9. PROOF CHARTS

The idea of presenting program proofs by diagrams was introduced by Lamport[77] and later developed by Owicki & Lamport[82] and Manna & Pnueli[82]. However because of a number of restrictions (such as impossibility of making infinite inductions) the method was not semantically complete.

This motivates our generalization which can be introduced by the self-explanatory:

Example 9-1:

The "à la Burstall" total correctness proof of program 2-1 considered at paragraph 5.2 can be presented as follows (we write at L as a shorthand for \([c=L]\)):

Propsition 1:

<table>
<thead>
<tr>
<th>at Start ( \land n \geq 0 \land n = n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>\quad \downarrow</td>
</tr>
<tr>
<td>at Loop ( \land n \geq 0 \land n = n \land p = 1 )</td>
</tr>
<tr>
<td>\quad \downarrow \quad \text{Lemma 0}</td>
</tr>
<tr>
<td>at Loop ( \land n = 0 \land p = 2^n )</td>
</tr>
<tr>
<td>\quad \downarrow</td>
</tr>
<tr>
<td>at Finish ( \land p = 2^n )</td>
</tr>
</tbody>
</table>

Lemma 0:

<table>
<thead>
<tr>
<th>at Loop ( \land n \geq 0 \land n = n \land p = p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>\quad \downarrow</td>
</tr>
<tr>
<td>at Loop ( \land n \geq 0 \land n = n \land p = p )</td>
</tr>
<tr>
<td>\qquad \quad \downarrow \quad \text{Lemma 0}, 0 \leq n \leq n</td>
</tr>
<tr>
<td>at Loop ( \land n = 0 \land p = 2^n )</td>
</tr>
<tr>
<td>\quad \downarrow</td>
</tr>
</tbody>
</table>
A proof chart for a transition system \( (S, tJ \) will be formalized using a finite set of finite well-structured single-entry single-exit labelled graphs. We write \( I^e \longrightarrow I^a \) to denote such a graph with a unique entry vertex labelled \( I^e \) and a unique exit vertex labelled \( I^a \).

The set of admissible graphs will be defined by a graph grammar. Elementary graphs are of the form \( I \longrightarrow J \) where \( I \) is the entry vertex, \( J \) is the exit vertex and there is a single edge from vertex \( I \) to vertex \( J \). There are different types of edges (drawn by different arrows) some of which can be labelled (the label is then written on the corresponding arrow). Composite graphs are obtained using the following graph composition operations:

1. If \( I \longrightarrow J \) and \( K \longrightarrow L \) are two graphs such that \( J = K \) then \( I \longrightarrow J \longrightarrow L \) denotes the graph such that the entry vertex \( K \) is identified with the exit vertex \( J \), there are no other mixtures of the vertices of the original graphs and the entry (respectively exit) vertex of the composite graph is the vertex labelled \( I \) (respectively \( L \)).

2. If \( I \longrightarrow J \) and \( K \longrightarrow L \) are two graphs such that \( I = K \) and \( J = L \) then \( I \longrightarrow J \longrightarrow K \longrightarrow L \) denotes the composite graph where the entry (respectively exit) vertices of the original graphs have been identified.

3. If \( I \longrightarrow J \) and \( K \longrightarrow L \) are two graphs such that \( I = K \) then the loop \( I \longrightarrow J \longrightarrow L \longrightarrow I \) is the composite graph with entry vertex \( I \) identified with \( K \), with exit vertex \( J \) and with a new arc from vertex \( L \) to entry vertex \( I \).

We write \( I(\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}, \mathbf{s}_0) \) (respectively \( I(\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}, \mathbf{s}_0) \) and \( I(s_0, s_1, s) \)) to mean that the label \( I \) attached to a graph vertex belongs to \( (S \times S \times S \times S \times \{ \# \}) \) where \( m = n \) (respectively \( m = n + 1, m = 0 \)) is the enclosing loops. Informally \( s_0 \) is the value of the state on program entry, \( s_1 \) (respectively \( s_1 \)) is the value of the state corresponding to the entry of the graph (respectively to the entry of the \( i \)-th enclosing loop in the graph) and \( s \) is the value of the current state.
DEFINITION 9-2  (Proof charts)

A proof chart for $(S,t)$ is a pair $((\Lambda,\vdash),(G_\lambda,(f,W_\lambda,\preceq_\lambda)):\lambda\in\Lambda)$ such that $(\Lambda,\vdash)$ is a finite well-founded set (of graph names) and for all $\lambda\in\Lambda$, $f_\lambda:G_\lambda(S^2+W_\lambda,\preceq_\lambda)$ and $G_\lambda$ is a well-formed chart.

