HKU CSIS Tech Report TR-A7-86

The Computer Journal **30** (4): 378−380 (1987)

Towards a Single Criterion for Identifying Program Unstructuredness*

T.H. Tse
Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong

ABSTRACT

We introduce the concepts of fully embedded skeletons and partially overlapping skeletons in program flowgraphs. We show that only one simple criterion is necessary and sufficient for the identification of program unstructuredness. Namely, a program flowgraph is unstructured if and only if it contains partially overlapping skeletons.

1. INTRODUCTION

In [2], we proposed a formal approach for studying the properties of program flowgraphs. We found two conditions for the identification of unstructuredness. In this paper, we shall extend our theory to include fully embedded skeletons and partially overlapping skeletons. We shall show that only one simple criterion is necessary and sufficient for the identification of unstructuredness in program flowgraphs.

2. DEFINING UNSTRUCTUREDNESS

A program flowgraph is defined as *unstructured* if and only if it contains at least one of the following:

(a) An Entry in the Middle of a Selection

Given a condition node n, the module M_n is said to be a *selection module* if and only if neither of its branches** $B_\alpha(n)$ or $B_{-\alpha}(n)$ contains n. A node m in M_n is defined as an *entry node* if and only if there exists some node p outside M_n such that m is a successor of p, i.e. $m = s_\beta(p)$. A node m is said to be an *entry in the middle of a selection* if and only if $m \neq n$ but is an entry node of a selection module M_n.

---

* Part of this research was done at the London School of Economics, University of London under a Commonwealth Academic Staff Scholarship. It was also supported in part by a University of Hong Kong Research Grant.

** The notations in this paper will follow those of [2].

(b) An Entry in the Middle of an Iteration

Given a condition node n, the module M_n is said to be an *iteration module* if and only if one of the branches $B_\alpha(n)$ contains n. Given an iteration module M_n, we define an *entry in the middle of an iteration* as an entry node m such that

(i) $m \neq n$ or $s_\alpha(n)$ or $s_{-\alpha}(n)$, or

(ii) there exists some other entry node $\neq m$.

(c) An Exit in the Middle of a Selection

A selection module M_n is said to have an *exit in the middle* if and only if its branches $B_\alpha(n)$ and $B_{-\alpha}(n)$ have a non-empty intersection.

(d) Multiple Exits in an Iteration

A decision node m is defined as an *exit of an iteration module* M_n if and only if:

(i) $M_m = M_n$;

(ii) m is in one of the branches $B_\gamma(m)$ but not in the opposite branch $B_{-\gamma}(m)$.

An iteration module is said to have *multiple exits* if and only if it has more than one exits.

3. PARTIALLY OVERLAPPING SKELETONS

In [2], we attempted to relate program unstructuredness to the properties of skeletons in flowgraphs. We shall extend our theory further in this paper. A skeleton $\mathbf{q}_\delta(v)$ is said to be *fully embedded* in another skeleton $\mathbf{q}_\gamma(u)$ if and only if $\mathbf{q}_\gamma(u)$ contains v as well as all the nodes of $\mathbf{q}_\delta(v)$. Two skeletons $\mathbf{q}_\gamma(u)$ and $\mathbf{q}_\delta(v)$ are said to be *partially overlapping* if and only if they are not fully embedded in one another but contain at least one common node m not equal to u or v.

Lemma 3.1

A skeleton $\mathbf{q}_\delta(v)$ is fully embedded in another skeleton $\mathbf{q}_\gamma(u)$ if and only if $\mathbf{q}_\gamma(u)$ contains both v and $s_\delta(v)$.

Proof:

If $\mathbf{q}_\delta(v)$ is fully embedded in $\mathbf{q}_\gamma(u)$, then the latter will of course contain both v and $s_\delta(v)$. Conversely, suppose $\mathbf{q}_\gamma(u)$ contains both v and $s_\delta(v)$. Then there exists a sequence of nodes $\langle w_0, \ldots, w_r \rangle$ such that

$w_0 = s_\gamma(u)$;
$w_i = s_\vee(w_{i-1})$ for $i = 1, \ldots, r$ (if $s_\gamma(u) \neq s_\delta(v)$);
$w_r = s_\delta(v)$.

For any node m ($\neq s_\delta(v)$) in $\mathbf{q}_\delta(v)$, there exists a sequence of nodes $\langle w_r, \ldots, w_t \rangle$ such that

$w_r = s_\delta(v)$;
$w_i = s_\vee(w_{i-1})$ for $i = r+1, \ldots, t$;
$w_t = m$.

Furthermore, $m \le s_{\vee}(v) \le s_{\vee}(u)$. Hence any m in $\mathbf{q}_\delta(v)$ must lie in $\mathbf{q}_\gamma(u)$. []

Lemma 3.2

If m is an entry node of M_n, then there exists a node u outside M_n such that m is in $\mathbf{q}_\gamma(u)$. Furthermore,

(a) $m = s_\gamma(u)$, or
(b) $m = s_{\vee}(v)$ for some node v in $\mathbf{q}_\gamma(u)$ but outside M_n.

Proof:

Suppose m is an entry node of M_n. Then there exists some node u_0 outside M_n such that $m = s_{\gamma_0}(u_0)$. If u_0 is a condition node, then (a) follows immediately. If, on the other hand, u_0 is an action node, then $s_{\vee}(u_0) = m$, and there exists a node u_1 outside M_n such that one of its skeletons $\mathbf{q}_{\gamma_1}(u_1)$ contains u_0. If $s_{\vee}(m) \le s_{\vee}(u_1)$, then (b) will follow. Otherwise $s_{\vee}(u_1) = m$, and there exists another node u_2 outside M_n such that one of its skeletons $\mathbf{q}_{\gamma_2}(u_2)$ contains u_1. Proceeding in this way, since the program flowgraph is finite, we shall arrive at some u_r outside M_n such that one of its skeletons $\mathbf{q}_{\gamma_r}(u_r)$ contains m, and $m = s_{\vee}(u_{r-1})$ for some node u_{r-1} in $\mathbf{q}_{\gamma_r}(u_r)$ but outside M_n. []

We can now derive the main theorems of the paper, thus connecting unstructuredness with partially overlapping skeletons.

Theorem 3.3

A program flowgraph is unstructured if there exist partially overlapping skeletons.

Proof:

Suppose the skeletons $\mathbf{q}_\gamma(u)$ and $\mathbf{q}_\delta(v)$ partially overlap at the node m. By Lemma 3.1, we have two cases:

(a) $v \notin \mathbf{q}_\gamma(u)$. Suppose M_v is an iteration module. Since there exists an elementary path from m to **end** not passing through v, by Lemma 5.1 of [2], there must be another exit. On the other hand, suppose M_v is a selection module. Since there exists an elementary path from $u \notin M_v$ to $m \in M_v$ not passing through v, we must have an entry in the middle of a selection. Hence we have unstructuredness in either case.

(b) $v \in \mathbf{q}_\gamma(u)$ and $s_\delta(v) \notin \mathbf{q}_\gamma(u)$. Let w be the first node in $\mathbf{q}_\delta(v)$ such that it is also in $\mathbf{q}_\gamma(u)$. Then w is an entry node for M_v. But since w cannot be v, $s_\delta(v)$ or $s_{-\delta}(v)$, it must be an entry in the middle of a selection or an entry in the middle of an iteration. In either case, we have unstructuredness. []

We shall prove that the converse of the theorem is also true, resulting in a necessary and sufficient condition for unstructuredness:

Theorem 3.4

A program flowgraph is unstructured if and only if there exist partially overlapping skeletons.

Proof:

Suppose a program flowgraph is unstructured. There are three possible anomalies:

(a) An Entry in the Middle of a Selection or Iteration

Let m be an entry in the middle of a module M_n. By Lemma 3.2, there exists a node u outside M_n such that one of its skeletons $\mathbf{q}_\gamma(u)$ contains m. Take the smallest submodule M_v in M_n such that m is also an entry in the middle of M_v. We have four cases:

(i) $m = s_\delta(v)$ and there is some other entry node $p = s_{-\delta}(v)$. Assume that there is no partially overlapping skeleton. Then v, $s_\delta(v)$ and $s_{-\delta}(v)$ are in $\mathbf{q}_\gamma(u)$. Since $s_\delta(v) \le v$ and $s_{-\delta}(v) \le v$, we must either have $s_\delta(v) \le s_{-\delta}(v) \le v$ or $s_{-\delta}(v) \le s_\delta(v) \le v$. Hence all paths from $s_\delta(v)$ to $s_{-\delta}(v)$ (or vice versa) do not pass through v, which is clearly a contradiction. Hence we must have partially overlapping skeletons.

