \in B_1 \mid \varphi(x) \wedge \neg\varphi(e) = 0\}$$

For the other direction suppose $e = \bigvee \{x \in B_1 \mid \varphi(x) \wedge u = 0\}$, which means that $e = \bigvee \{x \mid \varphi(x) \leq \neg u\}$ for some $u \in B_2$. By (vii) $e = \overline{e}$, so e is a safety element.

To show that e is actually the *largest* element x with $\varphi(x) \wedge u = 0$ we need only see that $\varphi(e) \wedge u = 0$. But

$$\begin{aligned} \varphi(e) &= \varphi\left(\bigvee \{x \in B_1 \mid \varphi(x) \wedge u = 0\}\right) \\ &= \bigvee \{\varphi(x) \mid x \in B_1, \varphi(x) \wedge u = 0\} \\ &\leq \bigvee \{v \in B_2 \mid v \wedge u = 0\} \\ &= \neg u. \end{aligned}$$

Proof of the Corollary : e_1 and e_2 are safety elements, so by theorem 2, $e_1 = \bigvee\{x \in B_1 \mid \varphi(x) \leq u_1\}$ and $e_2 = \bigvee\{x \in B_1 \mid \varphi(x) \leq u_2\}$. In particular, $\varphi(e_1) \leq u_1$ and $\varphi(e_2) \leq u_2$. Since φ is monotone, $\varphi(e_1 \wedge e_2) \leq \varphi(e_1) \wedge \varphi(e_2)$, hence $\varphi(e_1 \wedge e_2) \leq u_1 \wedge u_2$ and $e_1 \wedge e_2 \leq \bigvee\{x \in B_1 \mid \varphi(x) \leq u_1 \wedge u_2\}$. Conversely, if $\varphi(x) \leq u_1 \wedge u_2$ then $x \leq e_1 \wedge e_2$, so $e_1 \wedge e_2 = \bigvee\{x \in B_1 \mid \varphi(x) \leq u_1 \wedge u_2\}$, and by theorem 2 this is a safety element.

5 Conclusion

We have worked out the mathematical setup that is required to derive a result in the style of the Alpern-Schneider theorem characterizing safety and liveness. We have arrived at a more general result that directly yields the mentioned theorem, but that also broadens its application to transition systems whose temporal properties are given by properties of (one-step) transitions. In that case safety properties are those properties definable by a set of allowable state pairs (a_t, a_{t+1}) or, equivalently, by a set of such forbidden pairs. Progress properties are properties allowing each such possible pair, and every property is a conjunction of a safety property and a progress property.

References

[1] B. Alpern and F.B. Schneider. *Defining Liveness.* Information Processing Letters, 21 (1985), 181-185.

[2] G. Birkhoff. *Lattice theory.* A.M.S. Colloquium Publ., Providence, R.I. 1967.

[3] K.M. Chandy, J. Misra. *Parallel Program Design.* Addison Wesley, 1988.

[4] L. Lamport. *Proving the correctness of multiprocess programs.* IEEE Trans. on Software Engr. SE-3, 2(March), 125-143.

[5] M. Rem . *A Personal Perspective of the Alpern-Schneider Characterization of Safety and Liveness,* in W.H.J. Feijen, et al. *Beauty is Our Business*, Springer Verlag, 1990, 365-372.