$I^G_\lambda(s_0,s,s)\longrightarrow I^G_\lambda(s_0,s,s)$ generated by the following graph grammar:

$$J(s_0,s_0,s,s)\longrightarrow K(s_0,s_0,s,s)$$

when $\forall s_0,s,s\in S_0, s\in S_0, [J(s_0,s_0,s,s)\Rightarrow (\exists s\in S, t(s,s) \land \forall s\in S, (t(s,s) \Rightarrow K(s_0,s_0,s,s))]$

$$J(s_0,s_0,s,s)\longrightarrow L(s_0,s_0,s,s)\longrightarrow K(s_0,s_0,s,s)$$

when $\forall s_0,s,s\in S_0, s\in S_0, [J(s_0,s_0,s,s)\Rightarrow L^1(s_0,s_0,s,s)]$

$$L(s_0,s_0,s,s)\longrightarrow M(s_0,s_0,s,s)$$

when $f_\in (S^2 \rightarrow W) \land \forall s_0,s,s\in S_0, s\in S_0, [J(s_0,s_0,s,s)\Rightarrow (K(s_0,s_0,s,s) \lor L(s_0,s_0,s,s))] \land [M(s_0,s_0,s,s)\Rightarrow ((f(s_0,s_0,s) < f(s_0,s_0,s)) \land L(s_0,s_0,s,s)) \land [K(s_0,s_0,s,s)])$
(Observe that proof-charts are reducible graphs whence could also be formalized using a well-structured logical language).

We can prove the inevitability of $\psi$ for $(S,t,\phi)$ by showing that:

\[ (14) \exists t \in \mathcal{L} \text{ such that} \]

\[ \forall s_0, s \in S, (I^E_t(s_0,s,s) \equiv I^G_t(s_0,s,s), (t, w, \leq_t), t \in \mathcal{L}, \leq_t) \] and $\pi \in \mathcal{L}$ such that:

\[ \forall s_0, s, s \in S, (I^E_\pi(s_0, s, s) = [s_0 = s \land \phi(s)] \land I^G_\pi(s_0, s, s) = [s_0 = s \land \psi(s, s)]) \]

"À la Floyd" inevitability proofs can also be presented using proof charts as shown by the following:

**Example 9-3:**

An "à la Floyd" total correctness proof of program 2-1 can also be presented as follows:

\[
\begin{align*}
\text{at Start} \land n \geq 0 \land n = n \\
\downarrow \\
\text{at Loop} \land n \geq 0 \land n = n \land p = 1 \\
\downarrow \\
\text{at Loop} \land n \geq 0 \land n = n \land p = 2^{n-1} \\
\downarrow \\
\text{at Loop} \land n \geq 0 \land n = n \land p = 2^{n-1} \\
\downarrow \\
\text{at Loop} \land n = 0 \land p = 2^n \\
\downarrow \\
\text{at Finish} \land p = 2^n
\end{align*}
\]

**THEOREM 9-4** *(Soundness of proof charts)*

\[ (14) \Rightarrow (2), \text{ with } n \in \mathcal{L} \rightarrow \text{Ord} \]
Proof:

Let \((\Lambda, \prec)\), \(\{(G_\lambda, (f_\lambda, W_\lambda, \prec_\lambda)), \lambda \in \Lambda\}\) be a proof chart. Since \(W_\lambda \prec_\lambda\) we can assume without loss of generality that \(W_\lambda \in \text{Ord}\) and \(\prec_\lambda = \prec\) (otherwise we can use rank-functions). Since \(\Lambda\) is finite we can also assume that \(\Lambda \in \omega\) and \(\lambda \prec \lambda\). Since each graph \(G_\lambda\) is finite we can suppose that its vertices are named by elements of some finite set \(N_\lambda\), the vertex named \(j\) being labelled by \(j^3 \in (S \times S \times S(j)) \times S \to \{\text{true}, \text{false}\}\) where \(e(j)\) is the number of loops enclosing vertex \(j\). We let \(\varepsilon\) and \(\sigma\) be the respective names of the unique entry and exit vertices of \(G_\lambda\).