(ii) $m = v$. Again assume that there is no partially overlapping skeleton. Then $\mathbf{q}_\gamma(u)$ should contain both v and $s_\delta(v)$. But since $s_\delta(v) \le v$, $\mathbf{q}_\gamma(u)$ must be of the form $< \dots\ s_\delta(v), \dots , v, \dots >$. By definition, there exists a sequence of nodes $<w_0 \dots w_t>$ such that

$$w_0 = s_\delta(v);$$
$$w_i = s_\vee(w_{i-1}) \text{ for } i = 1, \dots , t;$$
$$w_t = v = m.$$

Furthermore, all of these nodes are in both $\mathbf{q}_\gamma(u)$ and M_v. This contradicts the fact that, by Lemma 3.2, $m = s_\vee(w)$ for some node w in $\mathbf{q}_\gamma(u)$ but outside M_n. Hence we must have partially overlapping skeletons.

(iii) $m \ne v$ or $s_\delta(v)$ or $s_{-\delta}(v)$, and m is in $\mathbf{q}_\delta(v)$. Then there exists a node w in $\mathbf{q}_\delta(v)$ such that $m = s_\vee(w)$. Assume that there is no partially overlapping skeleton. Then both m and w are in $\mathbf{q}_\gamma(u)$. The skeleton $\mathbf{q}_\gamma(u)$ must therefore be of the form $< \dots , w, m, \dots >$. This contradicts the fact that $m = s_\vee(w)$ for some node w in $\mathbf{q}_\gamma(u)$ but outside M_n. Hence we must have partially overlapping skeletons.

(iv) m is not in $\mathbf{q}_\delta(v)$. By definition, there exists a node w in M_v such that m is in $\mathbf{q}_\varepsilon(w)$. Then M_w contains two distinct entries: w and m. This contradicts the fact that M_v is the smallest submodule in M_n such that m is an entry in the middle of M_v. Hence we must have partially overlapping skeletons.

(b) An Exit in the Middle of a Selection

Since $B_\alpha(n)$ and $B_{-\alpha}(n)$ have a non-empty intersection, there exists u in $B_\alpha(n)$ and v in $B_{-\alpha}(n)$ such that $\mathbf{q}_\gamma(u)$ and $\mathbf{q}_\delta(v)$ contain a common node m. By definition, we have partially overlapping skeletons.

(c) Multiple Exits in an Iteration

Let M_n be the iteration module and suppose that there are more than one exits. We have two cases:

(i) n is an entry node of M_n. Then there exists a node u outside M_n such that one of its skeletons $\mathbf{q}_\gamma(u)$ contains n. On the other hand, by Corollary 5.6 of [2], n is in neither of its own skeletons. Therefore there exists another node v ($\ne n$) in M_n such that one of its skeletons $\mathbf{q}_\delta(v)$ contains n. By the definition of skeletons, $s_\vee(n) \le s_\vee(v)$. But since v is in M_n, $s_\vee(v) \le s_\vee(n)$. Hence $s_\vee(v) = s_\vee(n)$, and so v and n cannot both be in

$\mathbf{q}_\gamma(u)$. Thus we must have two partially overlapping skeletons.

(ii) n is not an entry node of M_n. Unless we are having an entry in the middle of an iteration, the entry node must be $s_\alpha(n)$. That is to say, there exists a node u outside M_n such that one of its skeletons $\mathbf{q}_\gamma(u)$ contains $s_\alpha(n)$. If $\mathbf{q}_\gamma(u)$ does not contain n, then we have two partially overlapping skeletons. If $\mathbf{q}_\gamma(u)$ contains n, then, by arguments similar to (i), we also have two partially overlapping skeletons. []

Following the line of [1], fully embedded skeletons and partially overlapping skeletons can be located in $O(N)$ time, where N is the total number of nodes. Hence unstructuredness in program flowgraphs can be identified in $O(N)$ time.

4. CONCLUSION

We have introduced the concepts of fully embedded skeletons and partially overlapping skeletons in program flowgraphs. We have shown that only one simple criterion is necessary and sufficient for the identification of program unstructuredness. Namely, a program flowgraph is unstructured if and only if it contains partially overlapping skeletons.

ACKNOWLEDGEMENTS

The author is grateful to R.K. Stamper of the London School of Economics, University of London and M.Y. Chan of the University of Hong Kong for their invaluable comments and suggestions.

REFERENCES

[1] R. Tarjan, "Depth first search and linear graph algorithms", *SIAM Journal on Computing* **1** (2): 146−160 (1972).

[2] T.H. Tse, "The identification of program unstructuredness: a formal approach", *The Computer Journal* **30** (6): 507−511 (1987).