Theoretical Computer Science 120 (1993) 123–155
Elsevier

"A la Burstall" intermittent assertions induction principles for proving inevitability properties of programs

P. Cousot

CNRS U.R.A. 1327, LIENS, Ecole Normale Supérieure, 45 Rue d'Ulm, F-75230 Paris Cedex 05, France

R. Cousot

CNRS U.R.A. 1439, LIX, Ecole Polytechnique, F-91128 Palaiseau Cedex, France

Communicated by R. Milner
Received January 1984
Revised January 1991

Abstract

Cousot, P. and R. Cousot, "A la Burstall" intermittent assertions induction principles for proving inevitability properties of programs, Theoretical Computer Science 120 (1993) 123–155.

We formalize Burstall's (1974) intermittent assertions method (initially conceived for proving total correctness of sequential programs) and generalize it to handle inevitability proofs for nondeterministic transition systems, hence (in particular) parallel programs.

Programs are modeled by transition systems, program executions by sets of complete traces and program properties by inevitability properties of transition systems (generalizing total correctness of programs). It follows that the study is independent of any particular programming language.

The basic proof principle that we derive from Burstall's and Manna and Waldinger's (1978) description of the intermittent assertions method is shown to be sound. It is also semantically complete under a condition on execution traces and inevitable properties. This condition is satisfied when considering inevitability properties such as total correctness or properties involving unary assertions on states. However, we conjecture that (even for deterministic programs) the basic proof principle is not complete when considering arbitrary binary inevitability properties (which relate state values at different "time instants").

This conjecture leads us to a generalization of Burstall's intermittent assertions method using transfinite induction (to handle unbounded nondeterminism) and using auxiliary or ghost variables in the limited form of ternary intermittent assertions (which can relate state values on program entry and at two other different "time instants").

Correspondence to: P. Cousot, CNRS U.R.A. 1327, LIENS, Ecole Normale Supérieure, 45 Rue d'Ulm, F-75230 Paris Cedex 05, France.

From this generalized proof principle we derive a series of induction principles so as to broaden the allowable forms of proofs. Also we obtain more abstract and hence better understood formalizations of Burstall's method.

All proof principles are sound and semantically complete (essentially, as noticed by Manna and Waldinger, because Burstall's method can be used to express "à la Floyd" proofs). However, we prove a stronger semantic completeness result in the sense that the propositions and lemmas involved in "à la Burstall" inevitability proofs can be chosen freely (at least under necessary and sufficient conditions that we specify accurately).

1. Introduction

We formalize Burstall's intermittent assertions method [3] initially conceived for proving total correctness of sequential programs and generalize it to handle inevitability properties of nondeterministic and parallel programs.

Programs are modeled by transition systems (Section 2) and program executions by sets of complete traces (Section 3) so that the study is independent of any particular programming language. We consider inevitability properties of programs such as total correctness, accessibility of a critical section, liveness of processes which must eventually progress, responsiveness to a request, etc. (Section 4).

In Section 5 we derive from examples in [3, 10] a basic proof principle which is a very concise formulation of Burstall's intermittent assertions method. The method is shown to be sound. Using transfinite (instead of finite) induction to handle unbounded nondeterminism, the basic induction principle is shown to be semantically complete under a sufficient (but not necessary) condition on execution traces and inevitable properties. This condition holds in particular when considering total correctness of programs as in [3]. It is also satisfied for unary inevitable properties which depend only upon final states (a restriction considered by Pnueli [11], Apt and Delporte [1] and Manna and Pnueli [9]).

When using unary inevitable properties, relationships between initial and final values of program variables can only be expressed by assigning initial values to auxiliary variables incorporated into states. The use of auxiliary variables has the disadvantage that the program has to be transformed. More importantly, the use of auxiliary variables is in a sense too flexible: one can relate any intermediate states during a computation and even store entire computations. Such a free use of auxiliary variables is not in the spirit of [3, 10], where lemmas are always of the form "if sometime $\phi(X_1, \ldots, X_n) \wedge X_1 = x_1 \wedge \cdots \wedge X_n = x_n$ at l then sometime $\psi(x_1, \ldots, x_n, X_1, \ldots, X_n)$ at l'" (where $X_1, \ldots, X_n$) are the program variables and $x_1, \ldots, x_n$ their respective symbolic values at program point l). This is captured in our basic induction principle using binary inevitable properties (better than by imposing adequate restrictions on the use of auxiliary variables that would depend upon the syntax of programs). However, we conjecture that even for deterministic programs there are inevitable properties for which the use of binary assertions is not semantically complete.

This conjecture leads us in Section 6 to a generalization of Burstall's intermittent assertions method using transfinite induction (to handle unbounded nondeterminism) and ternary intermittent assertions (thus allowing for lemmas of the more general form "if sometime $\phi(\underline{x}_1, \ldots, \underline{x}_n, X_1, \ldots, X_n) \wedge X_1 = x_1 \wedge \cdots \wedge X_n = x_n$ at l then sometime $\psi(\underline{x}_1, \ldots, \underline{x}_n, x_1, \ldots, x_n, X_1, \ldots, X_n)$ at l'", where $\underline{x}_1, \ldots, \underline{x}_n$ (resp. $x_1, \ldots, x_n$) denote the values of the program variables $X_1, \ldots, X_n$ on program entry (resp. at program point l). This generalized induction principle is then proved to be sound and semantically complete.

In Section 7 we derive a series of induction principles which are successive generalizations of the above proof principle. This broadens the range of application of the method [for example, when using infinite well-ordered sets of intermittent assertions (which can be given finite presentations by means of auxiliary termination variables) Burstall's method can be extended so as to incorporate Floyd's method [7]]. Moreover, the consideration of more and more abstract formalizations should lead to a better understanding of Burstall's method (for example, it is shown that hand-simulation and induction upon the data can be understood in a unified manner and reduced to computational induction in a form essentially more expressive than Floyd's method [7]). The successive generalizations introduce more flexibility to write proofs but no additional proof power, since all considered proof principles are shown to be sound and semantically complete and hence equivalent.

The completeness argument consists in showing that "à la Floyd" proofs can be reformulated using "à la Burstall" proofs (i.e. computational induction can be reduced to induction upon the data). However, this argument is not fully satisfactory because the style of the allowable proofs is fixed. Users of Burstall's method need a stronger completeness result since they want to know if the lemmas that they are going to use in their proofs can always be chosen freely. A positive answer is given in Section 8 (with the necessary and sufficient condition that each lemma involves a property which is inevitable for the program but also relatively to the other lemmas which are used in its proof).

2. Programs as transition systems

The operational semantics of a programming language associates a transition system $\langle S, t, \phi \rangle$ with each program of the language:

- S is a non-empty set of states,
- $t \in (S \times S \to \{\mathrm{tt}, \mathrm{ff}\})$ is a transition relation, understood as a function from pairs of states into truth values (tt is true and ff is false). $t(s, s')$ means that starting in state s and executing one program step can put the program into next state s'. A program is deterministic when a state may have none or one successor state. It is nondeterministic if a state s may have several different next states s',
- $\phi \in (S \to \{\mathrm{tt}, \mathrm{ff}\})$ characterizes initial states.

Example 2.1. Burstall [3] introduced the "intermittent assertions method" using examples. The first one was the following program, which computes 2^n when $n \geqslant 0$:

Start: $P := 1$;
Loop: **if** $N > 0$ **then begin** $P := 2 \times P$; $N := N - 1$;
goto Loop
end;
Finish:

Program states are of the form $\langle c, n, p \rangle$ where the control state c is a program label and the memory state associates integer values $n, p \in \mathbb{Z}$ with the program variables N, P:

- $S = \{\text{Start, Loop, Finish}\} \times \mathbb{Z} \times \mathbb{Z}$.

Execution should begin at program point "Start" with a positive initial value n for N and an arbitrary value p for P. Therefore,

- $\phi = \lambda \langle c, n, p \rangle . [c = \text{Start} \wedge n \geqslant 0]$.

The program is total and deterministic (all but the final states have a single successor state):

- $t(\langle c, n, p \rangle, \langle c', n', p' \rangle) = [(c = \text{Start} \wedge c' = \text{Loop} \wedge n' = n \wedge p' = 1)$
$\vee (c = \text{Loop} \wedge n > 0 \wedge c' = \text{Loop} \wedge n' = n - 1 \wedge p' = 2 \times p)$
$\vee (c = \text{Loop} \wedge n \leqslant 0 \wedge c' = \text{Finish} \wedge n' = n \wedge p' = p)]$.

3. Program executions as complete traces

Executions of a program $\langle S, t, \phi \rangle$ will be modeled as a set $\Sigma\langle S, t, \phi \rangle$ of sequences of states called complete execution traces. A sequence $p = p_0 p_1 p_2 \ldots$ in $\Sigma\langle S, t, \phi \rangle$ (where p_i is short for $p(i)$), represents an execution that starts in state p_0, performs the first program step to reach program state p_1, performs the next program step to reach program state p_2, etc. Since execution must start in some initial state p_0, this sequence cannot be empty. When execution does not terminate, this sequence is infinite. A finite sequence $p_0 \ldots p_n$ ends with a blocking state p_n which has no possible successor state. Therefore traces represent complete executions (as opposed to their prefixes which represent executions still in progress).

More formally,

- ω is the set of natural numbers;
- 0 is the empty set (also written $\emptyset$) or zero;
- if $n \in \omega$ and $n \neq 0$ then n will denote $\{0, \ldots, n-1\}$ (so that $m \in n$ is equivalent to $m < n$);
- if E is a set then $E \sim x = \{y \in E: y \neq x\}$ and $|E|$ is the cardinality of E;
- $\beta \in (S \to \{\text{tt}, \text{ff}\})$,
$\beta = \lambda s . [\forall s' \in S . \neg t(s, s')]$ characterizes blocking states;

- $\Sigma^0\langle S,t,\phi\rangle=\emptyset$, empty traces, are not considered;
- $\Sigma^n\langle S,t,\phi\rangle=\{p\in(n\to S)\colon \phi(p_0)\wedge\forall i\in(n-1).t(p_i,p_{i+1})\wedge\beta(p_{n-1})\}$: finite complete traces of length $n>0$;
- $\Sigma^\omega\langle S,t,\phi\rangle=\{p\in(\omega\to S)\colon \phi(p_0)\wedge\forall i\in\omega.t(p_i,p_{i+1})\}$: infinite traces;
- $\Sigma\langle S,t,\phi\rangle=\bigcup_{n\in\omega}\Sigma^n\langle S,t,\phi\rangle\cup\Sigma^\omega\langle S,t,\phi\rangle$: complete traces.

Example 3.1. A complete trace for the program in Example 2.1 with $N=3$ and $P=\underline{p}$ as starting condition would be

$$\langle\text{Start},3,\underline{p}\rangle\langle\text{Loop},3,1\rangle\langle\text{Loop},2,2\rangle\langle\text{Loop},1,4\rangle$$

$$\langle\text{Loop},0,8\rangle\langle\text{Finish},0,8\rangle.$$

4. Inevitability properties of programs

A property ψ is *inevitable* for a program if any program execution eventually leads to a state satisfying ψ. Termination, total correctness or absence of individual starvation of parallel processes are examples of inevitability properties of programs.

More formally, $\psi\in(S\times S\to\{\text{tt},\text{ff}\})$ is inevitable for $\langle S,t,\phi\rangle$ if and only if

$$\forall p\in\Sigma\langle S,t,\phi\rangle.\exists i\in Dom(p).\psi(p_0,p_i),$$

where $Dom(p)$ is the domain of function (or relation) p, $Rng(p)$ its range and $Fld(p)=Dom(p)\cup Rng(p)$ its field.

Extending Dijkstra's definitions of weak and strong termination [6], we say that $\psi\in(S\times S\to\{\text{tt},\text{ff}\})$ is *strongly inevitable* for $\langle S,t,\phi\rangle$ if and only if

$$\forall s\in S.\exists k\in\omega.\forall p\in\Sigma\langle S,t,\phi\rangle.[(p_0=s)\Rightarrow(\exists i\leqslant k.\psi(p_0,p_i))].$$

In other words, inevitability is strong when the number i of program steps necessary for reaching the "final" state p_i is bounded by an integer k depending only on the "initial" state p_0.

$\psi\in(S\times S\to\{\text{tt},\text{ff}\})$ is *weakly inevitable* for $\langle S,t,\phi\rangle$ if and only if ψ is inevitable for $\langle S,t,\phi\rangle$ but not strongly.

Example 4.1. The program in Example 2.1 computes $P=2^n$ when the initial value n of N is positive. This total correctness property can be expressed formally by the statement that

$$\psi=\lambda(\langle c,n,p\rangle,\langle c',n',p'\rangle).[c'=\text{Finish}\wedge p'=2^n]$$

is inevitable. Termination is strong since traces have $n+3$ states when initially $N=n\geqslant 0$.

Program properties are expressed using sets (or their characteristic functions) and not formal languages. This is because we want to get rid of those incompleteness

problems which are due to the inconvenient choice of assertion languages which are not expressive enough in order to describe these sets.

5. The basic induction principle underlying Burstall's intermittent assertions method

In this section we work out a basic induction principle which is a very concise formulation of Burstall's method. In the next section we shall relax a number of restrictions which cause incompleteness problems and derive more abstract and general induction principles which generalize Burstall's method.

The best way to convince the reader that our basic induction principle indeed corresponds to Burstall's method would be to derive it from an already existing formalization of the method. Since no such existing formalization is general enough and widely accepted, the best we can do is to start from Burstall's proof of the program in Example 2.1 [3] using Manna and Waldinger's notations [10]. The treatment of Burstall's other examples is similar but would be too long to be included here.

5.1. Proving inevitability properties of programs

The total correctness of the program in Example 2.1 is specified by the following proposition:

- if sometime $n \geqslant 0 \wedge N = n$ at Start then sometime $P = 2^n$ at Finish.

The proof of this proposition involves the following lemma:

- if sometime $n \geqslant 0 \wedge N = n \wedge P = p$ at Loop then sometime $N = 0 \wedge P = p \times 2^n$ at Loop.

Burstall observed that in the above statements n and p are mathematical variables whereas N and P are not since their meaning depends on context. The use of both mathematical and program variables in the same statement might be confusing (for example, from $N = n$ and $N = 0$ we cannot conclude that $n = 0$ in the above lemma). This confusion can be avoided if we get rid of program variables using different mathematical variables to denote values of program variables at different "time instants". For example, the lemma could be written as follows:

- if sometime $n \geqslant 0$ at Loop then sometime $n' = 0 \wedge p' = p \times 2^n$ at Loop.

This means:

"For all n, if $n \geqslant 0$ holds and execution of the program is started at label Loop with program variables N and P having values n and p then control will eventually pass through Loop with some values n' and p' of the program variables N and P such that $n' = 0 \wedge p' = p \times 2^n$ is satisfied".

Therefore the lemma simply asserts that

$$\theta_0 = \lambda(\langle c, n, p\rangle, \langle c', n', p'\rangle).[c' = \text{Loop} \wedge n' = 0 \wedge p' = p \times 2^n]$$

is inevitable for $\langle S, t, \varepsilon_0 \rangle$, where

$$\varepsilon_0 = \lambda \langle c, n, p \rangle . [c = \text{Loop} \wedge n \geqslant 0].$$

Similarly, the proposition asserts the inevitability of

$$\theta_1 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Finish} \wedge p' = 2^n]$$

for $\langle S, t, \varepsilon_1 \rangle$, where

$$\varepsilon_1 = \lambda \langle c, n, p \rangle . [c = \text{Start} \wedge n \geqslant 0].$$

More generally, for proving that ψ is inevitable for $\langle S, t, \phi \rangle$, Burstall's method consists in discovering auxiliary properties $\{\theta_l \in (S^2 \to \{\text{tt}, \text{ff}\}) : l \in \Lambda\}$ and corresponding initial conditions $\{\varepsilon_l \in (S \to \{\text{tt}, \text{ff}\}) : l \in \Lambda\}$ (such that $\exists \pi \in \Lambda . [\varepsilon_\pi = \phi \wedge \theta_\pi = \psi]$) which are all shown to be inevitable:

$$\forall l \in \Lambda . \forall p \in \Sigma \langle S, t, \varepsilon_l \rangle . \exists i \in Dom(p) . \theta_l(p_0, p_i).$$

Only a finite number, $|\Lambda|$, of lemmas should be used.

Remark. Since Burstall [3] considered only deterministic and total programs, the statement

if sometime $P(n, p)$ at L then sometime $Q(n, p, n', p')$ at L'

can also be understood as

$$\exists s \in \Sigma \langle S, t, \varepsilon \rangle . \exists i \in Dom(s) . \theta(s_0, s_i),$$

where

$$\varepsilon = \lambda \langle c, n, p \rangle . [c = L \wedge P(n, p)],$$

$$\theta = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = L' \wedge Q(n, p, n', p')].$$

All results in the present paper can very easily be adapted to this existential interpretation. However, we have chosen to develop the universal interpretation because it is more suitable for total correctness (and more generally inevitability properties) of nondeterministic programs with depth search execution, [8] hence parallel programs.

5.2. An example of proof

Now that we have obtained an abstract formalization of programs and inevitable properties of programs, let us come back to the example in order to capture the essence of Burstall's proof method.

The proof of proposition θ_1 is the following.

Assume:
  sometime $N \geqslant 0 \wedge N = n$ at Start $\lceil 13 \rfloor$
then by hand simulation:
  sometime $N \geqslant 0 \wedge N = n \wedge P = 1$ at Loop $\lceil 12 \rfloor$
then using lemma θ_0:
  sometime $N = 0 \wedge P = 2^n$ at Loop $\lceil 11 \rfloor$
then by hand simulation:
  sometime $P = 2^n$ at Finish $\lceil 10 \rfloor$
Q.E.D.

The proof of lemma θ_0 is by induction on n as follows:

Assume:
  sometime $N \geqslant 0 \wedge N = n \wedge P = p$ at Loop $\lceil 02 \rfloor$
either $N \leqslant 0$ and Q.E.D.
or $N > 0$ and then by hand simulation:
  sometime $N > 0 \wedge N = n - 1 \wedge P = p \times 2$ at Loop $\lceil 01 \rfloor$
then using lemma θ_0 as induction hypothesis for $n-1$
(such that $n > n - 1 \geqslant 0$):
  sometime $N = 0 \wedge P = p \times 2^n$ at Loop $\lceil 00 \rfloor$
Q.E.D.

5.3. *Intermittent assertions*

In [3], a proof of a lemma θ_l is a nonempty finite sequence $I_l^{n_l}, \ldots, I_l^1, I_l^0$ of intermittent assertions derived from one another by hand simulation or application of lemmas.

For example, the proof of proposition θ_1 involved the discovery of the following intermittent assertions:

$$I_1^3 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Start} \wedge n' \geqslant 0 \wedge n' = n],$$

$$I_1^2 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Loop} \wedge n' \geqslant 0 \wedge n' = n \wedge p' = 1],$$

$$I_1^1 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Loop} \wedge n' = 0 \wedge p' = 2^n],$$

$$I_1^0 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Finish} \wedge p' = 2^n],$$

whereas the proof of lemma θ_0 involves the discovery of

$$I_0^2 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Loop} \wedge n' \geqslant 0 \wedge n' = n \wedge p' = p],$$

$$I_0^1 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Loop} \wedge n' > 0 \wedge n' = n - 1 \wedge p' = p \times 2],$$

$$I_0^0 = \lambda(\langle c, n, p \rangle, \langle c', n', p' \rangle).[c' = \text{Loop} \wedge n' = 0 \wedge p' = p \times 2^n].$$

Remark 1. The intermittent assertions involved in the proof need not be different, as shown by the following counterexample, which is a valid proof of proposition θ_1 for all finite $k \geqslant 1$:

$$k \text{ times} \begin{cases} \text{sometime } N \geqslant 0 \wedge N = n \text{ at Start} & \text{(premise)} \\ \text{sometime } N \geqslant 0 \wedge N = n \wedge P = 1 \text{ at Loop} & \text{(hand-simulation)} \\ \text{sometime } N = 0 \wedge P = 2^n \text{ at Loop} & \text{(lemma)} \\ \ldots & \ldots \\ \text{sometime } N = 0 \wedge P = 2^n \text{ at Loop} & \text{(lemma)} \\ \text{sometime } P = 2^n \text{ at Finish} & \text{(conclusion).} \end{cases}$$

Remark 2. According to Burstall [3], "⌈sometime P at L⌋ says that there exists a state during the execution which is at L and has property P". Stated otherwise, all intermittent assertions I_l^i involved in the inevitability proof of lemma θ_l should be inevitable for $\langle S, t, \varepsilon_l \rangle$. This interpretation of intermittent assertions is inconsistent. For example I_0^1 never holds during execution when initially $N = 0$. More generally, Burstall treats tests by case analysis [3], so that the intermittent assertions involved in each case might not be inevitable for those initial states not corresponding to the considered case. We shall choose another interpretation of intermittent assertions so that case analysis causes no problem since only the disjunction of the intermittent assertions corresponding to all cases will have to be inevitable for all initial states.

5.4. Verification conditions

In a valid proof of inevitability of θ_l for $\langle S, t, \varepsilon_l \rangle$, intermittent assertions $I_l^{n_l}, \ldots, I_l^1, I_l^0$ are derived from one another according to rules (for computing the effect of an assignment or test, for using a lemma, etc.). Burstall's informal rules [3] can be understood as verification conditions that must be satisfied by the intermittent assertions. These verification conditions are now expressed formally.

5.4.1. Premises

All proofs in [3] start with the assumption of the premises ε_l of the proposition or lemma θ_l which is proved. Stated otherwise, $I_l^{n_l}$ should hold when the current state s' is an initial state s:

$$\forall s, s' \in S.([\varepsilon_l(s) \wedge s' = s] \Rightarrow I_l^{n_l}(s, s'))$$

or, more simply,

$$\forall s \in S.(\varepsilon_l(s) \Rightarrow I_l^{n_l}(s, s)).$$

For instance, the proof of proposition θ_1 starts with the check that

$$\forall \langle c, n, p \rangle \in S.(\varepsilon_1(\langle c, n, p \rangle) \Rightarrow I_1^3(\langle c, n, p \rangle, \langle c, n, p \rangle))$$

(where $c = \text{Start}$ and $n \geqslant 0$ or else ε_1 is false so that the verification condition is obviously true), whereas for lemma θ_0 we have

$$\forall \langle c, n, p \rangle \in S.(\varepsilon_0 \langle c, n, p \rangle) \Rightarrow I_0^2(\langle c, n, p \rangle, \langle c, n, p \rangle))$$

(where $c = \text{Loop}$, $n \geqslant 0$ in the nonevident case).

5.4.2. Hand simulation

Assume that the proof of proposition θ_l worked forward until reaching intermittent assertion I_l^i which is not the last one. The next step in the proof can be taken by hand simulation.

For total deterministic programs, Burstall's rules for computing the effect of an assignment or test [3] check that the current state s' satisfying I_l^i has a successor state s'' satisfying some intermittent assertion I_l^j which has to be taken into consideration later in the proof so that $j < i$:

$$[I_l^i(s, s') \wedge t(s', s'')] \Rightarrow \exists j < i. I_l^j(s, s'').$$

For example, in the proof of proposition θ_1, assignment $P := 1$ leads from $\lceil 13 \rfloor$ to $\lceil 12 \rfloor$ and corresponds to the following verification condition:

$$[I_1^3(\langle c, n, p \rangle, \langle c', n', p' \rangle) \wedge t(\langle c', n', p' \rangle, \langle c'', n'', p'' \rangle]$$
$$\Rightarrow I_1^2(\langle c, n, p \rangle, \langle c'', n'', p'' \rangle),$$

where $c' = \text{Start}$, $n' \geqslant 0$, $n' = n$ or the condition is obviously verified. The test $N \leqslant 0$ leads from $\lceil 11 \rfloor$ to $\lceil 10 \rfloor$ and corresponds to the verification condition

$$[I_1^1(\langle c, n, p \rangle, \langle c', n', p' \rangle) \wedge t(\langle c', n', p' \rangle, \langle c'', n'', p'' \rangle]$$
$$\Rightarrow I_1^0(\langle c, n, p \rangle, \langle c'', n'', p'' \rangle),$$

where $c' = \text{Loop}$, $n' \geqslant 0$, $p' = 2^n$, $c'' = \text{Finish}$, $n'' = n'$, $p'' = p'$).

In the proof of lemma θ_0, the loop body leads from $\lceil 02 \rfloor$ to $\lceil 01 \rfloor$. (In accordance with the operational semantics of the program in Example 2.1, the loop body should be treated as an atomic action.) The corresponding verification condition is

$$[I_0^2(\langle c, n, p \rangle, \langle c', n', p' \rangle) \wedge n' > 0 \wedge t(\langle c', n', p' \rangle, \langle c'', n'', p'' \rangle)]$$
$$\Rightarrow I_0^1(\langle c, n, p \rangle, \langle c'', n'', p'' \rangle),$$

where $c' = \text{Loop}$, $n' = n$, $p' = p$, $n' > 0$, $c'' = \text{Loop}$, $n'' = n' - 1$, $p'' = 2 \times p'$ or the condition is trivially satisfied.

Such verification conditions are not sufficient when nondeterminism is involved since it must also be proved that no blocking state is reachable. Hence, hand simulation should ensure the existence of at least one successor state:

$$\forall s, s' \in S.[I_l^i(s, s') \Rightarrow \exists s'' \in S.t(s', s'')]$$

and check that all possible successor states satisfy some intermittent assertion considered later:

$$\forall s, s', s'' \in S.[(I_l^i(s, s') \wedge t(s', s'')) \Rightarrow (\exists j < i. I_l^j(s, s''))].$$

5.4.3. Using lemmas in the proof of propositions

In the proof of proposition θ_1, intermittent assertion $\lceil 11 \rfloor$ is derived from $\lceil 12 \rfloor$ using lemma θ_0. It must first be checked that all current states s' satisfying $\lceil 11 \rfloor$ also satisfy the premises ε_0 of lemma θ_0. Then, by applying the lemma it must be proved that all successors s'' of s' by θ_0 satisfy $\lceil 12 \rfloor$. The corresponding verification conditions are the following:

$$[I_1^2(\langle c, n, p\rangle, \langle c', n', p'\rangle) \Rightarrow \varepsilon_0(\langle c', n', p'\rangle)] \wedge$$

$$[I_1^2(\langle c, n, p\rangle, \langle c', n', p'\rangle) \wedge \theta_0(\langle c', n', p'\rangle, \langle c'', n'', p''\rangle)$$

$$\Rightarrow I_1^1(\langle c, n, p\rangle, \langle c'', n'', p''\rangle)],$$

where in the nonbanal case $c' = \text{Loop}$, $n' \geqslant 0$, $n' = n$, $p' = 1$, $c'' = \text{Loop}$, $n'' = 0$, $p'' = p' \times 2^{n'}$.

More generally, the verification condition corresponding to the use of a lemma in the proof of a proposition is (temporarily)

$$\forall s, s' \in S.[I_l^i(s, s') \Rightarrow (\exists l' \in \Lambda.[\varepsilon_{l'}(s') \wedge \forall s'' \in S.[\theta_{l'}(s', s'') \Rightarrow \exists j < i. I_l^j(s, s'')]])].$$

Observe that (contrary to the case of hand simulation) the test that current states s' satisfy the premises $\varepsilon_{l'}$ of lemma $\theta_{l'}$ ensures the existence of at least one successor s'' to s' (unless improbably $\theta_{l'}$ is the identity lemma, i.e. $\theta_{l'}(s', s'') \Rightarrow (s' = s'')$). This is because lemma $\theta_{l'}$ is separately proved to be inevitable for $\Sigma\langle S, t, \varepsilon_{l'}\rangle$.

Apropos of the use of lemmas, notice that Burstall relies upon the mathematical culture of his readers and does not take the trouble to state elementary logical rules such as "proofs of lemmas and propositions should not be circular" [3]. Yet such rules have to be captured in the formalization of Burstall's method. A simple way consists in partially ordering the set Λ of lemmas by a well-founded ordering $\prec$ such that $l' \prec l$ is understood as "the inevitability proof of $\theta_{l'}$ does not depend upon the assumption that θ_l is inevitable". The (permanent) verification condition corresponding to the use of a lemma in the proof of a proposition is now the following:

$$\forall s, s' \in S.[I_l^i(s, s') \Rightarrow (\exists l' \in \Lambda.[l' \prec l \wedge \varepsilon_{l'}(s') \wedge \forall s'' \in S.[\theta_{l'}(s', s'')$$

$$\Rightarrow \exists j < i. I_l^j(s, s'')]])].$$

Moreover, since the set Λ is finite and $\prec$ is well-founded we can always (up to a rank function) choose Λ as a set of positive numbers and $\prec$ as the corresponding natural ordering $<$.

5.4.4. A little induction

Burstall proves lemmas using various forms of the principle of mathematical induction [3] which are all equivalent to the following:

$$\forall n' \in \omega.[(\forall n < n'.P(n)) \Rightarrow P(n')] \Rightarrow [\forall n' \in \omega.P(n')].$$

For example, in the proof of lemma θ_0, intermittent assertion $\lceil 00 \rfloor$ is derived from assertion $\lceil 01 \rfloor$ using lemma θ_0 as induction hypothesis. This is valid because

$$I_0^1(\langle c, n, p\rangle, \langle c', n', p'\rangle) \Rightarrow [\varepsilon_0(\langle c', n', p'\rangle) \wedge n' < n].$$

Then, by induction hypothesis, we derive intermittent assertion I_0^0 such that

$$[I_0^1(\langle c, n, p\rangle, \langle c', n', p'\rangle) \wedge \theta_0(\langle c', n', p'\rangle, \langle c'', n'', p''\rangle)]$$

$$\Rightarrow I_0^0(\langle c, n, p\rangle, \langle c'', n'', p''\rangle),$$

where $c' = \text{Loop}$, $n' > 0$, $n' = n - 1$, $p' = p \times 2$, $c'' = \text{Loop}$, $n'' = 0$, $p'' = p' \times 2^{n'}$.

This verification condition is specific of the example considered but, in general, Burstall specifies that the induction is on the data [3]. Since the above principle of mathematical induction applies to natural numbers, induction on data involves a map f_0 from the data into natural numbers. For example,

$$I_0^1(\langle c, n, p\rangle, \langle c', n', p'\rangle) \Rightarrow [\varepsilon_0(\langle c', n', p'\rangle) \wedge f_0(\langle c', n', p'\rangle) < f_0(\langle c, n, p\rangle)],$$

where

$$f_0 = \lambda\langle c, n, p\rangle.[n].$$

Since proofs of different lemmas are usually different, different maps f_l may have to be used, hence $f \in (\Lambda \to (S \to \omega))$. We infer from the example that the verification condition for the use of a lemma as induction hypothesis in the proof of this lemma should be of the form

$$\forall s, s' \in S.[I_l^i(s, s') \Rightarrow (\varepsilon_l(s') \wedge f_l(s') < f_l(s) \wedge \forall s'' \in S.[\theta_l(s', s'')$$

$$\Rightarrow \exists j < i.I_l^j(s, s'')])].$$

5.4.5. Conclusion

Starting from the premises ε_l of a lemma, a proof of this lemma ends when some intermittent assertion I_l^i has been derived which implies the conclusion θ_l of the lemma:

$$\forall s, s' \in S.[I_l^i(s, s') \Rightarrow \theta_l(s, s')].$$

For instance, the proof of proposition θ_1 ends with.

$$I_1^0(\langle c, n, p\rangle, \langle c', n', p'\rangle) \Rightarrow \theta_1(\langle c, n, p\rangle, \langle c', n', p'\rangle),$$

where $c' = \text{Finish}$, $p' = 2^n$ in the nontrivial case, whereas the proof of lemma θ_0 ends either with

$$[I_0^2(\langle c, n, p\rangle, \langle c', n', p'\rangle) \wedge n' \leqslant 0] \Rightarrow \theta_0(\langle c, n, p\rangle, \langle c', n', p'\rangle),$$

where $c' = \text{Loop}$, $n' = 0$, $n' = n$, $p' = p$,
or with

$$I_0^0(\langle c, n, p\rangle, \langle c', n', p'\rangle) \Rightarrow \theta_0(\langle c, n, p\rangle, \langle c', n', p'\rangle),$$

where $c' = \text{Loop}$, $n' = 0$, $p' = p \times 2^n$.

Finally, observe that in a proof all intermediate intermittent assertions should be processed (either by hand simulation or by using a lemma (in the proof of a proposition or as induction hypothesis)) or imply the conclusion.

5.5. *The basic induction principle formalizing Burstall's intermittent assertions method*

We can now sum up what we have learned from the example.
For proving

$$\forall p \in \Sigma \langle S, t, \phi\rangle . \exists i \in Dom(p) . \psi(p_0, p_i). \tag{0}$$

Burstall's method consists in proving the following:

$$[\exists \Lambda \in \omega, \varepsilon \in (\Lambda \to (S \to \{\text{tt}, \text{ff}\})), \theta \in (\Lambda \to (S^2 \to \{\text{tt}, \text{ff}\})), f \in (\Lambda \to (S \to \omega)),$$

$$n \in (\Lambda \to \omega).$$

$$(\exists \pi \in \Lambda . [\varepsilon_\pi = \phi \wedge \theta_\pi = \psi]) \ \wedge$$

$$(\forall l \in \Lambda . \exists I_l \in (n_l + 1 \to (S^2 \to \{\text{tt}, \text{ff}\})).$$

$$\forall i \leqslant n_l,\ s, s' \in S.$$

$$\text{(P)} \quad [\varepsilon_l(s) \Rightarrow I_l^{n_l}(s, s)] \ \wedge \tag{1}$$

$$[I_l^i(s, s') \Rightarrow$$

$$\text{(HS)} \quad (\exists s'' \in S . t(s', s'') \wedge \forall s'' \in S . [t(s', s'') \Rightarrow \exists j < i . I_l^j(s, s'')]) \ \vee$$

$$\text{(LI)} \quad (\exists l' \in \Lambda . [((l' < l) \vee (l' = l \wedge f_l(s') < f_l(s))) \wedge \varepsilon_{l'}(s') \wedge \forall s'' \in S . (\theta_{l'}(s', s'')$$

$$\Rightarrow \exists j < i . I_l^j(s, s''))]) \vee$$

$$\text{(C)} \quad \theta_l(s, s')])].$$

5.6. *Soundness and completeness issues about Burstall's method*

The question of soundness and completeness of Burstall's method has already been tackled partially. Representing programs by transition relations, the nondeterminism

of which is bounded, and giving a temporal interpretation of the intermittent assertions method, Pnueli proved the soundness and semantic completeness of a version of Burstall's method [11]. Similar arithmetical soundness and completeness results were obtained by Apt and Delporte for sequential deterministic structured programs [1]. Completeness results also follow informally from Manna and Waldinger's remark that the intermittent assertions method can be used to express conventional "à la Floyd" partial correctness and termination proofs that use the well-founded set approach [10], a method which is known to be semantically complete.

However, the exact scope of the above results should be interpreted very cautiously since these proofs only deal with the case of unary intermittent assertions (i.e. which assert a property of states, such as "if sometime $P(s)$ at L then sometime $Q(s')$ at L'") whereas Burstall's method and induction principle (1) make use of binary intermittent assertions (i.e. which relate states, such as "if sometime $P(s)$ at L then sometime $Q(s, s')$ at L'"). It is often argued that both approaches are equivalent because the effect of binary assertions can be obtained using auxiliary variables and unary assertions. Indeed, initial or intermediate values of program variables can be stored into auxiliary variables the value of which is part of the state. In fact, the use of auxiliary variables and unary assertions is more powerful than the use of binary assertions as in (1). This is because using auxiliary variables one can express relationships between values of the variables at any two different moments in the course of the computation (and even store entire execution traces into history variables). This is not possible with binary assertions since, for example, only the main proposition (and not all lemmas) can depend upon the initial values of the program variables in induction principle (1). However, the use of binary assertions appears to be much more disciplined because the question of when auxiliary variables do have to be introduced is solved once for all.

Our understanding with respect to soundness and completeness of induction principle (1) can be described as follows.

5.6.1. Soundness

Theorem 5.1 (soundness). (1) $\Rightarrow$ (0)

Proof. We introduce in Section 6 the induction principle (2), an obvious generalization of (1) (so that (1) $\Rightarrow$ (2)) and prove that (2) $\Rightarrow$ (0). □

5.6.2. Conjectures about semantic incompleteness

Although induction principle (1) only allows to use ranges of natural numbers, it can be used to prove termination of weakly but not strongly terminating programs. We show this using the following example (taken from [6, p. 356]):

Example 5.2. In general, the program (X and Y being natural constants),

$x, y := X, Y;$
do $x>0 \rightarrow x, y := x-1$, any natural number
$\square$ $y>0 \rightarrow y := y-1$
od

does not enjoy the property of strong termination, because for $X>0$ no upper bound for y can be given.

Weak termination can be proved by Floyd's method using the left lexicographic ordering on pairs of natural numbers $(x, y) \prec (x', y')$ if and only if $(x<x') \vee (x=x' \wedge y<y')$.

This can also be proved by induction principle (1). We have $S=\mathbb{Z}^2$, $t=\lambda(\langle x, y\rangle, \langle x', y'\rangle).[(x>0 \wedge x'=x-1) \vee (y>0 \wedge x'=x \wedge y'=y-1)]$, $\phi=\lambda\langle x, y\rangle.[x=X \geqslant 0 \wedge y=Y \geqslant 0]$, $\psi=\lambda(\langle x, y\rangle, \langle x', y'\rangle).[x'=y'=0]$ and choose $\Lambda=X+2$, $\varepsilon_{X+1}=\phi$, $\varepsilon_l(\langle x, y\rangle)=[x=l]$ for $l \in (X+1)$, $\theta_l=\psi$ for $l \in (X+2)$, $\pi=X+1$, $f_l(\langle x, y\rangle)=y$ for $l \in (X+2)$, $n_0=2$, $I_0^2(\langle x, y\rangle, \langle x', y'\rangle)=(\varepsilon_0(\langle x, y\rangle) \wedge \langle x', y'\rangle=\langle x, y\rangle)$ [(P), (C) when $y'=0$, (HS) when $y'>0$], $I_0^1(\langle x, y\rangle, \langle x', y'\rangle)=(x=0 \wedge y>0 \wedge t(\langle x, y\rangle, \langle x', y'\rangle))$ [(LI) with $l'=l=0$], $I_0^0=\theta_0$ [(C)], when $l=1, \ldots, X$, $n_l=2$, $I_l^2(\langle x, y\rangle, \langle x', y'\rangle)=(\varepsilon_l(\langle x, y\rangle) \wedge \langle x', y'\rangle=\langle x, y\rangle)$ [(P), (HS)], $I_l^1(\langle x, y\rangle, \langle x', y'\rangle)=(\varepsilon_l(\langle x, y\rangle) \wedge t(\langle x, y\rangle, \langle x', y'\rangle))$ [(LI) with $l'=l-1$ when $x'=x-1$, (LI) with $l'=l$ when $x'=x$ and $y'=y-1$], $I_l^0=\theta_l$ [(C)], $n_{X+1}=1$, $I_{X+1}^1(\langle x, y\rangle, \langle x', y'\rangle)=(\varepsilon_{X+1}(\langle x, y\rangle) \wedge \langle x', y'\rangle=\langle x, y\rangle)$ [(P), (LI) with $l'=X$], $I_{X+1}^0=\theta_{X+1}^0$ [(C)]. (The check of the verification conditions is left to the reader. We have indicated after each intermittent assertion which alternative should be chosen.)

As shown by the above example, the greater generality of Burstall's method restricted to natural numbers (which can be used to prove termination of weakly but not strongly terminating programs) over Floyd's method restricted to natural numbers (which can be used to prove only strong termination) is only seeming, because Burstall's method implicitly relies upon the lexicographic ordering on pairs of natural numbers as pointed out by induction principle (1).

Despite this apparent superiority, the order type of the lexicographic ordering on pairs of natural numbers involved in induction principle (1) is not as high as necessary when considering arbitrarily unbounded nondeterminism. Therefore we make the following semantic incompleteness conjecture: (0) $\not\Rightarrow$ (1).

By analogy with Floyd's method, two remedies can be considered in order to solve incompleteness problems related to unbounded nondeterminism. One consists in considering only bounded nondeterminism. The other consists in considering induction over arbitrary well-orderings (or up to an isomorphism over arbitrary ordinals). However, we risk the conjecture that induction principle (1) is not complete even with these simplifying hypotheses neither for bounded nondeterminism:

$[(0) \wedge \forall s \in S.(|\{s' \in S.t(s,s')\}| < \omega)] \not\Rightarrow (1)$ nor for arbitrary well-orderings: $(0) \not\Rightarrow [(1)$, where $f \in (\Lambda \to (S \to \Lambda))$, $\Lambda \in Ord]$ (Ord is the class of ordinals).

These conjectures follow from the remark that except for trivial examples (that can be handled by hand simulation) proofs involve a well-founded relation on the set of descendants of initial states corresponding to $(\Lambda \times S, \prec)$, where $(l', s') \prec (l, s)$ if and only if $(l' < l \vee (l' = l \wedge f_l(s') < f_l(s)))$. Although there is (by the inevitability assumption) a well-founded relation on the set of descendants of *each* initial state, there may exist no such well-founded relation on the set of descendants of *all* initial states as necessary in induction principle (1) because f_l does not depend upon initial states. This is the case for $S = \omega, t(x, x') = [x' = x + 1]$, $\phi(x) = \mathrm{tt}$, $\psi(x, x') = [x' = 2x]$.

5.6.3. A partial semantic completeness result

The above conjectures have only limited consequences because they do not apply in a great number of practical situations.

One such situation is when nondeterminism is bounded and the number of initial states is finite so that (at least in theory) proofs can be entirely done by hand simulation.

More interesting situations are those of total correctness of sequential programs considered in [3] or unary intermittent assertions considered in [11, 1]. Both situations can be coped with as particular cases of the following partial semantic completeness result.

We say that

- state s is *intermediate* for $\langle S, t, \phi, \psi \rangle$ when there is some execution trace such that ψ does not hold up to s;
- state s is a *goal* for $\langle S, t, \phi, \psi \rangle$ when ψ holds for the first time at s on some execution trace;
- state s is *accessible* for $\langle S, t, \phi, \psi \rangle$ when s is an intermediate or a goal state;
- the inevitability of ψ is *initial states independent* for $\langle S, t, \phi \rangle$ when no intermediate state can be a goal.

Definitions 5.3. (*Intermediate, goal and accessible states, initial states independence*).

- $Inter\langle S, t, \phi, \psi \rangle(\underline{s}) = \{ s \in S$: $\exists p \in \Sigma\langle S, t, \phi \rangle$, $i \in Dom(p)$. $p_0 = \underline{s} \wedge \forall j \leqslant i. \neg \psi(p_0, p_j) \wedge p_i = s \}$,

  $Inter\langle S, t, \phi, \psi \rangle = \bigcup \{ Inter\langle S, t, \phi, \psi \rangle(\underline{s})$: $\underline{s} \in S \}$,

- $Goal\langle S, t, \phi, \psi \rangle(\underline{s}) = \{ s \in S$: $\exists p \in \Sigma\langle S, t, \phi \rangle$, $i \in Dom(p)$. $p_0 = \underline{s} \wedge \forall j < i. \neg \psi(p_0, p_j) \wedge p_i = s \wedge \psi(p_0, p_i) \}$,

  $Goal\langle S, t, \phi, \psi \rangle = \bigcup \{ Goal\langle S, t, \phi, \psi \rangle(\underline{s})$: $\underline{s} \in S \}$,

- $Acc\langle S,t,\phi,\psi\rangle(\underline{s}) = Inter\langle S,t,\phi,\psi\rangle(\underline{s}) \cup Goal\langle S,t,\phi,\psi\rangle(\underline{s})$,

  $Acc\langle S,t,\phi,\psi\rangle = Inter\langle S,t,\phi,\psi\rangle \cup Goal\langle S,t,\phi,\psi\rangle$,
- $Isind\langle S,t,\phi,\psi\rangle = [Inter\langle S,t,\phi,\psi\rangle \cap Goal\langle S,t,\phi,\psi\rangle = \emptyset]$.

When this (sufficient but not necessary) initial states independence condition is satisfied, inevitability properties can be proved by (1) with $f \in (\Lambda \to (S \to \Delta))$ for some $\Delta \in Ord$.

Before proving this fact we must characterize the ordinal Δ which is necessary; stated otherwise, we propose a "measure" of the global nondeterminism of the program (as opposed to local characterizations of nondeterminism such as the so-called bounded nondeterminism [6]).

- We write $Rel(W, \prec)$ to state that $\prec$ is a relation on W represented by its characteristic function:

  $$Rel(W,\prec) = [W \times W \subseteq \mathrm{Dom}(\prec) \wedge Rng(\prec) = \{\mathrm{tt}, \mathrm{ff}\}].$$

- We write $Wf(W,\prec)$ to state that $\prec$ is a well-founded relation on W:

  $$Wf(W,\prec) = [Rel(W,\prec) \wedge \forall E \subseteq W.[E \neq 0 \Rightarrow \exists y \in E.(\forall z \in E.\neg(z \prec y))]]$$

  (this implies that there is no sequence $p \in (\omega \to W)$ such that $p_{i+1} \prec p_i$ for all $i \in \omega$. Assuming the axiom of choice, this property is equivalent to the above definition).
- The left restriction of relation t to E is written $t \restriction E$:

  $$t \restriction E(s,s') = [s \in E \wedge t(s,s')].$$

- t^{-1} is the inverse of relation t:

  $$t^{-1}(s',s) = t(s,s').$$

We first prove the following lemma.

Lemma 5.4 (Existence of a well-founded relation for inevitability proofs (with initial states independence hypothesis)).

$$[(0) \wedge Isind\langle S,t,\phi,\psi\rangle] \Rightarrow Wf(Acc\langle S,t,\phi,\psi\rangle, t \restriction Inter\langle S,t,\phi,\psi\rangle^{-1}).$$

Proof. Assume by reductio ad absurdum that $\exists p \in (\omega \to Acc\langle S,t,\phi,\psi\rangle).\forall i \in \omega.\ t \restriction Inter\langle S,t,\phi,\psi\rangle(p_i, p_{i+1})$. We can assume that $\phi(p_0)$ holds (else we can adjoin to the left of p a prefix $r_0 \ldots r_k$ of a trace of $\Sigma\langle Acc\langle S,t,\phi,\psi\rangle, t \restriction Inter\langle S,t,\phi,\psi\rangle, \phi\rangle$ such that $\phi(r_0)$ holds). By (0) there is a smallest $i \in Dom(p)$ such that $\psi(p_0, p_i)$ holds. Hence $p_i \in Goal\langle S,t,\phi,\psi\rangle$. Also $t \restriction Inter\langle S,t,\phi,\psi\rangle(p_i, p_{i+1})$ implies $p_i \in Inter\langle S,t,\phi,\psi\rangle$ in contradiction with $Isind\langle S,t,\phi,\psi\rangle$. □

Let E be a class of ordinals. $Sup(E) = \bigcup E$ will denote the least upper bound of E and $Sup^+(E)$ will denote the least strict upper bound of E.

The rank of $x \in W$ with respect to a well-founded relation $Wf(W, \prec)$ is an ordinal defined by transfinite recursion on W as follows:

$$rk(W, \prec)(x) = Sup^{+}\{rk(W, \prec)(y) : y \prec x\}.$$

The rank of a well-founded relation $Wf(W, \prec)$ is

$$rk(W, \prec) = Sup^{+}\{rk(W, \prec)(x) : x \in W\}.$$

The global nondeterminism of $\langle S, t, \phi \rangle$ with respect to ψ can be measured by the rank of the inverse of t left restricted to intermediate states:

Definition 5.5 (Rank of the global nondeterminism (with initial states independence hypothesis)). When (0) and $Isind\langle S, t, \phi, \psi \rangle$ hold, we define

$$rkgnd\langle S, t, \phi, \psi \rangle = rk(Acc\langle S, t, \phi, \psi \rangle, t \upharpoonright Inter\langle S, t, \phi, \psi \rangle^{-1}).$$

Remark. Observe that if the nondeterminism is locally bounded (i.e. $\forall s \in S.\ |\{s' : t(s, s')\}| < \omega$) then $rkgnd\langle S, t, \phi, \psi \rangle \leqslant \omega$. Similarly, if the nondeterminism is locally countable (i.e. $\forall s \in S.|\{s' : t(s, s')\}| \leqslant \omega$) then $rkgnd\langle S, t, \phi, \psi \rangle \leqslant \omega_1$. Finally, if t is recursive (i.e. effectively calculable) then $rkgnd\langle S, t, \phi, \psi \rangle \leqslant \omega_1^{CK}$ (where ω_1^{CK} is Church–Kleene's first nonrecursive ordinal [2]).

We can now state the partial completeness result concerning Burstall's method.

Theorem 5.6 (Partial semantic completeness).

$$[(0) \wedge Isind\langle S, t, \phi, \psi \rangle] \Rightarrow [(1) \text{ with } f \in (\Lambda \rightarrow (S \rightarrow rkgnd\langle S, t, \phi, \psi \rangle))].$$

Proof. Assume (0) and $Isind\langle S, t, \phi, \psi \rangle$. Let us choose $\Lambda = 2$, $\varepsilon_0(s) = [s \in Acc\langle S, t, \phi, \psi \rangle]$, $\theta_0(s, s') = \neg[\exists p \in \Sigma\langle S, t, \phi \rangle,\ i \in Dom(p).(\forall j < i. \neg\psi(p_0, p_j)) \wedge \psi(p_0, p_i) \wedge (\exists k \leqslant i. p_k = s) \wedge p_i = s']$, $f_0 \in (Acc\langle S, t, \phi, \psi \rangle \rightarrow rkgnd\langle S, t, \phi, \psi \rangle)$, $f_0(s) = rk(Acc\langle S, t, \phi, \psi \rangle, t \upharpoonright Inter\langle S, t, \phi, \psi \rangle^{-1})$, $n_0 = 2$, $I_0^2(s, s') = [\varepsilon_0(s) \wedge s' = s]$, $I_0^1(s, s') = [\varepsilon_0(s) \wedge \neg\theta_0(s, s) \wedge t(s, s')]$, $I_0^0 = \theta_0$, $\varepsilon_1 = \phi$, $\theta_1 = \psi$, $n_1 = 1$, $I_1^1(s, s') = [\varepsilon_1(s) \wedge s' = s]$, $I_1^0 = \theta_1$, $\pi = 1$. All verification conditions are obviously satisfied but for $\forall s, s' \in S.$ $(I_0^1(s, s') \wedge \neg\theta_0(s, s')) \Rightarrow (f_0(s') < f_0(s) \wedge \varepsilon_0(s') \wedge \forall s'' \in S.\theta_0(s', s'') \Rightarrow I_0^0(s, s''))$.

If $I_0^1(s, s') \wedge \neg\theta_0(s, s')$ holds, we have, by definition of I_0^1, ε_0 and (0), that $\exists p \in \Sigma\langle S, t, \phi \rangle, i \in Dom(p).(\forall j < i. \neg\psi(p_0, p_j)) \wedge \psi(p_0, p_i) \wedge \exists k.(s = p_k \wedge (k+1) < i \wedge s' = p_{k+1})$. Since $s, s' \in Inter\langle S, t, \phi, \psi \rangle$ and $t(s, s')$ holds, we have $f_0(s') < f_0(s) \wedge \varepsilon_0(s')$. If $\theta_0(s', s'')$ then $\exists q \in \Sigma\langle S, t, \phi \rangle$, $i' \in Dom(q)$. $(\forall j < i'. \neg\psi(p_0, p_j)) \wedge \psi(p_0, p_{i'}) \wedge (\exists k' \leqslant i'. q_{k'} = s') \wedge q_{i'} = s''$. We have $\forall j.(k' < j < i') \Rightarrow \neg\psi(p_0, q_j)$ since otherwise for the smallest j satisfying $k' < j < i' \wedge \psi(p_0, q_j)$, we would have $q_j \in Inter\langle S, t, \phi, \psi \rangle \cap Goal\langle S, t, \phi, \psi \rangle$. Observe that $q_{i'} \in Goal\langle S, t, \phi, \psi \rangle$ so that $\psi(p_0, q_{i'})$ holds since, otherwise, $q_{i'}$ would be an intermediate state of the trace $p_0 \ldots p_k$ $q_{k'} \ldots q_{i'} \ldots$. Since $s = p_k$ and $s'' = q_{i'}$ we conclude that $\theta_0(s, s'')$, hence $I_0^0(s, s'')$ hold. □

The above partial semantic completeness result applies to proofs of inevitability properties such that goal states have no successor state.

Theorem 5.7.

$$[(0) \wedge \forall s, s' \in S.(\psi(s, s') \Rightarrow \forall s'' \in S. \neg t(s', s''))] \Rightarrow Isind\langle S, t, \phi, \psi \rangle.$$

Proof. Assume that $s \in Goal\langle S, t, \phi, \psi \rangle$. We have $\forall s' \in S. \neg t(s, s')$. It follows that $s \notin Inter\langle S, t, \phi, \psi \rangle$ since otherwise there exists $n \in (\omega \sim 0)$, $p \in \Sigma^n\langle S, t, \phi \rangle$ such that $\forall i \in n. \neg\psi(p_0, p_i)$, in contradiction with (0). □

As corollary, we obtain that Burstall's [3] total correctness proof method for sequential programs (i.e. (1) with $f \in (\Lambda \rightarrow (S \rightarrow \omega))$) is semantically complete because program exit states have no successor states and only deterministic programs are considered.

Theorem 5.6 also applies to [11, 1] because they only consider unary intermittent assertions (i.e. relational intermittent assertions are expressed using auxiliary variables the value of which is part of the state).

Theorem 5.8.

$$\forall s, s' \in S.[\psi(s, s') \Rightarrow (\forall s'' \in S. \psi(s'', s'))] \Rightarrow Isind\langle S, t, \phi, \psi \rangle.$$

Proof. If $s \in Inter\langle S, t, \phi, \psi \rangle \cap Goal\langle S, t, \phi, \psi \rangle$, there are $s', s'' \in S$ such that $\neg\psi(s', s)$ and $\psi(s'', s)$, a contradiction. □

6. The basic induction principle generalizing Burstall's intermittent assertions method

Although induction principle (1) is sound and semantically complete in a great number of practical situations, we conjecture that it is not general enough to cope with some types of inevitability properties of programs, such as those considered in [10] for cyclic programs. Hence the necessity arises of generalizing induction principle (1).

The proposed generalization is quite simple. In order to ensure the existence of well-orderings to be used for induction, lemmas and intermittent assertions should depend upon initial states. Transfinite well-orderings should be used in order to cope with unbounded nondeterminism. These remarks lead from (1) to (2), the last induction principle (2) being later shown to be semantically complete.

$$[\exists \Lambda \in \omega,\ \varepsilon \in (\Lambda \rightarrow (S^2 \rightarrow \{\mathrm{tt}, \mathrm{ff}\})),\ \theta \in (\Lambda \rightarrow (S^3 \rightarrow \{\mathrm{tt}, \mathrm{ff}\})),$$

$$\Delta \in Ord,\ f \in (\Lambda \rightarrow (S^2 \rightarrow \Delta)),\ n \in (\Lambda \rightarrow \omega).$$

$$(\exists \pi \in \Lambda. \varepsilon_\pi = \lambda(\underline{s}, s).[s = \underline{s} \wedge \phi(\underline{s})] \wedge \theta_\pi = \lambda(\underline{s}, s, s').[\underline{s} = s \wedge \psi(s, s')]) \wedge$$

$$(\forall l \in \Lambda. \exists I_l \in (n_l + 1 \rightarrow (S^3 \rightarrow \{\mathrm{tt}, \mathrm{ff}\})).$$

$$\forall i \leqslant n_l,\ \underline{s}, s, s' \in S.$$

$$
\begin{aligned}
&(\mathrm{P}) \quad [\varepsilon_l(\underline{s}, s) \Rightarrow I_l^{n_l}(\underline{s}, s, s)] \wedge \qquad (2)\\
&\quad [I_l^i(\underline{s}, s, s') \Rightarrow\\
&(\mathrm{HS}) \quad (\exists s'' \in S.t(s', s'') \wedge \forall s'' \in S.[t(s', s'') \Rightarrow \exists j < i.I_l^j(\underline{s}, s, s'')]) \vee\\
&(\mathrm{LI}) \quad (\exists l' \in \Lambda.[((l' < l) \vee (l' = l \wedge f_l(\underline{s}, s') < f_l(\underline{s}, s))) \wedge \varepsilon_{l'}(\underline{s}, s') \wedge\\
&\qquad \forall s'' \in S.(\theta_{l'}(\underline{s}, s', s'') \Rightarrow \exists j < i.I_l^j(\underline{s}, s, s''))]) \vee\\
&(\mathrm{C}) \quad \theta_l(\underline{s}, s, s')])].
\end{aligned}
$$

In order to illustrate the use of this induction principle, let us consider the following example.

Example 6.1. $\psi(x, x') = [x' = 2x]$ is inevitable for $\langle \omega, t, \phi \rangle$ such that $t(x, x') = [x' = x + 1]$ and $\phi(x) = \mathrm{tt}$.

Observe that we do *not* have $Wf(Acc\langle S, t, \phi, \psi \rangle, t \restriction Inter\langle S, t, \phi, \psi \rangle^{-1})$.

The inevitability of ψ can be proved by induction principle (2) by choosing $\Lambda = 2$, $\pi = 1$, $\varepsilon_0(\underline{x}, x) = [\underline{x} \leqslant x \leqslant 2\underline{x}]$, $\theta_0(\underline{x}, x, x') = [\underline{x} \leqslant x \leqslant 2\underline{x} = x']$, $\Delta = \omega$, $f_0(\underline{x}, x) = [2\underline{x} - x]$, $\varepsilon_1(\underline{x}, x) = [\underline{x} = x]$, $\theta_1(\underline{x}, x, x') = [\underline{x} = x \wedge x' = 2x]$, $n_0 = 2$, $I_0^2(\underline{x}, x, x') = [\underline{x} \leqslant x = x' \leqslant 2\underline{x}]$ (satisfying (P) and (C) when $x' = 2\underline{x}$ or (HS) when $x' < 2x$), $I_0^1(\underline{x}, x, x') = [\underline{x} \leqslant x < x + 1 = x' \leqslant 2\underline{x}]$ (satisfying (LI) with $l' = l = 0$), $I_0^0 = \theta_0$ (C), $n_1 = 1$, $I_1^1(\underline{x}, x, x') = [\underline{x} = x = x']$ ((P), (LI) with $l' = 0$), $I_1^0 = \theta_1$ (C).

Induction principle (2) is an obvious generalization of (1).

Theorem 6.2 (Generalization of Burstall's method). (1) $\Rightarrow$ (2).

Before tackling the question of semantic completeness, we define which ordinals $\Delta \in Ord$ are sufficient in a proof by (2).

Definition 6.3 (*Rank of the global nondeterminism*). When (0) holds, we define

$$
\begin{aligned}
rkgnd\langle S, t, \phi, \psi \rangle = Sup^+ \{ & rk(Acc\langle S, t, \phi, \psi \rangle(\underline{s}),\\
& t \restriction Inter\langle S, t, \phi, \psi \rangle(\underline{s})^{-1}) : \underline{s} \in S \}.
\end{aligned}
$$

(This definition is justified by the fact that for all $\underline{s} \in S$, $t \restriction Inter\langle S, t, \phi, \psi \rangle(\underline{s})$ is well-founded on $Acc\langle S, t, \phi, \psi \rangle(\underline{s})$. This is proved in [4].)

The proof of semantic completeness of (2) follows from the remark that (2) can be used to express "à la Floyd" proofs.

Theorem 6.4 (Semantic completeness). (0) $\Rightarrow$ ((2), *with* $\Delta = rkgnd\langle S, t, \phi, \psi \rangle$).

Proof (hint). Choose $\Lambda = 2$, $\pi = 1$, $\varepsilon_1(\underline{s}, s) = [\underline{s} = s \wedge \phi(\underline{s})]$, $\theta_1(\underline{s}, s, s') = [\underline{s} = s \wedge \psi(s, s')]$, $\varepsilon_0(\underline{s}, s) = [s \in Acc\langle S, t, \phi, \psi \rangle(\underline{s})]$, $\theta_0(\underline{s}, s, s') = [\exists p \in \Sigma\langle S, t, \phi \rangle,\ i \in Dom(p).\forall j \in i. \neg \psi(p_0,$

$p_j) \wedge \psi(p_0, p_i) \wedge \underline{s} = p_0 \wedge \exists k \leqslant i . p_k = s \wedge p_i = s']$, f_1 is useless, $n_1 = 1$, $I_1^1(\underline{s}, s, s') = [\underline{s} = s = s' \wedge \phi(\underline{s})]$ (satisfies (P) and (LI) with $l' = 0$), $I_1^1 = \theta_1$, $n_0 = 2$, $I_0^2(\underline{s}, s, s') = [\varepsilon_0(\underline{s}, s) \wedge s' = s]$ (satisfies (P) and (C) or (HS)), $I_0^1(\underline{s}, s, s') = [\varepsilon_0(\underline{s}, s) \wedge \neg \theta_0(\underline{s}, s, s) \wedge t(s, s')]$ (satisfies (LI) with $l' = 0$ and $f_0(\underline{s}, s) = rk(Acc\langle S, t, \phi, \psi \rangle(\underline{s}), t \upharpoonright Inter\langle S, t, \phi, \psi \rangle(\underline{s})^{-1})(s)$ and $I_0^0 = \theta_0$ (satisfies (C)).

7. Equivalent induction principles generalizing Burstall's intermittent assertions method

We now derive a series of induction principles which are all shown to be sound and complete and hence equivalent to the basic induction principle (2). For the sake of conciseness, not all conceivable alternatives have been reported. One purpose of the series of induction principles is to propose more and more abstract formalizations that should lead to a better understanding of Burstall's method. The other purpose of the following proof principles is to broaden the allowed forms of proofs (so as to introduce more flexibility in writing proofs but no additional proof power since all principles are equivalent).

The number of lemmas $\langle \varepsilon_l, \theta_l \rangle$, $l \in \Lambda$ which can be used in induction principle (2) is finite. Hence an informal proposition such as

- if sometime $\underline{x} \leqslant X = x \leqslant 2\underline{x}$ then sometime $\underline{x} \leqslant x \leqslant 2\underline{x} = X$

has to be understood as a single lemma of name, say 0, such that $\varepsilon_0(\underline{x}, x) = [\underline{x} \leqslant x \leqslant 2\underline{x}]$ and $\theta_0(\underline{x}, x, x') = [\underline{x} \leqslant x \leqslant 2\underline{x} = x']$. Eliminating this restriction on names of lemmas, the above informal proposition can also be understood as a shorthand for an infinite number of lemmas of name $\underline{x}$ such that $\varepsilon_{\underline{x}}(x) = [\underline{x} \leqslant x \leqslant 2\underline{x}]$ and $\theta_{\underline{x}}(x, x') = [\underline{x} \leqslant x \leqslant 2\underline{x} = x']$. This point of view is consistent with the fact that the sole purpose of program initial state $\underline{s}$ in induction principle (2) is to offer the ability to use well-orderings for induction on the data that depend upon program initial states. These well-orderings can also be distinguished by giving them different names, one per program initial state. Also the main proposition $\langle \phi, \psi \rangle$ need not be the consequence of a single lemma $\langle \varepsilon_\pi, \theta_\pi \rangle$ as in (2) but could also be the consequence of different lemmas for different program initial states. These remarks lead to the following induction principle.

$$
\begin{aligned}
&[\exists \Lambda \in Ord,\ \varepsilon \in (\Lambda \to (S \to \{\mathrm{tt}, \mathrm{ff}\})),\ \theta \in (\Lambda \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})),\\
&\Delta \in Ord,\ f \in (\Lambda \to (S \to \Delta)),\ n \in (\Lambda \to \omega).\\
&\quad \forall s \in S . \exists \alpha \in \Lambda . (\varepsilon_\alpha(s) = \phi(s) \wedge \forall s' \in S . \theta_\alpha(s, s') = \psi(s, s')) \wedge\\
&\quad (\forall \alpha \in \Lambda . \exists I_\alpha \in (n_\alpha + 1 \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})).\\
&\quad \forall i \leqslant n_\alpha,\ s, s' \in S.\\
&\qquad [\varepsilon_\alpha(s) \Rightarrow I_\alpha^{n_\alpha}(s, s)] \wedge
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
&[I^i_\alpha(s,s') \Rightarrow \\
&\quad (\exists s'' \in S.t(s',s'') \wedge \forall s'' \in S.[t(s',s'') \Rightarrow \exists j<i.I^j_\alpha(s,s'')]) \vee \\
&\quad (\exists \alpha' \in \Lambda.[((\alpha'<\alpha) \vee (\alpha'=\alpha \wedge f_{\alpha'}(s')<f_\alpha(s))) \wedge \varepsilon_{\alpha'}(s') \wedge \\
&\qquad \forall s'' \in S.(\theta_{\alpha'}(s',s'') \Rightarrow \exists j<i.I^j_\alpha(s,s''))]) \vee \\
&\quad \theta_\alpha(s,s')])].
\end{aligned}
$$

Theorem 7.1. (2) $\Rightarrow$ (3).

Proof. The objects which are different in induction principles (2) and (3) but have been given the same name (such as $\Lambda, \varepsilon, \ldots$) will be referred to using indices 2 and 3 (such as $\Lambda_2, \Lambda_3, \varepsilon_2, \varepsilon_3, \ldots$) in the proof.

By the axiom of choice, there is an ordinal Σ and a one-to-one function δ that maps Σ into S. $\Sigma \times \Lambda_2$ well-ordered by the lexicographic ordering $\langle s',l' \rangle \prec \langle s,l \rangle$ if and only if $((s'<s) \vee (s'=s \wedge l'<l))$ is isomorphic with $\Lambda_2 \dot{\times} \Sigma$ ($\dot{\times}$ is ordinal multiplication) by the order isomorphism $\underline{\iota}=\lambda\langle s,l\rangle.[(\Lambda_2 \dot{\times} s)\dot{+}l]$, ($\dot{+}$ is ordinal addition). We let $\langle \underline{\sigma}, \underline{\lambda} \rangle$ be the inverse of $\underline{\iota}$ so that $\underline{\sigma} \in ((\Lambda_2 \dot{\times} \Sigma) \to \Sigma)$, $\underline{\lambda} \in ((\Lambda_2 \dot{\times} \Sigma) \to \Lambda_2)$ and $\forall \alpha \in (\Lambda_2 \dot{\times} \Sigma).[\alpha = \underline{\iota}(\langle \underline{\sigma}(\alpha), \underline{\lambda}(\alpha) \rangle)]$. We choose $\Lambda_3 = \Lambda_2 \dot{\times} \Sigma$, $\varepsilon_3 = \lambda\alpha.\lambda s.[\varepsilon_{2\underline{\lambda}(\alpha)}(\delta(\underline{\sigma}(\alpha)), s)]$, $\theta_3 = \lambda\alpha.\lambda(s,s').[\theta_{2\underline{\lambda}(\alpha)}(\delta(\underline{\sigma}(\alpha)), s, s')]$, $\Delta_3 = \Delta_2$, $f_3 = \lambda\alpha.\lambda s.[f_{2\underline{\lambda}(\alpha)}(\delta(\underline{\sigma}(\alpha)), s)]$, $n_3 = \lambda\alpha.n_{2\underline{\lambda}(\alpha)}$, $I^i_{3\alpha} = \lambda(s,s').I^i_{2\underline{\lambda}(\alpha)}(\delta(\underline{\sigma}(\alpha)), s, s')$. It follows that $\varepsilon_{3\underline{\iota}(\langle \delta^{-1}(s), \pi_2 \rangle)}(s) = \phi(s)$ and $\theta_{3\underline{\iota}(\langle \delta^{-1}(s), \pi_2 \rangle)}(s,s') = \psi(s,s')$. The other verification conditions are obvious to check. □

The names $\alpha \in \Lambda$ of the lemmas $\langle \varepsilon_\alpha, \theta_\alpha \rangle$ in (3) are well-ordered. For a given lemma $\langle \varepsilon_\alpha, \theta_\alpha \rangle$, the rôle of f_α is to introduce a well-ordering on the initial states of lemma $\langle \varepsilon_\alpha, \theta_\alpha \rangle$. The same effect can be obtained by considering not a single lemma $\langle \varepsilon_\alpha, \theta_\alpha \rangle$ but a family of lemmas $\{\langle \varepsilon_{\alpha, f_\alpha(s)}, \theta_{\alpha, f_\alpha(s)} \rangle : s \in S\}$. This point of view is more abstract in that only one well-ordering $(W, \prec)$ need to be used. It is defined by $\langle \alpha', f_{\alpha'}(s') \rangle \prec \langle \alpha, f_\alpha(s) \rangle$ if and only if $(\alpha'<\alpha \vee (\alpha'=\alpha \wedge f_{\alpha'}(s')<f_\alpha(s)))$ on $W = \{\langle \alpha, f_\alpha(s) \rangle : \alpha \in \Lambda \wedge s \in S\}$. Hence, up to an isomorphism we can use ordinals and rephrase induction principle (3) as follows:

$$
\begin{aligned}
&[\exists \Lambda \in Ord,\ \varepsilon \in (\Lambda \to (S \to \{\mathrm{tt}, \mathrm{ff}\})),\ \theta \in (\Lambda \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})),\ n \in (\Lambda \to \omega). \\
&\quad (\forall s \in S.\exists \alpha \in \Lambda.[(\varepsilon_\alpha(s) = \phi(s)) \wedge \forall s' \in S.(\theta_\alpha(s,s') = \psi(s,s'))]) \wedge \\
&\quad (\forall \alpha \in \Lambda.\exists I_\alpha \in (n_\alpha + 1 \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})). \\
&\quad \forall i \leqslant n_\alpha,\ s,s' \in S. \\
&\qquad [\varepsilon_\alpha(s) \Rightarrow I^{n_\alpha}_\alpha(s,s)] \wedge \qquad (4) \\
&\qquad [I^i_\alpha(s,s') \Rightarrow \\
&\qquad\quad (\exists s'' \in S.t(s',s'') \wedge \forall s'' \in S.[t(s',s'') \Rightarrow \exists j<i.I^j_\alpha(s,s'')]) \vee \\
&\qquad\quad (\exists \alpha' < \alpha.[\varepsilon_{\alpha'}(s') \wedge \forall s'' \in S.(\theta_{\alpha'}(s',s'') \Rightarrow \exists j<i.I^j_\alpha(s,s''))]) \vee \\
&\qquad\quad \theta_\alpha(s,s')])].
\end{aligned}
$$

Theorem 7.2. (3) $\Rightarrow$ (4).

Proof. $\langle \Lambda_3, \Lambda_3 \rangle$ well-ordered by the left lexicographic ordering is isomorphic with $\Lambda_3 \dot{\times} \Lambda_3$ by the order-isomorphism $\underline{\iota}(\alpha, \kappa) = [\Lambda_3 \dot{\times} \alpha \dot{+} \kappa]$ the inverse of which is $\langle \underline{\delta}, \underline{\lambda} \rangle$. We choose $\Lambda_4 = \Lambda_3 \dot{\times} \Lambda_3$, $\varepsilon_4 = \lambda\alpha.[\lambda s.[\varepsilon_{3\underline{\lambda}(\alpha)}(s) \wedge f_{3\underline{\lambda}(\alpha)}(s) = \underline{\delta}(\alpha)]]$, $\theta_4 = \lambda\alpha.[\lambda(s, s').[\theta_{3\underline{\lambda}(\alpha)}(s, s') \wedge f_{3\underline{\lambda}(\alpha)}(s) = \underline{\delta}(\alpha)]]$, $n_4 = \lambda\alpha.n_{3\underline{\lambda}(\alpha)}$, $I_{4\alpha}^{i} = \lambda(s, s').[I_{3\underline{\lambda}(\alpha)}^{i}(s, s') \wedge f_{3\underline{\lambda}(\alpha)}(s) = \underline{\delta}(\alpha)]$. □

Inevitability properties of programs have been specified as pairs $\langle \phi, \psi \rangle$ where ϕ is a condition on initial states and ψ a relationship between initial and final states so as to adhere to the method introduced by Burstall [3]. However, a single binary relation is enough because ψ is inevitable for $\langle S, t, \phi \rangle$ if and only if $\lambda(s, s').[\phi(s) \Rightarrow \psi(s, s')]$ is inevitable for $\langle S, t, \lambda s.\mathrm{tt} \rangle$. Hence we derive the following more abstract induction principle:

$$
\begin{aligned}
&[\exists \Lambda \in Ord,\ \theta \in (\Lambda \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})),\ n \in (\Lambda \to \omega).\\
&\quad ((\forall s \in S.\exists \alpha \in \Lambda.\forall s' \in S.[\theta_\alpha(s, s') = (\phi(s) \Rightarrow \psi(s, s'))]) \wedge\\
&\quad (\forall \alpha \in \Lambda.\exists I_\alpha \in (n_\alpha + 1 \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})).\\
&\quad \forall i \leqslant n_\alpha,\ s, s' \in S.\\
&\text{(P)} \qquad I_\alpha^{n_\alpha}(s, s) \wedge \qquad (5)\\
&\qquad [I_\alpha^{i}(s, s') \Rightarrow\\
&\text{(HS)} \qquad (\exists s'' \in S.t(s', s'') \wedge \forall s'' \in S.[t(s', s'') \Rightarrow \exists j < i.I_\alpha^{j}(s, s'')]) \vee\\
&\text{(LI)} \qquad (\exists \alpha' < \alpha.\forall s'' \in S.[\theta_{\alpha'}(s', s'') \Rightarrow \exists j < i.I_\alpha^{j}(s, s'')]) \vee\\
&\text{(C)} \qquad \theta_\alpha(s, s')])].
\end{aligned}
$$

Theorem 7.3. (4) $\Rightarrow$ (5).

Proof. Choose $\Lambda_5 = \Lambda_4$, $\theta_{5\alpha}(s, s') = [\varepsilon_{4\alpha}(s) \Rightarrow \theta_{4\alpha}(s, s')]$, $n_5 = n_4$ and $I_{5\alpha}^{i}(s, s') = [\varepsilon_{4\alpha}(s) \Rightarrow I_{4\alpha}^{i}(s, s')]$. □

If in induction principle (5) we consider the inevitability proof of a given lemma θ_α and this proof can be obtained without (LI) then the verification conditions (P), (HS) and (C) strongly resemble the verification conditions corresponding to Floyd's proof method [7] as formalized by induction principle (5) of Cousot and Cousot [4]. Stated otherwise, in invariant $I_\alpha^{i}(s, s')$, i plays the rôle of the nonnegative integer which is strictly decremented at each program step. By comparison with Floyd's method we observe that (5) imposes two unnecessary restrictions on i: n_α is a bound on the number of program steps and this number is independent of the considered initial state, n_α

(hence i) should be an integer (so that e.g. unbounded nondeterminism cannot be handled without (LI)).

We first relax the first limitation, choosing n_α as ordinal.

Theorem 7.4. (a) $(i) \Rightarrow (i \text{ with } n \in (\Lambda \to Ord))$, $i = 2, \ldots, 5$,
(b) $(i \text{ with } n \in (\Lambda \to Ord) \Rightarrow (i+1 \text{ with } n \in (\Lambda \to Ord))$, $i = 2, 3, 4$.

Proof. (a) is obvious because $\omega \in Ord$, hence $\omega \subseteq Ord$. (b) follows from the proofs of Theorems 7.1–7.3 which never use the fact that $n_l \in \omega$ but only that $(n_l + 1, <)$ is well-founded (this remains valid when $n_l \in Ord$ and $n_l + 1$ is the ordinal successor of n_l). □

We next relax the second limitation, choosing a possibly different maximum "number of program steps" for each initial state $\underline{s}$ (the "number of program steps" should not be understood to the letter but as $rk(Acc\langle S, t, \phi, \psi\rangle(\underline{s}), t^{-1} Inter\langle S, t, \phi, \psi\rangle(\underline{s})^{-1})(\underline{s})$):

$$
\begin{aligned}
&[\exists \Lambda \in Ord, \theta \in (\Lambda \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})), \Delta \in Ord, I \in (\Lambda \times \Delta \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})).\\
&\quad (\forall s \in S. \exists \pi \in \Lambda. \forall s' \in S. [\theta_\pi(s, s') = (\phi(s) \Rightarrow \psi(s, s'))]) \wedge\\
&\quad (\forall \alpha \in \Lambda,\ s, s' \in S,\ \delta' \in \Delta.\\
&\qquad [\exists \delta \in \Delta. I_\alpha^\delta(s, s)] \wedge \qquad (6)\\
&\qquad [I_\alpha^{\delta'}(s, s') \Rightarrow\\
&\qquad\quad (\exists s'' \in S. t(s', s'') \wedge \forall s'' \in S. [t(s', s'') \Rightarrow (\exists \delta'' < \delta'. I_\alpha^{\delta''}(s, s''))]) \vee\\
&\qquad\quad (\exists \alpha' < \alpha. \forall s'' \in S. [\theta_{\alpha'}(s', s'') \Rightarrow (\exists \delta'' < \delta'. I_\alpha^{\delta''}(s, s''))]) \vee\\
&\qquad\quad \theta_\alpha(s, s')])].
\end{aligned}
$$

Theorem 7.5. ((5) with $n \in (\Lambda \to Ord)) \Rightarrow (6)$.

Proof. Choose $\Lambda_6 = \Lambda_5$, $\theta_6 = \theta_5$, $\Delta_6 = \omega$, $I_{6\alpha}^{\delta}(s, s') = [\delta \leqslant n_{5\alpha} \wedge I_{5\alpha}^{\delta}(s, s')]$. □

The use of well-orderings (or up to order-isomorphisms of ordinals) in (6) is not mandatory. Well-founded relations can as well serve as a basis for induction.

Also as observed by Schwarz [12], Burstall's method can be explained as the mathematical deduction of theorems from axioms specifying the effect of elementary commands in the program. This informal explanation of Burstall's method can be formalized by considering the transition relation in the previous proof principles as a set of axioms or a given lemma from which other lemmas are derived. One difference (that had not to be taken into account by Schwarz [12], who considers only total deterministic programs) is that inevitability of t for $\langle S, t, \lambda s.\mathrm{tt}\rangle$ holds only for states

which have at least one successor. Moreover, the deduction process that Schwarz [12] left unspecified is always reducible to transfinite induction.

Finally, the main proposition $\lambda(s,s').[\phi(s) \Rightarrow \psi(s,s')]$ can always be chosen as one of the lemmas intervening in the proof.

Therefore, if we write $Wfi(W, \prec, \mu)$ to state that $\prec$ is a well-founded relation on W with minimal element μ, i.e.

$$Wfi(W, \prec, \mu) = Wf(W, \prec) \wedge \mu \in W \wedge \forall x \in W. \neg(x \prec \mu)$$

The above remarks lead to the following induction principle:

$$\begin{aligned}
&[\exists \Lambda, \prec, \mu \in \Lambda, \pi \in (\Lambda \sim \mu), \Delta, <, \theta \in (\Lambda \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})),\\
&I \in (\Lambda \times \Delta \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})).\\
&\quad Wfi(\Lambda, \prec, \mu) \wedge \theta_\mu = t \wedge Wf(\Delta, <) \wedge \theta_\pi = \lambda(s,s').[\phi(s) \Rightarrow \psi(s,s')] \wedge\\
&\quad (\forall \alpha \in (\Lambda \sim \mu), s, s' \in S, \delta' \in \Delta.\\
&\qquad [\exists \delta \in \Delta. I_\alpha^\delta(s,s)] \wedge \qquad (7)\\
&\qquad [I_\alpha^{\delta'}(s,s') \Rightarrow\\
&\qquad\quad (\exists \alpha' \in \Lambda.[\alpha' \prec \alpha \wedge ([\alpha' = \mu] \Rightarrow [\exists s'' \in S. \theta_{\alpha'}(s',s'')])\\
&\qquad\qquad \wedge \forall s'' \in S.(\theta_{\alpha'}(s',s'') \Rightarrow [\exists \delta'' \in \Delta.(\delta'' < \delta' \wedge I_\alpha^{\delta''}(s,s''))])] \vee\\
&\qquad\qquad \theta_\alpha(s,s')])].
\end{aligned}$$

Let us remark that condition $[\alpha' = \mu]$ under which $[\exists s'' \in S. \theta_\alpha(s',s'')]$ should hold is optional. When absent, the verification condition is simply redundant when $\alpha' \neq \mu$.

Theorem 7.6. (6) $\Rightarrow$ (7).

Proof. Choose $\mu \notin (\Lambda_6 \dot{+} 1)$ (e.g. $\Lambda_6 \dot{+} 2$) and $\Lambda_7 = (\Lambda_6 \dot{+} 1) \cup \{\mu\}$, $\alpha' \prec_7 \alpha$ if and only if $[\alpha \in (\Lambda_6 \dot{+} 1) \wedge ((\alpha' = \mu) \vee (\alpha' \in (\Lambda_6 \dot{+} 1) \wedge \alpha' < \alpha))]$, $\pi_7 = \Lambda_6$, $\Delta_7 = (\Delta_6 \cup 2)$, $<_7 = <_6 = <$, $\theta_{7\alpha}(s,s') = [(\alpha = \mu \wedge t(s,s')) \vee (\alpha = \Lambda_6 \wedge (\phi(s) \Rightarrow \psi(s,s'))) \vee (\alpha < \Lambda_6 \wedge \theta_{6\alpha}(s,s'))]$, $I_{7\alpha}^\delta(s,s') = [(\alpha = \mu) \vee (\alpha = \Lambda_6 \wedge \delta = 1 \wedge s' = s) \vee (\alpha = \Lambda_6 \wedge \delta = 0 \wedge (\phi(s) \Rightarrow \psi(s,s'))) \vee (\alpha < \Lambda_6 \wedge I_{6\alpha}^\delta(s,s'))]$. □

In induction principle (7) the verification condition $[\exists \delta \in \Delta. I_\alpha^\delta(s,s)]$ implies that lemma θ_α is inevitable for $\langle S, t, \lambda s.\mathrm{tt} \rangle$. But for the main proposition θ_π this property is not necessary. We need only the fact that θ_α should be inevitable for the particular states for which it is used. Hence the verification condition $[\exists \delta \in \Delta. I_\alpha^\delta(s,s)]$ of (7)

can be weakened in:

$$
\begin{aligned}
&[\exists \Lambda \in Ord,\ \theta \in (\Lambda \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})), \pi \in \Lambda,\ \Delta \in Ord,\ I \in (\Lambda \times \Delta \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})).\\
&\quad \theta_\pi = \dot{\lambda}(s,s').[\phi(s) \Rightarrow \psi(s,s')] \wedge\\
&\quad (\forall s \in S.\exists \delta \in \Delta.I_\pi^\delta(s,s)) \wedge\\
&\quad (\forall \alpha \in \Lambda,\ s,s' \in S,\ \delta' \in \Delta.\\
&\qquad [I_\alpha^{\delta'}(s,s') \Rightarrow \qquad (8)\\
&\qquad\quad (\exists s'' \in S.t(s',s'') \wedge \forall s'' \in S.[t(s',s'') \Rightarrow \exists \delta'' < \delta'.I_\alpha^{\delta''}(s,s'')]) \vee\\
&\qquad\quad (\exists \alpha' < \alpha.[\exists \delta \in \Delta.I_{\alpha'}^{\delta}(s',s') \wedge \forall s'' \in S.[\theta_{\alpha'}(s',s'')\\
&\qquad\qquad \Rightarrow \exists \delta'' < \delta'.I_\alpha^{\delta''}(s,s'')]]) \vee\\
&\qquad\quad \theta_\alpha(s,s')])].
\end{aligned}
$$

Theorem 7.7. (7) $\Rightarrow$ (8).

Proof. We first show that if $0<\varepsilon_0<\gamma$ and $0<\varepsilon_1<\gamma$ then $(\gamma \dot{\times} \delta_0)\dot{+}\varepsilon_0<(\gamma \dot{\times} \delta_1)\dot{+}\varepsilon_1$ if and only if $((\delta_0<\delta_1) \vee (\delta_0=\delta_1 \wedge \varepsilon_0<\varepsilon_1))$.

If $\delta_0<\delta_1$ then $(\gamma \dot{\times} \delta_0)\dot{+}\varepsilon_0<(\gamma \dot{\times} \delta_0)\dot{+}\gamma=\gamma \dot{\times}(\delta_0\dot{+}1)\leqslant\gamma \dot{\times} \delta_1<(\gamma \dot{\times} \delta_1)\dot{+}\varepsilon_1$. If $\delta_0=\delta_1 \wedge \varepsilon_0<\varepsilon_1$ then $\gamma \dot{\times} \delta_0=\gamma \dot{\times} \delta_1$, hence $(\gamma \dot{\times} \delta_0)\dot{+}\varepsilon_0<(\gamma \dot{\times} \delta_1)\dot{+}\varepsilon_1$.

If conversely, $\neg((\delta_0<\delta_1) \vee (\delta_0=\delta_1 \wedge \varepsilon_0<\varepsilon_1))$ then either $\delta_0=\delta_1$ and $\varepsilon_0=\varepsilon_1$ so that $\alpha=(\gamma \dot{\times} \delta_0)\dot{+}\varepsilon_0=(\gamma \dot{\times} \delta_1)\dot{+}\varepsilon_1=\beta$ and $\alpha \not< \beta$ or else $(\delta_0>\delta_1) \vee (\delta_0=\delta_1 \wedge \varepsilon_0>\varepsilon_1)$ so that by the first part of the proof (with 0 and 1 interchanged) we have $\beta<\alpha$, hence $\alpha \not< \beta$.

We next show that given a well-founded relation $\prec$ on W, there is an injective and order-preserving map $\iota(W,\prec)$ of W into the class $(Ord, <)$ of ordinals.

Let $E(W,\prec) \in (rk(W,\prec) \to \{X\colon X \subseteq W\})$ be defined by $E(W,\prec)(\alpha)=\{x \in W\colon rk(W,\prec)(x)=\alpha\}$. Observe that $\forall \alpha,\alpha' \in rk(W,\prec).[\alpha \neq \alpha' \Rightarrow E(W,\prec)(\alpha) \cap E(W,\prec)(\alpha')=\phi]$ and $\forall x \in W.\exists \alpha \in rk(W,\prec).[x \in E(W,\prec)(\alpha)]$.

By the axiom of choice, there is a linear ordering $\ll(W,\prec)(\alpha)$ which well-orders $E(W,\prec)(\alpha)$.

Define $\rho(W,\prec)(x,y)=rk[E(W,\prec)(rx), \ll(W,\prec)(rx)][y]$ where rx is $rk(W,\prec)(x)$ and $\varepsilon(W,\prec)(x)=\rho(W,\prec)(x,x)\dot{+}1$ so that $\forall x \in W.[0<\varepsilon(W,\prec)(x)]$. Define $\gamma(W,\prec)=Sup^+\{\varepsilon(W,\prec)(x)\colon x \in W\}$ so that $\forall x \in W.[\varepsilon(W,\prec)(x) \prec \gamma(W,\prec)]$ and $\iota(W,\prec)(x)=\gamma(W,\prec) \dot{\times} rk(W,\prec)(x)\dot{+}\varepsilon(W,\prec)(x)$.

If $x \prec y$ then $rk(W,\prec)(x)<rk(W,\prec)(y)$; hence by the lemma $\iota(W,\prec)(x) \prec \iota(W,\prec)(y)$. If $\iota x=\iota(W,\prec)(x)=\iota(W,\prec)(y)=\iota y$ then $\iota x \not< \iota y$ and $\iota y \not< \iota x$, so that by the lemma $rk(W,\prec)(x)=rk(W,\prec)(y)$ and $\varepsilon(W,\prec)(x)=\varepsilon(W,\prec)(y)$; therefore $\rho(W,\prec)(x,x)=\rho(W,\prec)(x,y)$. This implies that neither $x \ll(W,\prec)(rx)y$ nor $y \ll(W,\prec)(rx)\,x$ holds and, since $x,y \in E(W,\prec)(rx)$ which is linearly ordered by $\ll(W,\prec)(rx)$, we conclude that $x=y$.

To prove that (7)$\Rightarrow$(8) is now immediate when choosing $\Lambda_8 = Sup^+\{\iota(\Lambda_7 \sim \mu, \prec_7)(x)\colon x \in (\Lambda_7 \sim \mu)\}$, $\pi_8 = \iota(\Lambda_7 \sim \mu, \prec_7)(\pi_7)$, $\theta_{8\alpha}(s,s') = [\exists a \in (\Lambda_7 \sim \mu).\ (\alpha = \iota(\Lambda_7 \sim \mu, \prec_7)(a) \wedge \theta_{7a}(s,s'))]$, $\Delta_8 = Sup^+\{\iota(\Delta_7, <_7)(x)\colon x \in \Delta_7\}$ and $I_{8\alpha}^{\delta}(s,s') = [\exists a \in (\Lambda_7 \sim \mu), d \in \Delta_7.\ \alpha = \iota(\Lambda_7 \sim \mu, \prec_7)(a) \wedge \delta = \iota(\Delta_7, <_7)(d) \wedge I_{7a}^{d}(s,s')]$. □

The use of lemmas θ_α in induction principle (8) is redundant because we can use instead some intermittent assertion I_α^δ for some δ such that $I_\alpha^\delta(s,s') \Rightarrow \theta_\alpha(s,s')$. By convention, we can choose $\delta = 0$ so that induction principle (8) can be simplified as follows:

$$
\begin{aligned}
&[\exists \Lambda \in Ord,\ \pi \in \Lambda,\ \Delta \in Ord,\ I \in (\Lambda \times \Delta \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})).\\
&\quad (\forall s, s' \in S.\, I_\pi^0(s,s') = (\phi(s) \Rightarrow \psi(s,s'))) \wedge\\
&\quad (\forall s \in S.\, \exists \delta \in \Delta.\, I_\pi^\delta(s,s)) \wedge\\
&\quad (\forall \alpha \in \Lambda,\ s, s' \in S,\ \delta' \in (\Delta \sim 0).\\
&\qquad [I_\alpha^{\delta'}(s,s') \Rightarrow\\
&\qquad\quad (\exists s'' \in S.\, t(s',s'') \wedge \forall s'' \in S.\, [t(s',s'') \Rightarrow \exists \delta'' < \delta'.\, I_\alpha^{\delta''}(s,s'')]) \vee\\
&\qquad\quad (\exists \alpha' < \alpha.\, [\exists \delta \in \Delta.\, I_{\alpha'}^{\delta}(s',s') \wedge \forall s'' \in S.\, [I_{\alpha'}^{0}(s',s'')\\
&\qquad\qquad \Rightarrow \exists \delta'' < \delta'.\, I_\alpha^{\delta''}(s,s'')]])])].
\end{aligned}
\tag{9}
$$

Theorem 7.8. (8) $\Rightarrow$ (9).

Proof. Choose $\Lambda_9 = \Lambda_8$, $\pi_9 = \pi_8$, $\Delta_9 = \Delta_8$, $I_{9\alpha}^{\delta}(s,s') = [(\delta = 0 \wedge \theta_{8\alpha}(s,s')) \vee (\delta > 0 \wedge I_{8\alpha}^{\delta}(s,s'))]$. □

As shown by the succession of transformations, the proof that in (9) a state s' satisfying $I_\alpha^{\delta'}(s,s')$ inevitably leads to a state s'' such that $\theta_\alpha(s,s'')$ holds involve an induction along parts of computation paths modeled by δ' and an induction upon the data modeled by α. In order to make a comparison with Floyd's method, both cases can be reduced to computational induction using γ' measuring the "number of steps" remaining to be done between s' and s'':

$$
\begin{aligned}
&[\exists \Gamma \in Ord,\ I \in (\Gamma \to (S^2 \to \{\mathrm{tt}, \mathrm{ff}\})),\ \sigma \in (\Gamma \to \Gamma).\\
&\text{(P)}\quad (\forall s \in S.\, \exists \gamma \in \Gamma.\, [I_\gamma(s,s) \wedge \forall s' \in S.\, (I_{\sigma(\gamma)}(s,s') = [\phi(s) \Rightarrow \psi(s,s')]]) \wedge\\
&\qquad (\forall \gamma' \in \Gamma,\ s, s' \in S.\\
&\qquad\quad [I_{\gamma'}(s,s') \Rightarrow\\
&\text{(HS)}\qquad (\exists s'' \in S.\, t(s',s'') \wedge \forall s'' \in S.\, [t(s',s'') \Rightarrow \exists \gamma'' < \gamma'.\, (\sigma(\gamma'') = \sigma(\gamma')\\
&\qquad\qquad \wedge I_{\gamma''}(s,s''))]) \vee
\end{aligned}
\tag{10}
$$

(LI) $(\exists \gamma < \gamma'.[I_\gamma(s', s') \wedge \forall s'' \in S.[I_{\sigma(\gamma)}(s', s'') \Rightarrow \exists \gamma'' < \gamma'.(\sigma(\gamma'') = \sigma(\gamma') \wedge I_{\gamma''}(s, s''))]) \vee$

(C) $I_{\sigma(\gamma')}(s, s')]]$.

Theorem 7.9. (9) $\Rightarrow$ (10).

Proof. Let $\underline{\iota} = \lambda\langle\alpha, \delta\rangle.[(\Delta_9 \dot{\times} \alpha) \dot{+} \delta]$ be the order isomorphism between $\Delta_9 \times \Delta_9$ well-ordered by the left lexicographic ordering $\langle\alpha', \delta'\rangle \prec \langle\alpha, \delta\rangle$ if and only if $((\alpha' < \alpha) \vee (\alpha' = \alpha \wedge \delta' < \delta))$ and $\Gamma_{10} = \Delta_9 \dot{\times} \Delta_9$ well-ordered by $<$. We let $\langle \underline{\alpha}, \underline{\delta} \rangle$ be the inverse of $\underline{\iota}$ so that $\forall \alpha \in \Delta_9, \delta \in \Delta_9.(\alpha = \underline{\alpha}(\underline{\iota}(\langle\alpha, \delta\rangle)) \wedge \delta = \underline{\delta}(\underline{\iota}(\langle\alpha, \delta\rangle)))$ and $\forall \gamma \in \Gamma_{10}.\gamma = \underline{\iota}(\langle \underline{\alpha}(\gamma), \underline{\delta}(\gamma)\rangle)$. We choose $I_{10\gamma}(s, s') = I_{9\underline{\alpha}(\gamma)}^{\underline{\delta}(\gamma)}(s, s')$ and $\sigma(\gamma) = \underline{\iota}(\langle \underline{\alpha}(\gamma), 0\rangle)$. □

Using abstract generalization (10) of Burstall's method we can make a fair comparison with similar generalization of Floyd's method [4]. For Floyd's method, line (LI) is suppressed (so that one can always choose $\sigma(\gamma) = 0$). Hence the crucial difference between Floyd's and Burstall's methods is not the use of invariant versus intermittent assertions, nor the use of computational induction versus induction upon the data but, indeed, the introduction of recursion.

Equivalence of induction principles (2)–(10) follows from the following theorem.

Theorem 7.10 (soundness). (10) $\Rightarrow$ (0).

Proof. We prove by induction on $(\Gamma, <)$ that $\forall \gamma \in \Gamma.[\forall s \in S.\forall p \in \Sigma\langle S, t, \lambda s'.I_\gamma(s, s')\rangle.\ \exists i \in Dom(p).I_{\sigma(\gamma)}(s, p_i)]$. Assume this holds for $\gamma' < \gamma$. By reductio ad absurdum let $s \in S, p \in \Sigma\langle S, t, \lambda s'.I_\gamma(s, s')\rangle$ be such that $\forall i \in Dom(p).\neg I_{\sigma(\gamma)}(s, p_i)$. To get a contradiction we build an infinite sequence $\langle\langle i_k, \gamma_k\rangle : k \geqslant 0\rangle$ such that $\forall k \geqslant 0.[I_{\gamma_k}(s, p_{i_k}) \wedge \sigma(\gamma_k) = \sigma(\gamma) \wedge \gamma_k > \gamma_{k+1}]$. Choose $\gamma_0 = \gamma$ and $i_0 = 0$. If the sequence is built up to point k then $I_{\gamma_k}(s, p_{i_k})$ satisfies (HS), (LI) or (C). (C) is impossible (since $I_{\sigma(\gamma k)}(s, p_{i_k})$ would imply $I_{\sigma(\gamma)}(s, p_{i_k})$). In case (HS), $\exists s'' \in S.t(p_{i_k}, s'')$ implies that $i_{k+1} = (i_k + 1) \in Dom(p)$. Hence $t(p_{i_k}, p_{i_{k+1}})$ implies $\exists \gamma_{k+1} < \gamma_k.(\sigma(\gamma_{k+1}) = \sigma(\gamma_k) = \sigma(\gamma) \wedge I_{\gamma_{k+1}}(s, p_{i_{k+1}}))$. In case (LI) there exists $\gamma' < \gamma_k$ such that $I_{\gamma'}(p_{i_k}, p_{i_k})$. Hence by induction hypothesis $\exists j \in Dom(p^{+i_k}).I_{\sigma(\gamma')}(p_0^{+i_k}, p_j^{+i_k})$ (where p^{+j} is the subsequence $p_j\ p_{j+1} \ldots$ of p). If we let i_{k+1} be $i_k + j$ it follows that $I_{\sigma(\gamma')}(p_{i_k}, p_{i_{k+1}})$ holds whence $\exists \gamma_{k+1} < \gamma_k.[I_{\gamma_{k+1}}(s, p_{i_{k+1}}) \wedge \sigma(\gamma_{k+1}) = \sigma(\gamma_k) = \sigma(\gamma)]$. Q.E.D.

Now if $p \in \Sigma\langle S, t, \phi\rangle$ then $\exists \gamma \in \Gamma.I_\gamma(p_0, p_0)$ so that $p \in \Sigma\langle S, t, \lambda s'.I_\gamma(p_0, s')\rangle$ and by the above lemma $\exists i \in Dom(p).I_{\sigma(\gamma)}(p_0, p_i)$. By (P) this implies $\phi(p_0) \Rightarrow \psi(p_0, p_i)$, hence $\psi(p_0, p_i)$. □

8. Strong semantic completeness

The semantic completeness argument given in Theorem 6.4 is very weak because it essentially consists in saying that (2) can always be used to formulate "à la Floyd"

proofs (as suggested by Manna and Waldinger [10]). Having extended Burstall's method so as to incorporate Floyd's method (see Theorem 7.4 and (6)–(10)) the usual semantic completeness argument for Floyd's method can be transcribed for Burstall's method (e.g. (0) $\Rightarrow$ ((10) with (LI) suppressed (and $\sigma(\gamma)=0$)) as proved in [4]). However such completeness arguments are not in the spirit of Burstall [3], who encourages the decomposition of proofs of propositions into lemmas as opposed to Floyd [7], who proves a single proposition (decomposed into partial correctness, absence of blocking states and termination, a decomposition which is also applicable to each lemma involved in Burstall's method).

We now give a stronger semantic completeness result showing that the lemmas involved in "à la Burstall" proofs can always be chosen more freely.

First we have to introduce an induction principle (11) where the choice between "hand simulation" (HS) and "a little induction" (LI) is enforced. In particular, the lemmas that are to be used in (LI) should be imposed. For that purpose we consider a version of (6) where we introduce a choice relation $\iota_\alpha(s,s',\alpha')$ so that intermittent assertion $I_\alpha^{\delta'}(s,s')$ can be handled using lemma $\alpha'<\alpha$ if and only if $\iota_\alpha(s,s',\alpha')$ holds. (Observe that a dependence of ι on δ' would only be useful to impose the use of identity lemmas, a case of little importance that we exclude for simplicity).

To simplify later reasonings, (HS) will be treated in the style of (7) as a particular subcase of (LI) so that the transition relation t is viewed as a particular lemma, say θ_0, given as axiom.

Moreover, as observed for (8), the verification condition $[\exists\delta\in\Delta.I_\alpha^\delta(s,s)]$ of (6) or (7) implies that lemma θ_α is inevitable for $\langle S,t,\lambda s.\mathrm{tt}\rangle$. However, this is needed only for the particular states s for which θ_α may be used, i.e. when $\exists\alpha'<\alpha,\ s'\in S.\iota_{\alpha'}(s',s,\alpha)$.

Finally, since all lemmas enjoy the same kind of inevitability properties there is no real need to distinguish a particular main proposition.

These remarks lead to the following induction principle (where $\Lambda\in Ord$, $\theta\in(\Lambda\to(S^2\to\{\mathrm{tt},\mathrm{ff}\}))$, $\theta_0=t$ and $\iota\in(\Lambda\sim 0\to(S\times S\times\Lambda\to\{\mathrm{tt},\mathrm{ff}\}))$:

$$
\begin{aligned}
&[\exists\Delta\in Ord,\ I\in((\Lambda\sim 0)\times\Delta\to(S^2\to\{\mathrm{tt},\mathrm{ff}\})).\\
&\quad(\forall\alpha\in(\Lambda\sim 0),\ s\in S.[(\exists\alpha'\in(\Lambda\sim 0),\ s'\in S.\iota_{\alpha'}(s',s,\alpha))\Rightarrow\exists\delta\in\Delta.I_\alpha^\delta(s,s)])\wedge\\
&\quad(\forall\alpha\in(\Lambda\sim 0),\ s,s'\in S,\ \delta'\in\Delta.\\
&\quad[I_\alpha^{\delta'}(s,s')\Rightarrow\\
&\qquad([\exists\alpha'<\alpha.\iota_\alpha(s,s',\alpha')]\wedge\\
&\qquad[\forall\alpha'<\alpha.\iota_\alpha(s,s',\alpha')\Rightarrow\\
&\qquad\quad([\alpha'=0\Rightarrow\exists s''\in S.\theta_{\alpha'}(s',s'')]\wedge\\
&\qquad\qquad[\forall s''\in S.(\theta_{\alpha'}(s',s'')\Rightarrow\exists\delta''<\delta'.I_\alpha^{\delta''}(s,s''))]])\vee\\
&\qquad\theta_\alpha(s,s')]])].
\end{aligned}
\tag{11}
$$

We first show that (11) is yet another formulation of the induction principles generalizing Burstall's method.

Theorem 8.1 (Equivalence of the induction principles). $(6) \Rightarrow [\exists \Lambda, \theta, \iota.(\theta_0 = t \wedge (11))]$.

Proof. Choose $\Lambda_{11} = 1 \dot{+} \Lambda_6$, $\Delta_{11} = \Delta_6$, $\theta_{11_0} = t$, if $\alpha, \alpha' \in \Lambda_6$ then $\theta_{11(1\dot{+}\alpha)} = \theta_{6\alpha}$, $I_{11(1\dot{+}\alpha)}^{\delta'}(s, s') = [I_{6\alpha}^{\delta'}(s, s') \wedge \forall \delta \in \Delta_6.(I_{6\alpha}^{\delta}(s, s') \Rightarrow \delta' \leqslant \delta)]$, $\iota_{11(1\dot{+}\alpha)}(s, s', 0) = [\exists \delta' \in \Delta_6.$ $I_{11(1\dot{+}\alpha)}^{\delta'}(s, s') \wedge \neg \theta_{6\alpha}(s, s') \wedge \exists s'' \in S.t(s', s'') \wedge \forall s'' \in S.(t(s', s'') \Rightarrow [\exists \delta'' < \delta'.I_{6\alpha}^{\delta''}(s, s'')])]$ and $\iota_{11(1\dot{+}\alpha)}(s, s', 1 \dot{+} \alpha') = [\exists \delta' \in \Delta_6.I_{11(1\dot{+}\alpha)}^{\delta'}(s, s') \wedge \neg \theta_{6\alpha}(s, s') \wedge \forall s'' \in S.(\theta_{6\alpha'}(s', s'') \Rightarrow$ $[\exists \delta'' < \delta'.I_{6\alpha}^{\delta''}(s, s'')])]$. □

Theorem 8.2 (Equivalence of the induction principles (continued)).

$$[\exists \Lambda, \theta, \iota, \pi \in (\Lambda \sim 0).(\forall s \in S.\iota_\pi(s, s, \pi) = \mathrm{tt} \wedge \theta_\pi = \lambda(s, s').[\phi(s) \Rightarrow \psi(s, s')]$$

$$\wedge \theta_0 = t \wedge (11))] \Rightarrow (8).$$

Proof. Choose $\Lambda_8 = \Lambda_{11}$, $\theta_{8\alpha}(s, s') = \textit{if } \alpha = 0 \textit{ then } \mathrm{tt} \textit{ else } \theta_{11\alpha}(s, s')$, $\pi_8 = \pi_{11}$, $\Delta_8 = \Delta_{11}$, $I_{8\alpha}^{\delta}(s, s') = \textit{if } \alpha = 0 \textit{ then } \mathrm{ff} \textit{ else } I_{11\alpha}^{\delta}(s, s')$. □

Because we no longer distinguish a main proposition θ_π as in (0), soundness of (11) is better formulated as follows:

$$\forall \alpha \in (\Lambda \sim 0),\ p \in \Sigma\langle S, t, \lambda s.[\exists \alpha', s'.\iota_{\alpha'}(s', s, \alpha)]\rangle.\exists i \in Dom(p).\theta_\alpha(p_0, p_i). \qquad (12)$$

Theorem 8.3 (Soundness (relative to t)). $(11) \Rightarrow (12)$.

Proof. Follows from the later proved Theorems 8.5 and 8.6. □

We are mainly concerned about the semantic completeness of (11). The reciprocal of Theorem 8.3 is not true.

Theorem 8.4 (Insufficient completeness condition). $(12) \not\Rightarrow (11)$.

Proof. Consider the counterexample $S = \{a, b, c\}$, $t(s, s') = [(s = a \wedge s' = b) \vee (s = b \wedge s' = c)]$, $\Lambda = 3, \theta_0 = t, \theta_1(s, s') = [s = a \wedge s' = c]$, $\theta_2(s, s') = [s = a \wedge s = b]$, $\Delta \neq 0$, $\iota_\alpha(s, s', \alpha') = [(\alpha = 1 \wedge s = a \wedge s' \in \{a, b\} \wedge \alpha' = 0) \vee (\alpha = 2 \wedge s' = s = a \wedge \alpha' \in \{1, 2\})]$.

Obviously, (12) holds. If (11) were true then we would have $\iota_2(a, a, 2)$ hence $\exists \delta_1 \in \Delta.I_2^{\delta_1}(a, a)$ and by $\iota_2(a, a, 1)$ and $\theta_1(a, c)$ we would have $\exists \delta_2 < \delta_1.I_2^{\delta_2}(a, c)$. But $\neg \theta_2(a, c)$ and $\forall \alpha' < 2.\neg \iota_2(a, c, \alpha')$, a contradiction. □

In Burstall's method the use of lemma θ_l in the proof of proposition θ_p has the effect of covering a number of transitions by a single step θ_l. Hence θ_l can be used in the

proof of θ_p only if this reduction leaves θ_p inevitable. Stated otherwise, θ_p must be inevitable for transitions θ_t made up of the lemmas that are used in the proof of θ_p. This is expressed more formally by condition (where $\theta_0 = t$):

$$\forall \alpha \in (\Lambda \sim 0),\ s \in S.[(\exists \alpha' \in \Lambda, s' \in S.\iota_{\alpha'}(s', s, \alpha))$$
$$\Rightarrow (\forall p \in \Sigma\langle S, \tau_{\alpha s}, \lambda \underline{s}.[\underline{s} = s]\rangle.\exists i \in Dom(p).\theta_\alpha(p_0, p_i))], \tag{13}$$

where $\tau_{\alpha s}(s', s'') = [\exists \alpha' < \alpha.\iota_\alpha(s, s', \alpha') \wedge \theta_{\alpha'}(s', s'')]$.

Condition (13) is a necessary one for semantic completeness:

Theorem 8.5 (Soundness (relative to τ) – necessary completeness condition)). (11) $\Rightarrow$ (13).

Proof. Assume (11). If $\Lambda = 1$ or $\forall \alpha', s'.\neg \iota_{\alpha'}(s', s, \alpha)$ then (13) obviously holds, else we prove (13) by transfinite induction on $\alpha \in (\Lambda \sim 0)$. Given $\alpha \in (\Lambda \sim 0)$ and $s \in S$, assume by reductio ad absurdum that $\exists \alpha' \in \Lambda,\ s' \in S, p \in \Sigma\langle S, \tau_{\alpha s}, \lambda \underline{s}.[\underline{s} = s]\rangle.\iota_{\alpha'}(s', s, \alpha) \wedge \forall i \in Dom(p).\neg \theta_\alpha(p_0, p_i)$. To get a contradiction, we show that it is then possible to build an infinite sequence $\langle (\delta_k, i_k)\colon k \geqslant 0\rangle$ such that $\forall k \geqslant 0.I_\alpha^{\delta_k}(s, p_{i_k})$ holds and $\langle \delta_k\colon k \geqslant 0\rangle$ is an infinite strictly decreasing chain of ordinals.

We have $\iota_{\alpha'}(s', s, \alpha)$ so that by (11) we derive $I_\alpha^{\delta_0}(s, s)$ i.e. $I_\alpha^{\delta_0}(s, p_{i_0})$ with $i_0 = 0$. If the sequence has been built up to point k, then by (11) $I_\alpha^{\delta_k}(s, p_{i_k})$ implies

$$([\exists \underline{\alpha} < \alpha.\iota_\alpha(s, p_{i_k}, \underline{\alpha})] \wedge [\forall \alpha' < \alpha.\iota_\alpha(s, p_{i_k}, \alpha') \Rightarrow ([\alpha' = 0 \Rightarrow \exists s'' \in S.\theta_{\alpha'}(p_{i_k}, s'')]$$
$$\wedge [\forall s'' \in S.(\theta_{\alpha'}(p_{i_k}, s'') \Rightarrow \exists \delta'' < \delta_k.\ I_\alpha^{\delta''}(s, s''))])])$$

because we have assumed $\neg \theta_\alpha(p_0, p_{i_k})$ and $p_0 = s$. If $\underline{\alpha} = 0$ then $\exists s'' \in S.\theta_{\underline{\alpha}}(p_{i_k}, s'')$ whence $\exists s'' \in S.\tau_{\alpha s}(p_{i_k}, s'')$. Else $0 < \underline{\alpha} < \alpha$, so that by induction hypothesis $\forall s' \in S.[(\exists \alpha'', s''.\iota_{\alpha''}(s'', s', \underline{\alpha})) \Rightarrow (\forall p' \in \Sigma\langle S, \tau_{\underline{\alpha} s'}, \lambda \underline{s}.[\underline{s} = s']\rangle.\exists i \in Dom(p').\theta_{\underline{\alpha}}(p'_0, p'_i))]$. In particular for $s' = p_{i_k}$, $\iota_\alpha(s, p_{i_k}, \underline{\alpha})$ holds and $\Sigma\langle S, \tau_{\underline{\alpha} p_{i_k}}, \lambda \underline{s}.[\underline{s} = p_{i_k}]\rangle$ is not empty so that we derive $\exists s'' \in S.\theta_{\underline{\alpha}}(p_{i_k}, s'')$, whence $\exists s'' \in S.\tau_{\alpha s}(p_{i_k}, s'')$. Since p_{i_k} is not a blocking state $i_{k+1} = i_k + 1$ belongs to $Dom(p)$ and we have $\tau_{\alpha s}(p_{i_k}, p_{i_{k+1}})$. It follows that $\exists \alpha' < \alpha.\iota_\alpha(s, p_{i_k}, \alpha') \wedge \theta_{\alpha'}(p_{i_k}, p_{i_{k+1}})$, whence $\exists \delta_{k+1} < \delta_k.I_\alpha^{\delta_{k+1}}(s, p_{i_{k+1}})$. □

Condition (13) (i.e. each lemma is inevitable relatively to the lemmas used in its proof) implies condition (12) (i.e. each lemma is inevitable relatively to the transition system):

Theorem 8.6 (Inevitability relative to τ implies inevitability relative to t). (13) $\Rightarrow$ (12).

Proof. Assume (13), we prove (12) by transfinite induction on $\alpha \in (\Lambda \sim 0)$. Assume by reductio ad absurdum, that $\exists p \in \Sigma\langle S, t, \lambda s.[\exists \alpha', s'.\iota_{\alpha'}(s', s, \alpha)]\rangle.\ \forall i \in Dom(p).\ \neg \theta_\alpha(p_0, p_i)$. To get a contradiction, we shall build an infinite sequence $\langle i_k\colon k \geqslant 0\rangle$ such that $p_{i_0} p_{i_1} \ldots$ is a counterexample to (13), i.e. $\exists \alpha', s'.\iota_{\alpha'}(s', p_{i_0}, \alpha) \wedge \forall k \geqslant 0.[\tau_{\alpha p_{i_0}}(p_{i_k}, p_{i_{k+1}}) \wedge \neg \theta_\alpha(p_{i_0}, p_{i_k})]$. We let i_0 be 0. If the sequence is built up to i_k

then it can be extended since $\exists i_{k+1} \geqslant i_k . \tau_{\alpha p_{i_0}}(p_{i_k}, p_{i_{k+1}})$ (and $\neg\theta_\alpha(p_{i_0}, p_{i_k})$ by hypothesis and $i_0 = 0$). Otherwise, $\forall j \geqslant i_k . \neg\tau_{\alpha p_{i_0}}(p_{i_k}, p_j)$ so that by definition of $\tau_{\alpha p_{i_0}}$ we would have $\forall j \geqslant i_k . \forall \alpha' < \alpha . [\neg\iota_\alpha(p_0, p_{i_k}, \alpha') \vee \neg\theta_{\alpha'}(p_{i_k}, p_j)]$. If $\forall \alpha' < \alpha . \neg\iota_\alpha(p_0, p_{i_k}, \alpha')$ then $\forall s' \in S . \neg\tau_{\alpha p_{i_0}}(p_{i_k}, s')$ so that $p_{i_0} \ldots p_{i_k} \in \Sigma\langle S, \tau_{\alpha p_{i_0}}, \lambda \underline{s} . [\underline{s} = p_{i_0}]\rangle$, in contradiction with (13). Else $\exists \alpha' < \alpha . \iota_\alpha(p_0, p_{i_k}, \alpha')$; so for that $\alpha' < \alpha$ we derive $\forall j \geqslant i_k . \neg\theta_{\alpha'}(p_{i_k}, p_j)$, hence $p_{i_k} p_{i_k+1} \ldots \in \Sigma\langle S, t, \lambda s . [\exists \alpha'', s'' . \iota_{\alpha''}(s'', s, \alpha')]\rangle$, in contradiction with induction hypothesis (12). □

We can now give a necessary and sufficient semantic completeness conditionfor (11).

Theorem 8.7 (Necessary and sufficient strong completeness condition). (13) $\Leftrightarrow$ (11).

Proof. By Theorem 8.5 we just have to prove (13) $\Rightarrow$ (11). Given $\alpha \in (\Lambda \sim 0)$, $s \in S$, we define:

- $In_{\alpha s} = \bigcup \{ Inter\langle S, \tau_{\alpha s}, \lambda \underline{s} . \mathrm{tt}, \theta_\alpha\rangle(s) : \exists \alpha' \in \Lambda, s' \in S . \iota_{\alpha'}(s', s, \alpha) \}$,
- $Go_{\alpha s} = \bigcup \{ Goal\langle S, \tau_{\alpha s}, \lambda \underline{s} . \mathrm{tt}, \theta_\alpha\rangle(s) : \exists \alpha' \in \Lambda, s' \in S . \iota_{\alpha'}(s', s, \alpha) \}$,
- $Ac_{\alpha s} = In_{\alpha s} \cup Go_{\alpha s}$.

We first prove that (13) $\Rightarrow (\forall \alpha \in (\Lambda \sim 0), s \in S . Wf(Ac_{\alpha s}, \tau_{\alpha s} \restriction In_{\alpha s}^{-1}))$.

This is obvious when $\forall \alpha' \in \Lambda, s' \in S . \neg\iota_{\alpha'}(s', s, \alpha)$ since $Ac_{\alpha s}$ is empty. Else, given $\alpha \in (\Lambda \sim 0)$ and $s \in S$ such that $\exists \alpha' \in \Lambda, s' \in S . \iota_{\alpha'}(s', s, \alpha)$ assume, by reductio ad absurdum, that $\exists p \in (\omega \to Ac_{\alpha s}) . \forall i \in \omega . \tau_{\alpha s} \restriction In_{\alpha s}(p_i, p_{i+1})$. For all $i \in \omega$ we have $p_i \in In_{\alpha s}$ so that $\neg\theta_\alpha(s, p_i)$, hence $p_i \notin Go_{\alpha s}$. Since $p_0 \in In_{\alpha s}$ we can assume $p_0 = s$ (else we can adjoin to the left of p a prefix $r_0 \ldots r_k$ of some trace $r \in \Sigma\langle S, \tau_{\alpha s}, \lambda \underline{s} . \mathrm{tt}\rangle$ such that $r_0 = s \wedge \forall j \leqslant k . \neg\theta_\alpha(r_0, r_j) \wedge r_k = p_0$ so that $\forall i < k . \tau_{\alpha s} \restriction In_{\alpha s}(r_i, r_{i+1})$). We have $\exists \alpha' \in \Lambda, s' \in S . \iota_{\alpha'}(s', s, \alpha)$ and $p \in \Sigma\langle S, \tau_{\alpha s}, \lambda \underline{s} . [\underline{s} = s]\rangle$ and $\forall i \in Dom(p) . p_i \notin Go_{\alpha s}$ hence $\neg\theta_\alpha(p_0, p_i)$, in contradiction with (13). Q.E.D.

Assuming (13), by the previous lemma we can define:

- $\Lambda = Sup^{+} \{ rk(Ac_{\alpha s}, \tau_{\alpha s} \restriction In_{\alpha s}^{-1}) : \alpha \in (\Lambda \sim 0) \wedge s \in S \}$,
- $I_\alpha^{\delta'}(s, s') = [s' \in Ac_{\alpha s} \wedge \delta' = rk(Ac_{\alpha s}, \tau_{\alpha s} \restriction In_{\alpha s}^{-1})(s')]$.

If $\alpha \in (\Lambda \sim 0)$, $s \in S$ and $\exists \alpha' \in (\Lambda \sim 0)$, $s' \in S . \iota_{\alpha'}(s', s, \alpha)$ then $s \in Ac_{\alpha s}$ so that $I_\alpha^{\delta}(s, s)$ holds with $\delta = rk(Ac_{\alpha s}, \tau_{\alpha s} \restriction In_{\alpha s}^{-1})(s)$.

Assume $\alpha \in (\Lambda \sim 0)$, $s, s' \in S$, $\delta' \in \Lambda$ and $I_\alpha^{\delta'}(s, s')$. We have $s' \in Ac_{\alpha s}$. If $s' \in Go_{\alpha s}$ then $\theta_\alpha(s, s')$ holds. Else $s' \in In_{\alpha s}$ so that by (13) there exists $s'' \in S$ such that $\tau_{\alpha s}(s', s'')$, hence some $\alpha' < \alpha$ such that $\iota_\alpha(s, s', \alpha') \wedge \theta_{\alpha'}(s', s'')$. If $\alpha' = 0$ we conclude $\exists s'' \in S . t(s', s'')$ since $\theta_0 = t$. Else $\alpha' \neq 0$ and if $\forall s'' \in S . \neg t(s', s'')$ then $\langle s' \rangle \in \Sigma\langle S, t, \lambda s' . [\exists \alpha, s . \iota_\alpha(s, s', \alpha')]\rangle$, whence by (13) and Theorem 8.6 we conclude from (12) that $\theta_{\alpha'}(s, s')$; hence $\tau_{\alpha s}(s', s')$ holds. It follows from $s' \in In_{\alpha s}$ that $\exists \underline{\alpha}, \underline{s}, p \in \Sigma\langle S, \tau_{\alpha s}, \lambda \underline{s} . [\underline{s} = s]\rangle$, $i \in Dom(p) . \iota_{\underline{\alpha}}(\underline{s}, s, \alpha) \wedge \forall j \leqslant i . \neg\theta_\alpha(p_0, p_j) \wedge p_k = s'$, so that the infinite trace $p_0 \ldots p_k s' s' \ldots$ is a counterexample to (13). Hence, by reductio ad absurdum we conclude that $\exists s'' \in S . t(s', s'')$. Finally, given $\alpha' < \alpha$ and $s'' \in S$ such that $\iota_\alpha(s, s', \alpha')$ and $\theta_{\alpha'}(s', s'')$ we have $\tau_{\alpha s} \restriction In_{\alpha s}(s', s'')$ hence $s'' \in Ac_{\alpha s}$ and there exists $\delta'' = rk(Ac_{\alpha s}, \tau_{\alpha s} \restriction In_{\alpha s}^{-1})(s'') < rk(Ac_{\alpha s}, \tau_{\alpha s} \restriction In_{\alpha s}^{-1})(s') = \delta'$ such that $I_\alpha^{\delta''}(s, s'')$ holds. □

In a related paper [5], we show that Floyd's method (i.e. (2) without (LI)) and Burstall's method (more precisely (2)) are strongly equivalent in the sense that a proof by one method can be translated into a proof by the other method.

9. Conclusion

Our study and generalization of Burstall's method should be extended (e.g. in the style of [4]) so as to take fairness hypotheses for parallel programs into account. It should also be extended from a methodological point of view in order to obtain better presentations of Burstall's method and broader applications e.g. for logic programs.

Acknowledgment

We would like to thank L. Lamport and the other, anonymous, referee for their comments on the manuscript.

References

[1] K.R. Apt and C. Delporte, An axiomatization of the intermittent assertion method using temporal logic (extended abstract), in: *Proc. 10th ICALP*, Lecture Notes in Computer Science, Vol. 154 (Springer, Berlin, 1983) 15–27.

[2] K.R. Apt and G.D. Plotkin, Countable nondeterminism and random assignment. *J. ACM* **33** (1986) 724–767.

[3] R.M. Burstall, Program proving as hand simulation with a little induction, *Inform. Process.* **74** (1974) 308–312.

[4] P. Cousot and R. Cousot, "A la Floyd" induction principles for proving inevitability properties of programs, in: M. Nivat and J. Reynolds, eds, *Algebraic Methods in Semantics* (Cambridge Univ. Press, Cambridge, 1985) 277–312.

[5] P. Cousot, and R. Cousot, SOMETIME = ALWAYS + Recursion ≡ ALWAYS, on the equivalence of the intermittent and invariant assertions methods for proving inevitability properties of programs, *Acta Inform.* **24** (1987) 1–31.

[6] E.W. Dijkstra, *Selected Writings on Computing: a Personal Perspective* (Springer, Berlin, 1982).

[7] R.W. Floyd, Assigning meaning to programs, in: *Proc. Symp. Applied Math.*, Vol. 19 (AMS, Providence, RI, 1967) 19–32.

[8] D. Harel, *First Order Dynamic Logic*, Lecture Notes in Computer Science, Vol. 68 (Springer, Berlin, 1979).

[9] Z. Manna and A. Pnueli, Verification of concurrent programs: a temporal proof system, in: *Proc. 4th School on Advanced Programming*, Math. Centre Tract No. 159 (CWI, Amsterdam (1982) 163–255.

[10] Z. Manna, and R.J. Waldinger, Is SOMETIME sometimes better than ALWAYS?, intermittent assertions in proving program correctness, *Comm. ACM* **21** (1978) 159–172.

[11] Pnueli, A., The temporal logic of programs, in: *Proc. 18th Symp. on Foundations of Comp. Sci.*, Providence, RI (1977) 46–57.

[12] J. Schwarz, Event-based reasoning – a system for proving correct termination of programs, in: *Proc. 3rd ICALP*, Edinburgh (1976) 131–146.