For each \(\lambda \in \Lambda\) we consider the set \(T_\lambda\) of tuples \((j, \varepsilon, s, s', s)\) such that \(j \in N_\lambda\), \(s, s', s, s \in S\), \(s \in S(j)\) and \(j^3(s, s, s')\) holds. The binary relation \(\prec_\lambda\) on \(T_\lambda\) is defined by \((j, \varepsilon, s, s', s) \prec_\lambda (j, \varepsilon, s, s', s)\) if and only if either

- \(j^3 \rightarrow j^3 \wedge s_0 = s_0 \wedge s' = s' \wedge s = s \wedge s = s\) \(\wedge t(s, s')\)
- or \(j^3 \rightarrow j^3 \wedge s_0 = s_0 \wedge s' = s' \wedge s = s \wedge s = s\) \(\wedge j^3\)
- or \(j^3 \rightarrow j^3 \wedge s_0 = s_0 \wedge s' = s' \wedge s = s \wedge s = s\) \(\wedge j^3\)
- or \(j^3 \rightarrow j^3 \wedge s_0 = s_0 \wedge s' = s' \wedge s = s \wedge s = s\) \(\wedge j^3\)
- or \(j^3 \rightarrow j^3 \wedge s_0 = s_0 \wedge s' = s' \wedge s = s \wedge s = s\) \(\wedge j^3\)

Assume that \((j_\lambda, \varepsilon_\lambda, s_\lambda, s_\lambda, s_\lambda)\) is an infinite decreasing sequence for \(\prec_\lambda\). It follows that \((j_\lambda : \lambda \geq 0)\) is an infinite path in the finite graph \(G_\lambda\), hence a cycle. Therefore there is some vertex \(j\) of \(G_\lambda\) (of type \(j^3 \rightarrow j^3\)) such that the sequence \((j, \varepsilon, s, s, s)\) of elements of \((j, \varepsilon, s, s, s) : \lambda \geq 0)\) such that \(j_\lambda = j\) is infinite. This is in contradiction with \(W_\lambda \prec \prec_\lambda\). By reductio ad absurdum, we have \(W_\lambda \prec_\lambda\).
THEOREM 9-5  (Semantic completeness of proof charts)

(2) ⇒ (14)

Proof:

A proof by (2) can be presented by the proof chart \((G_\lambda, (f_\lambda, \text{Ord}, <)), \lambda \in \Lambda\)
where each \(G_\lambda, \lambda \in \Lambda\) is the following chart:
10. PROVING INEVITABILITY PROPERTIES OF PARALLEL PROGRAMS

Since parallel programs can be represented by non-deterministic transitions systems, proof charts can also be applied to inevitability proofs of parallel programs.

Example 9-6: (Total correctness of a parallel program)

We consider an asynchronous parallel version of program 2-1 to compute \(2^n\) when \(n \geq 0\):

1: \(N_1 := 0; N_2 := N;\)
2:  
   11: \(P_1 := 1;\)
   12: \(\text{if } N_1 + 1 < N_2 \text{ then}\)
   13: \(T_1 := N_1 + 1; P_1 := 2 \times P_1;\)
   14: \(N_1 := T_1;\)
   15: \(P_1; \text{goto 12;}\)
16:  
11  
21: \(P_2 := 1;\)
22: \(\text{if } N_1 + 1 < N_2 \text{ then}\)
23: \(T_2 := N_2 - 1; P_2 := 2 \times P_2;\)
24: \(N_2 := T_2;\)
25: \(P_2; \text{goto 22;}\)
26:  
3: \(P := \text{if } N_1 + 1 = N_2 \text{ then } 2 \times P_1 \times P_2 \text{ else } P_1 \times P_2 \text{ fi;}\)
4:  

We write \(\text{at } j\) (respectively \(\text{at } i,j\)) to stand for \(c = j\) \((c_i = j)\) where \(c\) \((c_i)\) is the program location counter \((\text{of process } i\) when control is in the parallel command\). We write \(\text{in } E\) for \(\forall \{\text{at } l : l \in E\}\), \((P \Rightarrow Q \mid R)\) is the abbreviation of \((P \land Q) \lor (P \land R)\) whereas if \(P\) holds then \((P \Rightarrow a | b)\) denotes value \(a\) else value \(b\). In particular \(\text{min}(a, b) = (a < b \Rightarrow a) \lor b\).

\[\begin{array}{c}
P \\ Q \\ P
\end{array}\]

will be shortened to

\[\begin{array}{c}
P \\ Q \lor R \\ Q \\ P
\end{array}\]
In the total correctness proof chart $P: (\text{at1} \land n_0 \geq 0 \rightarrow \text{at4} \land p = 2^{n_0})$

we distinguish two cases:

1. The case $n_0 \leq 1$ is handled by lemma $Z0: (\text{at1} \land 0 \leq n_0 \leq 1 \rightarrow \text{at4} \land p = 2^{n_0})$. This lemma can be proved by hand-simulation and the corresponding chart is left to the reader.

2. The main case $n_0 > 1$ is handled by lemma $L: (\text{Atloop} \land n1 = n1 \land n2 = n2 \land n1 + 1 < n2 \land \text{Inv} \rightarrow \text{at18} \land \text{at26} \land 0 \leq n1 \leq n2 \leq \min(n1+1, n2) \land p1 = 2^{n1} \land p2 = 2^{n0 - n2})$ where $\text{Atloop}$ stands for $([\text{at12} \land \text{in}(21, ..., 25)] \lor [\text{in}(11, ..., 15) \land \text{at22}])$ and $\text{Inv}$ is the following invariant:

\[ \text{Inv} = [(\text{at11} \Rightarrow n1<0) \land p1=2^{n1} \times (\text{at14} \land 2 + 1)] \land (\text{at14} \Rightarrow t1=n1+1) \land (\text{at21} \Rightarrow n2>0) \land p2=2^{n0-n2} \times (\text{at24} \land 2 + 1] \land (\text{at24} \Rightarrow t2=n2-1)] \]

The proof chart $P$ is the following:

\[
\begin{array}{c}
\text{at1} \land n_0 \geq 0 \\
\downarrow \\
\text{at1} \land 0 \leq n_0 \leq 1 \\
\downarrow \\
\text{at1} \land n_0 > 1 \\
\downarrow \\
\text{at2} \land n1 = 0 \land n2 \leq n_0 > 1 \\
\downarrow \\
\text{at11} \land \text{at21} \land n1 = 0 \land n2 \leq n_0 > 1 \\
\downarrow \\
\text{at12} \land \text{at21} \land n1 = 0 \land n2 \leq n_0 > 1 \\
\downarrow \\
\text{at16} \land \text{at26} \land 0 \leq n1 \leq n2 \leq \min(n1+1, n0) \land p1 = 2^{n1} \land p2 = 2^{n0-n2} \\
\downarrow \\
\text{at3} \land 0 \leq n1 \leq n2 \leq \min(n1+1, n0) \land p1 = 2^{n1} \land p2 = 2^{n0-n2} \\
\downarrow \\
\text{at4} \land p = 2^{n0}
\end{array}
\]
The proof of lemma L is by induction on $n_2-n_1$ which is strictly decreased after one iteration in the loop of one of the two processes. This iteration is described by lemma I: \[\text{Atloop } \land n_1=\bar{n}_1 \land n_2=\bar{n}_2 \land n_1+1<\bar{n}_2 \land \text{Inv} \implies \text{Atloop } \land 0\leq n_2-n_1 < n_2-n_1 \land n_1 \leq n_1 \land n_2 \leq n_2 \land n_1+1<\bar{n}_2 \land \text{Inv} \land (n_1=\bar{n}_2 \implies \text{at15 } \lor \text{at25}).\]

When execution is about to leave the loops we have $n_1 \leq n_2 < n_1+1$. The case $n_1=n_2$ is handled by lemma E: \[\{n\in\{1,2,15\} \land n\in\{25,26\} \land n_1=\bar{n}_1 \land p_1=\bar{p}_1 \land n_2=\bar{n}_2 \land p_2=\bar{p}_2 \implies \text{at16 } \lor \text{at26} \land n_1=\bar{n}_1 \land p_1=\bar{p}_1 \land n_2=\bar{n}_2 \land p_2=\bar{p}_2 \}.\]

The proof is trivial by hand-simulation and the corresponding chart is left to the reader. The case $n_2=n_1+1$ is handled by lemma B: \[\text{Atloop } \land n_1=\bar{n}_1 \land n_1+1=\bar{n}_2 \land \text{Inv} \implies \text{at16 } \lor \text{at26} \land n_1=n_1 \land n_2=n_2 \land p_1=\bar{p}_1 \land p_2=\bar{p}_2 \land n_1=n_2 \land n_1+1=n_2 \land \text{Inv}.\]

The proof chart L is the following:

There is no difficulty about the proofs of lemmas I and B which can entirely be done by hand simulation.

We let $\text{Inv}'$ be $\text{Inv} \land n_1+1<\bar{n}_2$ in the proof chart I: