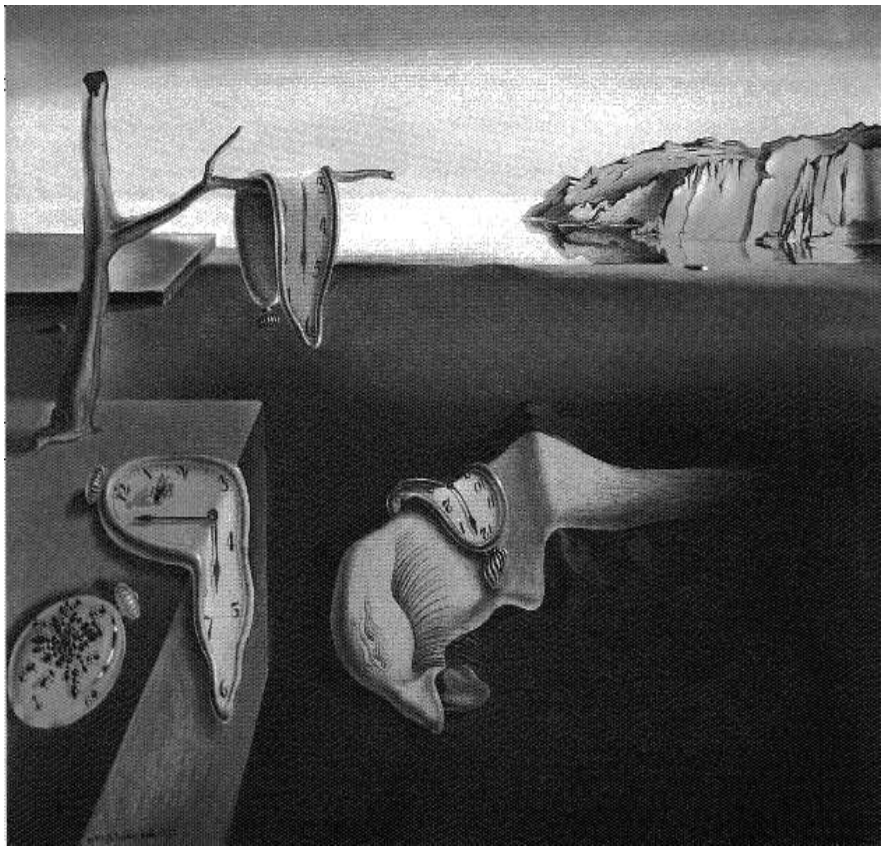
Institut für Automation
Abt. für Automatisierungssysteme

Technische Universität Wien

Projektbericht Nr. 183/1-137
2004

Tour de Spec — A Collection of Spec95 Program Paths and Associated Costs for Symbolic Evaluation

Bernd Burgstaller, Bernhard Scholz, and Johann Blieberger



Salvador Dali, "Die Beständigkeit der Erinnerung"

Tour de Spec — A Collection of Spec95 Program Paths and Associated Costs for Symbolic Evaluation

Bernd Burgstaller, Bernhard Scholz, and Johann Blieberger

October 26, 2004

Abstract

In this paper we dissect the entire SPEC95 benchmark suite [CPU95] in order to derive the corresponding resource requirements for symbolic evaluation. Measurements are based on a series of metrics on path expressions that capture the control flow information contained in the underlying control flow graphs. Path expressions as well as the metrics data itself are computed by a data-flow framework. Our measurements show that symbolic evaluation is a methodology capable of coping with the considerable problem sizes that arise from contemporary real-world applications such as those from the SPEC95 benchmark suite.

1 Introduction

It's a contest in purposeless suffering.
— Lance Armstrong, six-times* Tour de France winner,
in "It's Not About the Bike — My Journey Back to Life"

Without doubt the Tour de France is one of cycling's most prestigious road races: it lasts for over three weeks, and it comprises the entire circumference of France, mountains included. Good climbing abilities are a major requirement for any cyclist competing for overall tour victory, because it is during those long and steep climbs that huge leads are opened, with everything to be be gained or lost.

This paper is about a tour across the SPEC95 benchmark suite, a collection of 18 applications that was originally set up to assess the performance of computer hardware. For any static program analysis methodology the attempt to process a program suite of such kind is comparable to the participation of a renowned cyclist in the Tour de France: the problem size is huge, involving a sample of benchmark programs hand-picked from the entire range of contemporary real-world applications, and, depending on the computational complexity of the chosen algorithms, the time to get the work done is key to the success of the undertaking. In this way the mountains of the Pyrenees and Alps find their

*At the time of writing.

counterpart in the control flow graphs of the benchmark programs; Like mountains, these graphs can also pile up to insurmountable heights, with inefficient climbers fading already in the foothills. The ability to accomplish such a Tour de Spec can therefore be considered as an indicator regarding the practicality of a static program analysis methodology.

In this paper we survey the SPEC95 benchmark suite with respect to the resource requirements needed for symbolic evaluation. Symbolic evaluation is an advanced static program analysis methodology in which symbolic expressions are used to denote the values of program variables and computations (cf. [Bur04]). Program paths are represented by path expressions, which were introduced by Robert E. Tarjan in [Tar81] (a summary of the material relevant for this paper can be found in the Appendix). We start this paper with a few pre-race preliminaries presented in Section 2. In Section 3 we define a data-flow problem along with a solution procedure to compute path expressions for a given CFG. The computational effort of symbolic evaluation with respect to a given path expression is a function on the number of contained *acyclic* program paths. Acyclic program paths are introduced in Section 4, together with a series of metrics that compute the number of acyclic program paths from path expressions. Section 5 contains a proof that these metrics compute the minimal result for reducible flowgraphs. Furthermore we present an adversary argument showing that for irreducible flowgraphs these path expressions are not minimal in the general case. Section 6 starts out with a description of the experiment conducted to collect our input-data from the SPEC95 benchmark suite. We then describe the validation of the collected data and the results of our evaluation. The collected data itself is presented in Section 7. We cross the finishing line of this Tour de Spec in Section 8, where we draw our conclusions.

A final note is due in this introductory section regarding Sections 3–6: these are excerpts from [Bur04] that have been included here in order to complement the presentation of the data collected during the above-mentioned experiment.

2 Preliminaries

Definition 2.1
A *regular expression* R is *unambiguous* if each string in $\sigma(R)$ is represented uniquely in R. A precise definition of the term "unique representation" follows.

Let $a_1, a_2, \ldots, a_m$ be the symbols appearing in the regular expression in their natural order, and let f be the function relating the a_i to the symbols in R. The expression R' then denotes the corresponding regular expression over the a_i. For example, if $R = (0+11)^*0$, then $m = 4$, $R' = (a_1 + a_2 \cdot a_3)^* a_4$, and $\mathrm{f}(a_1) = \mathrm{f}(a_4) = 0$, and $\mathrm{f}(a_2) = \mathrm{f}(a_3) = 1$. If $t \in \sigma(R)$, $t = s_1 s_2 \ldots s_l$, then there are legitimate ways of assigning to each s_i an a_j. In our example, if $t = 01100$, then a legitimate assignment is $t' = a_1 a_2 a_3 a_1 a_4$ where $\mathrm{f}(t') = t$. We say that t is denoted in exactly one way if there exists a unique legitimate assignment. Formally, $t \in \sigma(R)$ is denoted by R in exactly k ways if there are exactly k distinct strings $t_1, \ldots, t_k$ in $\sigma(R')$ with $\mathrm{f}(t_i) = t$. It is clear that this discussion applies only to nonempty strings.

Definition 2.2
An *automaton* $A = (Q, \Sigma, \delta, q_0, F)$ is *unambiguous* if for each $t \in \sigma(A)$, $t \neq \Lambda$,

there exists a unique path in the state transition diagram[†] of A from q_0 to a state in F.

Definition 2.3
An *automaton* $A = (Q, \Sigma, \delta, q_0, F)$ is *primitive* if no input symbol appears more than once in the state transition diagram. That is

$$(\forall s \in \Sigma)(\forall q \in Q)\big[\,|\delta(q,s)| \leq 1\big], \text{ and} \tag{1}$$

$$(\forall s \in \Sigma)(\forall q, q' \in Q)\big[\,(q = q') \vee (\delta(q,s) = \emptyset) \vee (\delta(q',s) = \emptyset)\big]. \tag{2}$$

Therefore, if A is primitive, $|\Sigma|$ is equal to the number of edges in the state transition diagram. Clearly, a primitive automaton is unambiguous.

Definition 2.4
Given an automaton $A = (Q, \Sigma, \delta, q_0, F)$, we can rename the states of A in such a way that the resulting automaton A' has the set of states $Q' = \{1, \ldots, n\}$ for some integer n. Let us use $R^k_{i,j}$ as the name of a regular expression whose language is the set of strings such that each string r denotes a path from state i to state j in A', with the additional constraint that r contains no *intermediate* state whose number is greater than k. Note that the endpoints i and j are not considered "intermediate" since we require that an intermediate state must be entered and *then* left. The following inductive definition constructs the expressions $R^k_{i,j}$ for A'.

Basis: The basis is $k = 0$. Since states are numbered such that $Q' = \{1, \ldots, n\}$, the restriction on paths is that they must not contain intermediate states at all. We distinguish two cases:

$i \neq j$**:**

$$R^0_{i,j} = \begin{cases} s_1 + \ldots + s_k & \text{if } s_1, \ldots, s_k \text{ are labels from state } i \text{ to state } j, \\ \emptyset & \text{if there are no labels from } i \text{ to } j \text{ in } A'. \end{cases}$$

$i = j$**:**

$$R^0_{i,j} = \begin{cases} \Lambda + s_1 + \ldots + s_k & \text{if } s_1, \ldots, s_k \text{ are self-labels of state } i, \\ \Lambda & \text{if there are no self-labels of state } i \text{ in } A'. \end{cases}$$

The empty string Λ of the latter case accounts for the path of length zero from state i to i itself.

Induction: For $k > 0$ we have

$$R^k_{i,j} = R^{k-1}_{i,j} \; + \; R^{k-1}_{i,k} \, (R^{k-1}_{k,k})^* \, R^{k-1}_{k,j}.$$

Informally, the definition of $R^k_{i,j}$ above means that a path from state i to state j in A' that does not pass through an intermediate state higher than k is either

1) in $R^{k-1}_{i,j}$ (in which case it never passes through an intermediate state as high as k); or

[†]Confer [HU79, p.16].

2) composed of a path in $R_{i,k}^{k-1}$ (which takes A' from state i to state k for the first time), followed by a path in $(R_{k,k}^{k-1})^*$ (which takes A' zero or more times from state k to k itself without passing through a state higher than k), followed by a path in $R_{k,j}^{k-1}$ (which takes A' from state k to state j).

Definition 2.5
Given an automaton $A = (Q, \Sigma, \delta, q_0, F)$ and a set $M \subseteq Q$, we let $R_{i,j}^{M}$ denote the regular expression whose language is the set of strings r such that every path from state i to state j in A is represented by a string r (and vice versa), with the additional constraint that r must not contain *intermediate* states $q \notin M$. In particular, the regular expression with the empty set, $R_{i,j}^{\emptyset}$, denotes the language of strings without intermediate states and hence amounts to the set of direct edges from state i to state j (the empty set is a subset of every set, that is, $\emptyset \subseteq S$, whenever S is a set (cf. [Ros95, pp.40])). In this way we have $\sigma(R_{i,j}^{\emptyset}) \subseteq \sigma(R_{i,j}^{M})$, since every direct edge is considered a path from from state i to state j with zero intermediate states $q \in M$. Note that the endpoints i and j are not considered "intermediate" since we require that an intermediate state must be entered and *then* left.

Definition 2.6
A *control flow graph (CFG)* is a directed labeled graph $G = \langle N, E, n_e, n_x \rangle$ with node set N and edge set $E \subseteq N \times N$. *Entry (n_e)* and *Exit (n_x)* are distinguished nodes used to denote the start and terminal node. The start node n_e has no incoming edges ($\mathrm{in}(n_e) = \emptyset$), whereas the terminal node n_x has no outgoing edges ($\mathrm{out}(n_x) = \emptyset$). Furthermore we require that every node n is contained in a path from n_e to n_x. Note that since the set E of edges consists of ordered pairs of elements of N, the underlying undirected graph is *simple*.

The set of all successors of a node $n \in N$ is denoted by $\mathrm{Succ}(x)$, while the set of all predecessors of n is denoted by $\mathrm{Pred}(x)$.

Lemma 1. We can view a control flow graph $G = \langle N, E, n_e, n_x \rangle$ as the state transition diagram of an automaton $A_G = (N, E, \delta, n_e, \{n_x\})$, where the nodes of G correspond to the states of the automaton. Start node n_e and terminal node n_x denote the start- and accepting state of A_G. The edge set E corresponds to the alphabet of A_G, with the transition function

$$\delta(n, e) = m \Leftrightarrow (\mathrm{h}(e) = n \wedge \mathrm{t}(e) = m).$$

There are two direct consequences of the fact stated in Definition (2.6), that the set E of edges of a CFG consists of ordered pairs of elements of N.

(1) The transition function δ is a partial function.

(2) The automaton A_G is a primitive (and hence deterministic) automaton.

Thus for every CFG in the sense of Definition (2.6) there exists a corresponding primitive automation. The converse is not true since we require that a CFG has a single terminal node.

Definition 2.7
In a CFG, a node x *dominates* another node y iff all paths from the entry

node n_e to y always pass through x. We write $x \operatorname{dom} y$ to indicate that x dominates y. If $x \operatorname{dom} y$ and $x \neq y$, then x strictly dominates y, denoted by $x \operatorname{stdom} y$. A node x is said to *immediately dominate* another node y, denoted as $x = \operatorname{idom}(y)$, if $x \operatorname{stdom} y$ and there is no other node $z \neq x$ and $z \neq y$ such that $x \operatorname{stdom} z \operatorname{stdom} y$. The dominance relation is reflexive and transitive, and can be represented by the so-called *dominator tree*. An edge $e = (x, y)$ is an edge in the dominator tree of a CFG iff $x = \operatorname{idom}(y)$. Given a node x in the dominator tree, we define $\operatorname{sub}(x)$ to be the set of nodes of the dominator subtree rooted at x.

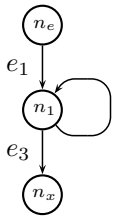
3 Path Expression Generation

In this section we define a forward data-flow problem along with a solution procedure that allows us to compute path expressions for a given CFG. The data-flow problem is based on the following equations.

$$R(n_e) = \Lambda \tag{3}$$

$$R(n) = \underset{n' \in \operatorname{Preds}(n)}{+} \left[R(n') \cdot R^{\emptyset}_{n',n} \right] \tag{4}$$

Recall that in Definition (2.6) on page 4 we required that $\operatorname{in}(n_e) = \emptyset$, so the empty string Λ is indeed the only regular expression that takes us from node n_e to n_e itself.



$$R(n_e) = \Lambda$$

$$R(n_1) = R(n_e) \cdot e_1 + R(n_1) \cdot e_2$$

$$R(n_e) = R(n_1) \cdot e_3$$

Figure 1: CFG with Infinite Number of Program Paths

Due to Equation(4) this data-flow problem cannot be solved by an iterative algorithm in the presence of loops. As an example consider the CFG and its associated data-flow equations depicted in Figure 1. The regular expression $e_1 \cdot e_2^* \cdot e_3$ is a valid path expression of type (n_e, n_x). Iterative algorithms fail to compute this expression once they attempt to determine the solution for the data-flow equation at node n_1. This is due the fact that the underlying lattice for this data-flow problem is infinite.

Application of an elimination-based algorithm can however solve this data-flow problem. Blieberger [Bli02, Section 3] mentions several elimination algorithms that exploit the structure of flowgraphs for improved time complexity. For our purposes we use Sreedhar's algorithm presented in [Sre95]. As pointed out in [Pau88], we define a normal form for our equations and set up a loop-breaking rule that allows us to handle circularities of the before-mentioned kind.

A data-flow equation E for our path expression data-flow problem is in

normal form, if it has the form

$$E : R(n) = \begin{cases} \underset{m \in M \subseteq \mathrm{sub}(\mathrm{idom}(n))}{+} \left[\, R(m) \cdot R_{m,n}^{S_m \subseteq \mathrm{sub}(m) \setminus \{m\}} \,\right] & \text{if } n \neq n_e, \\ \Lambda & \text{else.} \end{cases} \tag{5}$$

Equation (5) subsumes data-flow equations (3) and (4), which is not obvious for the latter equation. To explore this normal form in full detail we need to take into account the properties of our elimination algorithm (in fact the normal form of Equation (5) is tailored to the needs of this algorithm). Since the elimination algorithm is subject of Section 5.1 (pp. 16), full treatment of Equation (5) is postponed until then. In the meantime we only note that, contrary to Equation (4), it is possible for a data-flow equation in normal form, to depend on other nodes than its control flow predecessors. This is due to substitutions that occur during elimination (cf. [Pau88]). It can furthermore happen that due to substitutions the left-hand side of an equation E_i occurs several times on the right-hand side of an equation E_j, e.g.,

$$\begin{aligned} E_i &: R(n_i) = R(n_k) \cdot e_1 \\ E_j &: R(n_j) = R(n_i) R_{n_i,n_j}^{S_1} + \cdots + R(n_i) R_{n_i,n_j}^{S_l}. \end{aligned}$$

In such a case we employ the following rewrite-rule based on the distributive law of regular expression algebra (cf. e.g. [Sal66]) in order to transform E_j to normal form.

$$R_1 \cdot R_2 + R_1 \cdot R_3 \longrightarrow R_1 \cdot (R_2 + R_3) \tag{6}$$

Resuming the above example, we get $R(n_j) = R(n_i) \cdot (R_{n_i,n_j}^{S_1} + \cdots + R_{n_i,n_j}^{S_l})$ for the data-flow equation E_j.

It is the purpose of the *loopbreaking rule* to replace a data-flow equation E which depends on itself by an equivalent equation e for which the left-hand side of E does not appear on the right-hand side. Assume a data-flow equation for a given node n, for which the set S_n is defined as

$$S_n \subseteq \mathrm{sub}\big(\mathrm{idom}(n)\big) \setminus \big(\{\mathrm{idom}(n)\} \cup \mathrm{sub}(n)\big).$$

Then the data-flow equation under consideration is

$$E : R(n) = \underbrace{R(\mathrm{idom}(n)) \cdot R_{\mathrm{idom}(n),n}^{S_n}}_{t_1} + \underbrace{R(n) \cdot R_{n,n}^{S'_n \subseteq \mathrm{sub}(n) \setminus \{n\}}}_{t_2}. \tag{7}$$

Clearly this data-flow equation for node n contains a dependency on the immediate dominator of n, constituted by the term t_1. Moreover, it contains a dependency on n itself, constituted by the term t_2. By further dissecting terms t_1 and t_2, we arrive at

$$t_1 = R(\mathrm{idom}(n)) \cdot \underbrace{R_{\mathrm{idom}(n),n}^{S_n}}_{t_{1_1}}, \text{ and } t_2 = R(n) \cdot \underbrace{R_{n,n}^{S'_n \subseteq \mathrm{sub}(n) \setminus \{n\}}}_{t_{2_1}}.$$

Therein term t_{1_1} constitutes all program paths from $\mathrm{idom}(n)$ to n itself, possibly via intermediate nodes $n' \in S_n$. According to the definition of set S_n, no

program path of term t_{1_1} is allowed to contain idom(n), n, or a node dominated by n as an intermediate node. Term t_{2_1} constitutes all program paths from node n back to n itself, possibly via nodes strictly dominated by n.

We define our loopbreaking rule to replace Equation (7) by the equation

$$e : R(n) = \underbrace{R(\text{idom}(n)) \cdot R^{S_n}_{\text{idom}(n),n}}_{t_1} \cdot \underbrace{\left(R^{S'_n \subseteq \text{sub}(n)\setminus\{n\}}_{n,n}\right)^*}_{t_3}. \tag{8}$$

4 Program Path Metrics

In this section we define a series of metrics aiming at the determination of the space-requirements that can be expected for symbolic evaluation of programs. Programs are given as control flow graphs from which path-expressions are constructed. A path-expression of type (n_i, n_j) represents all program paths from node n_i to node n_j in the underlying control flow graph (CFG). Metrics are calculated by reinterpreting the operations $+$, $\cdot$, and * that are used to construct path expressions (cf. [Tar81]).

The number of program paths through a CFG advances from finite to infinite with the introduction of cycles. The symbolic evaluation method described in [Bur04] provides an abstraction mechanism that keeps the space requirements that are due to cycles finite. This is due to the fact that symbolic evaluation is based on *acyclic* program paths.

The term "acyclic program path" has already been coined in the literature, e.g., for path profiling ([BL96]), but we use a different approach of partitioning a given flowgraph into acyclic subgraphs. Moreover, our approach is based on path-expressions rather than on the flowgraphs themselves.

The metrics that we develop in the course of this section calculate the number of acyclic program paths through a given flow graph $G = \langle N, E, n_e, n_x\rangle$ from a path expression $R^N_{n_e,n_x}$ representing all program paths in G from the entry node n_e to its exit node n_x. Strictly speaking, these metrics determine the natural loops (cf. [ASU86, Section 10.4]) in $R^N_{n_e,n_x}$. The acyclic program paths then follow from the acyclic condensation (cf. [ZC91]) of these loops across all nesting levels.

Irreducible loops do not have the property of natural loops that they possess a unique header. In this way irreducibility introduces a kind of indeterminism in the path-expression generation data-flow problem of Section 3, because one of the possible loop entry nodes has then to be promoted as loop header. For the algorithms that calculate our metrics this indeterminism is of no concern since an operand of the * operator has a unique header per definition. As we will see in Section 5, this indeterminism will however be of concern with respect to the minimality of these metrics.

4.1 Loops as Black-Boxes (npp)

For the number of acyclic program paths that have to be considered by symbolic evaluation, the following metric calculates a lower bound. It treats subexpressions that correspond to cycles (loops) as black boxes with a resource-

requirement of 1. Apart from that, cycles (loops) are not considered further by this metric.

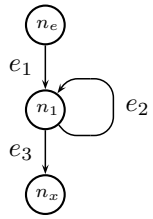


Figure 2: CFG with Infinite Number of Program Paths

For an example of a CFG with a cycle and hence an infinite number of program paths consider the graph depicted in Figure 2. The program paths from node n_e to node n_x are defined by the path expression $e_1 \cdot e_2^* \cdot e_3$ of type (n_e, n_x). The subexpression e_2^* corresponding to the cycle in Figure 2 accounts for the infinite number of program paths.

We can now define a mapping that allows us to compute the number of acyclic program paths (npp) from regular expressions representing sets of program paths. With the inductive definition below, P denotes a path (sub)expression, and e denotes an arbitrary CFG-edge.

$$\text{npp}(\Lambda) = 0, \tag{9}$$

$$\text{npp}(\emptyset) = 0, \tag{10}$$

$$\text{npp}(e) = 1, \tag{11}$$

$$\text{npp}(P_1 + P_2) = \text{npp}(P_1) + \text{npp}(P_2), \tag{12}$$

$$\text{npp}(P_1 \cdot P_2) = \text{npp}(P_1) * \text{npp}(P_2), \tag{13}$$

$$\text{npp}(P_1^*) = 1. \tag{14}$$

Equation (14) of this mapping assigns cycles a resource-requirement of 1 and is for this reason responsible for the black-box view of this metric with respect to loops.

Applying this metric to the example of Figure 2, we get

$$\begin{aligned} \text{npp}(e_1 \cdot e_2^* \cdot e_3) &= \text{npp}(e_1) * \text{npp}(e_2^*) * \text{npp}(e_3) \\ &= 1 * 1 * 1 \\ &= 1. \end{aligned} \tag{15}$$

A more elaborate example of a kite-shaped CFG is depicted in Figure 3. Equation (16) presents the path expression P of type (n_e, n_x). To facilitate reading we have used curly and square brackets in addition to the round brackets normally used for grouping regular expressions and path expressions.

$$\begin{aligned} P = {} & \overbrace{\big(\big(e_1 \cdot (e_2 \cdot e_3 + e_4)^* \cdot e_6\big) + \big(e_5 \cdot e_7\big)\big) \cdot e_{16}^*}^{R_1} \\ & \cdot \Big\{ \big[\overbrace{e_9 \cdot (((e_{10} \cdot e_{12}) + (e_{11} \cdot e_{17}^* \cdot e_{13})) \cdot e_{14})^*}^{R_2} \cdot \overbrace{(((e_{10} \cdot e_{12}) + (e_{11} \cdot e_{17}^* \cdot e_{13})) \cdot e_{15})}^{R_3} \big] \\ & + e_8 \Big\} \end{aligned} \tag{16}$$

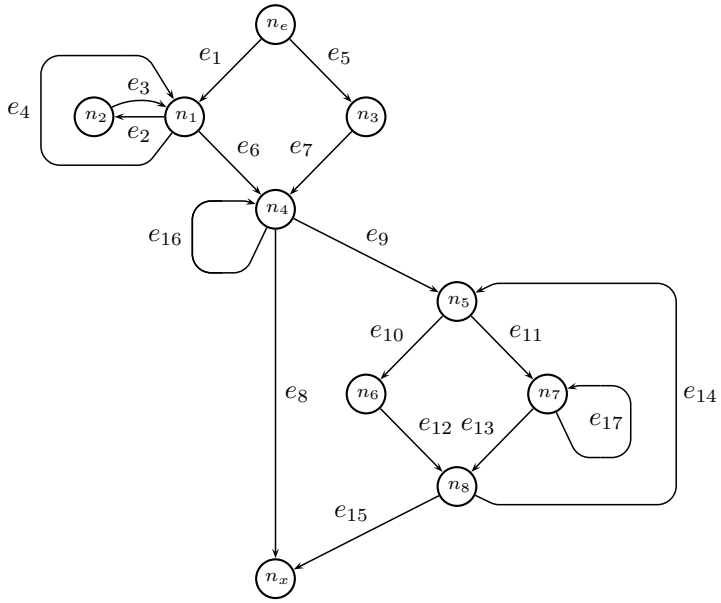


Figure 3: Elaborate Example: Kite-Shaped CFG

Applying the npp metric to the path expression of Equation (16), we get

$$\begin{aligned} \mathrm{npp}(P) &= \mathrm{npp}(R_1) * \big(\mathrm{npp}(R_2) * \mathrm{npp}(R_3) + \mathrm{npp}(e_8)\big) \\ &= 2 * (1 * 2 + 1) \\ &= 6. \end{aligned} \tag{17}$$

In relating this calculation to the CFG of Figure (3), it becomes clear that the npp-metric is only a lower bound for the resource-requirements needed for symbolic evaluation: the loops corresponding to path expressions e_{16}^*, e_{17}^*, $(e_2 \cdot e_3 + e_4)^*$ and $((e_{10} \cdot e_{12}) + (e_{11} \cdot e_{17}^* \cdot e_{13}) \cdot e_{14})^*$ are all estimated to have a resource requirement of 1. While this is true for loops e_{16}^* and e_{17}^* which both have only one path through its loop body, it underestimates the number of acyclic program paths for the latter two loop bodies. We will define metrics that compensate the under-estimate of the npp-metric in the next section.

4.2 Loop-Aware Path Metrics

Figure 4 contains a flowgraph with *nested* loops. The outer loop l_1 is along the program path $\pi_1 = \langle e_3, e_4 \rangle$, whereas the inner loop l_2 is along the path $\pi_2 = \langle e_2 \rangle$. Furthermore Figure 4 depicts two equivalent regular expressions R_1 and R_2 of type (s_e, s_x) for the respective flowgraph. From the npp-metric introduced in Section 4.1 we get

$$\mathrm{npp}(R_1) = \mathrm{npp}(R_2) = 1.$$

This result is an under-estimate in the sense that it does not take into account the subexpressions R_3 and R_4 (cf. Figure 4). This omission is due to the npp-rule

$$\mathrm{npp}(P_1^*) = 1$$

$$R_1 = e_1 \cdot (e_2^* \cdot e_3 \cdot e_4)^* e_2^* \cdot e_3 \cdot e_5$$

$$R_2 = e_1 \cdot (\underbrace{e_2}_{R_3})^* \cdot e_3 \cdot (\underbrace{e_4 \cdot e_2^* \cdot e_3}_{R_4})^* \cdot e_5$$

$$R_0 = e_1 \cdot (e_2 + e_3 \cdot e_4)^* \cdot e_3 \cdot e_5$$

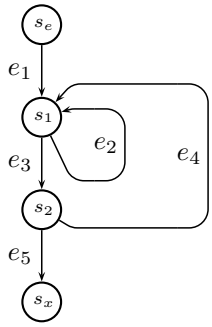


Figure 4: Example: Common Subexpression e_2^* in Path Expressions R_1 and R_2

of Equation (14). Informally, we can compensate this under-estimate if we add the cost for expressions R_3 and R_4 in terms of the npp-metric. Using path expression R_2 of Figure 4 we can calculate the precise result r as shown in the following equation.

$$\begin{aligned} r &= \mathrm{npp}(R_2) + \mathrm{npp}(e_2^*) + \mathrm{npp}(e_4 \cdot e_2^* \cdot e_3) \\ &= 1 + 1 + 2 \\ &= 4 \end{aligned} \tag{18}$$

Therein the first term corresponds to the original npp-metric, whereas the second and third term account for the calculation of the npp-values of subexpressions R_3 and R_4. On the other hand, if we take path expression R_0 of Figure 4, we get the following result r' in terms of our metric (observe that R_0 and R_2 are equivalent regular expressions).

$$\begin{aligned} r' &= \mathrm{npp}(R_0) + \mathrm{npp}(e_2 + e_3 \cdot e_4) \\ &= 1 + 2 \\ &= 3 \end{aligned} \tag{19}$$

From this example it becomes immediately clear that equivalent regular expressions need not yield the same result with respect to our metric. In this particular case it is the double occurrence of subexpression e_2^* in expression R_2 that imposes the extra cost of 1 compared to expression R_0. Regarding symbolic evaluation of a program corresponding to the CFG of Figure 4, the double occurrence of subexpression e_2^* in R_2 would mean that the loop represented by this subexpression is actually evaluated twice: once due to regular expression R_3, and once in the course of the evaluation of expression R_4.

We cannot expect that for each double occurrence there exists an equivalent regular expression without it. Moreover, the computational complexity of the required rewrite operations might be prohibitive. Double occurrences are not even restricted to loops — the edge e_3 occurs twice in expression R_2, and e_3 could in fact be replaced by an arbitrarily complex CFG.

In this way it is desirable to detect *common subexpressions* within path expressions in order to spend the effort for symbolic evaluation only once. Conceptually this corresponds to the introduction of so-called *meta-variables* to the algebra of path expressions. For example, if we define that meta-variable $M_1 = e_2^*$, then we get

$$R_2 = e_1 \cdot M_{1_a} \cdot e_3 \cdot (e_4 \cdot M_{1_b} \cdot e_3)^* \cdot e_5$$

for expression R_2 (note that we have used an additional level of subscripts to distinguish between the two occurrences of M_1 in the subsequent discussion). If, during symbolic evaluation of expression R_2, the meta-variable M_1 is encountered for the first time at M_{1_a}, the symbolic evaluation of the expression associated with M_1 will be carried out. For the second occurrence of meta-variable M_1 at M_{1_b} we can already use the result derived at M_{1_a}.

As already pointed out, the utilization of common regular subexpressions to reduce the effort associated with symbolic evaluation is not restricted to loops. In this way we might as well introduce an additional meta-variable for the edge e_3 in R_2. In the following discussion we will however focus on the use of common subexpressions representing loops.

Given the fact that regular expressions are not unique in the sense that *equivalent* regular expressions may describe the same set of program paths (cf. expressions R_1 and R_2 of Figure 4), we are faced with the problem of determining equivalence of regular expressions in order to spot common subexpressions. There exist algorithms that decide on the equivalence of regular sets (e.g., [Gin67, HU79, HMU01]), but the following two properties of the elimination algorithm of [Sre95] (cf. also Section 5.1) ensure that in order to show the equivalence of regular expressions representing loops it is sufficient to show their *identity*.

(1) Data-flow equations containing loops are inserted into other equations only after the path expression corresponding to the loop has been determined (this follows from the fact that the only possible remaining dependency for an equation at the time of insertion into other equations is one on its immediate dominator which clearly excludes dependencies on the loop body).

(2) A path expression corresponding to a loop body is not altered once it has been determined.

In other words, we can be sure that we will not encounter common subexpressions representing a loop body that are equivalent but not identical.

```
1   procedure Slice (P : PathExpr; S : in out Set) is
2   begin
3    if Match (P, RDot (a_, b_)) or Match (P, ROr (a_, b_))
4    then
5        Slice (a, S);
6        Slice (b, S);
7    elsif Match (P, RStar (a_)) then
8        S := S ∪ a;
9        Slice (a, S);
10   end if;
11  end Slice;
```

Figure 5: Slicing Procedure

The algorithm that calculates our metric for a given path-expression P consists of two steps. In the first step we extract natural loops P_1^* from P (P_1 denotes an arbitrary regular expression), and collect them in the set S. In the

second step we calculate the npp-metric for the path-expression P and for all elements in the set S and sum up the results. This two-step algorithm is depicted in pseudo-code in Figure 5 (Step 1) and Figure 6 (Step 2).

The slicing procedure of Figure 5 takes as input a path-expression P and a set S and recursively descends the syntax tree of P. It uses pattern matching (expressed through the call to procedure "Match" in lines 3 and 7) in order to determine the structure of P. In this particular case pattern matching is very simple since the matching decision is solely based on the operation symbol of the root node of the syntax (sub)tree of P. As a notational convention we append an *underscore* to a variable to denote the fact that this variable can match arbitrary regular expressions. On the contrary, the variable name without underscore denotes the corresponding matched regular expression. If during the recursive descent of expression P a natural loop is encountered (line 7), then the corresponding loop body is added to the set S (line 8).

Input:

a path-expression P

Output:

Metric (P)

(the loop-aware metric of acyclic program paths of P)

Algorithm:

$S := \{P\}$;

Slice (P, S);

Metric $(P) = \sum_{p \in S} \text{npp}(p)$;

Figure 6: Metric Algorithm

Note that the fact that S is a *set* ensures that common subexpressions denoting loop bodies are counted only once. If we declare S to be a multiset, with operator $\cup$ in line 8 of Figure 5 denoting the multiset union operation, than our metric counts each occurrence of a loop body. This leads us to the definitions of the following two loop-aware path metrics.

Definition 4.1
Given a path-expression P. With S being a multiset and $\cup$ the multiset union operation, the metric algorithm given in Figures 5 and 6 calculates a loop-aware npp-metric (cf. Section 4.1) that compensates the effort associated with a subexpression b that denotes a loop body for each occurrence of b in expression P. We call this metric **ncp**, the *number of considered acyclic program paths*.

Definition 4.2
With S being a set and $\cup$ the set union operation, the metric algorithm given in Figures 5 and 6 calculates a loop-aware npp-metric (cf. Section 4.1) that compensates the effort associated with a loop body only once. We call this metric **loncp** which stands for *loop-optimized* **ncp**.

Coming back to the kite-shaped CFG of Figure 3 on page 9, we can now calculate the ncp-metric for the corresponding path-expression P of type (s_e, s_x)

stated in Equation (16). Applying the algorithm of Figure 6, we get

$$\begin{aligned}\mathrm{ncp}(P) = &\ \mathrm{npp}(P)\\ &+ \mathrm{npp}(e_2 \cdot e_3 + e_4)\\ &+ \mathrm{npp}(e_{16})\\ &+ \mathrm{npp}((e_{10} \cdot e_{12} + e_{11} \cdot e_{17}^* \cdot e_{13}) \cdot e_{14})\\ &+ \mathrm{npp}(e_{17})\\ = &\ 13.\end{aligned} \tag{20}$$

Due to the double occurrence of the subexpression denoting the loop body e_{17}^*, the value that we get for the ncp-metric of expression P is one higher than the value calculated by the loncp-metric.

As already pointed out for the npp metric calculations of equations (18) and (19), equivalent regular expressions need not yield the same result with respect to the npp metric. Since the metrics ncp and loncp are based on the npp metric, they are also affected by this fact. Given a regular expression R, we often face the question whether there exists an equivalent regular expression R' that yields a lower value than R with respect to one of our metrics. Naturally this leads to the question whether a given regular expression R is *minimal* with respect to one of our metrics. In the following definition we specify the requirements for a regular expression to be npp-minimal. The definitions for the other two metrics are similar.

Definition 4.3
Let $\mathbb{A}$ denote the set of regular expressions. The equivalence relation $\sim \subseteq \mathbb{A} \times \mathbb{A}$ over regular expressions induces an equivalence class $[a]_\sim := \{a' \in \mathbb{A} \mid a \sim a'\}$ for each $a \in \mathbb{A}$. A given element a of an equivalence class $[a]_\sim$ is **npp-minimal**, if there exists no other element $b \in [a]_\sim$, $b \neq a$, such that $\mathrm{npp}(b) < \mathrm{npp}(a)$. Note that an npp-minimal element need not be *unique* in its equivalence class.

5 Loncp-Minimal Path-Expressions

In this section we show that the path-expressions generated by the data-flow framework defined in Section 3 are loncp-minimal for *reducible* flowgraphs. Finally we present an adversary argument showing that for *irreducible* flowgraphs these path expressions are not loncp-minimal in the general case.

We start out with a few lemmas that are needed to make our point. In Subsection 5.1 we show that the generated path expressions are *unambiguous*. This result is utilized in Subsection 5.2 to show that unambiguous path expressions are loncp-minimal for *reducible* flowgraphs. Subsection 5.3 contains the corresponding adversary argument for irreducible flowgraphs.

Lemma 2. Given two regular expressions $R_a = R_{x,y}^{M \cup \{y\}}$ and $R_b = R_{y,z}^{N \setminus \{y\}}$ over a primitive, Λ-free automaton $A = (Q, \Sigma, \delta, q_0, F)$. If the regular expressions R_a and R_b are unambiguous, then their concatenation, denoted by $R_a \cdot R_b$, is also unambiguous.

Proof. We utilize the concept of the proof of [BEGO71, Theorem 1]. According to Definition (2.3), a primitive automaton is unambiguous. Hence we

know from Definition (2.2) that no two different paths through automation A correspond to the same string $t \in \sigma(A)$, given that A is Λ-free.

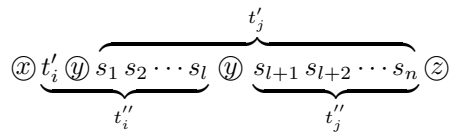


Figure 7: Decomposition of string $t = t'_i t'_j = t''_i t''_j$.

Assume, now, that string $t \in \sigma(R_a \cdot R_b)$ and that t can be decomposed in two ways: $t = t'_i t'_j = t''_i t''_j$, where $t'_i, t''_i \in \sigma(R_a)$, and $t'_j, t''_j \in \sigma(R_b)$. If $t'_i < t''_i$, then $t''_i = t'_i s_1 s_2 \cdots s_l$, where $l \geq 1$ and $s_k \in \Sigma$, $1 \leq k \leq l < n$. Figure 7 depicts string t and its corresponding decompositions. Note that the positions where t passes states x, y, and z have been marked with ⓧ,ⓨ, and ⓩ.

It follows from $t''_i \in \sigma(R_a)$, that in the state transition diagram of A, s_l is an edge entering state y. However, s_l is also a symbol in t'_j, and $t'_j \in \sigma(R_b)$ does not contain paths with intermediate state y. Thus, $t'_i = t''_i$, and consequently, $t'_j = t''_j$. According to the initial assumption that R_a and R_b are unambiguous, t'_i is denoted in exactly one way in R_a, and t'_j is denoted in exactly one way in R_b. As a consequence, $t = t'_i t'_j$ is denoted in exactly one way in $R_a \cdot R_b$. □

Corollary 1. Given two regular expressions $R_a = R_{x,y}^{M \setminus \{y\}}$ and $R_b = R_{y,z}^{N \setminus \{y\}}$ over a primitive, Λ-free automaton $A = (Q, \Sigma, \delta, q_0, F)$. If the regular expressions R_a and R_b are unambiguous, then their concatenation, denoted by $R_a \cdot R_b$, is also unambiguous.

The Corollary is a special case of Lemma (2) where the regular expression R_a must not contain an intermediate state y. Hence the decomposition $t = t''_i t''_j$ depicted in Figure 7 is illegal due to t''_i containing the intermediate state y. In fact there exists only one possible decomposition for $t \in \sigma(R_a \cdot R_b)$, namely $t = t'''_i t'''_j$, where $t'''_i \in \sigma(R_a)$, and $t'''_j \in \sigma(R_b)$. □

Lemma 3. Given a regular expression $R = R_{y,y}^{M \setminus \{y\}}$ over a primitive, Λ-free automaton $A = (Q, \Sigma, \delta, q_0, F)$. If the regular expression R is unambiguous, then the regular expression R^*, denoting the closure of $\sigma(R)$, is also unambiguous.

Proof. From the inductive definition of languages described by regular expressions (cf. Case (4), on p. 285), we have

$$\sigma(R^*) = (\sigma(R))^* = \bigcup_{k=0}^{\infty} \sigma(R)^k.$$

Therein $\sigma(R)^0 = \{\Lambda\}$, and $\sigma(R)^k = \sigma(R)^{k-1} \cdot \sigma(R)$. But it also holds that

$$\bigcup_{k=0}^{\infty} \sigma(R)^k = \sigma\big(\mathop{+}_{k=0}^{\infty} R^k\big),$$

where $R^0 = \Lambda$, and $R^k = R^{k-1} \cdot R$ (proof by induction on the number of operation symbols omitted). We can then prove the lemma in two steps.

(1) First we show that the regular expressions $R^k = \big(R_{y,y}^{M \setminus \{y\}}\big)^k$, for $k \geq 0$, are unambiguous.

(2) Based on (1) we show that the regular expression $R^* = \overset{\infty}{\underset{k=0}{+}} R^k$ is unambiguous.

AD (1): By structural induction on the unambiguity of R^k.

Basis: We use $k = 0$, $k = 1$, and $k = 2$ as base cases. $R^0 = \Lambda$ is unambiguous since it represents the empty string in exactly one way. Furthermore, $R^1 = R^0 \cdot R = \Lambda \cdot R = R$ is unambiguous due to the initial assumption of this lemma. $R^2 = R^1 \cdot R = R \cdot R$ is unambiguous due to Corollary (1) if we substitute state y for state x and state z.

Induction: Let $R^{k+1} = R^k \cdot R$, for $k \geq 2$, be a regular expression built by the inductive step of the definition. The regular expression R is unambiguous due to the initial assumption of this lemma. For the inductive step of the proof we may assume that R^k is unambiguous, too. It remains to show that $R^k \cdot R$ is an unambiguous regular expression.

The essential observation therein is that, although $R = R_{y,y}^{M\setminus\{y\}}$ is a regular expression without intermediate state y, the concatenation of $R_{y,y}^{M\setminus\{y\}}$ with itself yields a regular expression that contains intermediate state y. In this way R^k, for $k \geq 2$, qualifies as expression R_a in Lemma (2). Note that, strictly speaking, Definition (2.5) requires $R_a = R_{x,y}^{M\cup\{y\}}$ to contain *all* paths from state x to state y. This is not the case for R^k, since

$$\sigma(R^k) = \sigma(\big(R_{y,y}^{M\setminus\{y\}}\big)^k) \subseteq \sigma(\big(R_{y,y}^{M\setminus\{y\}}\big)^*) = \sigma(R_{y,y}^{M\cup\{y\}}).$$

The proof of Lemma (2) however includes also this sub-case. Substituting R for R_b in Lemma (2) finally shows that $R^k \cdot R$ is indeed unambiguous.

AD (2): From step (1) we already know that every term R^k in

$$R^* = \overset{\infty}{\underset{k=0}{+}} R^k$$

is unambiguous. In order to establish that R^* is unambiguous it remains to show that $\sigma(R^i) \cap \sigma(R^j) = \emptyset$, for $0 \leq i, j \leq \infty$ and $i \neq j$. Observe that $\sigma(R^0) \cap \sigma(R^j) = \emptyset$, for $0 \leq j \leq \infty$, since $\sigma(R^0) = \{\Lambda\}$, and R (and hence R^j) are Λ-free due to A being Λ-free. Moreover, $\sigma(R^k) \cap \sigma(R^{k+j}) = \emptyset$, for $k \geq 1$ and $j > 1$, since $\sigma(R^k)$ contains only paths that pass intermediate state y exactly $k-1$ times, whereas $\sigma(R^{k+j})$ contains only paths that pass intermediate state y exactly $k+j-1$ times.

□

Lemma 4. Given two regular expressions $R_a = R_{x,y}^{M\setminus\{y\}}$ and $R_b = R_{y,y}^{N\setminus\{y\}}$ over a primitive, Λ-free automaton $A = (Q, \Sigma, \delta, q_0, F)$. If the regular expressions R_a and R_b are unambiguous, then the regular expression denoted by $R_a \cdot R_b^*$ is also unambiguous.

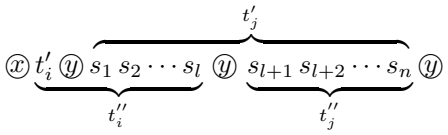


Figure 8: Decomposition of string $t = t'_i t'_j = t''_i t''_j$.

Proof. According to Definition (2.3), a primitive automaton is unambiguous. Hence we know from Definition (2.2) that no two different paths through automation A correspond to the same string $t \in \sigma(A)$, given that A is Λ-free.

Assume, now, that string $t \in \sigma(R_a \cdot R_b)$ and that t can be decomposed in two ways: $t = t'_i t'_j = t''_i t''_j$, where $t'_i, t''_i \in \sigma(R_a)$, and $t'_j, t''_j \in \sigma(R_b^*)$. If $t'_i < t''_i$, then $t''_i = t'_i s_1 s_2 \cdots s_l$, where $l \geq 1$ and $s_k \in \Sigma$, $1 \leq k \leq l < n$. Figure 8 depicts string t and its corresponding decompositions. Note that the positions where t passes states x and y have been marked with ⓧ and ⓨ.

It follows from $t'_i \in \sigma(R_a)$, that in the state transition diagram of A, s_1 is an edge emanating from state y. However, s_1 is also a symbol in t''_i, and $t''_i \in \sigma(R_a)$ does not contain paths with intermediate state y. Thus, $t'_i = t''_i$, and consequently, $t'_j = t''_j$. According to the initial assumption that R_a is unambiguous, t'_i is denoted in exactly one way in R_a. From the initial assumption that R_b is unambiguous and from Lemma (3) we know that t'_j is denoted in exactly one way in R_b^*. As a consequence, $t = t'_i t'_j$ is denoted in exactly one way in $R_a \cdot R_b$. □

5.1 Unambiguity of Path Expressions

Lemma 5. The path-expressions for a given reducible CFG as generated by the data-flow framework described in Section 3 are unambiguous in general.

Proof. The elimination algorithm described in [Sre95] operates on the DJ graph of a flowgraph rather than the flowgraph itself. The first step of this proof is therefore to show how the data-flow equations (3) and (4) can be applied to a DJ graph without affecting the underlying CFG-based data-flow problem. The elimination algorithm itself comprises two phases. The *elimination phase* performs DJ graph reduction and variable substitution of the equation system until the DJ graph is reduced to its dominator tree. After the first phase the equation at every node is expressed only in terms of its immediate dominator, with the root node n_e being the only exception (cf. Equation (3)). The second phase of the algorithm is concerned with the *propagation* of the solution at the root-node in a top-down fashion along the dominator tree to compute the solutions at the other nodes. In this way the latter two parts of the proof establish that both phases, *elimination* and *propagation*, leave us with unambiguous path expressions.

DJ Graph-Based Path Expression Generation. As pointed out in [Sre95, Corollary 3.1], we can construct a DJ graph from a flowgraph G by adding every missing *immediate dominance edge* (x, y) if this edge is not already present in G. As a consequence a DJ graph can contain more edges than the corresponding CFG.

Data-flow equations (3) and (4) are applicable to a DJ graph if we require that for an edge $e = (x, y)$ that is part of the CFG but not of the DJ graph,

$$R^{\emptyset}_{x,y} = \emptyset. \tag{21}$$

This ensures that edge e will not affect our CFG-based data-flow problem since the set of paths from node x to node y considered for path expression generation is now the empty set. Note that this is in line with Definition (2.5).

It is also worth mentioning that a DJ graph can be viewed as a primitive automation, which is a result of the following two facts.

1) According to Lemma (1), the underlying CFG can be viewed as a primitive automaton.
2) The immediate dominance edges that are added during transformation of the CFG to the DJ graph are unique, hence they occur only once as an input symbol in the state transition diagram derived from the DJ graph, as required by Definition (2.3).

Elimination. In this part of the proof we establish that the elimination phase of the elimination algorithm described in [Sre95] results in data-flow equations

$$E : R(n) = \begin{cases} R(\text{idom}(n)) \cdot R^{S_n \subseteq \text{sub}(\text{idom}(n)) \setminus \{\text{idom}(n)\}}_{\text{idom}(n),n} & \text{if } n \neq n_e, \\ \Lambda & \text{else,} \end{cases} \tag{22}$$

where

$$R^{S_n \subseteq \text{sub}(\text{idom}(n)) \setminus \{\text{idom}(n)\}}_{\text{idom}(n),n} \tag{23}$$

is an *unambiguous* path expression describing all program paths from node $\text{idom}(n)$ to node n that do not contain *strict* dominators of node n as intermediate nodes. We will achieve this result by structural induction on Property (5.1) below.

It is worth mentioning that Equation (22) is of the normal form defined in Equation (5). It is however more specific in the sense that it depends only on its immediate dominator (elimination has *reduced* it to a dependency on the immediate dominator). Taking this reduction into account, we will refer to the resulting normal form as the *reduced* normal form. Consequently, the form of Equation (5) is then referred to as the *general* normal form. Reduced normal form implies general normal form.

Property 5.1
Data-flow equations generated during the elimination phase are in general normal form, with the additional constraint, that the regular expressions

$$R^{S_m \subseteq \text{sub}(m) \setminus \{m\}}_{m,n}$$

contained in those normal-formed equations are unambiguous.

Note that, at first sight, Property (5.1) seems too weak since it requires only general normal form, whereas Equation (22) is in reduced normal form. It is however a property of the elimination algorithm that after elimination each equation is reduced to a dependency on its immediate dominator. In this way reduced normal form is already a consequence of elimination and need not be taken care of by our proof!

Basis: Our base case is is the system of data-flow equations we derive once we transform the CFG to the DJ graph and apply Equations (3) and (4) under the constraint specified in Equation (21). The resulting data-flow equations fulfill Property (5.1), since Equations (3) and (4) are in normal form, and the regular expressions

$$R^{\emptyset}_{n',n}, \text{ with } n' \in \text{Preds}(n)$$

contained in those normal-formed equations are unambiguous due to the automaton corresponding to a DJ graph being a primitive automaton.

Induction: For the inductive step, we may assume that our system of data-flow equations fulfills Property (5.1). We have now to consider the effects of the *ϵ-rules* E1 and E2 (cf. [Sre95, Section 4.1]) that carry out DJ graph reduction and variable substitution in order to show that the resulting equations again fulfill Property (5.1). This lengthy part of the proof is contained in Sections 5.1.1 (Lemma (7)) and Section 5.1.2 (Lemma (8)).

Propagation. Elimination leaves us with data-flow equations in reduced normal form as stated in Equation (22). Propagation is based on the principle that the concatenation of two path expressions R_1 and R_2, where R_1 denotes all program paths from node n_e to a given node idom(n), and R_2 denotes all program paths from node idom(n) to node n, results in a path expression $R_1 \cdot R_2$ that denotes all program paths from node n_e to n. Equation (23) states the side-condition that R_2 must not contain *strict* dominators of node n as intermediate nodes.

Lemma 6. Propagation of the solution $R(n_e)$ along the dominator tree results in unambiguous path expressions.

Proof. By structural induction on the unambiguity of the involved regular expressions.

Basis: For the base case we note that $R(n_e) = \Lambda$ denotes the empty string that corresponds to all program paths from node n_e to node n_e itself unambiguously. We can then propagate the solution at the root node to nodes $n_i \in \{n \mid \text{idom}(n) = n_e\}$ that are immediately dominated by it. Thus we get the following result

$$\begin{aligned} R(n_i) &= R(n_e) \cdot R^{S_{n_i} \subseteq \text{sub}(n_e) \setminus \{n_e\}}_{n_e,n_i} \\ &= \Lambda \cdot R^{S_{n_i} \subseteq \text{sub}(n_e) \setminus \{n_e\}}_{n_e,n_i} \\ &= R^{S_{n_i} \subseteq \text{sub}(n_e) \setminus \{n_e\}}_{n_e,n_i}, \end{aligned}$$

which is unambiguous due to Property (5.1).

Induction: For the inductive step, we may assume that in the equation

$$R(n) = \underbrace{R^{S'_n \subseteq \mathrm{sub}(n_e)\setminus\{n_e\}}_{n_e,\mathrm{idom}(n)}}_{t_1} \cdot \underbrace{R^{S_n \subseteq \mathrm{sub}(\mathrm{idom}(n))\setminus\{\mathrm{idom}(n)\}}_{\mathrm{idom}(n),n}}_{t_2}$$

term t_1 is unambiguous as it is the result of earlier propagations. Term t_2 is unambiguous due to Property (5.1). The unambiguity of $t_1 \cdot t_2$ follows then directly from Lemma (2) on page 13. □

The result that the propagation phase of our elimination algorithm leaves us with unambiguous path-expressions concludes also the proof of Lemma (5). □

5.1.1 Elimination Rule E1

Figure 9 illustrates the elimination of edge (y, y) due to an application of ϵ-rule E1. Dashed lines represent dominator edges, and solid lines represent join edges.

Edge (y, y) corresponds to a circular dependency and is therefore exactly the case for which the loopbreaking rule of our data-flow problem (cf. Section 3) has been designed. Equation (7) on page 6 holds for node y before the loopbreaking rule is applied, and Equation (8) holds afterwards.

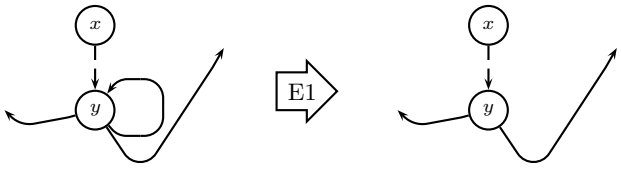


Figure 9: Graphical Illustration of Rule E1 for the Substitution "$y \to y$".

When setting up Property (5.1) on page 17, we concluded that it was sufficient to require general normal form, since the utilized elimination algorithm would, during its elimination phase, eventually reduce equations to reduced normal form. Figure 9 bears witness to this fact: before the application of the loopbreaking rule, the equation at note y is in general normal form, since it depends on its immediate dominator (node x), and on y itself. The loopbreaking rule removes the dependency on node y which reduces the equation to reduced normal form. Without the circular dependency at node y this would already have happened during an application of rule E2 that we will discuss in the next section. We conclude this section with the proof of the following lemma.

Lemma 7. The data-flow equation stated in Equation (8) fulfills Property (5.1).

Proof. As already observed, Equation (8) is in reduced normal form which implies general normal form. What remains to be checked for Property (5.1) is that the contained regular expressions have to be unambiguous. Unambiguity follows directly from Lemma (4) on page 15. □

5.1.2 Elimination Rule E2

The elimination algorithm described in [Sre95] actually contains two rules named E2a and E2b subsumed to rule E2. As pointed out in [Sre95, Section 10.4.1], the distinction between these two E2 rules is only minor. Our proof therefore considers only rule E2b, with the provision that the proof for rule E2a follows along the same line.

Figure 10 illustrates the elimination of edge (y, z) due to an application of ϵ-rule E2b. This corresponds to a substitution of the data-flow equation at node y into the equation at node z. The dotted line in Figure 10 from w to x represents the dominator tree path from w to x, dashed lines represent dominator edges, and solid lines represent join edges.

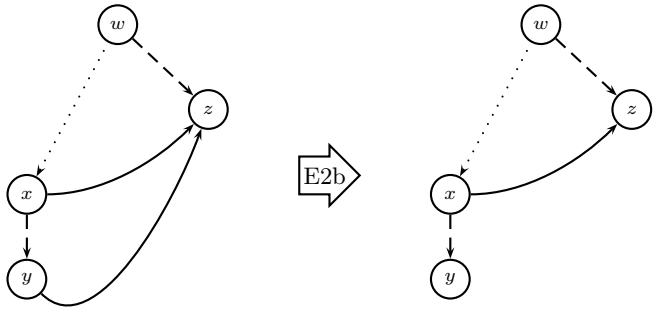


Figure 10: Graphical Illustration Rule E2b for the Substitution "$y \to z$".

Before application of ϵ-rule E2b we have the following equation at node z, which is also in *general* normal form.

$$\begin{aligned} R(z) = & \sum_{\substack{n \in S \subseteq \mathrm{sub}(\mathrm{idom}(z)) \\ \wedge n \notin \{x\} \cup \mathrm{sub}(y)}} R(n) \cdot R_{n,z}^{S_n \subseteq \mathrm{sub}(n) \setminus \{n\}} \\ & + R(x) \cdot R_{x,z}^{S_x \subseteq \mathrm{sub}(x) \setminus \left(\{x\} \cup \mathrm{sub}(y)\right)} \\ & + R(y) \cdot R_{y,z}^{S_y \subseteq \mathrm{sub}(y) \setminus \{y\}} \end{aligned} \tag{24}$$

Therein the third term (last line) represents a dependency on the equation $R(y)$ at node y. This dependency is represented by the edge (y, z) in Figure 10. To be specific, the regular expression

$$R_{y,z}^{S_y \subseteq \mathrm{sub}(y) \setminus \{y\}}$$

denotes paths from node y to node z that may contain only intermediate nodes that are *strictly* dominated by y. Moreover, the second term in Equation (24) represents a dependency on the equation $R(x)$ at node x. This dependency is represented by the edge (x, z) in Figure 10. The regular expression

$$R_{x,z}^{S_x \subseteq \mathrm{sub}(x) \setminus \left(\{x\} \cup \mathrm{sub}(y)\right)}$$

denotes paths from node x to node z that may contain only intermediate nodes that are *strictly* dominated by x and *not* dominated by y. Hence we have

$$S_x \cap \left\{ S_y \cup \{y\} \right\} = \emptyset. \tag{25}$$

The property stipulated in Equation (25) holds due to the facts that the equation at node y has not yet been substituted into node z, and that no equation into which node y has been substituted before this substitution has already been substituted into node z.

The data-flow equation at node z may furthermore contain dependencies on nodes as specified through the first term of Equation (24), but these dependencies have been omitted from Figure 10.

Given the data-flow equation $R(y)$ at node y,

$$R(y) = R(x) \cdot \underbrace{R_{x,y}^{S'_x \subseteq \mathrm{sub}(x) \setminus \{x\}}}_{t_1},$$

we may now perform the substitution $R(y) \to R(z)$, which yields

$$\begin{aligned} R(z) = & \sum_{\substack{n \in S \subseteq \mathrm{sub}(\mathrm{idom}(z)) \\ \wedge n \notin \{x\} \cup \mathrm{sub}(y)}} R(n) \cdot \underbrace{R_{n,z}^{S_n \subseteq \mathrm{sub}(n) \setminus \{n\}}}_{t_n} \\ & + R(x) \cdot \underbrace{R_{x,z}^{S_x \subseteq \mathrm{sub}(x) \setminus \left(\{x\} \cup \mathrm{sub}(y)\right)}}_{t_2} \\ & + R(x) \cdot \underbrace{R_{x,y}^{S'_x \subseteq \mathrm{sub}(x) \setminus \{x\}}}_{t_1} \cdot \underbrace{R_{y,z}^{S_y \subseteq \mathrm{sub}(y) \setminus \{y\}}}_{t_3}. \end{aligned} \tag{26}$$

Applying the rewrite-rule stated in Equation (6) on page 6, this can be brought to the general normal form

$$\begin{aligned} R(z) = & \sum_{\substack{n \in S \subseteq \mathrm{sub}(\mathrm{idom}(z)) \\ \wedge n \notin \{x\} \cup \mathrm{sub}(y)}} R(n) \cdot \underbrace{R_{n,z}^{S_n \subseteq \mathrm{sub}(n) \setminus \{n\}}}_{t_n} \\ & + R(x) \cdot \Big(\underbrace{R_{x,z}^{S_x \subseteq \mathrm{sub}(x) \setminus \left(\{x\} \cup \mathrm{sub}(y)\right)}}_{t_2} + \underbrace{R_{x,y}^{S'_x \subseteq \mathrm{sub}(x) \setminus \{x\}}}_{t_1} \cdot \underbrace{R_{y,z}^{S_y \subseteq \mathrm{sub}(y) \setminus \{y\}}}_{t_3} \Big). \end{aligned} \tag{27}$$

Lemma 8. The data-flow equation stated in Equation (27) fulfills Property (5.1).

Proof. As already observed, Equation (27) is in general normal form. What remains to be checked for Property (5.1) is that the contained regular expressions have to be unambiguous. It follows from the induction hypothesis that the regular expressions denoted by t_n and by $t_1 \cdots t_3$ are unambiguous. Moreover, the regular expression $t_1 \cdot t_3$ is unambiguous due to Lemma (2) on page 13. Finally, $t_2 + (t_1 \cdot t_3)$ is unambiguous since t_2 contains no path via y, and $t_1 \cdot t_3$ denotes only paths via y. Hence $\sigma(t_2) \cap \sigma(t_1 \cdot t_3) = \emptyset$. □

5.2 Reducible CFGs and Loncp-Minimality

Theorem. An unambiguous path-expression P of type (n_e, n_x) from a reducible CFG $G = \langle N, E, n_e, n_x \rangle$ is loncp-minimal in general.

Proof. If path-expression P is unambiguous, then each string in $\sigma(P)$ is represented uniquely in P (according to Definition 2.1 on page 2). Since each such

string corresponds to a program path through the flow graph G, each program path through G is in this case uniquely represented in P. We can now distinguish two cases.

(1) CFG G is acyclic. If G is acyclic, then the number of acyclic program paths as computed by the npp-metric corresponds to the actual number of program paths through G (proof by structural induction on the number of operation-symbols in P omitted). As the number of acyclic program paths cannot be lower than the number of program paths, we have established npp-minimality. Then loncp-minimality follows from the fact that for acyclic flowgraphs the npp- and the loncp-metric compute the same result.

(2) CFG G contains cycles. Due to G being irreducible, every cycle corresponds to a natural loop in G. It is the purpose of the slicing procedure depicted in Figure 5 on p. 11 to extract every natural loop, across all nesting levels. It is worth mentioning that a natural loop may contain other natural loops nested inside, and that these nested natural loops are not only extracted as part of the enclosing loop, but also as natural loops in their own respect (see also [Ram99]). When applying the npp-metric to a path-expression P corresponding to a natural loop nest, Equation (14) (p. 8) has the affect of condensing a nested loop to a single node. In this way the condensed version of P is again an acyclic graph to which Case (1) above applies. □

5.3 Irreducible CFGs and Loncp-Minimality

Theorem. Unambiguous path-expressions from irreducible CFGs are not loncp-minimal in general.

Proof. Indirect. Assume that unambiguous path-expressions from irreducible CFGs are loncp-minimal. Figure 11 depicts an irreducible CFG. For the unambiguous path-expression $R_1 = e_1 + (e_2 + e_1 \cdot e_4) \cdot (e_3 \cdot e_4)^* \cdot e_3$ of type (n_e, n_x) we have $\text{loncp}(R_1) = 4$. On the other hand, for the *equivalent* unambiguous path-expression $R_2 = (e_1 + e_2 \cdot e_3) \cdot (e_4 \cdot e_3)^*$ we get $\text{loncp}(R_2) = 3$. □

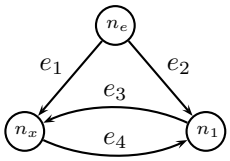


Figure 11: Potentially Loncp-Minimal Irreducible CFG.

As already pointed out, irreducible CFG-portions constitute loops that do not have a unique header. This not only separates them from reducible CFGs, but it also holds responsible for the above observation — depending on the node that we regard as the loop header (n_1 for R_1 vs. n_x for R_2), we get a different result from our loncp-metric.

6 Experimental Results

Finally I put to you for consideration by your discrete judgment that you should not wonder (but rather excuse me) if what I propound shall be not exactly comparable with the experiences and observations to be made by you and others, because there are may ways to err. One, almost inescapable, is some mistake in calculation; besides that, the smallness of these planets and their being observed with the telescope, which so greatly enlarges every object seen, means that an error even one second in the meetings and separations of these stars is made more apparent and noticeable than one thousand times as great an error in the aspects of other [naked eye] stars.
— Galileo Galilei, *Sunspot Letters*, Rome, 1613.

We shall now come to the description and evaluation of the experiments conducted on the SPEC95 benchmark suite. The following section describes the extraction of input-data and the basic setup of the experiment. Section 6.2 describes the validation of the data computed by our data-flow framework. In Section 6.3 we apply several data analysis methods from [Cle93] to evaluate the data collected in our experiment.

6.1 Basic Setup

Diagram (28) depicts the basic setup of our experiment. C- and Fortran source files (SRC) are input to the GCC compiler. The command line option "-dw" causes GCC to generate debug output (DBG) that contains, among other information, the control flow information of the input program. This option is only valid with optimization enabled (at least "-dw"), which in turn also affects the structure of the control flow information. We have used GCC 3.2 for the IA-32 architecture under RedHat 9.0 for this step.

$$\text{SRC} \xrightarrow{\text{GCC}} \text{DBG} \xrightarrow{\text{SCR}} \text{CFG} \xrightarrow{\text{FW}} R^N_{n_e,n_x} \xrightarrow{\text{M}} \text{Data} \qquad (28)$$

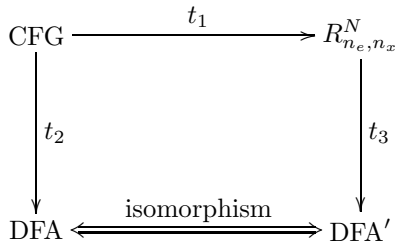
A combination of sed-, awk-, and perl scripts (SCR) is used to extract the control flow information from GCC's debug output in order to generate control flow graphs (CFG) that serve as input to our data-flow framework (FW).

It is the purpose of the data-flow framework to compute path expressions from control flow graphs according to the data-flow problem defined in Section 3. The metrics (M) defined in Section 4 have then been applied to these path expressions.

6.2 Validation

We have validated the implementation of our data-flow framework described in Section 3 in comparing the state transition diagrams corresponding to control flow graphs with the state transition diagrams corresponding to the generated path expressions. This approach is depicted in the following diagram which is

an extension of part of Diagram (28).



(29)

We take a given control flow graph CFG $= \langle N, E, n_e, n_x \rangle$ as input. According to Lemma 1 on page 4 we can view this graph as a deterministic finite automaton DFA (denoted by transformation t_2). Transformation t_1 denotes the conversion of the control flow graph to a path expression of type (n_e, n_x) by our data-flow framework (cf. also Diagram (28)). The generated path expression is denoted by $R^N_{n_e,n_x}$. Transformation t_3 is a compound transformation that comprises the following steps: the conversion of the path expression to a nondeterministic finite automaton, the subsequent conversion to a deterministic finite automaton, and the minimization of the result. The resulting deterministic finite automaton DFA′ is then checked for isomorphism against automaton DFA. This approach worked sufficiently well even for large flowgraphs. In fact we could validate 5048 out of 5053 graphs using this approach.

However, for the remaining 5 graphs validation turned out to be intractable. We investigated a different approach, based on [HU79, Theorem 3.8]. From there it follows that the languages $\sigma(A_1)$ and $\sigma(A_2)$ accepted by two finite automatons A_1 and A_2 are the same if

$$\big(\sigma(A_1) \cap \overline{\sigma(A_2)}\big) \cup \big(\overline{\sigma(A_1)} \cap \sigma(A_2)\big) = \emptyset.$$

In the above equation overlining denotes the complement of a language. Complementing a language required to compute the complement of a *deterministic* finite automaton. Making the automaton computed from the path expression $R^N_{n_e,n_x}$ deterministic turned out to be intractable for the remaining 5 graphs.

For the conversions and tests on automata and regular expressions described in this section we have used the Grail tool [RW94], a symbolic computation environment for finite-state automata and regular expressions. We have ported this tool to Linux, using the GNU toolchain with g++ 3.3. In addition we needed to add an *integer* alphabet to Grail in order to support automata with several thousand states and input symbols.

6.3 Data Evaluation

The SPEC CPU95 benchmark contains 18 benchmark programs divided into 8 integer (CINT95) and 10 floating point (CFP95) benchmarks. Tables 1 and 2 enumerate these benchmark programs along with the corresponding benchmark number and a short program description.

no.	name	description
099	go	an internationally ranked go-playing program
124	m88ksim	a chip simulator for the Motorola 88100 microprocessor
126	gcc	GNU C compiler (version 2.5.2)
129	compress	a in-memory version of the common Unix utility
130	li	Xlisp interpreter
132	ijpeg	image compression/decompression on in-memory images
134	perl	an interpreter for the perl language
147	vortex	an object oriented database

Table 1: SPEC CINT95 Benchmarks

no.	name	description
101	tomcatv	vectorized mesh generation
102	swim	shallow water equations
103	su2cor	Monte-Carlo method
104	hydro2d	Navier Stokes equations
107	mgrid	3d potential field
110	applu	partial differential equations
125	turb3d	turbulence modeling
141	apsi	weather prediction
145	fpppp	from Gaussian series of quantum chemistry benchmarks
146	wave5	Maxwell's equations

Table 2: SPEC CFP95 Benchmarks

All 18 benchmark programs together account for a total of 5053 procedures. Figure 12 depicts the distribution of these 5053 procedures among the benchmark programs.

We further differentiated between procedures with reducible and irreducible flowgraphs. From 5053 procedures, we found 5005 to contain reducible flowgraphs, while only 48 procedures turned out to be irreducible (the procedures with irreducible flowgraphs are listed in Table 3).

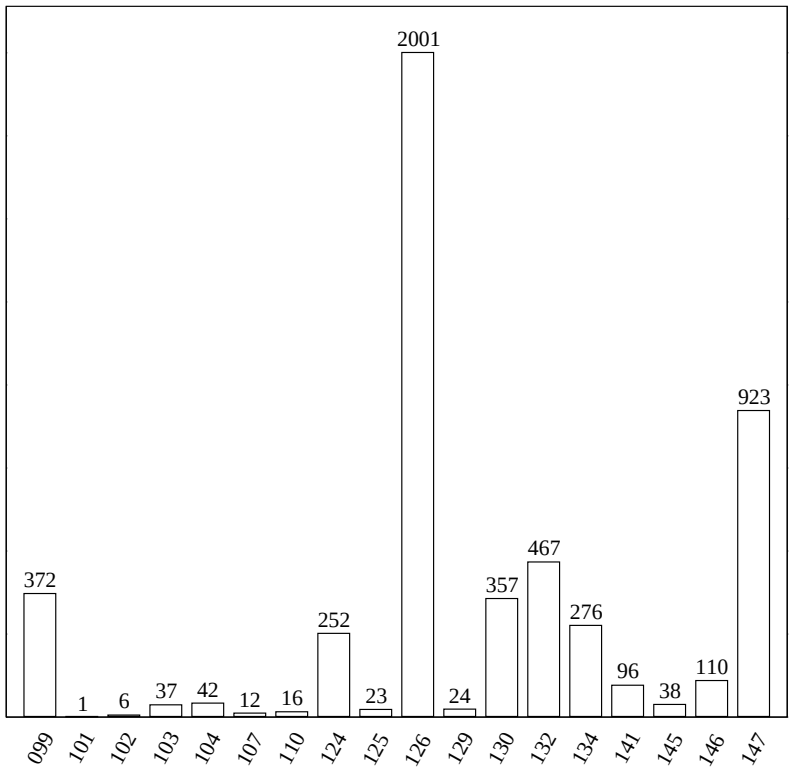


Figure 12: Number of Procedures Per Benchmark

#	benchmark #	procedure name
1	099	g2_c__readfile
2	124	asm_c__a_pname
3	124	cmdparser_c__pname
4	124	rm_c__rm
5	126	c-lex_c__yylex
6	126	caller-save_c__save_call_clobbered_regs
7	126	c-common_c__check_format_info
8	126	c-decl_c__store_parm_decls
9	126	combine_c__expand_field_assignment
10	126	c-parse_c__yyparse
11	126	cse_c__fold_rtx
12	126	cse_c__cse_insn
13	126	expr_c__get_pointer_alignment
14	126	flow_c__mark_used_regs
15	126	fold-const_c__div_and_round_double
16	126	loop_c__find_and_verify_loops

Table 3: SPEC95 Procedures with Irreducible Flow Graphs

#	benchmark #	procedure name
17	126	rtl_c__read_rtx
18	126	reorg_c__mark_target_live_regs
19	126	reorg_c__fill_simple_delay_slots
20	126	rtlanal_c__note_stores
21	126	rtl_c__read_skip_spaces
22	126	sched_c__schedule_block
23	126	stmt_c__check_for_full_enumeration_handling
24	126	stmt_c__bc_check_for_full_enumeration_handling
25	126	stmt_c__group_case_nodes
26	126	ucbqsort_c__qst
27	130	xllist_c__map
28	132	libpbm1_c__pm_init
29	132	libpbm5_c__pbm_dumpfont
30	132	rdgif_c__start_input_gif
31	134	cmd_c__cmd_exec
32	134	doarg_c__do_pack
33	134	eval_c__eval
34	134	doio_c__do_exec
35	134	dolist_c__do_match
36	134	dolist_c__do_split
37	134	dolist_c__do_unpack
38	134	dolist_c__do_kv
39	134	form_c__format
40	134	perly_c__yyparse
41	134	regcomp_c__regcomp
42	134	regcomp_c__regclass
43	134	str_c__intrpcompile
44	134	toke_c__yylex
45	134	toke_c__load_format
46	145	fmtgen_f__fmtgen_
47	145	fmtset_f__fmtset_
48	147	query_c__Query_AssertOnObject

Table 3: SPEC95 Procedures with Irreducible Flow Graphs

Observation 1: For the SPEC95 benchmark suite the share of irreducible control flow graphs is less than 1% and hence almost negligible.

Another topic of interest has been the edge/node ratio of SPEC95 control flow graphs. Figure 13 lists the ratio edges/nodes over nodes on a logarithmic scale.

Observation 2: It can be seen in Figure 13 that the edge/node ratio of SPEC95 control flow graphs is constant. This means that the number of edges is a linear

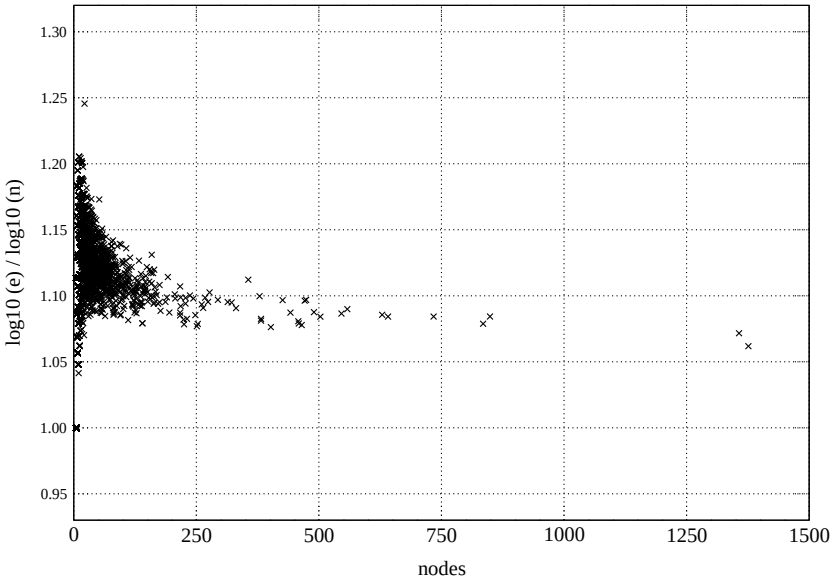


Figure 13: Edge/Node Ratio

function on the number of nodes.

Figure 14 graphs the loncp values of the SPEC95 procedures over the corresponding number of edges. Again we distinguish between reducible and irreducible control flow graphs. The fit for the data on reducible flowgraphs turns out to be a straight line for the logarithmic scale chosen for the loncp values. The fitted function is

$$\text{loncp}(e) = 2.9131 * \exp(0.076542 * e),$$

where e denotes the number of edges of a given procedure. The correlation coefficient of this function is 0.86041. This leads us to the following observation.

Observation 3: For the procedures of the SPEC95 benchmark suite the loncp metric is exponential in the number of CFG edges.

The fitted function for irreducible flowgraphs suggests that the loncp metric grows at a slower rate for irreducible graphs, but the sample of only 48 graphs of that kind is by no means representative.

Figure 15 (a) contains a quantile plot of the loncp values of the SPEC95 procedures. The f quantile $q(f)$ of a set of data is a value along the measurement scale of the data with the property that approximately a fraction f of the data are less than or equal to $q(f)$. The 0.25-, 0.50-, and 0.75-quantiles are distinguished values called lower quartile, median, and upper quartile respectively. If we order our observations of loncp values from smallest to largest, we can graph them against f. In this way quantile plots are an effective means of visualizing distributions. Coming back to our quantile plot depicted in Figure 15 (a), we arrive at the following observation.

Observation 4: The distribution of loncp values starts at the lowest possible value (1) and increases modestly up to the 0.94 quantile. Thereafter we can

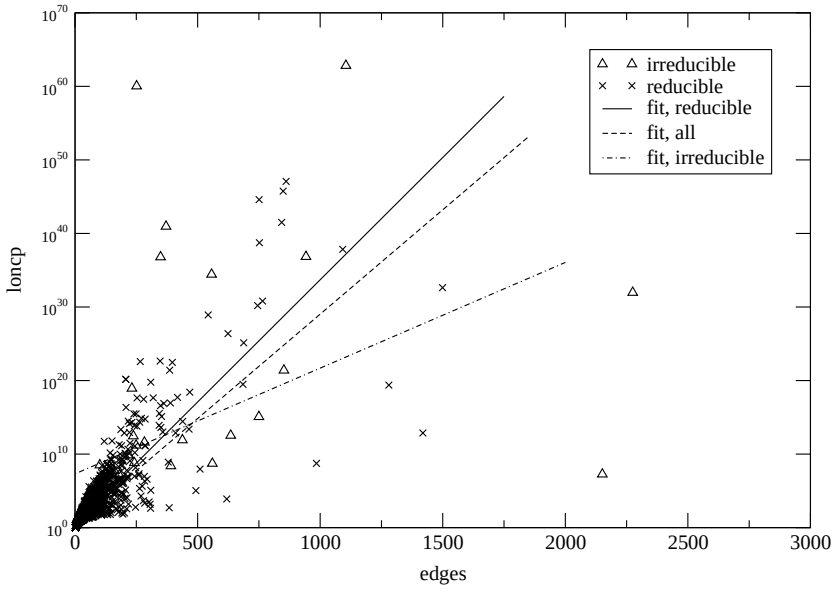


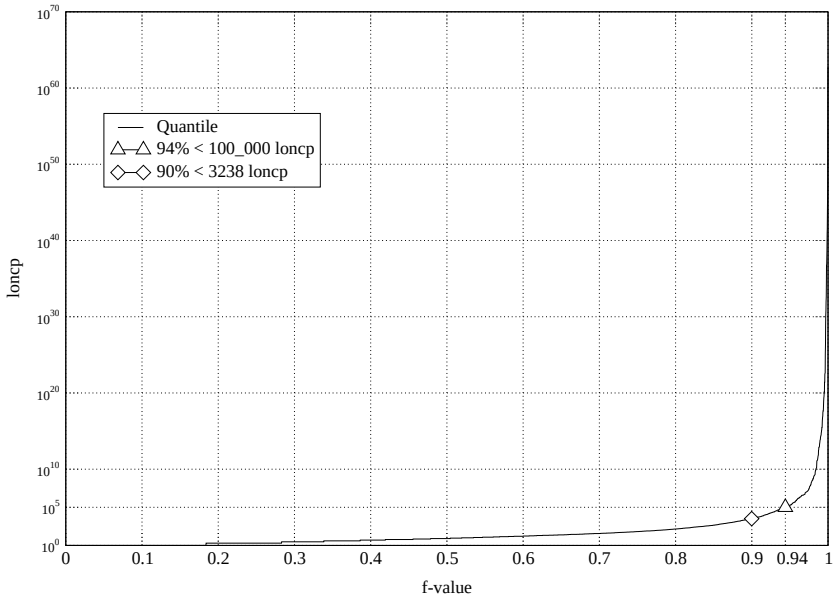
Figure 14: Regression

observe an excessive increase of quantiles which suggests that the final 6 percent of our distribution represent costly outliers. Figure 15 (b) has been scaled and excludes outliers above 10^6. The two distinguished data points in the upper right corner represent the 0.9 quantile and the 0.94 quantile. Hence we learn that for 90 percent of the distribution the loncp-value is below 3238, and for 0.94 percent it is still below 100000. The bottom line of this observation is that for the major part of SPEC95 procedures the loncp metric yields surprisingly low values, but that a few outliers turn out to be very costly.

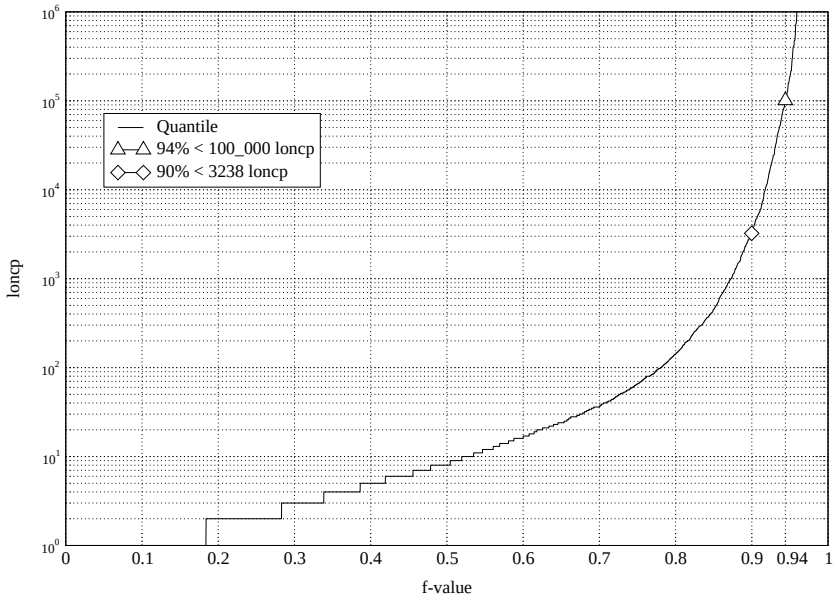
Observation (4) is also supported by the box plots of the loncp values depicted in Figure 16. For these plots the SPEC95 procedures have been grouped into the 18 benchmark programs, with an additional box plot titled "all" for the overall distribution. Note that benchmark programs 101 and 102 contain too few procedures to justify a box plot. For this reason their loncp values have been depicted as single data points. For benchmark 102 the data point at loncp = 17 actually represents three occurrences, which is in line with the total number of six procedures given in Figure 12.

Figure 16 (b) has again been scaled to exclude outliers above 10^6. In addition it contains a line connecting the medians of the respective plots. This line is helpful for locating the median for distributions like that of benchmark program 130 where the median falls together with one of the quartiles. Note that with all box plots the whiskers are drawn to the 0.1 and 0.9 quantiles.

Observation 5: Although the distributions of the SPEC95 benchmark programs have a large spread, their *center*, represented by the median, has a low loncp value. As an example consider the overall distribution depicted in the rightmost column of Figure 16. It spreads across the interval of $[1, 10^{63}]$, whereas the median is at 8 (cf. also the 0.5 quantile of Figure 15 (b)).

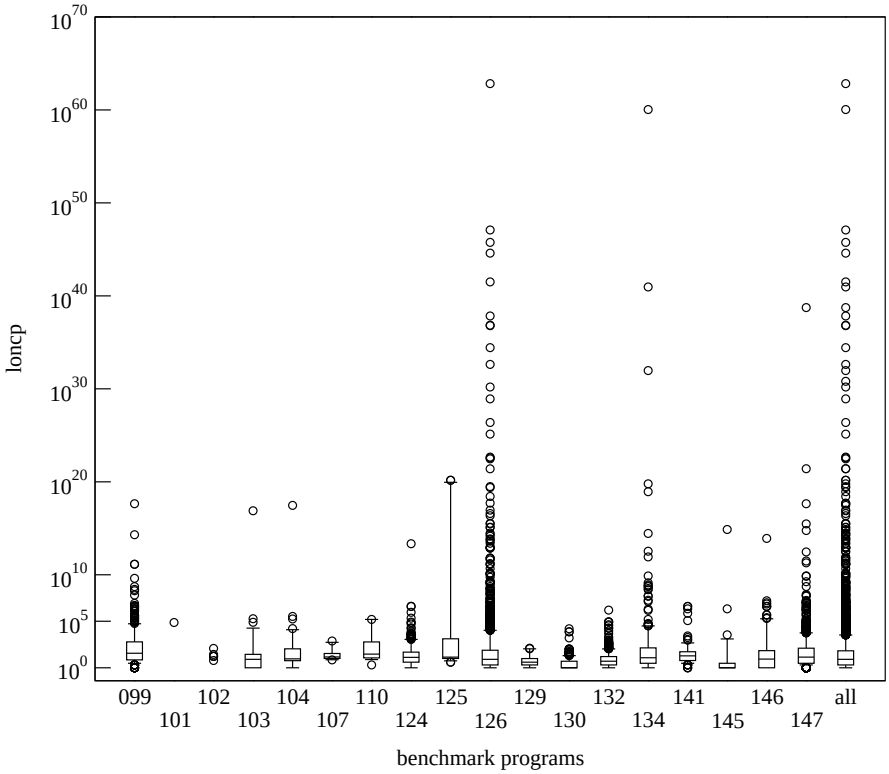


(a) Loncp Quantile Plot: complete distribution

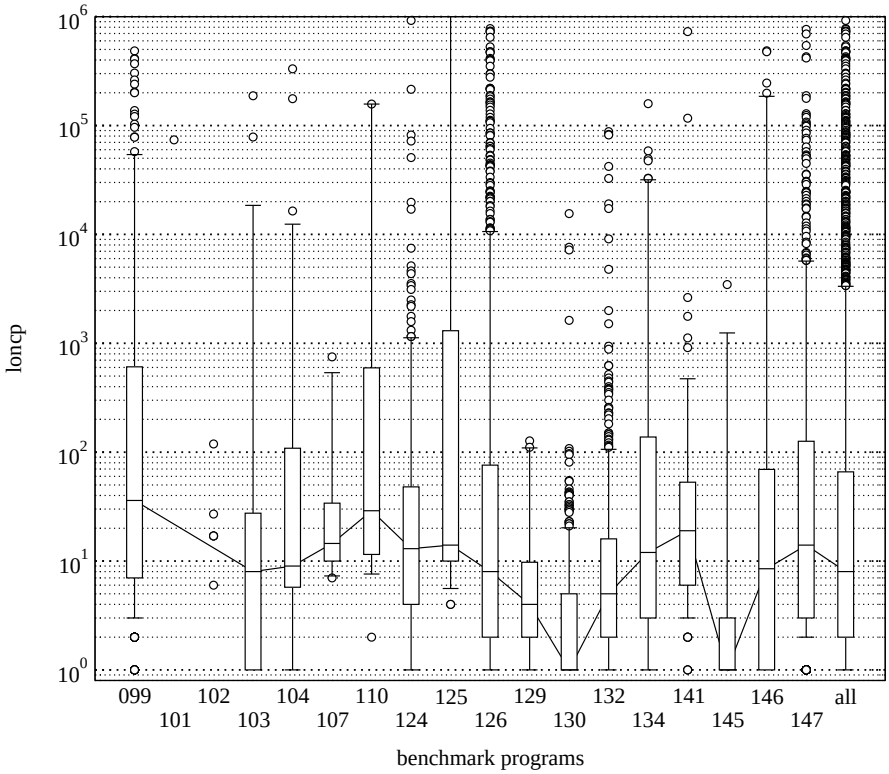


(b) Loncp Quantile Plot: outliers above 10^6 removed

Figure 15: Quantile Plot for SPEC95 Programs



(a) Loncp Box Plot: complete distribution



(b) Loncp Box Plot: outliers above 10^6 removed

Figure 16: Box Plot for SPEC95 Programs

Observation 6: The interquartile ranges of the SPEC95 benchmark programs are small. This means that the middle 50 percent of the data are tightly packed around the median. As an example we consider again the overall distribution depicted in the rightmost column of Figure 16 (b); It has a lower quartile of 2 and an upper quartile of 66 which results in an interquartile range of only 64.

Small interquartile ranges and low loncp values for the median support our argument stated in Observation (4) that the major part of SPEC95 procedures yield low loncp values, but that a few outliers are very costly.

Observation 7: The distributions of the majority of surveyed benchmark programs are skewed towards *larger* loncp values. This observation is supported by three facts regarding median, whiskers, and outliers of the plots.

1. For most benchmark programs the median is closer to the lower than to the upper quartile.
2. For every benchmark program the lower whisker ranging from the 0.1 to the 0.25 quantiles is smaller than the upper whisker ranging from the 0.75 to the 0.9 quantiles.
3. For every benchmark program the lower relative range of outliers (up to the 0.1 quantile) is by orders of magnitudes smaller than the corresponding upper relative range.

It should be noted that similarly to the interquartile range, the whiskers provide summaries of spread and shape of our benchmark data. But contrary to the interquartile range, the whiskers are not concerned with the center but with the extremes (or tails) of a distribution. The outliers, in turn, provide insight on the spread and shape in the extreme tails.

The remaining observations deal with the improvement of the loncp metric over the ncp metric. For all procedures except procedure `toke_c__yylex` from benchmark program 134 we have determined both ncp and loncp values[‡]. Figure 17 depicts the results of this survey.

Observation 8: From the 5052 examined SPEC95 procedures

- 18 percent had already the minimum metric value of 1,
- 22 percent yielded a smaller result for the loncp metric than for the ncp metric,
- and 60 percent could not be improved with the loncp metric.

In addition to the reduction in absolute values we also computed the *relative* reduction r' according to the formula

$$\text{r'} = \text{round}\big[10 * (100 - \frac{\text{loncp}(P)}{\text{ncp}(P) * 10^{-2}})\big] * 10^{-1}, \tag{30}$$

where "round" denotes the *round to nearest* rounding function. It became then apparent that there exist cases where the loncp metric achieves substantial improvements in terms of absolute values, but the relative reduction is comparatively small.

[‡]For procedure `toke_c__yylex` only the loncp value has been computed.

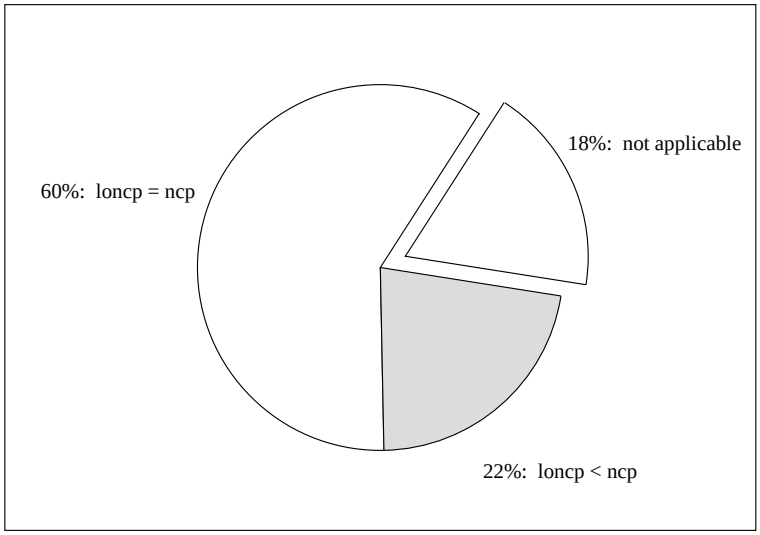


Figure 17: Loncp Metric: Number of Unaffected vs. Improved Procedures

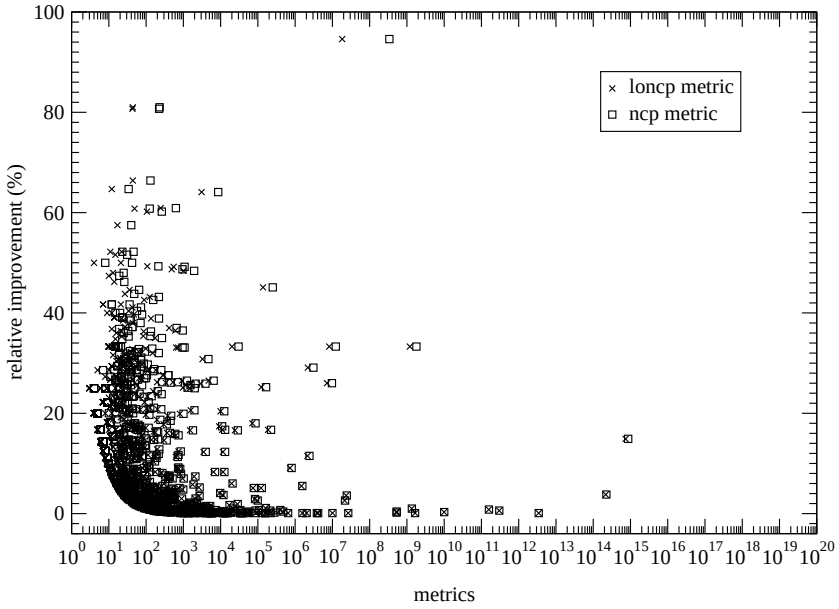


Figure 18: Relative Improvement of Loncp over Ncp Metric

Figure 18 lists the ncp and loncp values (x-axis) of all procedures with a relative reduction (y-axis) greater than zero. This figure already suggests that higher ncp values result in lower relative reduction.

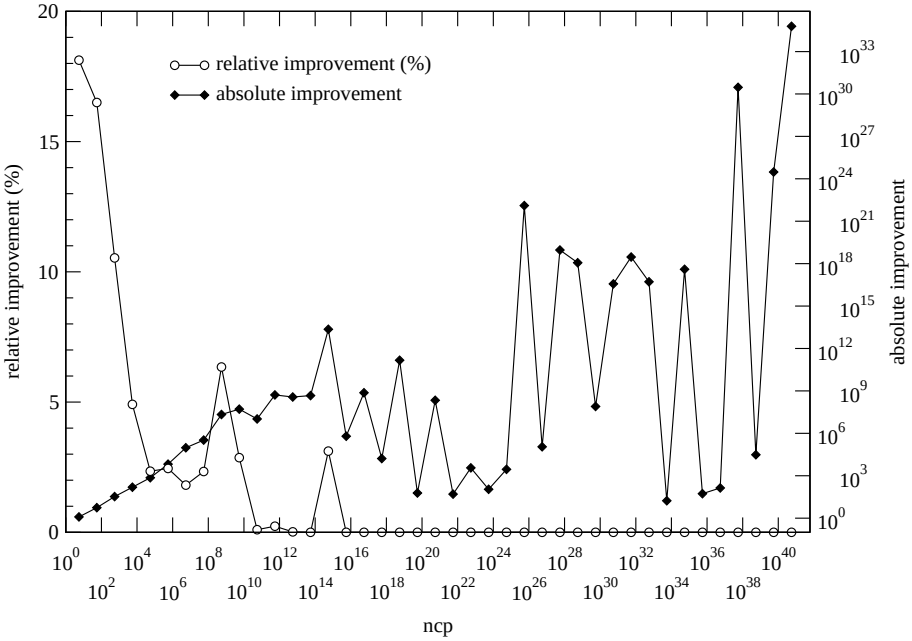


Figure 19: Comparison of Relative and Absolute Improvement

We derive the confirmation of this observation from Figure 19, where we directly compare absolute and relative reduction for ncp values. The x-axis of Figure 19 corresponds to the ncp values of procedures with an absolute reduction greater than zero. The y-axis on the left-hand side of this figure lists the relative reduction, whereas the y-axis on the right-hand side gives the absolute reduction. Every datapoint represents the average of the ncp-interval $[i, 10 * i)$, with $1 \leq i \leq 10^{40}$. Note that the 0.75 quantile for this distribution is below 10^5, which means that datapoints above this value represent comparatively few outliers. This is also in line with Observation 7. In this way the representative part of the graphs is located in the ncp interval $[1, 10^5]$. This leads us to our final observation.

Observation 9: For the majority of SPEC95 procedures that can be improved by the loncp metric, we have a negative correlation between ncp values and relative improvement, and a positive correlation between ncp values and absolute improvement.

The reason for this observation is at the moment unclear and needs further investigation. One possible cause could be that the optimization achieved with the loncp metric is based on the elimination of loops, and that programmer-supplied loops rarely exceed a given size. This in turn also limits the potential of a method targeting at loops. The loops contained in path expressions with high ncp values could be due to the use of `goto` statements. Another source for such loops (also possibly due to gotos) could be machine-generated code (e.g.,

scanners and parsers). It is likely that source code of this kind displays other characteristics as man-made goto-less code. It should however be noted that static program analysis methods are, for obvious reasons, mainly applied to the latter category.

7 Data

Mostly harmless.

— Ford Prefect, his reworked assessment of planet Earth for "The Guide",
in Douglas Adams' "The Hitchhiker's Guide to the Galaxy: a trilogy in 5 parts"

DATA: Spot, I have formulated a new mixture of foods
specifically tailored to your highly selective tastes.
Spot sniffs at the food... doesn't seem to like it... and walks away.
Data reacts.
DATA: Puzzling. I find it extremely difficult to predict what you will find acceptable.
(beat) Perhaps hunger will compel you to try it again.

— Star Trek – The Next Generation, "A Fistful of Datas", Season 6, Episode 8

This section contains the actual data collected from the SPEC95 benchmark programs. A table is devoted to each program. Tables are ordered by the SPEC95-specific benchmark number, following the distinction between integer (CINT95) and floating point (CFP95) benchmark programs. As pointed out in Section 5.3, for the irreducible graphs listed in Table 3 (pp. 26) these numbers constitute only upper bounds.

With the tables itself, we list the procedure number, the name of the procedure, the number of CFG nodes and edges, the calculated metrics data, and the relative reduction achieved by the loncp-metric compared to the ncp-metric. This reduction has been computed according to Equation (30) on page 32.

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	g22_c_lresurrect	13	19	6	19	15	21.1
2	g22_c_ldndate	61	99	4_421	4_440	4_431	0.2
3	g22_c_lsplit	21	34	144	153	152	0.7
4	g22_c_upldrflags	12	17	9	12	12	0
5	g22_c_lupdate	58	90	1_852	1_877	1_867	0.5
6	g22_c_lkilgrp	13	21	12	26	22	15.4
7	g22_c_lcombine	17	26	18	27	25	7.4
8	g23_c_fire_strat_rule	7	8	3	3	3	0
9	g23_c_fire_strat_score	9	11	5	5	5	0
10	g23_c_rule_fired	9	11	3	5	4	20
11	g23_c_empty_corner	17	24	17	24	22	8.3
12	g23_c_threethreecorner	12	17	27	27	27	0
13	g23_c_threefourcorner	25	38	1_296	2_592	2_592	0
14	g23_c_fourfourcorner	21	33	756	756	756	0
15	g23_c_stoneon44point	6	7	3	3	3	0
16	g23_c_stoneon34point	6	7	3	3	3	0
17	g23_c_stoneonhighpoint	6	7	3	3	3	0
18	g23_c_cornercolor	7	8	3	3	3	0
19	g23_c_stonefacing	7	8	3	3	3	0
20	g23_c_pincerstonefacing	6	6	2	2	2	0
21	g23_c_highpoint	8	9	3	3	3	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	g23_c_lowpoint	8	9	3	3	3	0
23	g23_c_spiral	9	13	10	10	10	0
24	g23_c_emptycorner	7	8	3	3	3	0
25	g23_c_fourfourpoint	3	2	1	1	1	0
26	g23_c_threefourpoint	5	5	2	2	2	0
27	g23_c_reflect	8	10	6	6	6	0
28	g23_c_which_corner	11	14	6	6	6	0
29	g23_c_shimari_kakari	23	33	113	121	121	0
30	g23_c_is_shimari	6	7	3	3	3	0
31	g23_c_makekakari	29	45	680	1_364	1_362	0.1
32	g23_c_makeshimari	9	12	4	8	8	0
33	g23_c_edge_shimari	11	16	7	7	7	0
34	g23_c_extedge	30	51	26	51	51	0
35	g23_c_urgextend	10	14	6	6	6	0
36	g23_c_findbestextension	175	286	1.32036×10^{11}	1.32036×10^{11}	1.32036×10^{11}	0
37	g23_c_ext_to_enemy	68	111	39_674	39_674	39_674	0
38	g23_c_extend	5	5	2	2	2	0
39	g23_c_contactfight	25	39	49	101	99	2
40	g23_c_threelibextend	24	37	186	201	201	0
41	g23_c_twolibextend	42	70	4_946	5_051	5_051	0
42	g23_c_cut_stones_val	18	29	30	43	40	7

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	g23_c_save_th_group	44	73	14_700	14_782	14_767	0.1
44	g23_c_canbecaptured	9	10	3	3	3	0
45	g23_c_savegroup	21	31	28	37	37	0
46	g23_c_make_eye_shape	32	52	120	255	202	20.8
47	g23_c_save_unsettled_army	48	78	18_432	18_461	18_455	0
48	g23_c_filloutside	22	31	40	46	46	0
49	g23_c_save_semeai	43	70	5_440	5_483	5_475	0.1
50	g23_c_runaway	14	20	8	12	11	8.3
51	g23_c_runhere	47	79	5_662	6_085	6_077	0.1
52	g23_c_save_weak_army	16	24	28	39	37	5.1
53	g23_c_remove_dead_stones	15	24	9	25	21	16
54	g23_c_killgroup	30	48	128	190	169	11.1
55	g23_c_attack_weak_army	59	99	369_105	369_213	369_179	0
56	g23_c_surround	44	74	523	559	559	0
57	g23_c_poke_eye	5	6	2	3	3	0
58	g23_c_poke_eye_ko	6	7	2	3	3	0
59	g23_c_getamoves	16	24	27	52	52	0
60	g23_c_approachmoves	18	30	18	53	42	20.8
61	g23_c_kill_th_group	40	63	9_738	9_757	9_757	0
62	g23_c_kothreat	17	29	12	29	26	10.3
63	g23_c_badkothreat	14	22	20	27	25	7.4

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	g23_c_connect_bamboo	14	22	10	42	21	50
65	g23_c_try_connect	23	35	190	190	190	0
66	g23_c_connect_2_links	6	7	3	3	3	0
67	g23_c_cut_connect	17	25	15	30	30	0
68	g23_c_canconnlkg	24	37	320	322	322	0
69	g23_c_canconnlink	60	108	3_516_722	3_516_736	3_516_728	0
70	g23_c_onelink_cut	11	15	12	12	12	0
71	g23_c_conn_val	38	62	1_560	1_560	1_560	0
72	g23_c_cut_val	60	95	2_665	2_665	2_665	0
73	g23_c_def_val	41	68	103_068	103_072	103_072	0
74	g23_c_atk_val	13	18	36	36	36	0
75	g23_c_direct_cut	57	94	1_771_560	1_771_597	1_771_597	0
76	g23_c_block_move	113	190	6.84288×10^{10}	1.36858×10^{11}	1.36858×10^{11}	0
77	g23_c_filldame	34	59	141	1_081	549	49.2
78	g23_c_get_reasons_for_moves	26	39	678	681	681	0
79	g23_c_squirm	22	37	127	255	254	0.4
80	g23_c_play_inside	11	16	6	11	11	0
81	g23_c_play_safe	48	85	36_864	37_423	37_400	0.1
82	g23_c_squirm_capture	5	6	2	3	3	0
83	g23_c_safe_capture	6	7	2	3	3	0
84	g23_c_hasaneye	10	13	4	8	6	25

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	g23_c_center	21	36	47	99	96	3
86	g23_c_isjunc	18	26	72	77	77	0
87	g23_c_threethreepoint	8	10	6	6	6	0
88	g23_c_close_corner	35	58	49	128	113	11.7
89	g23_c_isbadmove	15	20	6	12	9	25
90	g23_c_incorner	12	17	24	24	24	0
91	g23_c_sortnextmove	13	19	8	30	23	23.3
92	g23_c_getmovestotry	20	30	54	59	59	0
93	g23_c_tryamove	18	23	8	8	8	0
94	g23_c_bestmove	9	12	4	7	7	0
95	g23_c_strategy	40	62	15_732	15_736	15_736	0
96	g23_c_fixurgdefarmies	23	39	66	135	132	2.2
97	g23_c_compmove	18	28	112	119	119	0
98	g23_c_badmove	9	11	3	5	4	20
99	g23_c_check_killed	25	39	82	98	98	0
100	g23_c_deadinsidedead	10	13	5	8	6	25
101	g23_c_check_threatened	24	39	96	129	129	0
102	g23_c_check_sente	23	38	26	51	51	0
103	g23_c_cut_stones_th_val	13	21	8	27	21	22.2
104	g23_c_rule_applied	14	22	21	21	21	0
105	g23_c_stval	43	72	57_600	57_606	57_606	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	g23_c_new_th_group	19	28	63	69	69	0
107	g23_c_getscore	8	10	6	6	6	0
108	g23_c_initrtval	41	61	4_050	4_100	4_092	0.2
109	g23_c_radiateterr	13	21	8	21	18	14.3
110	g23_c_radiateweak	12	19	7	19	16	15.8
111	g23_c_radiateclear	6	7	2	3	3	0
112	g23_c_radiatescore	33	49	996	1_991	1_991	0
113	g23_c_radiatecorner	6	7	2	3	3	0
114	g23_c_radiatepiece	61	106	12_177	250_270	137_292	45.1
115	g23_c_cntterr	36	54	438	551	528	4.2
116	g23_c_evallibsterr	31	50	717	748	748	0
117	g25_c_markspot	31	44	72	85	82	3.5
118	g25_c_markpcls	38	60	392	483	459	5
119	g25_c_kill_ldrflags	5	6	2	3	3	0
120	g25_c_cantbecaptured	37	61	1_888	2_104	1_948	7.4
121	g25_c_canbethreatened	7	8	3	3	3	0
122	g25_c_cutstone	7	8	3	3	3	0
123	g25_c_findcaptured	31	50	630	689	684	0.7
124	g25_c_bdead	27	44	485	969	969	0
125	g25_c_newdeadgroup	23	35	90	100	98	2
126	g25_c_markgroup	46	71	125_809	127_381	127_287	0.1

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	g25_c_fixwasthgroup	17	26	20	30	28	6.7
128	g25_c_fixwasdeadgroup	12	16	8	11	11	0
129	g25_c_fixarmies	13	19	12	16	16	0
130	g25_c_life	3	2	1	1	1	0
131	g25_c_fixsmothered	59	96	6_591	13_273	13_219	0.4
132	g25_c_fixgralive	49	79	2_832	2_860	2_860	0
133	g25_c_initarmyalive	11	14	8	10	10	0
134	g25_c_startalive	10	13	4	6	6	0
135	g25_c_getterr	65	113	77_602	78_040	78_021	0
136	g25_c_pointeyes	31	53	586	1_022	950	7
137	g25_c_getarmynbp	16	25	11	39	28	28.2
138	g25_c_getarmylibs	6	7	2	3	3	0
139	g25_c_th_run	10	13	6	8	8	0
140	g25_c_getarmyth_pot	26	41	120	142	137	3.5
141	g25_c_th_terr	22	34	112	155	147	5.2
142	g25_c_th_conns	19	32	42	61	57	6.6
143	g25_c_getnumeyes	5	5	2	2	2	0
144	g25_c_deadgroups	14	21	12	25	21	16
145	g25_c_getarmyex_pot	74	125	5_019_651	5_029_455	5_029_455	0
146	g25_c_check_ex	95	160	412_283	412_283	412_283	0
147	g25_c_ucutdist	8	10	3	6	6	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	g25_c_canundercut	30	53	599	602	600	0.3
149	g25_c_undercut	27	44	3_136	4_704	4_704	0
150	g25_c_getarmyuc_pot	27	43	49	97	97	0
151	g25_c_doubleconnect	10	13	5	9	9	0
152	g25_c_getarmycn_pot	28	45	404	1_005	805	19.9
153	g25_c_cnthreat	17	26	25	35	35	0
154	g25_c_geteyespace	6	7	2	3	3	0
155	g25_c_getarmytv_pot	39	66	219	961	610	36.5
156	g25_c_canrunhere	58	96	1_926	1_964	1_959	0.3
157	g25_c_connect_run	16	27	13	47	36	23.4
158	g25_c_getarmyrn_pot	11	15	8	11	11	0
159	g25_c_miaialive	25	37	40	40	40	0
160	g25_c_bestpot	30	46	141	301	291	3.3
161	g25_c_totalpot	11	14	6	8	8	0
162	g25_c_secondbestpot	32	48	301	349	349	0
163	g25_c_moveispoteye	8	10	3	5	4	20
164	g25_c_adpot	64	100	160	177	175	1.1
165	g25_c_extendforeyes	50	81	2_720	2_740	2_732	0.3
166	g25_c_rmpot	17	24	8	10	10	0
167	g25_c_rmconnect	43	60	30	30	30	0
168	g25_c_blockextend	27	43	50	101	99	2

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	g25_c_blockuc	75	124	240_086	240_138	240_133	0
170	g25_c_findblock	14	20	12	48	36	25
171	g25_c_getarmywk_pot	10	13	6	8	8	0
172	g25_c_isseki	22	38	40	54	47	13
173	g25_c_semeailibs	49	77	33_600	33_713	33_683	0.1
174	g25_c_semeai_result	46	80	192	192	192	0
175	g25_c_win_semeai	13	20	14	24	19	20.8
176	g25_c_uns_semeai	11	15	8	11	11	0
177	g25_c_eyesifcapture	31	47	150	159	159	0
178	g25_c_semeaialive	10	13	5	5	5	0
179	g25_c_conntoalive	12	16	6	12	9	25
180	g25_c_weakalive	37	59	139	139	139	0
181	g25_c_newalive	9	12	4	7	7	0
182	g25_c_otherweakconn	17	26	12	26	19	26.9
183	g25_c_cnoneconn	50	83	1_817	1_845	1_841	0.2
184	g25_c_cnonepoint	64	112	1_250_256	1_250_283	1_250_273	0
185	g25_c_can_def_peep	43	71	202_500	202_504	202_504	0
186	g25_c_prot_two_point	54	93	484_712	484_719	484_716	0
187	g25_c_twoenemytwopoint	9	12	7	7	7	0
188	g25_c_cntwolkgs	25	36	21	21	21	0
189	g25_c_threepointjump	29	50	295	295	295	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	g25_c_largeknight	26	42	196	196	196	0
191	g25_c_cntwolinks	65	110	266_767_920	266_767_920	266_767_920	0
192	g25_c_checkpush	18	25	23	23	23	0
193	g25_c_protdiag	36	61	1_179	1_983	1_574	20.6
194	g25_c_fixcnprot	61	94	39_369	78_737	78_737	0
195	g25_c_combinearmy	11	15	5	7	7	0
196	g25_c_splitarmy	33	56	60	629	246	60.9
197	g25_c_cntplyhere	44	72	2_360	2_380	2_365	0.6
198	g26_c_dnn1	9	13	5	9	9	0
199	g26_c_upn1	23	37	280	297	297	0
200	g26_c_uscan	28	43	160	205	202	1.5
201	g26_c_incn1	5	5	2	2	2	0
202	g26_c_dscan	40	64	465	523	505	3.4
203	g26_c_cscan	8	10	3	5	5	0
204	g26_c_cuscan	10	14	5	9	9	0
205	g27a_c_upltr	49	76	2_322_432	2_322_448	2_322_443	0
206	g27a_c_chckcorner	20	31	85	85	85	0
207	g27a_c_edgeofterr	38	67	70	74	72	2.7
208	g27a_c_chckside	151	253	2.17918×10^{17}	4.35835×10^{17}	4.35835×10^{17}	0
209	g27b_c_upxy	43	68	88_519	98_893	96_309	2.6
210	g27b_c_dnxy	59	100	3_320_440	3_322_174	3_321_750	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	g28_c_dncons	27	43	142	321	302	5.9
212	g28_c_brkconns	27	42	200	215	215	0
213	g28_c_brconn	15	22	16	31	31	0
214	g28_c_moveconns	46	72	585	1_199	1_180	1.6
215	g28_c_adcons	30	49	336	375	369	1.6
216	g28_c_addconn	17	26	20	28	24	14.3
217	g28_c_chkcon	34	55	434	920	892	3
218	g28_c_addlks	9	13	5	9	9	0
219	g28_c_addlkgs	9	13	5	9	9	0
220	g28_c_brklks	10	15	8	12	12	0
221	g28_c_brklkgs	10	15	8	12	12	0
222	g28_c_rstrlks	19	31	66	195	163	16.4
223	g28_c_brklink	32	52	712	722	722	0
224	g28_c_brklkg	32	52	712	722	722	0
225	g28_c_realbrklink	29	46	190	200	197	1.5
226	g28_c_realbrklkg	29	46	190	200	197	1.5
227	g28_c_delconnrec	7	8	4	4	4	0
228	g28_c_addlink	17	26	20	28	24	14.3
229	g28_c_addlkg	17	26	20	28	24	14.3
230	g29_c_iscaptured	74	119	115_102	127_450	120_892	5.1
231	g29_c_defonelib	36	59	3_168	3_180	3_180	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	g29_c__defatari	39	64	1_925	3_923	2_906	25.9
233	g29_c__def_atk_nbr	73	120	4_572_612	4_607_427	4_607_301	0
234	g29_c__jump_to_escape	3	2	1	1	1	0
235	g29_c__play_next_to_group	77	127	52_416_004	65_520_074	65_520_037	0
236	g29_c__getefflibs	89	147	527_410_419	529_097_865	528_254_123	0.2
237	g29_c__attack2libs	29	46	1_106	2_301	2_256	2
238	g29_c__gendefmoves	58	95	1_300_339	1_300_387	1_300_383	0
239	g29_c__def_two_stone_wall	23	36	287	287	287	0
240	g29_c__jump	8	10	3	5	4	20
241	g29_c__deftwolibs	107	178	193_752_000	193_752_212	193_752_136	0
242	g29_c__atktwolibs	140	232	2.02367×10^{14}	2.02367×10^{14}	2.02367×10^{14}	0
243	g29_c__genatkmoves	44	69	46_116	46_154	46_150	0
244	g29_c__genrestatk	123	201	4_141_286_428	4_141_417_376	4_141_417_151	0
245	g29_c__livesordies	46	77	10_501	10_555	10_524	0.3
246	g2_c__main	8	10	2	4	4	0
247	g2_c__readfile	20	29	14	78	58	25.6
248	g2_c__check	33	49	113	113	113	0
249	g2_c__getmove	21	32	76	118	118	0
250	g2_c__init	3	2	1	1	1	0
251	g2_c__g2exit	3	2	1	2	2	0
252	g2_c__fixplaylevel	21	29	193	193	193	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
253	g2_c__initinit	3	2	1	1	1	0
254	g2_c__turnoffcplay	3	2	1	1	1	0
255	g2_c__outerror	3	2	1	1	1	0
256	g2_c__psqr	3	2	1	1	1	0
257	g2_c__ssqr	14	18	8	8	8	0
258	g2eye_c__fixli	28	43	31	47	47	0
259	g2eye_c__evalopenlineeye	68	113	8_424_651	8_424_676	8_424_660	0
260	g2eye_c__addeyerec	5	6	2	3	3	0
261	g2eye_c__findeyelist	33	51	1_803	2_461	2_431	1.2
262	g2eye_c__can_be_eye	10	15	12	12	12	0
263	g2eye_c__evalonepteye	3	2	1	1	1	0
264	g2eye_c__evalcornereyes	37	62	3_592	7_184	7_184	0
265	g2eye_c__deallocate_eye	11	17	10	17	16	5.9
266	g2eye_c__getcount	71	118	303_264	303_308	303_298	0
267	g2eye_c__deadshape	51	81	276	310	296	4.5
268	g2eye_c__evaldeadgroupeye	25	35	33	34	34	0
269	g2eye_c__evaloneptdeadeye	48	73	2_907	2_944	2_944	0
270	g2eye_c__evalmanyeyespots	24	35	104	105	105	0
271	g2eye_c__eval2pointeye	25	38	396	399	399	0
272	g2eye_c__eval_line_eye	76	120	406_944	406_956	406_953	0
273	g2eye_c__evalbigeye	72	118	199_838	199_884	199_884	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
274	g2eye_c__onelibnbr	9	11	3	5	4	20
275	g2jos_c__jupdate	6	7	2	3	3	0
276	g2jos_c__jupdatec	52	86	11_696	11_726	11_711	0.1
277	g2jos_c__getlastcolor	21	30	133	133	133	0
278	g2jos_c__getflag	7	8	3	3	3	0
279	g2jos_c__getxyjlib	9	10	3	3	3	0
280	g2jos_c__j2next	11	15	9	9	9	0
281	g2jos_c__j2skip	12	16	10	10	10	0
282	g2jos_c__j2more	21	30	38	56	56	0
283	g2jos_c__sibling	10	13	8	8	8	0
284	g2jos_c__joseki	56	92	8_428	12_637	11_587	8.3
285	g2jos_c__findurgentjoseki	34	52	426	849	744	12.4
286	g2jos_c__fire_joseki	43	73	206	206	206	0
287	g2jos_c__notjoseki	14	21	8	10	9	10
288	g2jos_c__evaljoseki	38	55	154	195	195	0
289	g2list_c__gtflist	6	6	2	2	2	0
290	g2list_c__comlist	12	16	2	6	6	0
291	g2list_c__cpylist	8	9	2	3	3	0
292	g2list_c__mrglist	28	40	21	26	26	0
293	g2list_c__andlist	13	18	8	11	11	0
294	g2list_c__addlist	21	29	10	12	11	8.3

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
295	g2list_c__adflist	6	6	2	2	2	0
296	g2list_c__dellist	23	31	10	13	11	15.4
297	g2list_c__killist	8	10	3	4	4	0
298	g2list_c__cntlist	6	7	2	3	3	0
299	g2list_c__newlist	10	14	5	11	10	9.1
300	g2list_c__inlist	10	13	4	6	5	16.7
301	g2s2_c__adplib	9	13	5	9	9	0
302	g2s2_c__deplib	11	15	5	9	9	0
303	g2s2_c__adlibs	11	15	5	9	9	0
304	g2s2_c__delibs	39	64	70	219	149	32
305	g2s2_c__resurrect	5	6	2	3	3	0
306	g2s2_c__newgrouparmy	12	16	8	11	11	0
307	g2s2_c__dndate	60	95	2_247_265	2_247_304	2_247_304	0
308	g2s2_c__make_army_free	12	16	6	9	9	0
309	g2s2_c__split	16	26	48	53	53	0
310	g2s2_c__real_update	7	8	3	3	3	0
311	g2s2_c__finds	17	24	18	67	51	23.9
312	g2s2_c__update	53	83	261_938	262_052	262_051	0
313	g2s2_c__makenewgroup	19	29	60	65	65	0
314	g2s2_c__kilgrp	14	21	32	35	35	0
315	g2s2_c__combine	43	71	44_801	44_829	44_827	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
316	g2s3_c__initkomi	10	13	5	5	5	0
317	g2s3_c__initbsizeconst	82	120	3_824_760	3_856_666	3_856_660	0
318	g2s3_c__initjflags	5	5	1	2	2	0
319	g2s3_c__initvars	48	73	132	160	157	1.9
320	g2shp_c__callfunc	46	84	40	40	40	0
321	g2shp_c__p_push_once	11	15	6	6	6	0
322	g2shp_c__p_break_out	11	14	7	7	7	0
323	g2shp_c__p_edge_connect	15	24	13	13	13	0
324	g2shp_c__p_clampkill	11	15	6	6	6	0
325	g2shp_c__p_crosscut	26	36	64	64	64	0
326	g2shp_c__extend_to_friend	7	9	3	5	5	0
327	g2shp_c__extend_along_edge	8	10	3	5	5	0
328	g2shp_c__cross_capture	7	9	3	5	5	0
329	g2shp_c__p_eyeprotect	23	37	81	81	81	0
330	g2shp_c__p_pullback	6	7	3	3	3	0
331	g2shp_c__p_two_point_edge	13	19	8	8	8	0
332	g2shp_c__p_twopoint	27	43	52	52	52	0
333	g2shp_c__p_jump_in_center	19	29	36	36	36	0
334	g2shp_c__p_knight_in_center	30	47	67	67	67	0
335	g2shp_c__p_outside_hane	6	7	3	3	3	0
336	g2shp_c__p_extend_3_libs	10	14	9	9	9	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
337	g2shp_c__p_overcut	24	38	398	398	398	0
338	g2shp_c__p_knights_from_behind	8	10	5	5	5	0
339	g2shp_c__cross_sector	21	32	58	58	58	0
340	g2shp_c__p_cross_sector_nodef	5	5	2	2	2	0
341	g2shp_c__p_cross_sector	3	2	1	1	1	0
342	g2shp_c__p_cross_def_sector	3	2	1	1	1	0
343	g2shp_c__p_invade3	14	19	8	8	8	0
344	g2shp_c__p_clamp	14	21	10	10	10	0
345	g2shp_c__p_defend_wall	24	38	30	30	30	0
346	g2shp_c__p_knightsmove	31	52	248	248	248	0
347	g2shp_c__p_edge_cut3	22	34	47	47	47	0
348	g2shp_c__p_edge_cut	22	34	47	47	47	0
349	g2shp_c__p_edge_cut2	22	34	47	47	47	0
350	g2shp_c__p_first_line	46	77	4_741	4_745	4_743	0
351	g2shp_c__p_jump_wall	14	22	21	21	21	0
352	g2shp_c__p_block_knights	15	22	11	11	11	0
353	g2shp_c__p_block_second	19	28	30	30	30	0
354	g2shp_c__p_edge_knights	22	33	28	28	28	0
355	g2shp_c__p_monkey_jump	28	44	32	32	32	0
356	g2shp_c__p_edge_hane	10	14	6	6	6	0
357	g2shp_c__p_double_diag	16	23	19	19	19	0

Table 4: SPEC CINT95 — 099.go

no.	function	nodes	edges	npp	ncp	loncp	rd.
358	g2shp_c__p_edge_kosumi	6	7	3	3	3	0
359	g2shp_c__p_hane	34	57	6_056	6_056	6_056	0
360	g2shp_c__p_stophane	39	62	10_754	10_754	10_754	0
361	g2shp_c__groupcanrunhere	11	14	3	8	6	25
362	g2shp_c__p_pushthru	27	42	88	94	94	0
363	g2shp_c__p_connect_hane	23	37	153	153	153	0
364	g2shp_c__initshapes	17	26	30	54	50	7.4
365	g2shp_c__sortshape	10	13	3	10	8	20
366	g2shp_c__newshape	7	9	3	5	4	20
367	g2shp_c__sameshape	14	20	8	19	12	36.8
368	g2shp_c__copyshape	40	60	600	822	749	8.9
369	g2shp_c__findshapes	77	127	307	745	679	8.9
370	g2shp_c__match	18	28	15	31	23	25.8
371	g2shp_c__match2	18	28	15	31	23	25.8
372	g2shp_c__evalshapes	7	10	3	7	6	14.3

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	addd_c_addd	20	30	39	39	39	0
2	adds_c_adds	10	12	5	5	5	0
3	alignd_c_alignd	10	14	4	6	6	0
4	aligns_c_aligns	12	14	4	4	4	0
5	asm_c_assembler	12	16	9	9	9	0
6	asm_c_reterr	3	2	1	1	1	0
7	asm_c_a_pname	33	53	12_375	17_319	17_019	1.7
8	asm_c_a_choice	7	9	3	5	4	20
9	asm_c_procopers	25	38	24	39	29	25.6
10	asm_c_mkwrd	3	2	1	1	1	0
11	asm_c_a_sfu1	19	31	19	19	19	0
12	asm_c_cksfu1	25	36	19	24	24	0
13	asm_c_a_ctl	20	28	10	10	10	0
14	asm_c_integer	28	42	89	89	89	0
15	asm_c_trpbci	9	12	7	7	7	0
16	asm_c_tbnd	11	14	5	5	5	0
17	asm_c_mem	14	19	7	7	7	0
18	asm_c_ckmem	45	68	312	312	312	0
19	asm_c_xfrbci	13	18	13	13	13	0
20	asm_c_logical	22	32	20	20	20	0
21	asm_c_xfrri	12	16	7	7	7	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	asm_c__ffirst	8	10	4	4	4	0
23	asm_c__rte	5	5	2	2	2	0
24	asm_c__xfra	12	16	7	7	7	0
25	asm_c__bits	11	14	5	5	5	0
26	asm_c__validreg	8	11	5	5	5	0
27	bf_c__bf	25	36	27	71	44	38
28	bm_c__bm	14	20	9	13	11	15.4
29	br_c__br	36	55	805	1_592	1_582	0.6
30	br_c__nobr	16	23	10	19	17	10.5
31	br_c__dbrks	15	21	21	42	42	0
32	br_c__ckbrkpts	16	24	22	26	24	7.7
33	br_c__brkptenb	5	5	1	2	2	0
34	br_c__settmpbrk	3	2	1	1	1	0
35	br_c__rsttmpbrk	3	2	1	1	1	0
36	bs_c__bs	30	45	39	155	89	42.6
37	ckiob_c__ckiob	11	14	5	5	5	0
38	classify_c__classify	7	8	3	3	3	0
39	cm_c__cm	33	46	222	222	222	0
40	cmdparser_c__parse	40	66	1_051	1_057	1_054	0.3
41	cmdparser_c__pname	32	53	16_379	20_991	19_742	6
42	cmdparser_c__strtolower	7	9	3	5	5	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	cmdparser_c_choice	7	9	3	5	4	20
44	cmdstruct_c_simexit	3	2	1	2	2	0
45	cmmu_atc_c_check_BATC	9	11	3	5	5	0
46	cmmu_atc_c_check_PATC	9	11	3	5	5	0
47	cmmu_atc_c_get_from_BATC	26	38	398	399	399	0
48	cmmu_atc_c_get_from_PATC	41	64	81_795	81_797	81_797	0
49	cmmu_atc_c_update_PATC	15	22	36	40	40	0
50	cmmu_c_load_data	96	142	215_396	215_396	215_396	0
51	cmmu_c_tablewalk	120	187	2.17916×10^{13}	2.17916×10^{13}	2.17916×10^{13}	0
52	cmmu_cache_c_load_from_cache	23	33	48	48	48	0
53	cmmu_cache_c_check_cache	14	21	17	23	20	13
54	cmmu_cache_c_cache_hit	7	9	4	5	5	0
55	cmmu_cache_c_mark_lru	12	15	5	5	5	0
56	cmmu_cache_c_find_change_order	9	12	6	7	7	0
57	cmmu_cache_c_cache_miss	47	68	2_176	2_193	2_185	0.4
58	cmmu_cache_c_store_data	29	45	60	70	68	2.9
59	cmmu_ctl_c_ctrl_space_read	111	163	3_536	3_540	3_540	0
60	cmmu_ctl_c_ctrl_space_write	95	140	460	460	460	0
61	cmmu_ctl_c_change_cache_set	47	72	26	27	27	0
62	cmmu_debug_c_cd	19	25	13	15	15	0
63	cmmu_debug_c_cs	17	23	128	128	128	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	cmmu_debug_c__pd	20	29	49	63	62	1.6
65	cmmu_debug_c__bd	15	20	9	10	10	0
66	cmmu_debug_c__cr	3	2	1	1	1	0
67	cmmu_func_c__cmmu_ctrl_func	83	124	1_296	1_329	1_317	0.9
68	cmmu_func_c__flush_all	10	13	4	8	8	0
69	cmmu_func_c__flush_byPandS	13	17	12	16	16	0
70	cmmu_func_c__write_line	7	8	2	3	3	0
71	cmmu_func_c__redo_set_lru	65	111	2_563	5_158	5_142	0.3
72	cmmu_init_c__cmmu_init	11	14	1	5	5	0
73	cmmu_init_c__cache_reset	9	11	1	5	4	20
74	cmmu_init_c__write_memory	3	2	1	1	1	0
75	cmmu_init_c__init_batc_HDWRD	3	2	1	1	1	0
76	converters_c__getexpr	28	41	494	518	502	3.1
77	converters_c__getrange	20	29	38	38	38	0
78	converters_c__getdata	8	10	4	4	4	0
79	converters_c__simatoi	28	42	158	164	164	0
80	converters_c__atosf	3	2	1	1	1	0
81	converters_c__str_toupper	8	10	3	5	5	0
82	ctlregs_c__wrctlregs	14	21	48	49	49	0
83	ctlregs_c__init_processor	13	20	9	19	17	10.5
84	dc_c__dc	5	5	2	2	2	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	dis_c__dis	14	21	16	16	16	0
86	dis_c__memi	16	21	11	11	11	0
87	dis_c__imm16	3	2	1	1	1	0
88	dis_c__logi	6	6	2	2	2	0
89	dis_c__inti	3	2	1	1	1	0
90	dis_c__sfu	9	11	4	4	4	0
91	dis_c__xfr	19	24	16	16	16	0
92	dis_c__rrr	20	28	14	14	14	0
93	dis_c__ctl	23	32	18	18	18	0
94	dis_c__sfu1	19	31	17	17	17	0
95	dis_c__gen1	17	22	7	7	7	0
96	dis_c__gen2	69	108	47	47	47	0
97	dis_c__rrrdisp	3	2	1	1	1	0
98	dis_c__rrrmdisp	12	15	5	5	5	0
99	dis_c__nameofbit	6	6	2	2	2	0
100	dis_c__cndtype	6	6	2	2	2	0
101	divd_c_divd	22	33	75	100	100	0
102	divs_c__divs	8	10	3	6	6	0
103	dmem_c_ldfrommem	55	80	770	770	770	0
104	dmem_c_sttomem	64	91	2_498	2_498	2_498	0
105	dmem_c_dacc	20	28	54	54	54	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	dmem_c__out_to_in	28	38	41	41	41	0
107	dmem_c__out_to_log	30	40	71	71	71	0
108	dpath_c__Data_path	37	54	648	648	648	0
109	dpath_c__addunsigned	30	49	1_768	1_768	1_768	0
110	dpath_c__sext	8	9	4	4	4	0
111	dpath_c__uext	8	9	4	4	4	0
112	dpath_c__make	7	8	3	3	3	0
113	dpath_c__cmmu_function1	31	45	1_154	1_154	1_154	0
114	dpath_c__cmmu_function2	8	10	6	6	6	0
115	dpath_c__execute	172	279	4_620	4_621	4_621	0
116	dpath_c__display_trace	24	33	22	22	22	0
117	fadd64_c__fadd64	18	25	34	34	34	0
118	fadd_c__fadd	38	59	140	140	140	0
119	fadds_c__fadds	14	19	14	14	14	0
120	fcdi_c__fcdi	34	51	273	275	275	0
121	fcds_c__fcds	14	19	12	12	12	0
122	fcid_c__fcid	10	12	4	5	5	0
123	fcis_c__fcis	8	9	4	4	4	0
124	fcmp64_c__fcmp64	35	57	3_390	3_390	3_390	0
125	fcmp_c__fcmp	26	37	48	48	48	0
126	fcmps_c__fcmps	33	52	2_262	2_262	2_262	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	fcsd_c_fcsd	7	8	3	3	3	0
128	fcsi_c__fcsi	23	32	55	55	55	0
129	fdiv64_c_fdiv64	19	27	47	47	47	0
130	fdiv_c__fdiv	38	59	140	140	140	0
131	fdivs_c__fdivs	18	25	29	29	29	0
132	floaterr_c__floaterr	29	42	35	35	35	0
133	floaterr_c__pre_except	17	22	36	36	36	0
134	floaterr_c__impre_except	11	13	8	8	8	0
135	flt_c_flt	14	17	12	12	12	0
136	fmul64_c__fmul64	14	18	10	10	10	0
137	fmul_c__fmul	38	59	140	140	140	0
138	fmuls_c_fmuls	11	14	6	6	6	0
139	fpunimp_c__fpunimp	11	13	8	8	8	0
140	fsub64_c__fsub64	18	25	34	34	34	0
141	fsub_c_fsub	38	59	140	140	140	0
142	fsubs_c__fsubs	14	19	14	14	14	0
143	go_c__go	16	23	16	16	16	0
144	go_c__gd	5	5	2	2	2	0
145	go_c__goexec	21	33	19	25	22	12
146	go_c__gn	3	2	1	1	1	0
147	go_c__gt	6	6	2	2	2	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	go_c__tr	8	9	3	3	3	0
149	go_c__tv	8	9	3	3	3	0
150	go_c__tx	3	2	1	1	1	0
151	go_c__tz	3	2	1	1	1	0
152	go_c__tt	6	6	2	2	2	0
153	go_c__trexec	53	85	50_704	50_902	50_803	0.2
154	he_c__he	16	24	12	34	24	29.4
155	id_c__ids	8	11	5	5	5	0
156	id_c__id	40	60	348	711	525	26.2
157	int_c__convert_int	16	21	20	20	20	0
158	interface_c__transinit	6	6	2	2	2	0
159	interface_c__rdwr	64	112	380_576	923_484	923_412	0
160	interface_c__ckquit	3	2	1	1	1	0
161	interface_c__checklmt	17	26	21	30	30	0
162	interface_c__getmemptr	5	5	2	2	2	0
163	interface_c__getpmem	14	19	14	15	15	0
164	interface_c__getmem	24	34	38	41	39	4.9
165	interface_c__releasemem	7	8	2	4	4	0
166	interface_c__releasepmem	7	8	2	4	4	0
167	interface_c__releaseseg	6	7	3	3	3	0
168	interface_c__dm	18	24	11	22	22	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	interface_c__intswap	3	2	1	1	1	0
170	interface_c__shortswap	3	2	1	1	1	0
171	lo_c__lo	126	194	4_143_904	4_147_718	4_144_666	0.1
172	lo_c__rlo	121	188	3_818_813	3_822_637	3_819_575	0.1
173	lo_c__pr_opt_hdr	5	5	1	2	2	0
174	lo_c__loadmem	38	64	1_790	4_384	4_358	0.6
175	main_c__main	64	100	7_296	7_483	7_469	0.2
176	main_c__presetsim	11	14	2	5	5	0
177	map_c__map	17	23	17	17	17	0
178	md_c__mds	8	11	5	5	5	0
179	md_c__md	45	67	768	1_554	1_156	25.6
180	mm_c__mm	62	91	1_458	3_122	3_122	0
181	multd_c__multd	16	21	4	10	9	10
182	mults_c__mults	8	9	4	4	4	0
183	normalized_c__normalized	12	19	17	21	21	0
184	normalizes_c__normalizes	11	17	9	13	13	0
185	opn_output_c__open_output	12	17	4	14	13	7.1
186	pc_c__Pc	33	49	189	189	189	0
187	pc_c__checkforjump	16	25	26	26	26	0
188	pc_c__checkfortrap	20	30	59	59	59	0
189	rd_c__rd	19	26	34	34	34	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	rd_c_rdexec	18	25	16	24	24	0
191	rd_c_dispSFU	9	11	4	8	8	0
192	reserved_c_reserved	16	22	14	14	14	0
193	reserves_c_reserves	13	16	8	8	8	0
194	returnd_c_return_double	14	18	13	13	13	0
195	returns_c_return_single	13	16	7	7	7	0
196	rm_c_rm	72	109	62	829	828	0.1
197	rm_c_rmSFU	25	36	2	29	29	0
198	rm_c_getregval	31	51	684	685	685	0
199	round_c_round	24	40	58	58	58	0
200	roundd_c_roundd	12	17	21	21	21	0
201	rounds_c_rounds	11	15	14	14	14	0
202	runsim_c_run	20	30	17	22	19	13.6
203	runsim_c_setargs	3	2	1	1	1	0
204	runsim_c_runsilent	11	14	3	7	6	14.3
205	runsim_c_dumpcore	39	60	155	189	167	11.6
206	runsim_c_loadcore	43	63	52	93	63	32.3
207	runsim_c_rstsys	5	5	2	2	2	0
208	runsim_c_cacheoff	3	2	1	1	1	0
209	runsim_c_rrn	20	32	61	68	64	5.9
210	runsim_c_cp_ptrs	5	6	2	3	3	0

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	sdsr_c__sr	3	2	1	1	1	0
212	sdsr_c__sd	3	2	1	1	1	0
213	show_c__show	32	46	51	69	69	0
214	signals_c__sig_handler	6	6	2	2	2	0
215	signals_c__sig_set	5	5	2	2	2	0
216	sim_io_c__getarg	6	6	2	2	2	0
217	sim_io_c__copystr	10	12	2	5	4	20
218	sim_io_c__stdio_enable	8	10	4	4	4	0
219	simload_c__makesim	7	8	4	4	4	0
220	sim_printf_c__sim_printf	23	34	11	33	21	36.4
221	simtime_c__test_issue	11	16	10	13	13	0
222	simtime_c__do_issue	3	2	1	1	1	0
223	simtime_c__killtime	17	23	20	24	24	0
224	simtime_c__readtime	3	2	1	1	1	0
225	simtime_c__check_scoreboard	38	63	72_000	72_002	72_002	0
226	stats_c__statreset	3	2	1	1	1	0
227	stats_c__printstats	6	7	3	3	3	0
228	stats_c__Statistics	17	28	13	13	13	0
229	symbols_c__symbol	32	52	40	46	43	6.5
230	symbols_c__findsym	24	40	49	52	51	1.9
231	symbols_c__symcreate	25	35	14	25	23	8

Table 5: SPEC CINT95 — 124.m88ksim

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	symbols_c__symcopy	6	7	2	3	3	0
233	symbols_c__symfree	7	8	4	4	4	0
234	symbols_c__prtsym	22	32	16	22	22	0
235	symbols_c__initsymptrs	13	20	12	14	13	7.1
236	symbols_c__find_next_symbol	10	15	8	11	9	18.2
237	sysface_c__pa	14	17	5	5	5	0
238	sysface_c__nopa	5	5	2	2	2	0
239	sysface_c__PPrintf	9	11	5	5	5	0
240	sysface_c__Eprintf	5	5	2	2	2	0
241	sysface_c__FFgets	3	2	1	1	1	0
242	sysface_c__cwd	3	2	1	1	1	0
243	sysface_c__pwd	3	2	1	1	1	0
244	sysVbcs_c__sysVbcs	30	42	25	31	27	12.9
245	sysVbcs_c__sysVupfil	10	12	4	5	5	0
246	sysVbcs_c__sysVclose	7	8	2	4	4	0
247	table_c__lookupdisasm	9	13	6	9	7	22.2
248	table_c__init_disasm	7	9	2	4	4	0
249	table_c__install	5	5	2	2	2	0
250	trap_c__vector	29	46	73	73	73	0
251	trap_c__exception	20	28	146	146	146	0
252	updstat_c__upd_status	11	14	7	7	7	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	aux-output_c_reg_or_0_operand	13	20	15	15	15	0
2	aux-output_c_fp_zero_operand	3	2	1	1	1	0
3	aux-output_c_restore_operand	8	11	5	5	5	0
4	aux-output_c_call_operand	12	18	7	8	8	0
5	aux-output_c_call_operand_address	10	15	7	7	7	0
6	aux-output_c_symbolic_operand	15	20	7	7	7	0
7	aux-output_c_symbolic_memory_operand	12	17	12	12	12	0
8	aux-output_c_reg_or_nonsymb_mem_operand	7	9	4	4	4	0
9	aux-output_c_sparc_operand	17	24	13	13	13	0
10	aux-output_c_move_operand	26	41	47	47	47	0
11	aux-output_c_move_pic_label	7	8	3	3	3	0
12	aux-output_c_memop	9	11	4	4	4	0
13	aux-output_c_eq_or_neq	3	2	1	1	1	0
14	aux-output_c_normal_comp_operator	10	14	6	6	6	0
15	aux-output_c_noov_compare_op	11	14	5	5	5	0
16	aux-output_c_extend_op	3	2	1	1	1	0
17	aux-output_c_cc_arithop	3	2	1	1	1	0
18	aux-output_c_cc_arithopn	3	2	1	1	1	0
19	aux-output_c_arith_operand	7	9	4	4	4	0
20	aux-output_c_arith_double_operand	17	29	79	79	79	0
21	aux-output_c_shift_operand	7	9	4	4	4	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	aux-output_c__small_int	6	7	3	3	3	0
23	aux-output_c__uns_small_int	9	13	9	9	9	0
24	aux-output_c__uns_arith_operand	6	7	3	3	3	0
25	aux-output_c__clobbered_register	6	7	3	3	3	0
26	aux-output_c__gen_compare_reg	9	11	4	4	4	0
27	aux-output_c__leaf_return_peephole_ok	3	2	1	1	1	0
28	aux-output_c__eligible_for_epilogue_delay	37	64	231	231	231	0
29	aux-output_c__short_branch	3	2	1	1	1	0
30	aux-output_c__reg_unused_after	19	32	46	86	66	23.3
31	aux-output_c__check_pic	8	11	3	5	5	0
32	aux-output_c__pic_address_needs_scratch	9	13	6	6	6	0
33	aux-output_c__legitimize_pic_address	35	53	23	41	41	0
34	aux-output_c__initialize_pic	3	2	1	1	1	0
35	aux-output_c__finalize_pic	7	8	2	3	3	0
36	aux-output_c__sparc_address_cost	12	16	6	6	6	0
37	aux-output_c__emit_move_sequence	52	86	42_528	42_528	42_528	0
38	aux-output_c__singlemove_string	19	27	9	11	11	0
39	aux-output_c__mem_aligned_8	25	40	111	111	111	0
40	aux-output_c__output_move_double	87	137	206_520_192	206_520_300	206_520_300	0
41	aux-output_c__output_move_quad	83	130	105_515_136	105_633_612	105_633_612	0
42	aux-output_c__output_fp_move_double	21	33	16	37	37	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	aux-output_c_output_fp_move_quad	21	33	16	35	35	0
44	aux-output_c_find_addr_reg	24	40	44	221	135	38.9
45	aux-output_c_output_scc_insn	21	31	112	114	114	0
46	aux-output_c_compute_frame_size	21	33	108	116	116	0
47	aux-output_c_output_function_prologue	80	119	80_784	80_811	80_811	0
48	aux-output_c_output_function_epilogue	84	124	76_296	85_326	85_299	0
49	aux-output_c_sparc_builtin_saveregs	10	13	6	8	8	0
50	aux-output_c_output_cbranch	53	81	920	920	920	0
51	aux-output_c_output_return	24	33	11	11	11	0
52	aux-output_c_order_regs_for_local_alloc	5	5	2	2	2	0
53	aux-output_c_registers_ok_for_ldd_peep	8	10	4	4	4	0
54	aux-output_c_addrs_ok_for_ldd_peep	19	30	27	27	27	0
55	aux-output_c_register_ok_for_ldd	8	9	3	3	3	0
56	aux-output_c_print_operand	84	132	130	130	130	0
57	aux-output_c_output_double_int	16	21	6	7	7	0
58	aux-output_c_sparc_type_code	57	98	71	91	87	4.4
59	aux-output_c_sparc_frw_compute_frame_size	39	62	18_252	18_424	18_424	0
60	aux-output_c_sparc_frw_save_restore	36	54	1_185	1_206	1_206	0
61	aux-output_c_sparc_frw_output_function_prologue	19	24	36	36	36	0
62	aux-output_c_sparc_frw_output_function_epilogue	35	49	784	976	976	0
63	aux-output_c_sparc_frw_epilogue_delay_slots	8	9	4	4	4	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	aux-output_c__sparc_frw_eligible_for_epilogue_delay	7	9	4	4	4	0
65	bc-emit_c__prsym	6	6	2	2	2	0
66	bc-emit_c__hash	5	6	2	3	3	0
67	bc-emit_c__sym_lookup	7	9	3	5	4	20
68	bc-emit_c__bc_sym_write	13	18	5	18	14	22.2
69	bc-emit_c__seg_create	3	2	1	1	1	0
70	bc-emit_c__seg_align	7	8	2	3	3	0
71	bc-emit_c__seg_data	7	8	2	3	3	0
72	bc-emit_c__seg_skip	7	8	2	3	3	0
73	bc-emit_c__seg_defsym	6	6	2	2	2	0
74	bc-emit_c__seg_refsym	3	2	1	1	1	0
75	bc-emit_c__seg_concat	13	18	9	11	11	0
76	bc-emit_c__bc_seg_write	27	44	244	864	796	7.9
77	bc-emit_c__bc_initialize	15	21	18	37	37	0
78	bc-emit_c__bc_define_pointer	3	2	1	1	1	0
79	bc-emit_c__bc_begin_function	3	2	1	1	1	0
80	bc-emit_c__bc_align_bytecode	3	2	1	1	1	0
81	bc-emit_c__bc_emit_bytecode_const	5	5	2	2	2	0
82	bc-emit_c__bc_get_bytecode_label	3	2	1	1	1	0
83	bc-emit_c__bc_emit_bytecode_labeldef	7	8	3	3	3	0
84	bc-emit_c__bc_emit_bytecode_labelref	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	bc-emit_c__bc_emit_code_labelref	3	2	1	1	1	0
86	bc-emit_c__bc_end_function	11	17	10	14	14	0
87	bc-emit_c__bc_align_const	3	2	1	1	1	0
88	bc-emit_c__bc_emit_const	3	2	1	1	1	0
89	bc-emit_c__bc_emit_const_skip	3	2	1	1	1	0
90	bc-emit_c__bc_emit_const_labeldef	3	2	1	1	1	0
91	bc-emit_c__bc_emit_const_labelref	3	2	1	1	1	0
92	bc-emit_c__bc_align_data	3	2	1	1	1	0
93	bc-emit_c__bc_emit_data	3	2	1	1	1	0
94	bc-emit_c__bc_emit_data_skip	3	2	1	1	1	0
95	bc-emit_c__bc_emit_data_labeldef	3	2	1	1	1	0
96	bc-emit_c__bc_emit_data_labelref	3	2	1	1	1	0
97	bc-emit_c__bc_emit_common	6	6	2	2	2	0
98	bc-emit_c__bc_globalize_label	3	2	1	1	1	0
99	bc-emit_c__bc_text	3	2	1	1	1	0
100	bc-emit_c__bc_data	3	2	1	1	1	0
101	bc-emit_c__bc_align	6	6	2	2	2	0
102	bc-emit_c__bc_emit	6	6	2	2	2	0
103	bc-emit_c__bc_emit_skip	6	6	2	2	2	0
104	bc-emit_c__bc_emit_labeldef	6	6	2	2	2	0
105	bc-emit_c__bc_emit_labelref	6	6	2	2	2	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	bc-emit_c__bc_write_file	3	2	1	1	1	0
107	bc-emit_c__bc_gen_rtx	5	5	2	2	2	0
108	bc-emit_c__bc_print_rtl	3	2	1	1	1	0
109	bc-emit_c__bc_emit_bytecode	6	7	3	3	3	0
110	bc-emit_c__bc_emit_instruction	21	35	14	41	28	31.7
111	bc-emit_c__bc_emit_trampoline	3	2	1	1	1	0
112	bc-emit_c__bc_xstrdup	3	2	1	1	1	0
113	bc-optab_c__conversion_reasonable_p	85	147	1_428_433_129	1_428_449_338	1_428_441_233	0
114	bc-optab_c__deduce_conversion	57	90	581_760	647_622	646_745	0.1
115	bc-optab_c__emit_typecode_conversion	7	9	4	5	5	0
116	bc-optab_c__bc_init_mode_to_code_map	5	5	1	2	2	0
117	bc-optab_c__preferred_typecode	10	12	4	8	8	0
118	bc-optab_c__bc_expand_conversion	3	2	1	1	1	0
119	bc-optab_c__bc_expand_truth_conversion	3	2	1	1	1	0
120	bc-optab_c__bc_expand_binary_operation	14	19	9	34	26	23.5
121	bc-optab_c__bc_expand_unary_operation	12	16	5	18	14	22.2
122	bc-optab_c__bc_expand_increment	7	9	1	5	4	20
123	caller-save_c__choose_hard_reg_mode	27	41	350	362	362	0
124	caller-save_c__init_caller_save	41	63	1_440	1_529	1_496	2.2
125	caller-save_c__init_save_areas	7	8	1	4	3	25
126	caller-save_c__setup_save_areas	42	66	414	514	478	7

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	caller-save_c__save_call_clobbered_regs	63	101	45_645_029	319_848_454	319_708_950	0
128	caller-save_c__set_reg_live	12	17	12	13	13	0
129	caller-save_c__clear_reg_live	10	14	6	7	7	0
130	caller-save_c__restore_referenced_regs	29	46	45	53	52	1.9
131	caller-save_c__insert_save_restore	41	66	135	205	155	24.4
132	calls_c__calls_function	3	2	1	1	1	0
133	calls_c__calls_function_1	48	81	325	337	330	2.1
134	calls_c__prepare_call_address	9	11	8	8	8	0
135	calls_c__emit_call_1	16	23	16	35	33	5.7
136	calls_c__expand_call	458	751	3.95352×10^{44}	3.95352×10^{44}	3.95352×10^{44}	0
137	calls_c__emit_library_call	80	130	360_984_576	361_065_506	361_036_130	0
138	calls_c__emit_library_call_value	136	222	1.39295×10^{14}	1.39295×10^{14}	1.39295×10^{14}	0
139	calls_c__store_one_arg	73	113	11_028_961	11_029_020	11_029_018	0
140	c-aux-info_c__concat	7	8	4	4	4	0
141	c-aux-info_c__concat3	9	11	8	8	8	0
142	c-aux-info_c__affix_data_type	11	13	2	4	4	0
143	c-aux-info_c__gen_formal_list_for_type	21	30	41	45	45	0
144	c-aux-info_c__deserves_ellipsis	11	16	12	13	13	0
145	c-aux-info_c__gen_formal_list_for_func_def	18	26	70	76	76	0
146	c-aux-info_c__gen_type	60	93	196	203	203	0
147	c-aux-info_c__gen_decl	22	30	194	194	194	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	c-aux-info_c_gen_aux_info_record	12	15	9	9	9	0
149	c-common_c_declare_function_name	11	13	5	5	5	0
150	c-common_c_declare_hidden_char_array	5	5	2	2	2	0
151	c-common_c_combine_strings	43	63	2_048	2_059	2_058	0
152	c-common_c_decl_attributes	64	102	198	293	286	2.4
153	c-common_c_init_function_format_info	3	2	1	1	1	0
154	c-common_c_record_function_format	10	14	5	9	7	22.2
155	c-common_c_check_function_format	10	13	5	7	7	0
156	c-common_c_check_format_info	207	349	5.76466×10^{36}	6.24505×10^{36}	6.24505×10^{36}	0
157	c-common_c_constant_expression_warning	6	7	3	3	3	0
158	c-common_c_overflow_warning	5	5	2	2	2	0
159	c-common_c_unsigned_conversion_warning	11	15	6	6	6	0
160	c-common_c_convert_and_check	14	21	15	15	15	0
161	c-common_c_c_expand_expr_stmt	11	16	20	20	20	0
162	c-common_c_check_case_value	13	19	13	14	14	0
163	c-common_c_type_for_size	40	57	19	19	19	0
164	c-common_c_type_for_mode	48	70	24	24	24	0
165	c-common_c_binary_op_error	28	50	24	24	24	0
166	c-common_c_shorten_compare	151	244	1.57043×10^{10}	1.57043×10^{10}	1.57043×10^{10}	0
167	c-common_c_truthvalue_conversion	38	63	39	39	39	0
168	c-common_c_get_directive_line	26	40	36	364	364	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	c-common_c__c_build_type_variant	5	5	2	2	2	0
170	c-convert_c__convert	26	36	12	12	12	0
171	c-decl_c__c_decode_option	128	195	69	69	69	0
172	c-decl_c__print_lang_decl	3	2	1	1	1	0
173	c-decl_c__print_lang_type	3	2	1	1	1	0
174	c-decl_c__print_lang_identifier	3	2	1	1	1	0
175	c-decl_c__finish_incomplete_decl	8	11	5	5	5	0
176	c-decl_c__make_binding_level	3	2	1	1	1	0
177	c-decl_c__global_bindings_p	3	2	1	1	1	0
178	c-decl_c__keep_next_level	3	2	1	1	1	0
179	c-decl_c__kept_level_p	10	15	11	11	11	0
180	c-decl_c__declare_parm_level	3	2	1	1	1	0
181	c-decl_c__in_parm_level_p	3	2	1	1	1	0
182	c-decl_c__pushlevel	12	16	16	16	16	0
183	c-decl_c__poplevel	62	101	1_088_640	1_088_659	1_088_659	0
184	c-decl_c__delete_block	10	13	6	8	8	0
185	c-decl_c__insert_block	3	2	1	1	1	0
186	c-decl_c__set_block	3	2	1	1	1	0
187	c-decl_c__push_label_level	6	6	2	2	2	0
188	c-decl_c__pop_label_level	20	29	30	41	41	0
189	c-decl_c__pushtag	11	15	12	13	13	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	c-decl_c_duplicate_decls	206	353	1.29847×10^{15}	1.29847×10^{15}	1.29847×10^{15}	0
191	c-decl_c_pushdecl	145	244	1.22491×10^{11}	1.22491×10^{11}	1.22491×10^{11}	0
192	c-decl_c_pushdecl_top_level	3	2	1	1	1	0
193	c-decl_c_implicitly_declare	13	18	21	21	21	0
194	c-decl_c_redeclaration_error_message	27	44	23	23	23	0
195	c-decl_c_lookup_label	11	14	5	5	5	0
196	c-decl_c_shadow_label	5	5	2	2	2	0
197	c-decl_c_define_label	9	11	6	6	6	0
198	c-decl_c_getdecls	3	2	1	1	1	0
199	c-decl_c_gettags	3	2	1	1	1	0
200	c-decl_c_storedecls	3	2	1	1	1	0
201	c-decl_c_storetags	3	2	1	1	1	0
202	c-decl_c_lookup_tag	15	22	9	21	14	33.3
203	c-decl_c_pending_xref_error	5	5	2	2	2	0
204	c-decl_c_lookup_tag_reverse	10	14	4	12	7	41.7
205	c-decl_c_lookup_name	7	8	3	3	3	0
206	c-decl_c_lookup_name_current_level	10	14	5	6	6	0
207	c-decl_c_init_decl_processing	23	31	288	288	288	0
208	c-decl_c_builtin_function	13	18	36	36	36	0
209	c-decl_c_shadow_tag	3	2	1	1	1	0
210	c-decl_c_shadow_tag_warned	21	32	78	90	90	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	c-decl_c__groktypename	5	5	2	2	2	0
212	c-decl_c__groktypename_in_parm_context	5	5	2	2	2	0
213	c-decl_c__start_decl	40	62	2_496	2_496	2_496	0
214	c-decl_c__finish_decl	89	143	1.43686×10^{10}	1.43686×10^{10}	1.43686×10^{10}	0
215	c-decl_c__maybe_build_cleanup	3	2	1	1	1	0
216	c-decl_c__push_parm_decl	3	2	1	1	1	0
217	c-decl_c__clear_parm_order	3	2	1	1	1	0
218	c-decl_c__complete_array_type	24	34	56	58	58	0
219	c-decl_c__grokdeclarator	459	745	1.56156×10^{30}	1.56156×10^{30}	1.56156×10^{30}	0
220	c-decl_c__grokparms	30	46	253	262	262	0
221	c-decl_c__get_parm_info	38	60	5_281	5_307	5_307	0
222	c-decl_c__parmlist_tags_warning	18	27	26	51	51	0
223	c-decl_c__xref_tag	12	16	10	10	10	0
224	c-decl_c__start_struct	14	19	20	20	20	0
225	c-decl_c__grokfield	3	2	1	1	1	0
226	c-decl_c__field_decl_cmp	3	2	1	1	1	0
227	c-decl_c__finish_struct	145	242	7.18561×10^{13}	7.18561×10^{13}	7.18561×10^{13}	0
228	c-decl_c__layout_array_type	5	5	2	2	2	0
229	c-decl_c__start_enum	12	16	24	24	24	0
230	c-decl_c__finish_enum	39	60	9_720	9_729	9_729	0
231	c-decl_c__build_enumerator	25	39	1_020	1_022	1_021	0.1

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	c-decl_c__start_function	50	82	299_905	299_905	299_905	0
233	c-decl_c__c_mark_varargs	3	2	1	1	1	0
234	c-decl_c__store_parm_decls	112	181	13_653_368	13_653_745	13_653_623	0
235	c-decl_c__combine_parm_decls	39	62	7_520	7_623	7_622	0
236	c-decl_c__finish_function	27	41	1_134	1_134	1_134	0
237	c-decl_c__push_c_function_context	5	5	2	2	2	0
238	c-decl_c__pop_c_function_context	9	12	6	8	8	0
239	c-iterate_c__init_iterators	9	11	8	8	8	0
240	c-iterate_c__iterator_for_loop_start	3	2	1	1	1	0
241	c-iterate_c__iterator_for_loop_end	3	2	1	1	1	0
242	c-iterate_c__iterator_expand	3	2	1	1	1	0
243	c-iterate_c__expand_stmt_with_iterators_1	6	6	2	2	2	0
244	c-iterate_c__collect_iterators	46	70	64	66	66	0
245	c-iterate_c__iterator_loop_prologue	9	11	8	8	8	0
246	c-iterate_c__iterator_loop_epilogue	9	11	8	8	8	0
247	c-iterate_c__top_level_ixpansion_p	3	2	1	1	1	0
248	c-iterate_c__isn_append	8	9	3	3	3	0
249	c-iterate_c__istack_sublevel_to_current	11	15	11	11	11	0
250	c-iterate_c__push_iterator_stack	9	11	8	8	8	0
251	c-iterate_c__pop_iterator_stack	5	5	1	2	2	0
252	c-iterate_c__add_ixpansion	12	16	17	17	17	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
253	c-iterate_c__delete_ixpansion	18	27	22	47	45	4.3
254	c-lang_c__lang_decode_option	3	2	1	1	1	0
255	c-lang_c__lang_init	3	2	1	1	1	0
256	c-lang_c__lang_finish	3	2	1	1	1	0
257	c-lang_c__lang_identify	3	2	1	1	1	0
258	c-lang_c__print_lang_statistics	3	2	1	1	1	0
259	c-lang_c__lookup_interface	3	2	1	1	1	0
260	c-lang_c__is_class_name	3	2	1	1	1	0
261	c-lang_c__maybe_objc_check_decl	3	2	1	1	1	0
262	c-lang_c__maybe_objc_comptypes	3	2	1	1	1	0
263	c-lang_c__maybe_objc_method_name	3	2	1	1	1	0
264	c-lang_c__maybe_building_objc_message_expr	3	2	1	1	1	0
265	c-lang_c__recognize_objc_keyword	3	2	1	1	1	0
266	c-lang_c__build_objc_string	3	2	1	2	2	0
267	c-lang_c__GNU_xref_begin	3	2	1	1	1	0
268	c-lang_c__GNU_xref_end	3	2	1	1	1	0
269	c-lex_c__make_pointer_declarator	3	2	1	1	1	0
270	c-lex_c__forget_protocol_qualifiers	7	8	2	4	4	0
271	c-lex_c__remember_protocol_qualifiers	15	20	6	12	12	0
272	c-lex_c__init_lex	97	151	6.18494×10^{11}	6.18494×10^{11}	6.18494×10^{11}	0
273	c-lex_c__reinit_parse_for_function	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
274	c-lex_c__yyprint	13	18	8	8	8	0
275	c-lex_c__skip_white_space	24	34	3	12	12	0
276	c-lex_c__position_after_white_space	6	6	2	2	2	0
277	c-lex_c__extend_token_buffer	3	2	1	1	1	0
278	c-lex_c__check_newline	113	199	1_099_227	1_099_266	1_099_242	0
279	c-lex_c__readescape	60	97	37	134	134	0
280	c-lex_c__yyerror	16	20	6	6	6	0
281	c-lex_c__yylex	465	750	1.20264×10^{15}	1.20265×10^{15}	1.20265×10^{15}	0
282	c-lex_c__set_yydebug	3	2	1	1	1	0
283	c-lex_c__is_reserved_word	12	17	13	13	13	0
284	combine_c__combine_instructions	43	72	7_260	7_442	7_393	0.7
285	combine_c__init_reg_last_arrays	31	44	16_384	16_384	16_384	0
286	combine_c__setup_incoming_promotions	3	2	1	1	1	0
287	combine_c__set_nonzero_bits_and_sign_copies	27	42	68	68	68	0
288	combine_c__can_combine_p	77	142	4_937_988	4_938_025	4_938_000	0
289	combine_c__combinable_i3pat	33	58	1_264	1_270	1_267	0.2
290	combine_c__try_combine	490	843	3.14075×10^{41}	3.14075×10^{41}	3.14075×10^{41}	0
291	combine_c__undo_all	9	12	6	8	8	0
292	combine_c__find_split_point	137	243	9_694_460	9_694_460	9_694_460	0
293	combine_c__subst	849	1499	4.24045×10^{32}	4.24045×10^{32}	4.24045×10^{32}	0
294	combine_c__expand_compound_operation	33	51	76	76	76	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
295	combine_c_expand_field_assignment	34	52	21_088	21_871	21_657	1
296	combine_c_make_extraction	127	204	1.3219×10^{10}	1.3219×10^{10}	1.3219×10^{10}	0
297	combine_c_make_compound_operation	113	193	1_007_974	1_007_978	1_007_978	0
298	combine_c_get_pos_from_mask	7	8	3	3	3	0
299	combine_c_force_to_mode	154	269	20_076	20_076	20_076	0
300	combine_c_known_cond	54	91	2_040	2_054	2_051	0.1
301	combine_c_make_field_assignment	58	98	24_230	24_230	24_230	0
302	combine_c_apply_distributive_law	42	71	270	270	270	0
303	combine_c_simplify_and_const_int	39	62	702	702	702	0
304	combine_c_nonzero_bits	111	187	1_152	1_152	1_152	0
305	combine_c_num_sign_bit_copies	110	189	313	313	313	0
306	combine_c_extended_count	11	14	5	5	5	0
307	combine_c_merge_outer_ops	57	90	5_432	5_432	5_432	0
308	combine_c_simplify_shift_const	237	410	7.59209×10^{12}	7.59211×10^{12}	7.5921×10^{12}	0
309	combine_c_recog_for_combine	42	67	3_901	3_923	3_911	0.3
310	combine_c_gen_lowpart_for_combine	39	59	319	319	319	0
311	combine_c_gen_rtx_combine	25	40	48	69	59	14.5
312	combine_c_gen_binary	33	55	2_379	2_379	2_379	0
313	combine_c_gen_unary	5	5	2	2	2	0
314	combine_c_simplify_comparison	379	685	3.08255×10^{19}	3.08255×10^{19}	3.08255×10^{19}	0
315	combine_c_reversible_comparison_p	17	25	11	11	11	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
316	combine_c__update_table_tick	16	23	9	12	12	0
317	combine_c__record_value_for_reg	31	48	2_880	2_885	2_885	0
318	combine_c__record_dead_and_set_regs_1	17	25	10	10	10	0
319	combine_c__record_dead_and_set_regs	23	34	33	47	46	2.1
320	combine_c__get_last_value_validate	26	40	36	46	41	10.9
321	combine_c__get_last_value	30	50	317	321	318	0.9
322	combine_c__use_crosses_set_p	28	44	27	43	34	20.9
323	combine_c__reg_dead_at_p_1	15	20	13	13	13	0
324	combine_c__reg_dead_at_p	33	50	294	314	300	4.5
325	combine_c__remove_death	5	5	2	2	2	0
326	combine_c__move_deaths	47	77	88	100	96	4
327	combine_c__reg_bitfield_target_p	25	39	65	67	66	1.5
328	combine_c__distribute_notes	130	224	598_430_833	1_795_292_667	1_196_861_694	33.3
329	combine_c__distribute_links	32	57	353	725	714	1.5
330	combine_c__dump_combine_stats	3	2	1	1	1	0
331	combine_c__dump_combine_total_stats	3	2	1	1	1	0
332	convert_c__convert_to_pointer	17	23	8	10	10	0
333	convert_c__convert_to_real	18	23	7	7	7	0
334	convert_c__convert_to_integer	71	116	887	887	887	0
335	convert_c__convert_to_complex	18	23	7	7	7	0
336	c-parse_c__yyparse	546	942	1.27698×10^{35}	6.96127×10^{36}	6.96127×10^{36}	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
337	cse_c_rtx_cost	53	95	345	351	350	0.3
338	cse_c_new_basic_block	21	30	128	133	132	0.8
339	cse_c_make_new_qty	5	5	1	2	2	0
340	cse_c_make_regs_eqv	33	57	740	864	861	0.3
341	cse_c_delete_reg_equiv	11	13	5	5	5	0
342	cse_c_mention_regs	34	55	178	187	186	0.5
343	cse_c_insert_regs	22	34	27	31	29	6.5
344	cse_c_free_element	3	2	1	1	1	0
345	cse_c_get_element	6	6	2	2	2	0
346	cse_c_remove_from_table	27	41	289	293	293	0
347	cse_c_lookup	11	16	7	13	10	23.1
348	cse_c_lookup_for_remove	16	25	9	17	13	23.5
349	cse_c_lookup_as_function	11	15	5	9	7	22.2
350	cse_c_insert	88	151	26_808_320	26_808_327	26_808_326	0
351	cse_c_merge_equiv_classes	21	31	51	100	100	0
352	cse_c_invalidate	37	58	86	121	111	8.3
353	cse_c_remove_invalid_refs	10	14	4	14	11	21.4
354	cse_c_rehash_using_reg	24	37	32	69	57	17.4
355	cse_c_invalidate_memory	13	20	8	30	23	23.3
356	cse_c_invalidate_for_call	22	34	30	62	51	17.7
357	cse_c_use_related_value	22	35	152	163	157	3.7

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
358	cse_c_canon_hash	55	94	66	220	125	43.2
359	cse_c_safe_hash	3	2	1	1	1	0
360	cse_c_exp_equiv_p	80	138	1_485	1_533	1_506	1.8
361	cse_c_refers_to_p	24	37	18	37	25	32.4
362	cse_c_set_nonvarying_address_components	24	39	492	492	492	0
363	cse_c_refers_to_mem_p	26	40	24	42	30	28.6
364	cse_c_cse_rtx_addr_varies_p	17	28	34	34	34	0
365	cse_c_canon_reg	28	48	30	42	41	2.4
366	cse_c_find_best_addr	125	216	4.59642×10^{10}	4.59643×10^{10}	4.59643×10^{10}	0
367	cse_c_find_comparison_args	58	104	118_261	215_463	214_166	0.6
368	cse_c_simplify_unary_operation	127	212	10_992	11_376	11_376	0
369	cse_c_simplify_binary_operation	277	493	109_382	109_653	109_653	0
370	cse_c_simplify_plus_minus	87	150	75_634_780	75_676_459	75_643_729	0
371	cse_c_cse_gen_binary	28	46	840	840	840	0
372	cse_c_simplify_relational_operation	162	294	3_777_393	3_777_408	3_777_408	0
373	cse_c_simplify_ternary_operation	27	42	40	44	44	0
374	cse_c_fold_rtx	471	852	2.4973×10^{21}	2.4973×10^{21}	2.4973×10^{21}	0
375	cse_c_equiv_constant	31	54	140	144	142	1.4
376	cse_c_gen_lowpart_if_possible	16	22	34	34	34	0
377	cse_c_record_jump_equiv	13	17	10	10	10	0
378	cse_c_record_jump_cond	94	150	11_333_259	11_333_259	11_333_259	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
379	cse_c__cse_insn	641	1105	6.70107×10^{62}	6.70107×10^{62}	6.70107×10^{62}	0
380	cse_c__note_mem_written	20	30	27	27	27	0
381	cse_c__invalidate_from_clobbers	35	57	230	241	241	0
382	cse_c__cse_process_notes	28	48	29	31	31	0
383	cse_c__cse_around_loop	42	75	2_798	2_840	2_836	0.1
384	cse_c__invalidate_skipped_set	18	27	38	38	38	0
385	cse_c__invalidate_skipped_block	11	16	8	11	11	0
386	cse_c__cse_check_loop_start	10	15	8	8	8	0
387	cse_c__cse_set_around_loop	77	135	520_512	527_750	527_744	0
388	cse_c__cse_end_of_basic_block	79	141	33_728_490	34_606_018	34_605_999	0
389	cse_c__cse_main	50	78	22_400	22_449	22_449	0
390	cse_c__cse_basic_block	58	94	78_606	83_905	81_497	2.9
391	cse_c__count_reg_usage	23	38	20	26	25	3.8
392	cse_c__delete_dead_from_cse	39	65	872	1_007	994	1.3
393	c-typeck_c__require_complete_type	6	7	3	3	3	0
394	c-typeck_c__incomplete_type_error	22	32	18	25	23	8
395	c-typeck_c__qualify_type	9	12	9	9	9	0
396	c-typeck_c__common_type	89	146	1_227	1_272	1_258	1.1
397	c-typeck_c__comptypes	45	74	151	151	151	0
398	c-typeck_c__comp_target_types	7	9	4	4	4	0
399	c-typeck_c__function_types_compatible_p	17	26	21	21	21	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
400	c-typeck_c__type_lists_compatible_p	36	59	70	108	104	3.7
401	c-typeck_c__self_promoting_args_p	17	28	20	28	24	14.3
402	c-typeck_c__self_promoting_type_p	12	18	8	8	8	0
403	c-typeck_c__unsigned_type	14	19	7	7	7	0
404	c-typeck_c__signed_type	14	19	7	7	7	0
405	c-typeck_c__signed_or_unsigned_type	25	35	12	12	12	0
406	c-typeck_c__c_sizeof	21	30	11	11	11	0
407	c-typeck_c__c_sizeof_nowarn	11	14	5	5	5	0
408	c-typeck_c__c_size_in_bytes	11	14	5	5	5	0
409	c-typeck_c__c_alignof	9	11	4	4	4	0
410	c-typeck_c__c_alignof_expr	25	36	26	28	28	0
411	c-typeck_c__decl_constant_value	13	21	10	10	10	0
412	c-typeck_c__default_conversion	73	118	165_096	165_098	165_098	0
413	c-typeck_c__lookup_field	27	43	27	49	34	30.6
414	c-typeck_c__build_component_ref	34	50	98	98	98	0
415	c-typeck_c__build_indirect_ref	21	31	54	54	54	0
416	c-typeck_c__build_array_ref	45	72	1_096	1_097	1_097	0
417	c-typeck_c__build_function_call	29	45	344	345	345	0
418	c-typeck_c__convert_arguments	76	123	4_271_444	5_339_300	5_339_300	0
419	c-typeck_c__parser_build_binary_op	56	98	4_500_625	4_500_625	4_500_625	0
420	c-typeck_c__build_binary_op	268	465	2.47751×10^{13}	2.47751×10^{13}	2.47751×10^{13}	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
421	c-typeck_c__pointer_int_sum	26	39	196	196	196	0
422	c-typeck_c__pointer_diff	13	18	36	36	36	0
423	c-typeck_c__build_unary_op	124	204	9_489	9_493	9_491	0
424	c-typeck_c__lvalue_p	21	31	19	19	19	0
425	c-typeck_c__lvalue_or_else	5	5	2	2	2	0
426	c-typeck_c__unary_complex_lvalue	8	9	3	3	3	0
427	c-typeck_c__pedantic_lvalue_warning	8	10	4	4	4	0
428	c-typeck_c__readonly_warning	15	19	12	12	12	0
429	c-typeck_c__mark_addressable	23	33	17	18	18	0
430	c-typeck_c__build_conditional_expr	86	132	476_568	476_568	476_568	0
431	c-typeck_c__build_compound_expr	3	2	1	1	1	0
432	c-typeck_c__internal_build_compound_expr	15	21	12	12	12	0
433	c-typeck_c__build_c_cast	70	115	189_053	189_055	189_054	0
434	c-typeck_c__build_modify_expr	41	65	1_026	1_026	1_026	0
435	c-typeck_c__convert_for_assignment	115	190	43_134	43_140	43_137	0
436	c-typeck_c__warn_for_assignment	11	14	7	7	7	0
437	c-typeck_c__initializer_constant_valid_p	41	73	223	223	223	0
438	c-typeck_c__valid_compound_expr_initializer	10	13	5	5	5	0
439	c-typeck_c__store_init_value	12	18	13	14	14	0
440	c-typeck_c__push_string	8	9	3	3	3	0
441	c-typeck_c__push_member_name	11	13	6	6	6	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
442	c-typeck_c__push_array_bounds	8	9	3	3	3	0
443	c-typeck_c__spelling_length	9	11	3	5	5	0
444	c-typeck_c__print_spelling	11	16	6	13	12	7.7
445	c-typeck_c__get_spelling	9	11	5	5	5	0
446	c-typeck_c__error_init	8	9	4	4	4	0
447	c-typeck_c__pedwarn_init	8	9	4	4	4	0
448	c-typeck_c__digest_init	94	157	41_799	41_831	41_815	0
449	c-typeck_c__start_init	17	25	48	48	48	0
450	c-typeck_c__finish_init	7	9	4	5	5	0
451	c-typeck_c__really_start_incremental_init	19	25	40	40	40	0
452	c-typeck_c__push_init_level	36	55	1_400	1_403	1_403	0
453	c-typeck_c__check_init_type_bitfields	9	13	5	6	6	0
454	c-typeck_c__pop_init_level	61	94	1_350	1_412	1_411	0.1
455	c-typeck_c__set_init_index	25	40	280	282	282	0
456	c-typeck_c__set_init_label	17	23	11	19	13	31.6
457	c-typeck_c__output_init_element	70	111	2_871_792	2_871_795	2_871_795	0
458	c-typeck_c__output_pending_init_elements	42	64	22	209	178	14.8
459	c-typeck_c__process_init_element	88	149	111_771	111_805	111_805	0
460	c-typeck_c__c_expand_asm_operands	24	37	98	107	107	0
461	c-typeck_c__c_expand_return	19	27	22	22	22	0
462	c-typeck_c__c_expand_start_case	13	19	25	25	25	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
463	dbxout_c_dbxout_init	15	22	36	36	36	0
464	dbxout_c_dbxout_typedefs	8	11	5	5	5	0
465	dbxout_c_dbxout_source_file	7	9	4	4	4	0
466	dbxout_c_dbxout_source_line	3	2	1	1	1	0
467	dbxout_c_dbxout_finish	3	2	1	1	1	0
468	dbxout_c_dbxout_continue	3	2	1	1	1	0
469	dbxout_c_dbxout_type_fields	40	62	1_454	4_720	3_267	30.8
470	dbxout_c_dbxout_type_method_1	17	22	36	36	36	0
471	dbxout_c_dbxout_type_methods	57	99	10_217	11_942	11_498	3.7
472	dbxout_c_dbxout_range_type	13	16	12	12	12	0
473	dbxout_c_dbxout_type	130	206	127_536	127_696	127_694	0
474	dbxout_c_print_int_cst_octal	11	14	8	8	8	0
475	dbxout_c_print_octal	5	6	2	3	3	0
476	dbxout_c_dbxout_type_name	9	11	2	4	4	0
477	dbxout_c_dbxout_symbol	67	115	5_607	5_611	5_611	0
478	dbxout_c_dbxout_symbol_location	48	73	374	376	375	0.3
479	dbxout_c_dbxout_symbol_name	12	15	16	16	16	0
480	dbxout_c_dbxout_prepare_symbol	3	2	1	1	1	0
481	dbxout_c_dbxout_finish_symbol	9	11	6	6	6	0
482	dbxout_c_dbxout_syms	5	6	2	3	3	0
483	dbxout_c_dbxout_parms	64	98	491	981	981	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
484	dbxout_c__dbxout_reg_parms	20	33	55	109	109	0
485	dbxout_c__dbxout_args	5	6	2	3	3	0
486	dbxout_c__dbxout_types	8	11	4	7	7	0
487	dbxout_c__dbxout_block	22	35	110	221	220	0.5
488	dbxout_c__dbxout_really_begin_function	5	5	2	2	2	0
489	dbxout_c__dbxout_begin_function	3	2	1	1	1	0
490	dbxout_c__dbxout_function	3	2	1	1	1	0
491	emit-rtl_c__gen_rtx	39	61	33	51	44	13.7
492	emit-rtl_c__gen_rtvec	7	9	3	4	4	0
493	emit-rtl_c__gen_rtvec_v	7	9	3	4	4	0
494	emit-rtl_c__gen_reg_rtx	22	30	19	21	21	0
495	emit-rtl_c__mark_reg_pointer	3	2	1	1	1	0
496	emit-rtl_c__max_reg_num	3	2	1	1	1	0
497	emit-rtl_c__max_label_num	7	8	3	3	3	0
498	emit-rtl_c__get_first_label_num	3	2	1	1	1	0
499	emit-rtl_c__gen_lowpart_common	96	158	734_305	734_405	734_405	0
500	emit-rtl_c__gen_realpart	3	2	1	1	1	0
501	emit-rtl_c__gen_imagpart	3	2	1	1	1	0
502	emit-rtl_c__gen_lowpart	15	20	17	18	18	0
503	emit-rtl_c__gen_highpart	29	44	28	35	35	0
504	emit-rtl_c__subreg_lowpart_p	13	16	6	6	6	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
505	emit-rtl_c__operand_subword	67	103	1_128	1_129	1_129	0
506	emit-rtl_c__operand_subword_force	9	12	4	7	7	0
507	emit-rtl_c__reverse_comparison	11	13	6	6	6	0
508	emit-rtl_c__change_address	13	18	12	21	21	0
509	emit-rtl_c__gen_label_rtx	6	6	2	2	2	0
510	emit-rtl_c__gen_inline_header_rtx	3	2	1	1	1	0
511	emit-rtl_c__set_new_first_and_last_insn	3	2	1	1	1	0
512	emit-rtl_c__set_new_first_and_last_label_num	3	2	1	1	1	0
513	emit-rtl_c__save_emit_status	3	2	1	1	1	0
514	emit-rtl_c__restore_emit_status	5	5	1	2	2	0
515	emit-rtl_c__unshare_all_rtl	7	9	3	5	5	0
516	emit-rtl_c__copy_rtx_if_shared	48	85	902	913	912	0.1
517	emit-rtl_c__reset_used_flags	13	21	12	18	17	5.6
518	emit-rtl_c__make_safe_from	23	37	66	69	69	0
519	emit-rtl_c__get_insns	3	2	1	1	1	0
520	emit-rtl_c__get_last_insn	3	2	1	1	1	0
521	emit-rtl_c__set_last_insn	5	5	1	2	2	0
522	emit-rtl_c__get_last_insn_anywhere	10	13	4	6	5	16.7
523	emit-rtl_c__get_max_uid	3	2	1	1	1	0
524	emit-rtl_c__next_insn	8	11	5	5	5	0
525	emit-rtl_c__previous_insn	8	11	5	5	5	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
526	emit-rtl_c__next_nonnote_insn	6	8	3	4	4	0
527	emit-rtl_c__prev_nonnote_insn	6	8	3	4	4	0
528	emit-rtl_c__next_real_insn	8	12	5	6	6	0
529	emit-rtl_c__prev_real_insn	8	12	5	6	6	0
530	emit-rtl_c__next_active_insn	12	19	8	10	10	0
531	emit-rtl_c__prev_active_insn	12	19	8	10	10	0
532	emit-rtl_c__next_label	6	8	3	4	4	0
533	emit-rtl_c__prev_label	6	8	3	4	4	0
534	emit-rtl_c__try_split	22	33	72	75	75	0
535	emit-rtl_c__make_insn_raw	3	2	1	1	1	0
536	emit-rtl_c__make_jump_insn_raw	3	2	1	1	1	0
537	emit-rtl_c__add_insn	7	8	4	4	4	0
538	emit-rtl_c__add_insn_after	16	24	21	23	23	0
539	emit-rtl_c__delete_insns_since	6	6	2	2	2	0
540	emit-rtl_c__reorder_insns	15	20	64	64	64	0
541	emit-rtl_c__find_line_note	9	13	5	7	7	0
542	emit-rtl_c__reorder_insns_with_line_notes	8	10	5	5	5	0
543	emit-rtl_c__emit_insn_before	9	12	5	6	6	0
544	emit-rtl_c__emit_jump_insn_before	6	6	2	2	2	0
545	emit-rtl_c__emit_call_insn_before	3	2	1	1	1	0
546	emit-rtl_c__emit_barrier_before	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
547	emit-rtl_c__emit_note_before	3	2	1	1	1	0
548	emit-rtl_c__emit_insn_after	9	12	5	6	6	0
549	emit-rtl_c__emit_insn_after_with_line_notes	7	8	4	4	4	0
550	emit-rtl_c__emit_jump_insn_after	6	6	2	2	2	0
551	emit-rtl_c__emit_barrier_after	3	2	1	1	1	0
552	emit-rtl_c__emit_label_after	5	5	2	2	2	0
553	emit-rtl_c__emit_note_after	3	2	1	1	1	0
554	emit-rtl_c__emit_line_note_after	7	8	3	3	3	0
555	emit-rtl_c__emit_insn	9	12	5	6	6	0
556	emit-rtl_c__emit_insns	5	6	2	3	3	0
557	emit-rtl_c__emit_insns_before	5	6	2	3	3	0
558	emit-rtl_c__emit_insns_after	12	17	9	11	11	0
559	emit-rtl_c__emit_jump_insn	6	6	2	2	2	0
560	emit-rtl_c__emit_call_insn	6	6	2	2	2	0
561	emit-rtl_c__emit_label	5	5	2	2	2	0
562	emit-rtl_c__emit_barrier	3	2	1	1	1	0
563	emit-rtl_c__emit_line_note	6	6	2	2	2	0
564	emit-rtl_c__emit_note	14	20	16	16	16	0
565	emit-rtl_c__emit_line_note_force	3	2	1	1	1	0
566	emit-rtl_c__force_next_line_note	3	2	1	1	1	0
567	emit-rtl_c__classify_insn	23	36	16	28	20	28.6

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
568	emit-rtl_c__emit	15	20	6	7	7	0
569	emit-rtl_c__start_sequence	6	6	2	2	2	0
570	emit-rtl_c__push_to_sequence	7	10	4	5	5	0
571	emit-rtl_c__push_topmost_sequence	5	6	2	3	3	0
572	emit-rtl_c__pop_topmost_sequence	5	6	2	3	3	0
573	emit-rtl_c__end_sequence	3	2	1	1	1	0
574	emit-rtl_c__in_sequence_p	3	2	1	1	1	0
575	emit-rtl_c__gen_sequence	15	22	26	28	28	0
576	emit-rtl_c__restore_reg_data	14	21	21	29	29	0
577	emit-rtl_c__restore_reg_data_1	32	51	50	57	56	1.8
578	emit-rtl_c__init_emit	9	11	4	5	5	0
579	emit-rtl_c__init_emit_once	33	50	320	357	352	1.4
580	explow_c__plus_constant_wide	36	59	51	52	52	0
581	explow_c__plus_constant_for_output_wide	6	6	2	2	2	0
582	explow_c__eliminate_constant_term	14	21	23	23	23	0
583	explow_c__find_next_ref	19	30	14	36	22	38.9
584	explow_c__expr_size	6	7	3	3	3	0
585	explow_c__break_out_memory_refs	23	39	278	278	278	0
586	explow_c__copy_all_regs	14	20	10	10	10	0
587	explow_c__memory_address	157	286	1.47306×10^{11}	1.47306×10^{11}	1.47306×10^{11}	0
588	explow_c__memory_address_noforce	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
589	explow_c__validize_mem	7	8	3	3	3	0
590	explow_c__stabilize	12	16	8	8	8	0
591	explow_c__copy_to_reg	7	8	4	4	4	0
592	explow_c__copy_addr_to_reg	3	2	1	1	1	0
593	explow_c__copy_to_mode_reg	10	13	8	10	10	0
594	explow_c__force_reg	14	20	12	12	12	0
595	explow_c__force_not_mem	7	8	3	3	3	0
596	explow_c__copy_to_suggested_reg	7	8	3	3	3	0
597	explow_c__adjust_stack	6	7	3	3	3	0
598	explow_c__anti_adjust_stack	6	7	3	3	3	0
599	explow_c__round_push	7	8	3	3	3	0
600	explow_c__emit_stack_save	18	24	12	14	14	0
601	explow_c__emit_stack_restore	8	9	4	4	4	0
602	explow_c__allocate_dynamic_stack_space	13	18	24	24	24	0
603	explow_c__hard_function_value	7	8	3	3	3	0
604	explow_c__hard_libcall_value	7	8	3	3	3	0
605	explow_c__rtx_to_tree_code	12	18	8	8	8	0
606	expmed_c__init_expmed	11	14	8	16	16	0
607	expmed_c__negate_rtx	10	12	4	4	4	0
608	expmed_c__store_bit_field	67	103	43_776	43_783	43_783	0
609	expmed_c__store_fixed_bit_field	57	85	29_955	29_957	29_957	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
610	expmed_c__store_split_bit_field	32	49	2_940	4_476	4_428	1.1
611	expmed_c__extract_bit_field	70	109	96_336	109_308	109_303	0
612	expmed_c__extract_fixed_bit_field	43	63	619	620	620	0
613	expmed_c__mask_rtx	15	19	32	32	32	0
614	expmed_c__lshift_value	10	12	6	6	6	0
615	expmed_c__extract_split_bit_field	26	37	388	604	508	15.9
616	expmed_c__expand_inc	5	5	2	2	2	0
617	expmed_c__expand_dec	5	5	2	2	2	0
618	expmed_c__expand_shift	54	84	6_624	13_236	11_032	16.7
619	expmed_c__synth_mult	54	86	64_264	64_269	64_269	0
620	expmed_c__expand_mult	58	92	23_121	48_231	48_153	0.2
621	expmed_c__expand_divmod	139	222	3.77976×10^{12}	8.65687×10^{12}	8.65687×10^{12}	0
622	expmed_c__make_tree	32	47	17	17	17	0
623	expmed_c__expand_mult_add	6	6	2	2	2	0
624	expmed_c__expand_and	17	23	18	24	24	0
625	expmed_c__emit_store_flag	120	197	985_227_264	985_236_560	985_236_560	0
626	expr_c__bc_init_mode_to_opcode_maps	5	5	1	2	2	0
627	expr_c__init_expr_once	21	33	71	210	176	16.2
628	expr_c__init_expr	3	2	1	1	1	0
629	expr_c__save_expr_status	3	2	1	1	1	0
630	expr_c__restore_expr_status	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
631	expr_c_enqueue_insn	3	2	1	1	1	0
632	expr_c_protect_from_queue	25	38	39	39	39	0
633	expr_c_queued_subexp_p	12	17	7	7	7	0
634	expr_c_emit_queue	5	6	2	3	3	0
635	expr_c_init_queue	5	5	1	2	2	0
636	expr_c_convert_move	163	263	112_824	219_893	219_889	0
637	expr_c_convert_to_mode	3	2	1	1	1	0
638	expr_c_convert_modes	34	57	2_296	2_296	2_296	0
639	expr_c_move_by_pieces	50	84	368_448	736_934	736_913	0
640	expr_c_move_by_pieces_ninsns	19	28	38	57	55	3.5
641	expr_c_move_by_pieces_1	15	20	17	33	33	0
642	expr_c_emit_block_move	30	50	227	360	296	17.8
643	expr_c_move_block_to_reg	13	21	30	31	31	0
644	expr_c_move_block_from_reg	10	14	3	7	6	14.3
645	expr_c_use_regs	5	6	2	3	3	0
646	expr_c_group_insns	6	6	2	2	2	0
647	expr_c_clear_storage	6	6	2	2	2	0
648	expr_c_emit_move_insn	37	66	5_400	10_802	10_802	0
649	expr_c_emit_move_insn_1	37	58	32	422	390	7.6
650	expr_c_push_block	18	27	30	30	30	0
651	expr_c_gen_push_operand	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
652	expr_c__emit_push_insn	75	117	138_096	138_110	138_110	0
653	expr_c__expand_assignment	49	75	356	362	362	0
654	expr_c__store_expr	71	113	85_706	85_706	85_706	0
655	expr_c__store_constructor	47	70	176	327	302	7.6
656	expr_c__store_field	54	88	3_176	3_176	3_176	0
657	expr_c__get_inner_reference	55	81	576	736	668	9.2
658	expr_c__force_operand	44	71	1_545	1_545	1_545	0
659	expr_c__save_noncopied_parts	10	13	4	7	7	0
660	expr_c__init_noncopied_parts	8	10	3	5	5	0
661	expr_c__safe_from_p	58	105	5_237	5_254	5_248	0.1
662	expr_c__fixed_type_p	10	14	6	6	6	0
663	expr_c__expand_expr	835	1419	7.16615×10^{12}	7.43767×10^{12}	7.43767×10^{12}	0
664	expr_c__bc_expand_expr	78	133	54	60	60	0
665	expr_c__get_pointer_alignment	37	56	41	51	48	5.9
666	expr_c__string_constant	26	44	908	913	911	0.2
667	expr_c__c_strlen	20	29	14	16	15	6.2
668	expr_c__expand_builtin	187	308	416	429	425	0.9
669	expr_c__apply_args_register_offset	7	9	4	4	4	0
670	expr_c__apply_args_size	37	55	603	1_297	972	25.1
671	expr_c__apply_result_size	33	49	302	954	638	33.1
672	expr_c__result_vector	25	35	101	202	202	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
673	expr_c_expand_builtin_apply_args	24	34	164	205	205	0
674	expr_c_expand_builtin_apply	29	43	264	817	804	1.6
675	expr_c_expand_builtin_return	3	2	1	1	1	0
676	expr_c_expand_increment	57	92	725_761	725_761	725_761	0
677	expr_c_preexpand_calls	40	68	262	269	269	0
678	expr_c_init_pending_stack_adjust	3	2	1	1	1	0
679	expr_c_clear_pending_stack_adjust	11	17	17	17	17	0
680	expr_c_do_pending_stack_adjust	7	8	3	3	3	0
681	expr_c_expand_cleanups_to	5	6	2	3	3	0
682	expr_c_jumpifnot	3	2	1	1	1	0
683	expr_c_jumpif	3	2	1	1	1	0
684	expr_c_do_jump	101	164	1_824	1_838	1_838	0
685	expr_c_do_jump_by_parts_greater	31	45	2_688	2_715	2_715	0
686	expr_c_do_jump_by_parts_greater_rtx	31	45	2_688	2_715	2_715	0
687	expr_c_do_jump_by_parts_equality	18	25	64	67	67	0
688	expr_c_do_jump_by_parts_equality_rtx	18	25	64	67	67	0
689	expr_c_do_jump_for_compare	24	36	27	54	48	11.1
690	expr_c_compare	9	10	4	4	4	0
691	expr_c_compare_from_rtx	22	37	664	664	664	0
692	expr_c_do_store_flag	116	194	268_069_762	311_077_798	311_077_796	0
693	expr_c_do_tablejump	7	8	4	4	4	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
694	expr_c__bc_load_memory	13	18	5	9	9	0
695	expr_c__bc_store_memory	12	16	4	9	9	0
696	expr_c__bc_allocate_local	13	17	12	13	13	0
697	expr_c__bc_allocate_variable_array	5	5	2	2	2	0
698	expr_c__bc_load_externaddr	3	2	1	1	1	0
699	expr_c__bc_strdup	3	2	1	1	1	0
700	expr_c__bc_load_externaddr_id	5	5	1	2	2	0
701	expr_c__bc_load_localaddr	3	2	1	1	1	0
702	expr_c__bc_load_parmaddr	5	5	2	2	2	0
703	expr_c__bc_canonicalize_array_ref	5	5	2	2	2	0
704	expr_c__bc_expand_component_address	18	24	15	17	17	0
705	expr_c__bc_push_offset_and_size	3	2	1	1	1	0
706	expr_c__bc_expand_address	32	50	33	35	35	0
707	expr_c__bc_runtime_type_code	8	10	2	4	4	0
708	expr_c__bc_gen_constr_label	9	11	8	8	8	0
709	expr_c__bc_expand_constructor	18	27	20	22	22	0
710	expr_c__bc_store_field	10	12	5	5	5	0
711	expr_c__bc_store_bit_field	3	2	1	1	1	0
712	expr_c__bc_load_bit_field	3	2	1	1	1	0
713	expr_c__bc_adjust_stack	12	15	5	5	5	0
714	final_c__init_final	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
715	final_c_end_final	51	80	216_001	216_011	216_011	0
716	final_c_app_enable	5	5	2	2	2	0
717	final_c_app_disable	5	5	2	2	2	0
718	final_c_dbr_sequence_length	6	6	2	2	2	0
719	final_c_init_insn_lengths	3	2	1	1	1	0
720	final_c_get_attr_length	28	43	43	44	44	0
721	final_c_shorten_branches	57	89	2_652	2_779	2_746	1.2
722	final_c_asm_insn_count	8	11	4	7	7	0
723	final_c_final_start_function	14	18	24	24	24	0
724	final_c_profile_after_prologue	7	8	4	4	4	0
725	final_c_profile_function	5	5	2	2	2	0
726	final_c_final_end_function	8	9	4	4	4	0
727	final_c_add_bb	3	2	1	1	1	0
728	final_c_add_bb_string	13	18	12	15	13	13.3
729	final_c_final	18	29	96	103	103	0
730	final_c_final_scan_insn	167	283	7_536_389	9_275_236	9_275_204	0
731	final_c_output_source_line	10	14	12	12	12	0
732	final_c_alter_subreg	13	17	12	12	12	0
733	final_c_walk_alter_subreg	14	18	6	6	6	0
734	final_c_output_operand_lossage	5	5	1	2	2	0
735	final_c_output_asm_insn	69	108	917	1_149	1_147	0.2

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
736	final_c_output_asm_label	8	9	3	3	3	0
737	final_c_output_operand	10	14	7	9	9	0
738	final_c_output_address	35	50	27	30	30	0
739	final_c_output_addr_const	38	57	19	22	21	4.5
740	final_c_asm_fprintf	23	36	20	78	47	39.7
741	final_c_split_double	14	18	6	6	6	0
742	final_c_leaf_function_p	19	32	28	43	34	20.9
743	final_c_only_leaf_regs_used	10	13	5	8	8	0
744	final_c_leaf_renumber_regs	11	16	9	13	13	0
745	final_c_leaf_renumber_regs_insn	22	36	17	35	28	20
746	flow_c_flow_analysis	20	31	66	99	99	0
747	flow_c_find_basic_blocks	100	177	1_370_348_928	1_562_194_030	1_562_193_319	0
748	flow_c_mark_label_ref	20	33	13	24	22	8.3
749	flow_c_life_analysis	145	240	2.98657×10^{15}	2.98657×10^{15}	2.98657×10^{15}	0
750	flow_c_allocate_for_life_analysis	27	39	4_096	4_097	4_097	0
751	flow_c_init_regset_vector	6	7	2	3	3	0
752	flow_c_propagate_block	109	184	9_950_682_000	1.00099×10^{10}	9_980_298_002	0.3
753	flow_c_insn_dead_p	44	76	5_545	5_565	5_555	0.2
754	flow_c_libcall_dead_p	23	35	10	26	21	19.2
755	flow_c_regno_uninitialized	12	16	10	10	10	0
756	flow_c_regno_clobbered_at_setjmp	18	25	45	45	45	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
757	flow_c__mark_set_regs	12	18	7	10	10	0
758	flow_c__mark_set_1	91	148	173_869_900	173_870_554	173_870_233	0
759	flow_c__mark_used_regs	107	177	2_038_086	2_039_357	2_038_305	0.1
760	flow_c__find_use_as_address	28	48	410	442	424	4.1
761	flow_c__dump_flow_info	46	71	24_704	25_170	25_161	0
762	fold-const_c__encode	5	5	1	2	2	0
763	fold-const_c__decode	5	5	1	2	2	0
764	fold-const_c__force_fit_type	25	35	81	81	81	0
765	fold-const_c__add_double	5	5	1	2	2	0
766	fold-const_c__neg_double	6	6	2	2	2	0
767	fold-const_c__mul_double	34	48	46	56	51	8.9
768	fold-const_c__lshift_double	12	16	5	8	7	12.5
769	fold-const_c__rshift_double	12	17	8	13	12	7.7
770	fold-const_c__lrotate_double	12	16	5	8	7	12.5
771	fold-const_c__rrotate_double	9	12	4	7	6	14.3
772	fold-const_c__div_and_round_double	85	136	1_541_232	1_636_928	1_636_671	0
773	fold-const_c__split_tree	35	56	656	657	657	0
774	fold-const_c__const_binop	104	168	401	416	416	0
775	fold-const_c__size_int	10	12	4	4	4	0
776	fold-const_c__size_binop	20	32	46	46	46	0
777	fold-const_c__fold_convert	27	39	30	30	30	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
778	fold-const_c__non_lvalue	14	20	15	15	15	0
779	fold-const_c__pedantic_non_lvalue	5	5	2	2	2	0
780	fold-const_c__invert_tree_comparison	12	17	6	7	7	0
781	fold-const_c__swap_tree_comparison	10	14	5	6	6	0
782	fold-const_c__operand_equal_p	60	100	22_097	22_099	22_099	0
783	fold-const_c__operand_equal_for_comparison_p	13	21	22	22	22	0
784	fold-const_c__twoval_comparison_p	48	77	272	272	272	0
785	fold-const_c__eval_subst	44	64	410	410	410	0
786	fold-const_c__omit_one_operand	6	6	2	2	2	0
787	fold-const_c__invert_truthvalue	37	62	28	31	31	0
788	fold-const_c__distribute_bit_expr	19	26	13	13	13	0
789	fold-const_c__make_bit_field_ref	3	2	1	1	1	0
790	fold-const_c__optimize_bit_field_compare	40	64	237	237	237	0
791	fold-const_c__decode_field_reference	31	55	4_609	4_617	4_612	0.1
792	fold-const_c__all_ones_mask_p	3	2	1	1	1	0
793	fold-const_c__simple_operand_p	16	27	36	37	37	0
794	fold-const_c__range_test	41	66	192	192	192	0
795	fold-const_c__fold_truthop	100	165	3_475_217	3_475_217	3_475_217	0
796	fold-const_c__fold	734	1280	2.40306×10^{19}	2.40306×10^{19}	2.40306×10^{19}	0
797	function_c__find_function_data	7	9	1	5	4	20
798	function_c__push_function_context	5	5	2	2	2	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
799	function_c_pop_function_context	7	9	4	5	5	0
800	function_c_get_frame_size	3	2	1	1	1	0
801	function_c_assign_stack_local	23	31	52	52	52	0
802	function_c_assign_outer_stack_local	20	27	26	26	26	0
803	function_c_assign_stack_temp	25	41	240	249	249	0
804	function_c_combine_temp_slots	26	39	73	161	153	5
805	function_c_preserve_temp_slots	21	36	19	25	25	0
806	function_c_free_temp_slots	9	13	5	9	9	0
807	function_c_push_temp_slots	3	2	1	1	1	0
808	function_c_pop_temp_slots	8	11	4	7	7	0
809	function_c_put_var_into_stack	25	39	323	324	324	0
810	function_c_put_reg_into_stack	21	30	80	80	80	0
811	function_c_fixup_var_refs	10	15	8	12	12	0
812	function_c_find_fixup_replacement	9	13	6	9	7	22.2
813	function_c_fixup_var_refs_insns	36	58	668	1_385	1_359	1.9
814	function_c_fixup_var_refs_1	105	176	3_788_561	3_788_578	3_788_571	0
815	function_c_fixup_memory_subreg	14	19	24	25	25	0
816	function_c_walk_fixup_memory_subreg	15	23	16	24	23	4.2
817	function_c_fixup_stack_1	19	31	16	24	23	4.2
818	function_c_optimize_bit_field	57	97	40_928	42_931	42_930	0
819	function_c_instantiate_virtual_regs	7	9	3	5	5	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
820	function_c__instantiate_decls	9	12	8	9	9	0
821	function_c__instantiate_decls_1	7	10	4	6	6	0
822	function_c__instantiate_decl	26	46	96	100	98	2
823	function_c__instantiate_virtual_regs_1	129	218	907	2_138	1_584	25.9
824	function_c__delete_handlers	13	20	17	33	33	0
825	function_c__nonlocal_label_rtx_list	5	6	2	3	3	0
826	function_c__use_variable	11	15	6	6	6	0
827	function_c__use_variable_after	11	15	6	6	6	0
828	function_c__max_parm_reg_num	3	2	1	1	1	0
829	function_c__get_first_nonparm_insn	6	6	2	2	2	0
830	function_c__get_first_block_beg	8	11	1	8	6	25
831	function_c__aggregate_value_p	19	28	32	34	33	2.9
832	function_c__assign_parms	217	347	4.49356×10^{22}	4.49473×10^{22}	4.49473×10^{22}	0
833	function_c__locate_and_pad_parm	33	47	1_440	1_440	1_440	0
834	function_c__pad_to_arg_alignment	14	18	12	12	12	0
835	function_c__pad_below	23	32	24	24	24	0
836	function_c__round_down	3	2	1	1	1	0
837	function_c__uninitialized_vars_warning	21	36	84	126	126	0
838	function_c__setjmp_args_warning	9	13	5	9	9	0
839	function_c__setjmp_protect	13	21	14	21	21	0
840	function_c__setjmp_protect_args	10	15	6	11	11	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
841	function_c__lookup_static_chain	12	17	4	8	7	12.5
842	function_c__fix_lexical_addr	26	40	12	31	29	6.5
843	function_c__trampoline_address	26	39	75	88	80	9.1
844	function_c__round_trampoline_addr	3	2	1	1	1	0
845	function_c__identify_blocks	12	17	7	12	12	0
846	function_c__reorder_blocks	14	20	9	16	16	0
847	function_c__blocks_nreverse	5	6	2	3	3	0
848	function_c__all_blocks	7	9	4	5	5	0
849	function_c__bc_build_calldesc	5	6	2	3	3	0
850	function_c__init_function_start	12	16	9	9	9	0
851	function_c__mark_varargs	3	2	1	1	1	0
852	function_c__expand_main_function	5	5	2	2	2	0
853	function_c__bc_expand_function_start	18	25	48	54	52	3.7
854	function_c__bc_expand_function_end	3	2	1	1	1	0
855	function_c__expand_function_start	51	79	69_121	69_125	69_125	0
856	function_c__expand_function_end	44	71	45_145	45_172	45_169	0
857	function_c__record_insns	8	10	3	4	4	0
858	function_c__contains	19	28	10	19	16	15.8
859	function_c__thread_prologue_and_epilogue_insns	3	2	1	1	1	0
860	function_c__reposition_prologue_and_epilogue_notes	3	2	1	1	1	0
861	getpwd_c__getpwd	17	28	17	26	25	3.8

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
862	global_c__global_alloc	136	217	3.03873×10^{14}	3.03873×10^{14}	3.03873×10^{14}	0
863	global_c__allocno_compare	6	6	2	2	2	0
864	global_c__global_conflicts	53	88	13_857	28_723	28_181	1.9
865	global_c__expand_preferences	37	60	61	255	182	28.6
866	global_c__prune_preferences	37	56	27	105	76	27.6
867	global_c__find_reg	224	349	3.86478×10^{16}	3.86478×10^{16}	3.86478×10^{16}	0
868	global_c__retry_global_alloc	9	12	7	7	7	0
869	global_c__record_one_conflict	18	26	11	21	21	0
870	global_c__record_conflicts	9	13	3	9	7	22.2
871	global_c__mark_reg_store	26	39	164	165	165	0
872	global_c__mark_reg_clobber	22	33	43	44	44	0
873	global_c__mark_reg_conflicts	16	24	30	31	31	0
874	global_c__mark_reg_death	18	26	20	21	21	0
875	global_c__mark_reg_live_nc	8	10	4	5	5	0
876	global_c__set_preference	41	67	2_046	3_224	3_214	0.3
877	global_c__mark_elimination	19	27	69	137	137	0
878	global_c__dump_conflicts	41	64	4_005	4_547	4_502	1
879	global_c__dump_global_regs	12	17	8	13	13	0
880	insn-attrtab_c__insn_current_length	7	9	4	4	4	0
881	insn-attrtab_c__insn_variable_length_p	7	9	4	4	4	0
882	insn-attrtab_c__insn_default_length	85	147	110	110	110	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
883	insn-attrtab_c_result_ready_cost	63	107	66	66	66	0
884	insn-attrtab_c_fp_mds_unit_ready_cost	13	18	7	7	7	0
885	insn-attrtab_c_fp_mds_unit_blockage_range	13	18	7	7	7	0
886	insn-attrtab_c_fp_alu_unit_ready_cost	7	9	4	4	4	0
887	insn-attrtab_c_memory_unit_ready_cost	7	9	4	4	4	0
888	insn-attrtab_c_function_units_used	63	107	72	72	72	0
889	insn-attrtab_c_num_delay_slots	21	31	13	13	13	0
890	insn-attrtab_c_get_attr_in_annul_branch_delay	77	140	413	413	413	0
891	insn-attrtab_c_get_attr_in_uncond_branch_delay	77	140	413	413	413	0
892	insn-attrtab_c_get_attr_in_branch_delay	77	140	413	413	413	0
893	insn-attrtab_c_get_attr_in_call_delay	77	140	413	413	413	0
894	insn-attrtab_c_get_attr_type	99	177	123	123	123	0
895	insn-attrtab_c_get_attr_use_clobbered	7	9	4	4	4	0
896	insn-attrtab_c_eligible_for_delay	159	309	118_963	119_003	119_003	0
897	insn-attrtab_c_eligible_for_annul_false	107	187	5_473	5_513	5_513	0
898	insn-attrtab_c_const_num_delay_slots	3	2	1	1	1	0
899	insn-emit_c_gen_cmpsi	3	2	1	1	1	0
900	insn-emit_c_gen_cmpsf	3	2	1	1	1	0
901	insn-emit_c_gen_cmpdf	3	2	1	1	1	0
902	insn-emit_c_gen_cmptf	3	2	1	1	1	0
903	insn-emit_c_gen_seq_special	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
904	insn-emit_c__gen_sne_special	3	2	1	1	1	0
905	insn-emit_c__gen_seq	6	6	2	2	2	0
906	insn-emit_c__gen_sne	6	6	2	2	2	0
907	insn-emit_c__gen_sgt	3	2	1	1	1	0
908	insn-emit_c__gen_slt	3	2	1	1	1	0
909	insn-emit_c__gen_sge	3	2	1	1	1	0
910	insn-emit_c__gen_sle	3	2	1	1	1	0
911	insn-emit_c__gen_sgtu	9	12	7	7	7	0
912	insn-emit_c__gen_sltu	3	2	1	1	1	0
913	insn-emit_c__gen_sgeu	3	2	1	1	1	0
914	insn-emit_c__gen_sleu	9	12	7	7	7	0
915	insn-emit_c__gen_beq	3	2	1	1	1	0
916	insn-emit_c__gen_bne	3	2	1	1	1	0
917	insn-emit_c__gen_bgt	3	2	1	1	1	0
918	insn-emit_c__gen_bgtu	3	2	1	1	1	0
919	insn-emit_c__gen_blt	3	2	1	1	1	0
920	insn-emit_c__gen_bltu	3	2	1	1	1	0
921	insn-emit_c__gen_bge	3	2	1	1	1	0
922	insn-emit_c__gen_bgeu	3	2	1	1	1	0
923	insn-emit_c__gen_ble	3	2	1	1	1	0
924	insn-emit_c__gen_bleu	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
925	insn-emit_c_gen_movqi	5	5	2	2	2	0
926	insn-emit_c_gen_movhi	5	5	2	2	2	0
927	insn-emit_c_gen_movsi	5	5	2	2	2	0
928	insn-emit_c_gen_movdi	5	5	2	2	2	0
929	insn-emit_c_gen_movsf	5	5	2	2	2	0
930	insn-emit_c_gen_movdf	5	5	2	2	2	0
931	insn-emit_c_gen_split_87	3	2	1	1	1	0
932	insn-emit_c_gen_movtf	5	5	2	2	2	0
933	insn-emit_c_gen_zero_extendhisi2	5	5	2	2	2	0
934	insn-emit_c_gen_zero_extendqihi2	3	2	1	1	1	0
935	insn-emit_c_gen_zero_extendqisi2	3	2	1	1	1	0
936	insn-emit_c_gen_extendhisi2	5	5	2	2	2	0
937	insn-emit_c_gen_extendqihi2	9	11	8	8	8	0
938	insn-emit_c_gen_extendqisi2	5	5	2	2	2	0
939	insn-emit_c_gen_extendsfdf2	3	2	1	1	1	0
940	insn-emit_c_gen_extendsftf2	3	2	1	1	1	0
941	insn-emit_c_gen_extenddftf2	3	2	1	1	1	0
942	insn-emit_c_gen_truncdfsf2	3	2	1	1	1	0
943	insn-emit_c_gen_trunctfsf2	3	2	1	1	1	0
944	insn-emit_c_gen_trunctfdf2	3	2	1	1	1	0
945	insn-emit_c_gen_floatsisf2	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
946	insn-emit_c__gen_floatsidf2	3	2	1	1	1	0
947	insn-emit_c__gen_floatsitf2	3	2	1	1	1	0
948	insn-emit_c__gen_fix_truncsfsi2	3	2	1	1	1	0
949	insn-emit_c__gen_fix_truncdfsi2	3	2	1	1	1	0
950	insn-emit_c__gen_fix_trunctfsi2	3	2	1	1	1	0
951	insn-emit_c__gen_adddi3	3	2	1	1	1	0
952	insn-emit_c__gen_addsi3	3	2	1	1	1	0
953	insn-emit_c__gen_subdi3	3	2	1	1	1	0
954	insn-emit_c__gen_subsi3	3	2	1	1	1	0
955	insn-emit_c__gen_mulsi3	3	2	1	1	1	0
956	insn-emit_c__gen_mulsidi3	10	14	6	6	6	0
957	insn-emit_c__gen_const_mulsidi3	3	2	1	1	1	0
958	insn-emit_c__gen_umulsidi3	10	14	6	6	6	0
959	insn-emit_c__gen_const_umulsidi3	3	2	1	1	1	0
960	insn-emit_c__gen_divsi3	3	2	1	1	1	0
961	insn-emit_c__gen_udivsi3	3	2	1	1	1	0
962	insn-emit_c__gen_anddi3	3	2	1	1	1	0
963	insn-emit_c__gen_andsi3	3	2	1	1	1	0
964	insn-emit_c__gen_split_146	3	2	1	1	1	0
965	insn-emit_c__gen_iordi3	3	2	1	1	1	0
966	insn-emit_c__gen_iorsi3	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
967	insn-emit_c_gen_split_152	3	2	1	1	1	0
968	insn-emit_c_gen_xordi3	3	2	1	1	1	0
969	insn-emit_c_gen_xorsi3	3	2	1	1	1	0
970	insn-emit_c_gen_split_158	3	2	1	1	1	0
971	insn-emit_c_gen_split_159	3	2	1	1	1	0
972	insn-emit_c_gen_negdi2	3	2	1	1	1	0
973	insn-emit_c_gen_negsi2	3	2	1	1	1	0
974	insn-emit_c_gen_one_cmpldi2	3	2	1	1	1	0
975	insn-emit_c_gen_one_cmplsi2	3	2	1	1	1	0
976	insn-emit_c_gen_addtf3	3	2	1	1	1	0
977	insn-emit_c_gen_adddf3	3	2	1	1	1	0
978	insn-emit_c_gen_addsf3	3	2	1	1	1	0
979	insn-emit_c_gen_subtf3	3	2	1	1	1	0
980	insn-emit_c_gen_subdf3	3	2	1	1	1	0
981	insn-emit_c_gen_subsf3	3	2	1	1	1	0
982	insn-emit_c_gen_multf3	3	2	1	1	1	0
983	insn-emit_c_gen_muldf3	3	2	1	1	1	0
984	insn-emit_c_gen_mulsf3	3	2	1	1	1	0
985	insn-emit_c_gen_divtf3	3	2	1	1	1	0
986	insn-emit_c_gen_divdf3	3	2	1	1	1	0
987	insn-emit_c_gen_divsf3	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
988	insn-emit_c__gen_negtf2	3	2	1	1	1	0
989	insn-emit_c__gen_negdf2	3	2	1	1	1	0
990	insn-emit_c__gen_negsf2	3	2	1	1	1	0
991	insn-emit_c__gen_abstf2	3	2	1	1	1	0
992	insn-emit_c__gen_absdf2	3	2	1	1	1	0
993	insn-emit_c__gen_abssf2	3	2	1	1	1	0
994	insn-emit_c__gen_sqrttf2	3	2	1	1	1	0
995	insn-emit_c__gen_sqrtdf2	3	2	1	1	1	0
996	insn-emit_c__gen_sqrtsf2	3	2	1	1	1	0
997	insn-emit_c__gen_ashlsi3	3	2	1	1	1	0
998	insn-emit_c__gen_ashrsi3	3	2	1	1	1	0
999	insn-emit_c__gen_lshrsi3	3	2	1	1	1	0
1000	insn-emit_c__gen_jump	3	2	1	1	1	0
1001	insn-emit_c__gen_tablejump	6	6	2	2	2	0
1002	insn-emit_c__gen_pic_tablejump	3	2	1	1	1	0
1003	insn-emit_c__gen_call	12	14	4	4	4	0
1004	insn-emit_c__gen_call_value	3	2	1	1	1	0
1005	insn-emit_c__gen_untyped_call	3	2	1	1	1	0
1006	insn-emit_c__gen_untyped_return	6	6	2	2	2	0
1007	insn-emit_c__gen_update_return	3	2	1	1	1	0
1008	insn-emit_c__gen_return	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1009	insn-emit_c_gen_nop	3	2	1	1	1	0
1010	insn-emit_c_gen_indirect_jump	3	2	1	1	1	0
1011	insn-emit_c_gen_nonlocal_goto	5	5	2	2	2	0
1012	insn-emit_c_gen_flush_register_windows	3	2	1	1	1	0
1013	insn-emit_c_gen_goto_handler_and_restore	3	2	1	1	1	0
1014	insn-emit_c_gen_flush	3	2	1	1	1	0
1015	insn-emit_c_gen_ffssi2	3	2	1	1	1	0
1016	insn-emit_c_gen_split_231	3	2	1	1	1	0
1017	insn-emit_c_gen_split_232	3	2	1	1	1	0
1018	insn-emit_c_gen_split_233	3	2	1	1	1	0
1019	insn-emit_c_gen_split_234	3	2	1	1	1	0
1020	insn-emit_c_gen_split_235	3	2	1	1	1	0
1021	insn-emit_c_gen_split_236	3	2	1	1	1	0
1022	insn-emit_c_gen_split_237	3	2	1	1	1	0
1023	insn-emit_c_gen_split_238	3	2	1	1	1	0
1024	insn-emit_c_gen_split_239	3	2	1	1	1	0
1025	insn-emit_c_gen_split_240	3	2	1	1	1	0
1026	insn-emit_c_gen_split_241	3	2	1	1	1	0
1027	insn-emit_c_gen_split_242	3	2	1	1	1	0
1028	insn-emit_c_gen_split_243	3	2	1	1	1	0
1029	insn-emit_c_gen_split_244	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1030	insn-emit_c__add_clobbers	17	24	7	9	9	0
1031	insn-emit_c__init_mov_optab	3	2	1	1	1	0
1032	insn-extract_c_insn_extract	89	144	59	61	61	0
1033	insn-opinit_c_init_all_optabs	81	119	5.49756×10^{11}	5.49756×10^{11}	5.49756×10^{11}	0
1034	insn-output_c_output_44	3	2	1	1	1	0
1035	insn-output_c_output_55	6	7	3	3	3	0
1036	insn-output_c_output_56	6	7	3	3	3	0
1037	insn-output_c_output_58	3	2	1	1	1	0
1038	insn-output_c_output_62	3	2	1	1	1	0
1039	insn-output_c_output_66	3	2	1	1	1	0
1040	insn-output_c_output_68	10	12	3	4	4	0
1041	insn-output_c_output_76	13	20	17	17	17	0
1042	insn-output_c_output_77	5	5	2	2	2	0
1043	insn-output_c_output_78	11	14	5	5	5	0
1044	insn-output_c_output_80	3	2	1	1	1	0
1045	insn-output_c_output_81	3	2	1	1	1	0
1046	insn-output_c_output_83	13	16	5	5	5	0
1047	insn-output_c_output_85	13	20	17	17	17	0
1048	insn-output_c_output_86	3	2	1	1	1	0
1049	insn-output_c_output_88	6	6	2	2	2	0
1050	insn-output_c_output_89	13	16	5	5	5	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1051	insn-output_c_output_91	13	20	17	17	17	0
1052	insn-output_c_output_92	13	20	17	17	17	0
1053	insn-output_c_output_93	6	6	2	2	2	0
1054	insn-output_c_output_97	3	2	1	1	1	0
1055	insn-output_c_output_99	3	2	1	1	1	0
1056	insn-output_c_output_110	3	2	1	1	1	0
1057	insn-output_c_output_123	10	13	5	5	5	0
1058	insn-output_c_output_127	10	13	5	5	5	0
1059	insn-output_c_output_144	10	13	5	5	5	0
1060	insn-output_c_output_150	10	13	5	5	5	0
1061	insn-output_c_output_156	10	13	5	5	5	0
1062	insn-output_c_output_173	10	13	5	5	5	0
1063	insn-output_c_output_191	3	2	1	1	1	0
1064	insn-output_c_output_192	3	2	1	1	1	0
1065	insn-output_c_output_194	3	2	1	1	1	0
1066	insn-output_c_output_195	3	2	1	1	1	0
1067	insn-output_c_output_211	3	2	1	1	1	0
1068	insn-output_c_output_212	3	2	1	1	1	0
1069	insn-output_c_output_213	3	2	1	1	1	0
1070	insn-output_c_output_214	3	2	1	1	1	0
1071	insn-output_c_output_216	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1072	insn-output_c__output_217	3	2	1	1	1	0
1073	insn-output_c__output_219	3	2	1	1	1	0
1074	insn-output_c__output_220	3	2	1	1	1	0
1075	insn-output_c__output_223	3	2	1	1	1	0
1076	insn-output_c__output_256	5	5	1	2	2	0
1077	insn-output_c__output_257	5	5	1	2	2	0
1078	insn-output_c__output_258	3	2	1	1	1	0
1079	insn-output_c__output_260	6	6	2	2	2	0
1080	insn-output_c__output_261	6	6	2	2	2	0
1081	insn-output_c__output_262	6	6	2	2	2	0
1082	insn-output_c__output_263	6	6	2	2	2	0
1083	insn-output_c__output_264	6	6	2	2	2	0
1084	insn-output_c__output_266	3	2	1	1	1	0
1085	insn-output_c__output_267	3	2	1	1	1	0
1086	insn-peep_c__peephole	356	688	1.36116×10^{25}	1.36116×10^{25}	1.36116×10^{25}	0
1087	insn-recog_c__recog_1	164	302	3_182	3_182	3_182	0
1088	insn-recog_c__recog_2	106	191	100	100	100	0
1089	insn-recog_c__recog_3	558	985	544_329_036	544_329_036	544_329_036	0
1090	insn-recog_c__recog_4	133	247	606	606	606	0
1091	insn-recog_c__recog_5	294	510	91_368_754	91_368_754	91_368_754	0
1092	insn-recog_c__recog	165	284	2_159	2_159	2_159	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1093	insn-recog_c__split_1	107	202	97	97	97	0
1094	insn-recog_c__split_insns	61	97	588	588	588	0
1095	integrate_c__function_cannot_inline_p	38	61	561	569	564	0.9
1096	integrate_c__initialize_for_inline	19	29	60	74	74	0
1097	integrate_c__finish_inline	3	2	1	1	1	0
1098	integrate_c__adjust_copied_decl_tree	7	9	4	5	5	0
1099	integrate_c__save_for_inline_copying	57	90	449_280	466_625	466_599	0
1100	integrate_c__copy_decl_list	11	15	7	9	9	0
1101	integrate_c__copy_decl_tree	7	9	4	5	5	0
1102	integrate_c__copy_decl_rtls	10	15	8	12	12	0
1103	integrate_c__save_for_inline_nocopy	16	23	18	28	28	0
1104	integrate_c__save_constants	25	37	17	32	25	21.9
1105	integrate_c__note_modified_parmregs	9	13	6	6	6	0
1106	integrate_c__copy_for_inline	71	125	4_995	5_009	5_008	0
1107	integrate_c__expand_inline_function	244	418	5.13645×10^{17}	5.13774×10^{17}	5.13774×10^{17}	0
1108	integrate_c__integrate_parm_decls	5	6	2	3	3	0
1109	integrate_c__integrate_decl_tree	15	22	36	39	39	0
1110	integrate_c__copy_rtx_and_substitute	105	171	228	272	260	4.4
1111	integrate_c__try_constants	12	18	7	13	13	0
1112	integrate_c__subst_constants	112	199	37_752	37_826	37_794	0.1
1113	integrate_c__mark_stores	18	26	40	42	42	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1114	integrate_c__restore_constants	25	36	11	17	16	5.9
1115	integrate_c__set_block_origin_self	8	12	5	7	7	0
1116	integrate_c__set_decl_origin_self	9	13	6	7	7	0
1117	integrate_c__set_block_abstract_flags	7	10	4	6	6	0
1118	integrate_c__set_decl_abstract_flags	9	12	5	6	6	0
1119	integrate_c__output_inline_function	30	43	3_073	3_075	3_075	0
1120	jump_c__jump_optimize	426	765	5.01269×10^{30}	6.37979×10^{30}	6.37979×10^{30}	0
1121	jump_c__duplicate_loop_exit_test	78	131	21_596	21_994	21_899	0.4
1122	jump_c__squeeze_notes	12	18	9	17	17	0
1123	jump_c__find_cross_jump	55	100	350_112	350_443	350_442	0
1124	jump_c__do_cross_jump	17	27	36	58	51	12.1
1125	jump_c__get_label_before	14	23	23	24	24	0
1126	jump_c__get_label_after	14	23	23	24	24	0
1127	jump_c__jump_back_p	19	29	23	23	23	0
1128	jump_c__can_reverse_comparison_p	18	28	24	24	24	0
1129	jump_c__reverse_condition	16	25	10	11	11	0
1130	jump_c__swap_condition	14	22	9	10	10	0
1131	jump_c__unsigned_condition	10	14	5	6	6	0
1132	jump_c__signed_condition	10	14	5	6	6	0
1133	jump_c__comparison_dominates_p	17	28	16	16	16	0
1134	jump_c__simplejump_p	8	11	5	5	5	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1135	jump_c__condjump_p	15	24	14	14	14	0
1136	jump_c__sets_cc0_p	3	2	1	2	2	0
1137	jump_c__follow_jumps	22	36	26	60	44	26.7
1138	jump_c__tension_vector_labels	10	14	5	9	9	0
1139	jump_c__mark_jump_label	39	70	154	170	169	0.6
1140	jump_c__delete_jump	6	7	3	3	3	0
1141	jump_c__delete_computation	27	46	35	77	73	5.2
1142	jump_c__delete_insn	68	120	2_724_004	2_724_040	2_724_040	0
1143	jump_c__next_nondeleted_insn	5	6	2	3	3	0
1144	jump_c__delete_for_peephole	11	14	5	10	10	0
1145	jump_c__invert_jump	7	9	3	4	4	0
1146	jump_c__invert_exp	17	27	13	29	20	31
1147	jump_c__redirect_jump	19	29	86	86	86	0
1148	jump_c__delete_from_jump_chain	18	27	23	24	24	0
1149	jump_c__redirect_exp	29	47	157	173	164	5.2
1150	jump_c__redirect_tablejump	9	12	8	8	8	0
1151	jump_c__delete_labelref_insn	14	20	13	17	15	11.8
1152	jump_c__rtx_renumbered_equal_p	65	106	313	399	370	7.3
1153	jump_c__true_regnum	12	16	6	6	6	0
1154	jump_c__mark_modified_reg	15	22	18	32	30	6.2
1155	jump_c__thread_jumps	68	119	4_980	12_522	9_972	20.4

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1156	jump_c__rtx_equal_for_thread_p	57	100	77	117	91	22.2
1157	local-alloc_c__alloc_qty	3	2	1	1	1	0
1158	local-alloc_c__alloc_qty_for_scratch	26	43	445	463	455	1.7
1159	local-alloc_c__local_alloc	44	68	12_864	12_946	12_939	0.1
1160	local-alloc_c__validate_equiv_mem_from_store	8	11	7	7	7	0
1161	local-alloc_c__validate_equiv_mem	19	33	34	72	50	30.6
1162	local-alloc_c__memref_referenced_p	22	36	27	39	32	17.9
1163	local-alloc_c__memref_used_between_p	8	11	4	8	6	25
1164	local-alloc_c__optimize_reg_copy_1	53	88	79	405	405	0
1165	local-alloc_c__optimize_reg_copy_2	25	43	50	73	73	0
1166	local-alloc_c__update_equiv_regs	70	122	14_777_392	14_909_348	14_909_343	0
1167	local-alloc_c__block_alloc	146	253	4.63833×10^{13}	6.18491×10^{13}	6.18467×10^{13}	0
1168	local-alloc_c__qty_compare	3	2	1	1	1	0
1169	local-alloc_c__qty_compare_1	5	5	2	2	2	0
1170	local-alloc_c__combine_regs	63	105	404_049	404_056	404_054	0
1171	local-alloc_c__reg_meets_class_p	6	7	3	3	3	0
1172	local-alloc_c__reg_classes_overlap_p	14	19	4	8	8	0
1173	local-alloc_c__update_qty_class	7	8	4	4	4	0
1174	local-alloc_c__reg_is_set	8	10	5	5	5	0
1175	local-alloc_c__reg_is_born	15	20	16	16	16	0
1176	local-alloc_c__wipe_dead_reg	17	27	32	37	37	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1177	local-alloc_c__find_free_reg	63	101	18_001	18_061	18_024	0.2
1178	local-alloc_c__mark_life	13	18	8	10	10	0
1179	local-alloc_c__post_mark_life	20	30	16	24	22	8.3
1180	local-alloc_c__no_conflict_p	20	33	88	98	93	5.1
1181	local-alloc_c__requires_inout_p	10	14	5	9	7	22.2
1182	local-alloc_c__dump_local_alloc	7	9	3	5	5	0
1183	loop_c__init_loop	3	2	1	1	1	0
1184	loop_c__loop_optimize	43	70	64_516	64_595	64_592	0
1185	loop_c__scan_loop	159	277	91_392	1_357_535_312	1_344_093_389	1
1186	loop_c__record_excess_regs	17	27	14	20	19	5
1187	loop_c__libcall_other_reg	8	10	3	5	5	0
1188	loop_c__reg_in_basic_block_p	14	22	5	18	12	33.3
1189	loop_c__libcall_benefit	11	16	6	11	11	0
1190	loop_c__skip_consec_insns	9	13	4	9	8	11.1
1191	loop_c__ignore_some_movables	12	19	6	25	16	36
1192	loop_c__force_movables	18	28	12	27	25	7.4
1193	loop_c__combine_movables	43	78	555	743	656	11.7
1194	loop_c__regs_match_p	16	27	81	83	83	0
1195	loop_c__rtx_equal_for_loop_p	68	120	7_849	9_746	9_342	4.1
1196	loop_c__add_label_notes	20	33	20	28	27	3.6
1197	loop_c__move_movables	160	264	2.07221×10^{14}	2.25354×10^{14}	2.1672×10^{14}	3.8

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1198	loop_c__count_nonfixed_reads	18	28	18	26	25	3.8
1199	loop_c__prescan_loop	20	30	11	20	20	0
1200	loop_c__find_and_verify_loops	85	148	22_322_674	26_670_805	26_632_842	0.1
1201	loop_c__mark_loop_jump	32	55	38	49	47	4.1
1202	loop_c__labels_in_range_p	13	17	3	7	6	14.3
1203	loop_c__note_addr_stored	18	28	21	23	23	0
1204	loop_c__invariant_p	40	72	73	95	82	13.7
1205	loop_c__consec_sets_invariant_p	31	51	176	392	284	27.6
1206	loop_c__find_single_use_in_loop	21	32	13	22	21	4.5
1207	loop_c__count_loop_regs_set	68	117	221_980	278_524	277_995	0.2
1208	loop_c__loop_reg_used_before_p	10	14	6	14	10	28.6
1209	loop_c__strength_reduce	274	468	2.63631×10^{18}	2.63631×10^{18}	2.63631×10^{18}	0
1210	loop_c__valid_initial_value_p	16	25	11	11	11	0
1211	loop_c__find_mem_givs	17	29	16	22	21	4.5
1212	loop_c__record_biv	12	15	12	12	12	0
1213	loop_c__record_giv	65	106	34_221	38_298	38_248	0.1
1214	loop_c__check_final_value	49	81	726	994	946	4.8
1215	loop_c__update_giv_derive	31	53	126	659	415	37
1216	loop_c__basic_induction_var	52	92	255	267	257	3.7
1217	loop_c__general_induction_var	25	35	42	44	44	0
1218	loop_c__simplify_giv_expr	89	151	1_148	1_388	1_388	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1219	loop_c__consec_sets_giv	30	51	179	399	289	27.6
1220	loop_c__express_from	16	23	10	10	10	0
1221	loop_c__combine_givs_p	19	26	23	23	23	0
1222	loop_c__combine_givs	23	39	28	213	108	49.3
1223	loop_c__emit_iv_add_mult	5	5	2	2	2	0
1224	loop_c__product_cheap_p	39	62	1_080	1_092	1_086	0.5
1225	loop_c__check_dbra_loop	77	138	154_368	154_416	154_392	0
1226	loop_c__maybe_eliminate_biv	19	28	37	49	43	12.2
1227	loop_c__maybe_eliminate_biv_1	89	165	53_231	53_661	53_443	0.4
1228	loop_c__last_use_this_basic_block	13	19	7	9	8	11.1
1229	loop_c__record_initial	8	11	5	5	5	0
1230	loop_c__update_reg_last_use	15	24	21	27	26	3.7
1231	loop_c__get_condition	80	136	3_946_323	3_947_261	3_946_792	0
1232	loop_c__get_condition_for_loop	8	10	4	4	4	0
1233	optabs_c__add_equal_note	23	38	100	102	101	1
1234	optabs_c__widen_operand	10	13	5	5	5	0
1235	optabs_c__expand_binop	382	624	2.35386×10^{26}	2.38206×10^{26}	2.38206×10^{26}	0
1236	optabs_c__sign_expand_binop	19	29	34	34	34	0
1237	optabs_c__expand_twoval_binop	56	89	1_511_424	1_806_340	1_806_338	0
1238	optabs_c__expand_unop	81	130	1_092_290	1_116_395	1_116_315	0
1239	optabs_c__expand_complex_abs	48	73	7_488	7_510	7_494	0.2

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1240	optabs_c__emit_unop_insn	16	23	96	96	96	0
1241	optabs_c__emit_no_conflict_block	43	67	4_682	4_755	4_723	0.7
1242	optabs_c__emit_libcall_block	24	38	64	80	80	0
1243	optabs_c__emit_clr_insn	3	2	1	1	1	0
1244	optabs_c__emit_0_to_1_insn	3	2	1	1	1	0
1245	optabs_c__emit_cmp_insn	58	93	149_760	175_108	175_105	0
1246	optabs_c__can_compare_p	8	9	2	3	3	0
1247	optabs_c__emit_float_lib_cmp	46	77	30	34	33	2.9
1248	optabs_c__emit_indirect_jump	5	5	2	2	2	0
1249	optabs_c__gen_add2_insn	7	9	1	4	4	0
1250	optabs_c__have_add2_insn	3	2	1	1	1	0
1251	optabs_c__gen_sub2_insn	7	9	1	4	4	0
1252	optabs_c__have_sub2_insn	3	2	1	1	1	0
1253	optabs_c__gen_move_insn	27	39	64	77	75	2.6
1254	optabs_c__can_extend_p	3	2	1	1	1	0
1255	optabs_c__gen_extend_insn	3	2	1	1	1	0
1256	optabs_c__can_fix_p	8	9	3	3	3	0
1257	optabs_c__can_float_p	3	2	1	1	1	0
1258	optabs_c__expand_float	77	119	1_952	2_125	2_062	3
1259	optabs_c__ftruncify	3	2	1	1	1	0
1260	optabs_c__expand_fix	101	153	3_854	4_047	3_988	1.5

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1261	optabs_c__init_optab	7	8	2	3	3	0
1262	optabs_c__init_libfuncs	10	15	5	13	11	15.4
1263	optabs_c__init_integral_libfuncs	3	2	1	1	1	0
1264	optabs_c__init_floating_libfuncs	3	2	1	1	1	0
1265	optabs_c__init_complex_libfuncs	3	2	1	1	1	0
1266	optabs_c__init_optabs	15	24	16	22	22	0
1267	print-rtl_c__print_rtx	67	102	4_514	4_801	4_753	1
1268	print-rtl_c__debug_rtx	3	2	1	1	1	0
1269	print-rtl_c__debug_rtx_list	15	24	36	38	38	0
1270	print-rtl_c__debug_rtx_find	10	14	6	9	7	22.2
1271	print-rtl_c__print_rtl	11	14	4	5	5	0
1272	print-tree_c__debug_tree	3	2	1	1	1	0
1273	print-tree_c__print_node_brief	32	47	321	322	322	0
1274	print-tree_c__indent_to	7	9	4	5	5	0
1275	print-tree_c__print_node	225	344	3.38514×10^{15}	3.38514×10^{15}	3.38514×10^{15}	0
1276	real_c__endian	13	17	4	6	6	0
1277	real_c__earith	20	30	12	12	12	0
1278	real_c__etrunci	6	6	2	2	2	0
1279	real_c__etruncui	6	6	2	2	2	0
1280	real_c__ereal_atof	15	19	6	6	6	0
1281	real_c__ereal_negate	6	6	2	2	2	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1282	real_c__efixi	6	6	2	2	2	0
1283	real_c__efixui	6	6	2	2	2	0
1284	real_c__ereal_from_int	9	11	6	6	6	0
1285	real_c__ereal_from_uint	3	2	1	1	1	0
1286	real_c__ereal_to_int	11	14	7	7	7	0
1287	real_c__ereal_ldexp	6	6	2	2	2	0
1288	real_c__target_isinf	3	2	1	1	1	0
1289	real_c__target_isnan	3	2	1	1	1	0
1290	real_c__target_negative	3	2	1	1	1	0
1291	real_c__real_value_truncate	14	20	6	8	8	0
1292	real_c__debug_real	3	2	1	1	1	0
1293	real_c__etartdouble	3	2	1	1	1	0
1294	real_c__etarldouble	3	2	1	1	1	0
1295	real_c__etardouble	3	2	1	1	1	0
1296	real_c__etarsingle	3	2	1	1	1	0
1297	real_c__ereal_to_decimal	3	2	1	1	1	0
1298	real_c__ereal_cmp	3	2	1	1	1	0
1299	real_c__ereal_isneg	3	2	1	1	1	0
1300	real_c__einit	3	2	1	1	1	0
1301	real_c__eclear	5	5	1	2	2	0
1302	real_c__emov	5	5	1	2	2	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1303	real_c__eabs	3	2	1	1	1	0
1304	real_c__eneg	5	5	2	2	2	0
1305	real_c__eisneg	5	5	2	2	2	0
1306	real_c__eisinf	5	5	2	2	2	0
1307	real_c__eisnan	10	12	3	4	4	0
1308	real_c__einfin	5	5	1	2	2	0
1309	real_c__enan	5	5	1	2	2	0
1310	real_c__emovi	17	22	6	9	9	0
1311	real_c__emovo	12	15	6	7	7	0
1312	real_c__ecleaz	5	5	1	2	2	0
1313	real_c__ecleazs	5	5	1	2	2	0
1314	real_c__emovz	5	5	1	2	2	0
1315	real_c__einan	3	2	1	1	1	0
1316	real_c__eiisnan	9	11	3	5	4	20
1317	real_c__eiinfin	3	2	1	1	1	0
1318	real_c__eiisinf	5	5	2	2	2	0
1319	real_c__ecmpm	10	12	3	4	4	0
1320	real_c__eshdn1	9	11	4	8	8	0
1321	real_c__eshup1	9	11	4	8	8	0
1322	real_c__eshdn8	5	5	1	2	2	0
1323	real_c__eshup8	5	5	1	2	2	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1324	real_c__eshup6	5	5	1	2	2	0
1325	real_c__eshdn6	5	5	1	2	2	0
1326	real_c__eaddm	5	5	1	2	2	0
1327	real_c__esubm	5	5	1	2	2	0
1328	real_c__m16m	10	12	2	5	5	0
1329	real_c__edivm	18	25	12	21	21	0
1330	real_c__emulm	12	15	2	6	6	0
1331	real_c__emdnorm	61	95	184_347	184_350	184_350	0
1332	real_c__esub	13	17	6	6	6	0
1333	real_c__eadd	13	17	6	6	6	0
1334	real_c__eadd1	34	48	55	56	56	0
1335	real_c__ediv	38	54	69	73	71	2.7
1336	real_c__emul	35	50	65	70	67	4.3
1337	real_c__e53toe	20	29	24	24	24	0
1338	real_c__e64toe	16	22	4	8	8	0
1339	real_c__e113toe	18	24	10	12	12	0
1340	real_c__e24toe	18	25	20	20	20	0
1341	real_c__etoe113	8	9	3	3	3	0
1342	real_c__toe113	12	15	5	6	6	0
1343	real_c__etoe64	8	9	3	3	3	0
1344	real_c__toe64	10	12	3	4	4	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1345	real_c__etoe53	8	9	3	3	3	0
1346	real_c__toe53	13	16	7	7	7	0
1347	real_c__etoe24	8	9	3	3	3	0
1348	real_c__toe24	13	16	7	7	7	0
1349	real_c__ecmp	25	35	13	15	15	0
1350	real_c__eround	3	2	1	1	1	0
1351	real_c__ltoe	9	10	4	4	4	0
1352	real_c__ultoe	6	6	2	2	2	0
1353	real_c__eifrac	22	30	17	19	18	5.3
1354	real_c__euifrac	20	26	17	18	18	0
1355	real_c__eshift	21	35	33	39	39	0
1356	real_c__enormlz	24	37	20	29	24	17.2
1357	real_c__e24toasc	3	2	1	1	1	0
1358	real_c__e53toasc	3	2	1	1	1	0
1359	real_c__e64toasc	3	2	1	1	1	0
1360	real_c__e113toasc	3	2	1	1	1	0
1361	real_c__etoasc	89	142	296_762	297_288	297_248	0
1362	real_c__asctoe24	3	2	1	1	1	0
1363	real_c__asctoe53	3	2	1	1	1	0
1364	real_c__asctoe64	3	2	1	1	1	0
1365	real_c__asctoe113	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1366	real_c__asctoe	3	2	1	1	1	0
1367	real_c__asctoeg	98	160	48_328	48_794	48_785	0
1368	real_c__efloor	14	20	10	12	12	0
1369	real_c__efrexp	5	5	2	2	2	0
1370	real_c__eldexp	3	2	1	1	1	0
1371	real_c__eremain	14	19	7	7	7	0
1372	real_c__eiremain	8	10	3	5	5	0
1373	real_c__mtherr	5	5	2	2	2	0
1374	real_c__make_nan	13	19	8	11	11	0
1375	real_c__ereal_from_float	3	2	1	1	1	0
1376	real_c__ereal_from_double	3	2	1	1	1	0
1377	real_c__uditoe	8	9	2	3	3	0
1378	real_c__ditoe	15	20	12	15	15	0
1379	real_c__etoudi	22	30	7	10	10	0
1380	real_c__etodi	24	34	12	17	17	0
1381	real_c__esqrt	33	49	164	185	180	2.7
1382	recog_c__init_recog_no_volatile	3	2	1	1	1	0
1383	recog_c__init_recog	3	2	1	1	1	0
1384	recog_c__recog_memoized	5	5	2	2	2	0
1385	recog_c__check_asm_operands	11	15	5	7	6	14.3
1386	recog_c__validate_change	15	22	14	16	16	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1387	recog_c_apply_change_group	28	46	86	117	116	0.9
1388	recog_c_num_validated_changes	3	2	1	1	1	0
1389	recog_c_cancel_changes	9	12	4	7	7	0
1390	recog_c_validate_replace_rtx_1	62	110	21_206	21_212	21_211	0
1391	recog_c_validate_replace_rtx	3	2	1	1	1	0
1392	recog_c_find_single_use_1	45	74	273	357	305	14.6
1393	recog_c_find_single_use	20	33	17	31	23	25.8
1394	recog_c_general_operand	115	212	4_341_882	4_341_882	4_341_882	0
1395	recog_c_address_operand	3	2	1	1	1	0
1396	recog_c_register_operand	15	23	33	33	33	0
1397	recog_c_scratch_operand	8	11	5	5	5	0
1398	recog_c_immediate_operand	22	37	445	445	445	0
1399	recog_c_const_int_operand	3	2	1	1	1	0
1400	recog_c_const_double_operand	14	21	21	21	21	0
1401	recog_c_nonimmediate_operand	10	15	7	7	7	0
1402	recog_c_nonmemory_operand	33	56	338	338	338	0
1403	recog_c_push_operand	8	10	4	4	4	0
1404	recog_c_memory_address_p	80	149	69_880	69_880	69_880	0
1405	recog_c_memory_operand	14	19	16	16	16	0
1406	recog_c_indirect_operand	23	36	56	56	56	0
1407	recog_c_comparison_operator	7	9	5	5	5	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1408	recog_c__asm_noperands	31	50	56	63	59	6.3
1409	recog_c__decode_asm_operands	73	114	958	1_038	1_038	0
1410	recog_c__find_constant_term_loc	26	46	114	114	114	0
1411	recog_c__offsettable_memref_p	6	7	3	3	3	0
1412	recog_c__offsettable_nonstrict_memref_p	6	7	3	3	3	0
1413	recog_c__offsettable_address_p	19	29	34	34	34	0
1414	recog_c__mode_dependent_address_p	3	2	1	1	1	0
1415	recog_c__mode_independent_operand	5	5	2	2	2	0
1416	recog_c__adj_offsettable_operand	15	23	11	12	12	0
1417	recog_c__constrain_operands	155	284	1_889_450	2_421_519	2_143_185	11.5
1418	recog_c__reg_fits_class_p	13	18	8	9	9	0
1419	regclass_c__init_reg_sets	138	221	1.17846×10^{10}	1.17892×10^{10}	1.17876×10^{10}	0
1420	regclass_c__init_reg_sets_1	31	45	192	224	224	0
1421	regclass_c__fix_register	10	12	4	4	4	0
1422	regclass_c__reg_preferred_class	6	6	2	2	2	0
1423	regclass_c__reg_alternate_class	6	6	2	2	2	0
1424	regclass_c__regclass_init	3	2	1	1	1	0
1425	regclass_c__regclass	112	192	828_071_833	1_659_775_798	1_657_959_698	0.1
1426	regclass_c__record_reg_classes	147	261	13_810_178	21_866_597	21_290_816	2.6
1427	regclass_c__copy_cost	40	66	6_238	6_238	6_238	0
1428	regclass_c__record_address_regs	24	39	33	36	36	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1429	regclass_c__reg_scan	28	43	3_840	3_859	3_859	0
1430	regclass_c__reg_scan_mark_refs	58	107	24_539	24_548	24_547	0
1431	regclass_c__reg_class_subset_p	10	14	6	8	7	12.5
1432	regclass_c__reg_classes_intersect_p	14	20	5	8	8	0
1433	reload1_c__init_reload	16	21	16	18	18	0
1434	reload1_c__reload	503	849	5.65617×10^{45}	5.65617×10^{45}	5.65617×10^{45}	0
1435	reload1_c__possible_group_p	31	52	260	365	365	0
1436	reload1_c__count_possible_groups	31	52	186	430	398	7.4
1437	reload1_c__modes_equiv_for_class_p	9	12	4	7	7	0
1438	reload1_c__spill_failure	5	5	1	2	2	0
1439	reload1_c__new_spill_reg	19	30	144	147	147	0
1440	reload1_c__delete_dead_insn	9	13	6	6	6	0
1441	reload1_c__alter_reg	33	50	418	418	418	0
1442	reload1_c__mark_home_live	8	11	5	6	6	0
1443	reload1_c__mark_scratch_live	7	9	4	5	5	0
1444	reload1_c__set_label_offsets	62	101	76	93	92	1.1
1445	reload1_c__eliminate_regs	175	306	1_434	1_507	1_477	2
1446	reload1_c__eliminate_regs_in_insn	60	102	1_641_084	1_641_130	1_641_124	0
1447	reload1_c__mark_not_eliminable	15	23	16	23	23	0
1448	reload1_c__spill_hard_reg	45	73	1_420	1_507	1_501	0.4
1449	reload1_c__scan_paradoxical_subregs	15	25	14	20	19	5

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1450	reload1_c__hard_reg_use_compare	5	5	2	2	2	0
1451	reload1_c__order_regs_for_reload	32	47	72	89	88	1.1
1452	reload1_c__reload_as_needed	76	127	1_384_144	1_560_724	1_474_135	5.5
1453	reload1_c__forget_old_reloads_1	21	34	90	98	98	0
1454	reload1_c__reload_reg_class_lower	11	15	9	9	9	0
1455	reload1_c__mark_reload_reg_in_use	17	27	20	29	29	0
1456	reload1_c__clear_reload_reg_in_use	17	27	20	29	29	0
1457	reload1_c__reload_reg_free_p	59	100	85	105	95	9.5
1458	reload1_c__reload_reg_free_before_p	52	89	48	65	56	13.8
1459	reload1_c__reload_reg_reaches_end_p	52	89	68	90	76	15.6
1460	reload1_c__allocate_reload_reg	53	88	81_144	106_858	106_140	0.7
1461	reload1_c__choose_reload_regs	228	389	6.78727×10^{16}	8.95074×10^{16}	8.95074×10^{16}	0
1462	reload1_c__emit_reload_insns	314	543	8.28762×10^{28}	8.28763×10^{28}	8.28763×10^{28}	0
1463	reload1_c__gen_input_reload	55	92	9_566	9_566	9_566	0
1464	reload1_c__delete_output_reload	36	64	968	992	982	1
1465	reload1_c__inc_for_reload	19	26	72	72	72	0
1466	reload1_c__constraint_accepts_reg_p	20	27	2	15	15	0
1467	reload1_c__count_occurrences	19	31	22	29	28	3.4
1468	reload_c__find_secondary_reload	52	85	95_040	164_160	164_160	0
1469	reload_c__get_secondary_mem	18	24	30	30	30	0
1470	reload_c__clear_secondary_mem	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1471	reload_c__push_reload	474	861	1.20584×10^{47}	1.20584×10^{47}	1.20584×10^{47}	0
1472	reload_c__push_replacement	5	5	2	2	2	0
1473	reload_c__transfer_replacements	8	10	3	5	5	0
1474	reload_c__combine_reloads	80	141	394_831	400_175	397_511	0.7
1475	reload_c__find_dummy_reload	60	101	413_954	413_958	413_958	0
1476	reload_c__earlyclobber_operand_p	9	11	3	5	4	20
1477	reload_c__hard_reg_set_here_p	18	29	28	32	30	6.2
1478	reload_c__strict_memory_address_p	94	177	776_670	776_670	776_670	0
1479	reload_c__operands_match_p	54	86	503	559	544	2.7
1480	reload_c__n_occurrences	7	9	3	5	5	0
1481	reload_c__decompose	60	94	1_454	2_895	2_895	0
1482	reload_c__immune_p	42	74	877	878	878	0
1483	reload_c__safe_from_earlyclobber	3	2	1	1	1	0
1484	reload_c__find_reloads	629	1092	6.93036×10^{37}	6.93036×10^{37}	6.93036×10^{37}	0
1485	reload_c__alternative_allows_memconst	16	26	16	20	19	5
1486	reload_c__find_reloads_toplev	47	79	4_955	4_977	4_977	0
1487	reload_c__make_memloc	5	5	2	2	2	0
1488	reload_c__find_reloads_address	85	146	61_608	61_608	61_608	0
1489	reload_c__subst_reg_equivs	14	23	16	18	18	0
1490	reload_c__form_sum	49	81	7_908	7_908	7_908	0
1491	reload_c__subst_indexed_address	30	46	701	701	701	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1492	reload_c__find_reloads_address_1	100	168	52_487	52_490	52_490	0
1493	reload_c__find_reloads_address_part	35	63	9_839	9_839	9_839	0
1494	reload_c__subst_reloads	17	25	14	41	28	31.7
1495	reload_c__copy_replacements	16	23	6	18	15	16.7
1496	reload_c__find_replacement	19	28	10	16	13	18.8
1497	reload_c__refers_to_regno_for_reload_p	63	105	313	454	370	18.5
1498	reload_c__reg_overlap_mentioned_for_reload_p	30	46	17	19	19	0
1499	reload_c__refers_to_mem_for_reload_p	16	24	9	13	11	15.4
1500	reload_c__find_equiv_reg	192	350	4.1351×10^{13}	4.1351×10^{13}	4.1351×10^{13}	0
1501	reload_c__find_inc_amount	19	32	64	80	71	11.2
1502	reload_c__regno_clobbered_p	15	22	13	19	16	15.8
1503	reorg_c__mark_referenced_resources	69	119	323	356	347	2.5
1504	reorg_c__mark_set_resources	63	111	248	947	486	48.7
1505	reorg_c__stop_search_p	15	22	8	9	9	0
1506	reorg_c__resource_conflicts_p	14	21	19	21	20	4.8
1507	reorg_c__insn_references_resource_p	5	5	1	2	2	0
1508	reorg_c__insn_sets_resource_p	5	5	1	2	2	0
1509	reorg_c__find_end_label	24	39	103	106	106	0
1510	reorg_c__emit_delay_sequence	30	46	1_536	3_086	3_077	0.3
1511	reorg_c__add_to_delay_list	12	16	5	6	6	0
1512	reorg_c__delete_from_delay_slot	19	29	128	131	131	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1513	reorg_c__delete_scheduled_jump	3	2	1	1	1	0
1514	reorg_c__note_delay_statistics	5	5	2	2	2	0
1515	reorg_c__optimize_skip	27	44	54	54	54	0
1516	reorg_c__get_jump_flags	26	38	42	56	56	0
1517	reorg_c__rare_destination	20	30	25	43	34	20.9
1518	reorg_c__mostly_true_jump	36	67	2_771	2_782	2_773	0.3
1519	reorg_c__get_branch_condition	26	43	167	167	167	0
1520	reorg_c__condition_dominates_p	14	21	9	9	9	0
1521	reorg_c__redirect_with_delay_slots_safe_p	10	14	7	10	10	0
1522	reorg_c__steal_delay_list_from_target	25	43	85	122	98	19.7
1523	reorg_c__steal_delay_list_from_fallthrough	20	33	29	32	32	0
1524	reorg_c__try_merge_delay_insns	54	92	456	585	560	4.3
1525	reorg_c__redundant_insn_p	67	118	5_137	5_267	5_186	1.5
1526	reorg_c__own_thread_p	21	36	39	51	44	13.7
1527	reorg_c__find_basic_block	19	30	18	29	22	24.1
1528	reorg_c__update_block	6	7	3	3	3	0
1529	reorg_c__reorg_redirect_jump	5	5	2	2	2	0
1530	reorg_c__update_reg_dead_notes	11	17	6	19	15	21.1
1531	reorg_c__update_live_status	19	28	34	36	36	0
1532	reorg_c__next_insn_no_annul	12	19	17	18	18	0
1533	reorg_c__mark_target_live_regs	169	283	3.99814×10^{11}	3.99814×10^{11}	3.99814×10^{11}	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1534	reorg_c__fill_simple_delay_slots	146	249	2_834_444_091	1.58716×10^{11}	1.57466×10^{11}	0.8
1535	reorg_c__fill_slots_from_thread	128	211	55_945_731	59_924_737	59_922_170	0
1536	reorg_c__fill_eager_delay_slots	28	42	49	146	98	32.9
1537	reorg_c__relax_delay_slots	110	186	172_442_588	344_885_177	344_885_176	0
1538	reorg_c__make_return_insns	40	65	1_680	1_765	1_755	0.6
1539	reorg_c__dbr_schedule	90	146	2_090_206_081	2_090_327_152	2_090_327_098	0
1540	rtlanal_c__rtx_unstable_p	18	28	11	15	13	13.3
1541	rtlanal_c__rtx_varies_p	17	28	15	19	17	10.5
1542	rtlanal_c__rtx_addr_can_trap_p	24	35	13	13	13	0
1543	rtlanal_c__rtx_addr_varies_p	14	21	8	12	10	16.7
1544	rtlanal_c__get_integer_term	12	16	14	14	14	0
1545	rtlanal_c__get_related_value	10	14	8	8	8	0
1546	rtlanal_c__reg_mentioned_p	27	46	40	52	45	13.5
1547	rtlanal_c__no_labels_between_p	8	10	3	5	4	20
1548	rtlanal_c__reg_used_between_p	11	15	5	9	7	22.2
1549	rtlanal_c__reg_referenced_p	29	50	30	34	32	5.9
1550	rtlanal_c__reg_referenced_between_p	11	15	5	9	7	22.2
1551	rtlanal_c__reg_set_between_p	11	15	5	9	7	22.2
1552	rtlanal_c__reg_set_p_1	6	7	3	3	3	0
1553	rtlanal_c__reg_set_p	11	15	7	7	7	0
1554	rtlanal_c__modified_between_p	17	30	34	50	41	18

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1555	rtlanal_c__modified_in_p	17	30	34	50	41	18
1556	rtlanal_c__single_set	15	23	10	18	14	22.2
1557	rtlanal_c__find_last_value	19	31	34	52	43	17.3
1558	rtlanal_c__refers_to_regno_p	62	102	155	218	192	11.9
1559	rtlanal_c__reg_overlap_mentioned_p	33	51	20	25	23	8
1560	rtlanal_c__reg_set_last_1	17	25	22	22	22	0
1561	rtlanal_c__reg_set_last	29	49	111	119	115	3.4
1562	rtlanal_c__rtx_equal_p	35	62	25	57	34	40.4
1563	rtlanal_c__note_stores	28	50	15	241	196	18.7
1564	rtlanal_c__dead_or_set_p	16	22	10	13	12	7.7
1565	rtlanal_c__dead_or_set_regno_p	54	85	2_460	2_738	2_595	5.2
1566	rtlanal_c__find_reg_note	10	14	5	9	7	22.2
1567	rtlanal_c__find_regno_note	16	23	10	22	16	27.3
1568	rtlanal_c__remove_note	10	13	2	6	5	16.7
1569	rtlanal_c__volatile_insn_p	16	28	33	56	40	28.6
1570	rtlanal_c__volatile_refs_p	16	28	33	56	40	28.6
1571	rtlanal_c__side_effects_p	17	30	34	57	41	28.1
1572	rtlanal_c__may_trap_p	26	47	106	123	111	9.8
1573	rtlanal_c__inequality_comparisons_p	16	26	16	28	21	25
1574	rtlanal_c__replace_rtx	13	20	7	13	12	7.7
1575	rtlanal_c__replace_regs	32	54	60	68	67	1.5

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1576	rtl_c__rtvec_alloc	11	15	16	17	17	0
1577	rtl_c__rtx_alloc	7	8	2	3	3	0
1578	rtl_c__rtx_free	7	8	3	3	3	0
1579	rtl_c__copy_rtx	24	40	42	69	60	13
1580	rtl_c__copy_most_rtx	23	37	20	43	33	23.3
1581	rtl_c__dump_and_abort	8	10	4	10	9	10
1582	rtl_c__read_skip_spaces	21	36	6	59	40	32.2
1583	rtl_c__read_name	19	30	22	23	23	0
1584	rtl_c__read_rtx	76	121	472_840	477_679	475_668	0.4
1585	rtl_c__init_rtl	17	24	6	12	12	0
1586	sched_c__canon_rtx	17	24	25	25	25	0
1587	sched_c__init_alias_analysis	21	34	144	157	157	0
1588	sched_c__rtx_equal_for_memref_p	31	55	40	74	51	31.1
1589	sched_c__find_symbolic_term	12	18	7	11	9	18.2
1590	sched_c__memrefs_conflict_p	116	190	59_994	59_994	59_994	0
1591	sched_c__read_dependence	6	7	3	3	3	0
1592	sched_c__true_dependence	21	35	131	131	131	0
1593	sched_c__anti_dependence	20	33	66	66	66	0
1594	sched_c__output_dependence	17	29	65	65	65	0
1595	sched_c__add_dependence	20	33	39	43	41	4.7
1596	sched_c__remove_dependence	12	16	4	14	11	21.4

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1597	sched_c__clear_units	3	2	1	1	1	0
1598	sched_c__potential_hazard	17	25	13	15	15	0
1599	sched_c__priority	46	68	1_923	1_955	1_955	0
1600	sched_c__free_pending_lists	19	30	81	85	85	0
1601	sched_c__add_insn_mem_dependence	9	10	4	4	4	0
1602	sched_c__flush_pending_lists	9	13	8	10	10	0
1603	sched_c__sched_analyze_1	66	110	4_089	4_145	4_137	0.2
1604	sched_c__sched_analyze_2	68	118	328	383	369	3.7
1605	sched_c__sched_analyze_insn	23	36	49	56	55	1.8
1606	sched_c__sched_analyze	23	34	38	100	87	13
1607	sched_c__sched_note_set	50	79	5_209	5_244	5_244	0
1608	sched_c__rank_for_schedule	45	67	13_124	13_132	13_126	0
1609	sched_c__schedule_insn	48	70	8_864	8_957	8_957	0
1610	sched_c__schedule_select	70	113	1_090_836	1_093_391	1_092_880	0
1611	sched_c__create_reg_dead_note	25	35	72	77	77	0
1612	sched_c__attach_deaths	67	110	94_271	94_325	94_308	0
1613	sched_c__attach_deaths_insn	17	23	7	10	10	0
1614	sched_c__unlink_notes	17	25	42	62	62	0
1615	sched_c__new_sometimes_live	5	5	1	2	2	0
1616	sched_c__finish_sometimes_live	5	6	2	3	3	0
1617	sched_c__schedule_block	322	557	2.70192×10^{34}	2.70379×10^{34}	2.70379×10^{34}	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1618	sched_c__regno_use_in	16	25	15	27	20	25.9
1619	sched_c__split_hard_reg_notes	17	24	10	19	17	10.5
1620	sched_c__new_insn_dead_notes	45	80	1_265	2_350	2_299	2.2
1621	sched_c__update_n_sets	17	27	24	28	28	0
1622	sched_c__update_flow_info	141	247	145_504_128	145_864_377	145_864_192	0
1623	sched_c__schedule_insns	124	201	7.40711×10^{12}	7.40711×10^{12}	7.40711×10^{12}	0
1624	sched_c__prepare_unit	9	12	4	6	6	0
1625	sched_c__schedule_unit	9	12	4	6	6	0
1626	sched_c__actual_hazard	11	15	5	7	7	0
1627	sched_c__adjust_priority	34	53	166	170	169	0.6
1628	sched_c__birthing_insn_p	18	27	17	19	18	5.3
1629	sched_c__actual_hazard_this_instance	13	17	9	9	9	0
1630	stmt_c__init_stmt	3	2	1	1	1	0
1631	stmt_c__init_stmt_for_function	3	2	1	1	1	0
1632	stmt_c__save_stmt_status	3	2	1	1	1	0
1633	stmt_c__restore_stmt_status	3	2	1	1	1	0
1634	stmt_c__emit_nop	8	11	5	5	5	0
1635	stmt_c__label_rtx	8	9	2	3	3	0
1636	stmt_c__emit_jump	3	2	1	1	1	0
1637	stmt_c__expand_computed_goto	6	6	2	2	2	0
1638	stmt_c__expand_label	13	17	8	8	8	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1639	stmt_c__declare_nonlocal_label	5	5	2	2	2	0
1640	stmt_c__expand_goto	10	12	4	4	4	0
1641	stmt_c__expand_goto_internal	24	36	40	45	45	0
1642	stmt_c__bc_expand_goto_internal	21	32	97	105	101	3.8
1643	stmt_c__expand_fixup	40	64	4_392	4_398	4_396	0
1644	stmt_c__bc_expand_fixup	3	2	1	1	1	0
1645	stmt_c__fixup_gotos	45	78	10_155	11_100	11_094	0.1
1646	stmt_c__bc_fixup_gotos	10	14	7	13	13	0
1647	stmt_c__expand_asm	8	9	3	3	3	0
1648	stmt_c__expand_asm_operands	70	115	219_689	219_860	219_732	0.1
1649	stmt_c__expand_expr_stmt	32	48	235	236	236	0
1650	stmt_c__warn_if_unused_value	19	34	30	30	30	0
1651	stmt_c__clear_last_expr	3	2	1	1	1	0
1652	stmt_c__expand_start_stmt_expr	6	6	2	2	2	0
1653	stmt_c__expand_end_stmt_expr	19	28	11	11	11	0
1654	stmt_c__in_try_block	13	18	6	12	9	25
1655	stmt_c__in_except_block	13	18	6	12	9	25
1656	stmt_c__in_exception_handler	7	10	4	5	5	0
1657	stmt_c__expand_raise	8	9	3	3	3	0
1658	stmt_c__expand_start_try	12	15	16	16	16	0
1659	stmt_c__expand_end_try	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1660	stmt_c__expand_start_except	17	26	25	29	27	6.9
1661	stmt_c__expand_escape_except	8	10	3	5	4	20
1662	stmt_c__expand_end_except	37	56	3_456	3_668	3_655	0.4
1663	stmt_c__expand_catch	10	13	5	5	5	0
1664	stmt_c__expand_catch_default	6	6	2	2	2	0
1665	stmt_c__expand_end_catch	7	8	3	3	3	0
1666	stmt_c__expand_start_cond	15	19	32	32	32	0
1667	stmt_c__expand_start_elseif	5	5	2	2	2	0
1668	stmt_c__expand_start_else	8	9	4	4	4	0
1669	stmt_c__expand_end_cond	27	38	960	1_152	1_152	0
1670	stmt_c__bc_expand_start_cond	3	2	1	1	1	0
1671	stmt_c__bc_expand_end_cond	3	2	1	1	1	0
1672	stmt_c__bc_expand_start_else	3	2	1	1	1	0
1673	stmt_c__expand_start_loop	12	15	16	16	16	0
1674	stmt_c__expand_start_loop_continue_elsewhere	3	2	1	1	1	0
1675	stmt_c__expand_loop_continue_here	6	6	2	2	2	0
1676	stmt_c__bc_expand_end_loop	21	29	192	384	384	0
1677	stmt_c__expand_end_loop	64	106	4_793_089	4_793_897	4_793_897	0
1678	stmt_c__expand_continue_loop	6	7	3	3	3	0
1679	stmt_c__expand_exit_loop	6	7	3	3	3	0
1680	stmt_c__expand_exit_loop_if_false	9	11	5	5	5	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1681	stmt_c__preserve_subexpressions_p	12	16	6	6	6	0
1682	stmt_c__expand_exit_something	8	10	3	5	4	20
1683	stmt_c__expand_null_return	10	14	5	6	6	0
1684	stmt_c__expand_value_return	12	18	24	25	25	0
1685	stmt_c__expand_null_return_1	14	20	16	16	16	0
1686	stmt_c__expand_return	47	77	13_754	13_754	13_754	0
1687	stmt_c__drop_through_at_end_p	10	15	12	13	13	0
1688	stmt_c__tail_recursion_args	29	48	105	122	116	4.9
1689	stmt_c__expand_start_bindings	21	29	256	256	256	0
1690	stmt_c__remember_end_note	3	2	1	1	1	0
1691	stmt_c__expand_end_bindings	68	108	29_568_001	29_568_203	29_568_203	0
1692	stmt_c__bc_expand_end_bindings	44	65	147_456	147_842	147_842	0
1693	stmt_c__expand_decl	61	95	3_424	3_436	3_436	0
1694	stmt_c__bc_expand_decl	17	24	8	9	9	0
1695	stmt_c__expand_decl_init	18	27	11	11	11	0
1696	stmt_c__bc_expand_variable_local_init	3	2	1	1	1	0
1697	stmt_c__bc_expand_decl_init	16	23	13	25	25	0
1698	stmt_c__expand_decl_cleanup	7	8	3	3	3	0
1699	stmt_c__expand_anon_union_decl	16	22	9	26	18	30.8
1700	stmt_c__expand_cleanups	11	15	6	11	11	0
1701	stmt_c__move_cleanups_up	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1702	stmt_c__last_cleanup_this_contour	6	6	2	2	2	0
1703	stmt_c__any_pending_cleanups	14	21	16	18	17	5.6
1704	stmt_c__expand_start_case	17	22	48	48	48	0
1705	stmt_c__bc_expand_start_case	3	2	1	1	1	0
1706	stmt_c__expand_start_case_dummy	9	11	8	8	8	0
1707	stmt_c__expand_end_case_dummy	21	29	192	384	384	0
1708	stmt_c__case_index_expr_type	6	6	2	2	2	0
1709	stmt_c__pushcase	39	62	262	306	294	3.9
1710	stmt_c__pushcase_range	36	59	457	525	517	1.5
1711	stmt_c__bc_pushcase	18	28	25	27	26	3.7
1712	stmt_c__check_for_full_enumeration_handling	41	68	3_776	33_768	33_548	0.7
1713	stmt_c__bc_check_for_full_enumeration_handling	22	37	155	227	198	12.8
1714	stmt_c__expand_end_case	122	194	1_874_719_105	1_874_719_477	1_874_719_436	0
1715	stmt_c__bc_expand_end_case	41	61	15_360	15_558	15_558	0
1716	stmt_c__bc_new_uid	3	2	1	1	1	0
1717	stmt_c__do_jump_if_equal	11	14	6	6	6	0
1718	stmt_c__estimate_case_costs	27	42	95	119	108	9.2
1719	stmt_c__group_case_nodes	11	18	25	45	36	20
1720	stmt_c__balance_case_nodes	32	48	56	69	67	2.9
1721	stmt_c__node_has_low_bound	12	17	6	8	7	12.5
1722	stmt_c__node_has_high_bound	12	17	6	8	7	12.5

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1723	stmt_c__node_is_bounded	6	7	3	3	3	0
1724	stmt_c__emit_jump_if_reachable	5	5	2	2	2	0
1725	stmt_c__emit_case_nodes	69	102	1_024	1_024	1_024	0
1726	stmt_c__find_loop_tree_blocks	3	2	1	1	1	0
1727	stmt_c__unroll_block_trees	3	2	1	1	1	0
1728	stor-layout_c__get_pending_sizes	5	6	2	3	3	0
1729	stor-layout_c__variable_size	13	16	5	5	5	0
1730	stor-layout_c__mode_for_size	10	14	7	9	8	11.1
1731	stor-layout_c__smallest_mode_for_size	8	10	1	5	4	20
1732	stor-layout_c__round_up	3	2	1	1	1	0
1733	stor-layout_c__layout_decl	51	81	24_769	24_782	24_782	0
1734	stor-layout_c__layout_record	78	120	10_862_628	11_767_846	11_767_846	0
1735	stor-layout_c__layout_union	39	56	1_408	1_451	1_451	0
1736	stor-layout_c__layout_type	89	151	13_651	13_678	13_667	0.1
1737	stor-layout_c__make_signed_type	14	17	16	16	16	0
1738	stor-layout_c__make_unsigned_type	5	5	2	2	2	0
1739	stor-layout_c__fixup_signed_type	12	14	8	8	8	0
1740	stor-layout_c__fixup_unsigned_type	9	10	4	4	4	0
1741	stor-layout_c__get_best_mode	28	45	291	312	302	3.2
1742	stor-layout_c__save_storage_status	3	2	1	1	1	0
1743	stor-layout_c__restore_storage_status	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1744	stupid_c__stupid_life_analysis	65	104	11_919_360	11_919_479	11_919_474	0
1745	stupid_c__stupid_reg_compare	6	7	3	3	3	0
1746	stupid_c__stupid_find_reg	37	57	168	191	187	2.1
1747	stupid_c__stupid_mark_refs	46	74	217	229	228	0.4
1748	toplev_c__get_run_time	3	2	1	1	1	0
1749	toplev_c__print_time	3	2	1	1	1	0
1750	toplev_c__count_error	14	19	7	7	7	0
1751	toplev_c__pfatal_with_name	3	2	1	2	2	0
1752	toplev_c__fatal_io_error	3	2	1	2	2	0
1753	toplev_c__fatal_insn_not_found	40	57	196_608	393_216	393_216	0
1754	toplev_c__decl_name	3	2	1	1	1	0
1755	toplev_c__announce_function	8	9	3	3	3	0
1756	toplev_c__report_error_function	23	34	156	158	158	0
1757	toplev_c__error	3	2	1	1	1	0
1758	toplev_c__error_with_file_and_line	6	6	2	2	2	0
1759	toplev_c__error_with_decl	6	6	2	2	2	0
1760	toplev_c__error_for_asm	13	19	13	13	13	0
1761	toplev_c__fatal	3	2	1	2	2	0
1762	toplev_c__warning_with_file_and_line	8	9	3	3	3	0
1763	toplev_c__warning	3	2	1	1	1	0
1764	toplev_c__warning_with_decl	8	9	3	3	3	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1765	toplev_c__warning_for_asm	13	19	13	13	13	0
1766	toplev_c__pedwarn	6	6	2	2	2	0
1767	toplev_c__pedwarn_with_decl	6	6	2	2	2	0
1768	toplev_c__pedwarn_with_file_and_line	6	6	2	2	2	0
1769	toplev_c__sorry	6	6	2	2	2	0
1770	toplev_c__really_sorry	6	6	2	2	2	0
1771	toplev_c__fancy_abort	3	2	1	1	1	0
1772	toplev_c__do_abort	3	2	1	2	2	0
1773	toplev_c__botch	3	2	1	2	2	0
1774	toplev_c__xmalloc	5	5	2	2	2	0
1775	toplev_c__xrealloc	5	5	2	2	2	0
1776	toplev_c__exact_log2_wide	9	12	4	5	5	0
1777	toplev_c__floor_log2_wide	5	6	2	3	3	0
1778	toplev_c__set_float_handler	5	5	2	2	2	0
1779	toplev_c__push_float_handler	5	5	2	2	2	0
1780	toplev_c__pop_float_handler	5	5	2	2	2	0
1781	toplev_c__float_signal	5	5	1	3	3	0
1782	toplev_c__pipe_closed	3	2	1	1	1	0
1783	toplev_c__strip_off_ending	28	49	767	767	767	0
1784	toplev_c__output_quoted_string	8	11	4	7	7	0
1785	toplev_c__output_file_directive	6	8	3	4	4	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1786	toplev_c__output_lang_identify	3	2	1	1	1	0
1787	toplev_c__compile_file	164	266	3.83463×10^{22}	3.83463×10^{22}	3.83463×10^{22}	0
1788	toplev_c__rest_of_decl_compilation	27	44	90	90	90	0
1789	toplev_c__rest_of_type_compilation	6	7	3	3	3	0
1790	toplev_c__rest_of_compilation	132	208	1.80609×10^{16}	2.25723×10^{16}	2.25723×10^{16}	0
1791	toplev_c__main	218	358	5.0259×10^{12}	1.00518×10^{13}	1.00518×10^{13}	0
1792	toplev_c__set_target_switch	12	15	6	9	9	0
1793	toplev_c__print_single_switch	5	5	2	2	2	0
1794	toplev_c__print_switch_values	17	24	16	24	24	0
1795	tree_c__init_obstacks	27	38	4_096	4_096	4_096	0
1796	tree_c__gcc_obstack_init	3	2	1	1	1	0
1797	tree_c__save_tree_status	11	14	16	16	16	0
1798	tree_c__restore_tree_status	11	14	9	9	9	0
1799	tree_c__temporary_allocation	3	2	1	1	1	0
1800	tree_c__end_temporary_allocation	3	2	1	1	1	0
1801	tree_c__resume_temporary_allocation	3	2	1	1	1	0
1802	tree_c__saveable_allocation	3	2	1	1	1	0
1803	tree_c__push_obstacks	9	11	8	8	8	0
1804	tree_c__push_obstacks_nochange	9	11	8	8	8	0
1805	tree_c__pop_obstacks	7	8	3	3	3	0
1806	tree_c__allocation_temporary_p	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1807	tree_c__permanent_allocation	19	26	81	81	81	0
1808	tree_c__preserve_data	9	11	8	8	8	0
1809	tree_c__preserve_initializer	21	29	512	512	512	0
1810	tree_c__rtl_in_current_obstack	3	2	1	1	1	0
1811	tree_c__rtl_in_saveable_obstack	3	2	1	1	1	0
1812	tree_c__oballoc	9	11	8	8	8	0
1813	tree_c__obfree	7	8	3	3	3	0
1814	tree_c__permalloc	9	11	8	8	8	0
1815	tree_c__perm_calloc	9	11	8	8	8	0
1816	tree_c__savealloc	9	11	8	8	8	0
1817	tree_c__print_obstack_name	27	39	140	144	144	0
1818	tree_c__debug_obstack	3	2	1	1	1	0
1819	tree_c__object_permanent_p	3	2	1	1	1	0
1820	tree_c__push_momentary	9	11	8	8	8	0
1821	tree_c__clear_momentary	7	8	3	3	3	0
1822	tree_c__pop_momentary	7	8	3	3	3	0
1823	tree_c__pop_momentary_nofree	3	2	1	1	1	0
1824	tree_c__suspend_momentary	3	2	1	1	1	0
1825	tree_c__resume_momentary	5	5	2	2	2	0
1826	tree_c__init_tree_codes	3	2	1	1	1	0
1827	tree_c__make_node	58	94	29_376	29_378	29_378	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1828	tree_c__copy_node	26	43	1_056	1_058	1_058	0
1829	tree_c__copy_list	7	9	3	4	4	0
1830	tree_c__get_identifier	32	52	2_316	2_619	2_610	0.3
1831	tree_c__start_identifier_warnings	3	2	1	1	1	0
1832	tree_c__set_identifier_size	3	2	1	1	1	0
1833	tree_c__build_int_2_wide	3	2	1	1	1	0
1834	tree_c__build_real	3	2	1	1	1	0
1835	tree_c__real_value_from_int_cst	6	6	2	2	2	0
1836	tree_c__build_real_from_int_cst	3	2	1	1	1	0
1837	tree_c__build_string	9	11	8	8	8	0
1838	tree_c__build_complex	3	2	1	1	1	0
1839	tree_c__make_tree_vec	13	18	32	33	33	0
1840	tree_c__integer_zerop	11	17	16	17	17	0
1841	tree_c__integer_onep	11	17	16	17	17	0
1842	tree_c__integer_all_onesp	23	35	44	50	49	2
1843	tree_c__integer_pow2p	17	27	64	65	65	0
1844	tree_c__real_zerop	10	15	12	13	13	0
1845	tree_c__real_onep	10	15	12	13	13	0
1846	tree_c__real_twop	10	15	12	13	13	0
1847	tree_c__really_constant_p	5	6	2	3	3	0
1848	tree_c__value_member	7	9	3	5	4	20

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1849	tree_c__purpose_member	7	9	3	5	4	20
1850	tree_c__binfo_member	7	9	3	5	4	20
1851	tree_c__chain_member	7	9	3	5	4	20
1852	tree_c__list_length	5	6	2	3	3	0
1853	tree_c__chainon	10	14	3	9	7	22.2
1854	tree_c__tree_last	6	8	3	4	4	0
1855	tree_c__nreverse	5	6	2	3	3	0
1856	tree_c__listify	8	10	3	5	5	0
1857	tree_c__build_tree_list	3	2	1	1	1	0
1858	tree_c__build_decl_list	3	2	1	1	1	0
1859	tree_c__tree_cons	13	17	16	17	17	0
1860	tree_c__decl_tree_cons	3	2	1	1	1	0
1861	tree_c__perm_tree_cons	3	2	1	1	1	0
1862	tree_c__temp_tree_cons	3	2	1	1	1	0
1863	tree_c__saveable_tree_cons	3	2	1	1	1	0
1864	tree_c__size_in_bytes	10	12	4	4	4	0
1865	tree_c__int_size_in_bytes	11	14	5	5	5	0
1866	tree_c__array_type_nelts	6	6	2	2	2	0
1867	tree_c__staticp	16	26	12	12	12	0
1868	tree_c__save_expr	11	16	10	11	11	0
1869	tree_c__contains_placeholder_p	35	56	62	63	63	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1870	tree_c__substitute_in_expr	34	54	14	25	24	4
1871	tree_c__substitute_in_type	40	69	286	333	333	0
1872	tree_c__stabilize_reference	13	19	8	8	8	0
1873	tree_c__stabilize_reference_1	22	37	17	17	17	0
1874	tree_c__build	33	52	64	70	70	0
1875	tree_c__build1	18	26	160	161	161	0
1876	tree_c__build_nt	5	6	2	3	3	0
1877	tree_c__build_parse_node	5	6	2	3	3	0
1878	tree_c__build_decl	7	8	3	3	3	0
1879	tree_c__build_block	3	2	1	1	1	0
1880	tree_c__build_type_variant	9	13	5	9	7	22.2
1881	tree_c__change_main_variant	11	16	6	7	7	0
1882	tree_c__build_type_copy	3	2	1	1	1	0
1883	tree_c__type_hash_list	5	6	2	3	3	0
1884	tree_c__type_hash_lookup	20	34	35	87	61	29.9
1885	tree_c__type_hash_add	3	2	1	1	1	0
1886	tree_c__type_hash_canon	16	21	9	9	9	0
1887	tree_c__type_list_equal	14	21	8	15	11	26.7
1888	tree_c__tree_int_cst_equal	12	18	8	8	8	0
1889	tree_c__tree_int_cst_lt	12	17	9	9	9	0
1890	tree_c__simple_cst_list_equal	11	16	5	9	7	22.2

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1891	tree_c__simple_cst_equal	54	94	78	80	80	0
1892	tree_c__build_pointer_type	5	5	2	2	2	0
1893	tree_c__build_index_type	12	16	9	9	9	0
1894	tree_c__build_range_type	11	14	8	8	8	0
1895	tree_c__build_index_2_type	3	2	1	1	1	0
1896	tree_c__index_type_equal	11	17	8	8	8	0
1897	tree_c__build_array_type	9	11	6	6	6	0
1898	tree_c__build_function_type	7	8	4	4	4	0
1899	tree_c__build_reference_type	7	8	3	3	3	0
1900	tree_c__build_method_type	7	8	2	3	3	0
1901	tree_c__build_offset_type	5	5	2	2	2	0
1902	tree_c__build_complex_type	5	5	2	2	2	0
1903	tree_c__get_unwidened	32	55	6_120	6_146	6_146	0
1904	tree_c__get_narrower	23	37	77	81	81	0
1905	tree_c__type_precision	8	10	4	4	4	0
1906	tree_c__int_fits_type_p	22	38	36	36	36	0
1907	tree_c__decl_function_context	20	28	17	25	22	12
1908	tree_c__decl_type_context	13	18	5	15	9	40
1909	tree_c__print_obstack_statistics	5	6	2	3	3	0
1910	tree_c__dump_tree_statistics	3	2	1	1	1	0
1911	tree_c__get_file_function_name	17	26	24	30	30	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1912	ucbqsort_c__qst	39	59	352	1_079	722	33.1
1913	ucbqsort_c__qsort	24	39	91	107	102	4.7
1914	unroll_c__unroll_loop	248	397	1.97529×10^{22}	2.96294×10^{22}	2.96294×10^{22}	0
1915	unroll_c__precondition_loop_p	45	70	106	114	114	0
1916	unroll_c__init_reg_map	7	9	2	4	4	0
1917	unroll_c__calculate_giv_inc	20	28	12	89	65	27
1918	unroll_c__initial_reg_note_copy	10	12	3	4	4	0
1919	unroll_c__final_reg_note_copy	7	9	3	5	5	0
1920	unroll_c__copy_loop_body	102	168	159_210	162_799	161_003	1.1
1921	unroll_c__emit_unrolled_add	5	5	2	2	2	0
1922	unroll_c__back_branch_in_range_p	17	26	24	35	29	17.1
1923	unroll_c__fold_rtx_mult_add	18	27	128	135	135	0
1924	unroll_c__biv_total_increment	10	14	5	7	6	14.3
1925	unroll_c__iteration_info	21	30	11	11	11	0
1926	unroll_c__approx_final_value	18	27	10	11	11	0
1927	unroll_c__find_splittable_regs	48	79	10_083	30_266	20_174	33.3
1928	unroll_c__find_splittable_givs	87	144	4_114_306	12_342_932	8_228_627	33.3
1929	unroll_c__reg_dead_after_loop	18	29	19	65	36	44.6
1930	unroll_c__final_biv_value	23	34	22	22	22	0
1931	unroll_c__final_giv_value	26	40	22	33	31	6.1
1932	unroll_c__loop_iterations	65	106	3_487	3_760	3_754	0.2

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1933	varasm_c__text_section	8	9	3	3	3	0
1934	varasm_c__data_section	8	9	3	3	3	0
1935	varasm_c__readonly_data_section	3	2	1	1	1	0
1936	varasm_c__in_text_section	3	2	1	1	1	0
1937	varasm_c__make_function_rtl	19	27	23	23	23	0
1938	varasm_c__bc_make_decl_rtl	22	32	61	61	61	0
1939	varasm_c__strip_reg_name	6	7	3	3	3	0
1940	varasm_c__decode_reg_name	23	35	49	53	53	0
1941	varasm_c__make_decl_rtl	80	126	92_911_519	92_911_520	92_911_520	0
1942	varasm_c__make_var_volatile	5	5	1	2	2	0
1943	varasm_c__assemble_constant_align	13	18	21	21	21	0
1944	varasm_c__assemble_asm	8	9	3	3	3	0
1945	varasm_c__assemble_destructor	5	5	2	2	2	0
1946	varasm_c__assemble_constructor	5	5	2	2	2	0
1947	varasm_c__assemble_gc_entry	5	5	2	2	2	0
1948	varasm_c__assemble_start_function	15	21	42	42	42	0
1949	varasm_c__assemble_end_function	3	2	1	1	1	0
1950	varasm_c__assemble_zeros	8	10	4	4	4	0
1951	varasm_c__assemble_align	10	13	7	7	7	0
1952	varasm_c__assemble_string	24	36	36	100	85	15
1953	varasm_c__bc_output_ascii	15	22	16	31	31	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1954	varasm_c__assemble_variable	125	206	5_443_203_096	5_443_203_096	5_443_203_096	0
1955	varasm_c__contains_pointers_p	12	18	7	9	8	11.1
1956	varasm_c__bc_output_constructor	8	10	2	4	4	0
1957	varasm_c__bc_output_data_constructor	6	7	2	3	3	0
1958	varasm_c__assemble_external	3	2	1	1	1	0
1959	varasm_c__assemble_external_libcall	3	2	1	1	1	0
1960	varasm_c__assemble_global	3	2	1	1	1	0
1961	varasm_c__assemble_label	5	5	2	2	2	0
1962	varasm_c__assemble_name	10	12	4	4	4	0
1963	varasm_c__assemble_static_space	16	21	64	64	64	0
1964	varasm_c__assemble_trampoline_template	13	17	16	17	17	0
1965	varasm_c__assemble_integer	33	50	104	221	213	3.6
1966	varasm_c__assemble_real	16	20	6	10	10	0
1967	varasm_c__immed_double_const	32	51	1_993	2_007	2_004	0.1
1968	varasm_c__immed_real_const_1	9	11	4	4	4	0
1969	varasm_c__immed_real_const	3	2	1	1	1	0
1970	varasm_c__clear_const_double_mem	7	9	3	4	4	0
1971	varasm_c__decode_addr_const	26	38	16	44	38	13.6
1972	varasm_c__const_hash	36	53	21	25	25	0
1973	varasm_c__compare_constant	3	2	1	1	1	0
1974	varasm_c__compare_constant_1	51	80	51	60	55	8.3

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1975	varasm_c__record_constant	11	14	16	16	16	0
1976	varasm_c__record_constant_1	55	79	236	239	239	0
1977	varasm_c__defer_addressed_constants	3	2	1	1	1	0
1978	varasm_c__output_deferred_addressed_constants	7	9	3	4	4	0
1979	varasm_c__copy_constant	13	20	6	9	9	0
1980	varasm_c__output_constant_def	32	47	673	675	675	0
1981	varasm_c__output_constant_def_contents	33	52	672	672	672	0
1982	varasm_c__init_const_rtx_hash_table	3	2	1	1	1	0
1983	varasm_c__save_varasm_status	3	2	1	1	1	0
1984	varasm_c__restore_varasm_status	3	2	1	1	1	0
1985	varasm_c__decode_rtx_const	24	38	48	60	59	1.7
1986	varasm_c__simplify_subtraction	6	6	2	2	2	0
1987	varasm_c__const_hash_rtx	5	5	1	2	2	0
1988	varasm_c__compare_constant_rtx	7	8	2	3	3	0
1989	varasm_c__record_constant_rtx	13	17	32	32	32	0
1990	varasm_c__force_const_mem	32	48	3_121	3_122	3_122	0
1991	varasm_c__find_pool_constant	7	9	1	5	4	20
1992	varasm_c__get_pool_constant	3	2	1	1	1	0
1993	varasm_c__get_pool_mode	3	2	1	1	1	0
1994	varasm_c__get_pool_offset	3	2	1	1	1	0
1995	varasm_c__get_pool_size	3	2	1	1	1	0

Table 6: SPEC CINT95 — 126.gcc

no.	function	nodes	edges	npp	ncp	loncp	rd.
1996	varasm_c__output_constant_pool	32	50	505	2_017	1_513	25
1997	varasm_c__output_addressed_constants	24	37	15	18	18	0
1998	varasm_c__output_byte_asm	6	6	2	2	2	0
1999	varasm_c__output_constant	35	55	147	161	160	0.6
2000	varasm_c__bc_assemble_integer	33	51	120	194	189	2.6
2001	varasm_c__output_constructor	94	145	21_048_504	24_277_076	23_399_806	3.6

Table 7: SPEC CINT95 — 129.compress

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	compress95_c_spec_select_action	17	23	12	24	24	0
2	compress95_c_compress	31	49	54	161	127	21.1
3	compress95_c_output	29	41	110	111	111	0
4	compress95_c_decompress	28	42	8	141	108	23.4
5	compress95_c_getcode	19	26	56	56	56	0
6	compress95_c_rindex	7	9	3	5	5	0
7	compress95_c_onintr	3	2	1	2	2	0
8	compress95_c_oops	5	5	2	4	4	0
9	compress95_c_cl_block	12	14	6	6	6	0
10	compress95_c_cl_hash	7	9	2	4	4	0
11	compress95_c_prratio	8	9	4	4	4	0
12	compress95_c_version	3	2	1	1	1	0
13	compress95_c_getbyte	6	6	2	2	2	0
14	compress95_c_putbyte	3	2	1	1	1	0
15	compress95_c_readbytes	10	13	5	6	6	0
16	compress95_c_writebytes	5	6	2	3	3	0
17	harness_c_add_line	5	6	2	3	3	0
18	harness_c_fill_text_buffer	17	24	4	14	11	21.4
19	harness_c_print_buffer	5	6	2	3	3	0
20	harness_c_getranchar	13	16	4	6	6	0
21	harness_c_ran2	3	2	1	1	1	0

Table 7: SPEC CINT95 — 129.compress

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	harness_c__ran	6	6	2	2	2	0
23	harness_c__compare_buffer	11	14	5	5	5	0
24	harness_c__main	5	5	1	2	2	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	os_c__osinit	3	2	1	1	1	0
2	os_c__osrand	3	2	1	1	1	0
3	os_c__osgetc	3	2	1	1	1	0
4	os_c__osputc	3	2	1	1	1	0
5	os_c__oscheck	3	2	1	1	1	0
6	os_c__osfinit	3	2	1	1	1	0
7	os_c__osfinish	3	2	1	1	1	0
8	xlbfun_c__xeval	3	2	1	1	1	0
9	xlbfun_c__xapply	6	7	3	3	3	0
10	xlbfun_c__xfuncall	6	7	3	3	3	0
11	xlbfun_c__xquote	3	2	1	1	1	0
12	xlbfun_c__xfunction	12	16	8	8	8	0
13	xlbfun_c__xlambda	3	2	1	1	1	0
14	xlbfun_c__xbquote	3	2	1	1	1	0
15	xlbfun_c__bquote1	30	45	44	46	46	0
16	xlbfun_c__xset	3	2	1	1	1	0
17	xlbfun_c__xsetq	5	6	2	3	3	0
18	xlbfun_c__xsetf	12	17	6	11	11	0
19	xlbfun_c__placeform	34	54	31	34	32	5.9
20	xlbfun_c__xdefun	3	2	1	1	1	0
21	xlbfun_c__xdefmacro	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	xlbfun_c__defun	3	2	1	1	1	0
23	xlbfun_c__xgensym	10	12	4	4	4	0
24	xlbfun_c__xmakesymbol	3	2	1	1	1	0
25	xlbfun_c__xintern	3	2	1	1	1	0
26	xlbfun_c__makesymbol	6	6	2	2	2	0
27	xlbfun_c__xsymname	3	2	1	1	1	0
28	xlbfun_c__xsymvalue	5	6	2	3	3	0
29	xlbfun_c__xsymplist	3	2	1	1	1	0
30	xlbfun_c__xget	3	2	1	1	1	0
31	xlbfun_c__xputprop	3	2	1	1	1	0
32	xlbfun_c__xremprop	3	2	1	1	1	0
33	xlbfun_c__xhash	10	13	5	5	5	0
34	xlbfun_c__xaref	6	7	3	3	3	0
35	xlbfun_c__xmkarray	3	2	1	1	1	0
36	xlcont_c__xcond	10	14	4	6	6	0
37	xlcont_c__xcase	14	22	16	20	20	0
38	xlcont_c__keypresent	9	13	5	7	6	14.3
39	xlcont_c__xand	7	9	3	4	4	0
40	xlcont_c__xor	7	9	3	4	4	0
41	xlcont_c__xif	9	10	4	4	4	0
42	xlcont_c__xlet	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	xlcont_c__xletstar	3	2	1	1	1	0
44	xlcont_c__let	10	13	8	9	9	0
45	xlcont_c__xprog	3	2	1	1	1	0
46	xlcont_c__xprogstar	3	2	1	1	1	0
47	xlcont_c__prog	7	8	4	4	4	0
48	xlcont_c__xgo	3	2	1	1	1	0
49	xlcont_c__xreturn	6	6	2	2	2	0
50	xlcont_c__xprog1	3	2	1	1	1	0
51	xlcont_c__xprog2	3	2	1	1	1	0
52	xlcont_c__progx	9	12	4	6	6	0
53	xlcont_c__xprogn	6	7	2	3	3	0
54	xlcont_c__xdo	3	2	1	1	1	0
55	xlcont_c__xdostar	3	2	1	1	1	0
56	xlcont_c__doloop	18	26	40	42	42	0
57	xlcont_c__xdolist	14	20	20	21	21	0
58	xlcont_c__xdotimes	12	16	12	13	13	0
59	xlcont_c__xcatch	8	10	3	4	4	0
60	xlcont_c__xthrow	6	6	2	2	2	0
61	xlcont_c__xerror	6	6	2	2	2	0
62	xlcont_c__xcerror	6	6	2	2	2	0
63	xlcont_c__xbreak	9	10	4	4	4	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	xlcont_c__xcleanup	3	2	1	1	1	0
65	xlcont_c__xcontinue	3	2	1	1	1	0
66	xlcont_c__xerrset	8	9	4	4	4	0
67	xlcont_c__xevalhook	6	6	2	2	2	0
68	xlcont_c__dobindings	15	22	12	17	17	0
69	xlcont_c__doupdates	16	25	24	30	30	0
70	xlcont_c__tagblock	16	23	17	20	20	0
71	xldbug_c__xlfail	3	2	1	1	1	0
72	xldbug_c__xlabort	3	2	1	1	1	0
73	xldbug_c__xlbreak	3	2	1	1	1	0
74	xldbug_c__xlerror	3	2	1	1	1	0
75	xldbug_c__xlcerror	3	2	1	1	1	0
76	xldbug_c__xlerrprint	8	9	4	4	4	0
77	xldbug_c__doerror	5	5	2	2	2	0
78	xldbug_c__breakloop	28	39	88	95	95	0
79	xldbug_c__stacktop	6	6	2	2	2	0
80	xldbug_c__xlbaktrace	10	14	3	15	11	26.7
81	xldbug_c__xldinit	5	5	1	2	2	0
82	xldmem_c__cons	3	2	1	1	1	0
83	xldmem_c__consa	3	2	1	1	1	0
84	xldmem_c__consd	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	xldmem_c_cvstring	3	2	1	1	1	0
86	xldmem_c_cvcstring	3	2	1	1	1	0
87	xldmem_c_cvsymbol	3	2	1	1	1	0
88	xldmem_c_cvcsymbol	3	2	1	1	1	0
89	xldmem_c_cvsubr	3	2	1	1	1	0
90	xldmem_c_cvfile	3	2	1	1	1	0
91	xldmem_c_cvfixnum	3	2	1	1	1	0
92	xldmem_c_cvflonum	3	2	1	1	1	0
93	xldmem_c_newstring	3	2	1	1	1	0
94	xldmem_c_newobject	3	2	1	1	1	0
95	xldmem_c_newvector	6	7	3	3	3	0
96	xldmem_c_newnode	6	7	3	3	3	0
97	xldmem_c_stralloc	6	7	3	3	3	0
98	xldmem_c_strsave	3	2	1	1	1	0
99	xldmem_c_findmem	5	5	2	2	2	0
100	xldmem_c_gc	5	6	2	3	3	0
101	xldmem_c_mark	18	24	3	11	9	18.2
102	xldmem_c_vmark	5	6	2	3	3	0
103	xldmem_c_sweep	25	37	12	43	33	23.3
104	xldmem_c_addseg	11	14	4	5	5	0
105	xldmem_c_livecar	9	12	3	5	5	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	xldmem_c_livecdr	8	10	2	4	4	0
107	xldmem_c_stats	3	2	1	1	1	0
108	xldmem_c_xlminit	5	5	1	2	2	0
109	xleval_c_xleval	17	23	22	22	22	0
110	xleval_c_xlxeval	10	13	5	5	5	0
111	xleval_c_xlapply	18	24	10	10	10	0
112	xleval_c_evform	27	39	33	33	33	0
113	xleval_c_evalhook	3	2	1	1	1	0
114	xleval_c_xlevlist	10	13	5	9	9	0
115	xleval_c_xlunbound	3	2	1	1	1	0
116	xleval_c_evfun	12	17	18	19	19	0
117	xleval_c_xlabind	46	82	7_593	7_637	7_624	0.2
118	xleval_c_iskeyword	7	9	4	4	4	0
119	xleval_c_xlsave	7	9	3	5	5	0
120	xlfio_c_xread	14	17	16	16	16	0
121	xlfio_c_xprint	3	2	1	1	1	0
122	xlfio_c_xprin1	3	2	1	1	1	0
123	xlfio_c_xprinc	3	2	1	1	1	0
124	xlfio_c_xterpri	6	6	2	2	2	0
125	xlfio_c_printit	8	9	4	4	4	0
126	xlfio_c_xflatsize	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	xlfio_c__xflatc	3	2	1	1	1	0
128	xlfio_c__flatsize	3	2	1	1	1	0
129	xlfio_c__xopeni	3	2	1	1	1	0
130	xlfio_c__xopeno	3	2	1	1	1	0
131	xlfio_c__openit	13	17	10	10	10	0
132	xlfio_c__xclose	5	5	2	2	2	0
133	xlfio_c__xrdchar	9	10	4	4	4	0
134	xlfio_c__xpkchar	19	26	23	32	29	9.4
135	xlfio_c__xwrchar	6	6	2	2	2	0
136	xlfio_c__xreadline	24	36	96	111	108	2.7
137	xlinit_c_xlinit	5	6	2	3	3	0
138	xlio_c_xlgetc	26	40	41	41	41	0
139	xlio_c_xlpeek	12	17	7	7	7	0
140	xlio_c_xlputc	10	12	4	4	4	0
141	xlio_c_xlflush	5	6	2	3	3	0
142	xlisp_c__main	20	28	25	60	55	8.3
143	xlisp_c__stdprint	3	2	1	1	1	0
144	xlisp_c__stdputstr	3	2	1	1	1	0
145	xljump_c_xlbegin	3	2	1	1	1	0
146	xljump_c_xlend	3	2	1	1	1	0
147	xljump_c_xljump	3	2	1	2	2	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	xljump_c__xltoplevel	3	2	1	1	1	0
149	xljump_c__xlcleanup	3	2	1	1	1	0
150	xljump_c__xlcontinue	3	2	1	1	1	0
151	xljump_c__xlgo	12	19	8	19	17	10.5
152	xljump_c__xlreturn	7	9	3	5	5	0
153	xljump_c__xlthrow	8	11	4	7	7	0
154	xljump_c__xlsignal	10	14	5	9	9	0
155	xljump_c__findtarget	7	9	3	5	5	0
156	xllist_c__xcar	3	2	1	1	1	0
157	xllist_c__xcdr	3	2	1	1	1	0
158	xllist_c__xcaar	3	2	1	1	1	0
159	xllist_c__xcadr	3	2	1	1	1	0
160	xllist_c__xcdar	3	2	1	1	1	0
161	xllist_c__xcddr	3	2	1	1	1	0
162	xllist_c__xcaaar	3	2	1	1	1	0
163	xllist_c__xcaadr	3	2	1	1	1	0
164	xllist_c__xcadar	3	2	1	1	1	0
165	xllist_c__xcaddr	3	2	1	1	1	0
166	xllist_c__xcdaar	3	2	1	1	1	0
167	xllist_c__xcdadr	3	2	1	1	1	0
168	xllist_c__xcddar	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	xllist_c__xcdddr	3	2	1	1	1	0
170	xllist_c__xcaaaar	3	2	1	1	1	0
171	xllist_c__xcaaadr	3	2	1	1	1	0
172	xllist_c__xcaadar	3	2	1	1	1	0
173	xllist_c__xcaaddr	3	2	1	1	1	0
174	xllist_c__xcadaar	3	2	1	1	1	0
175	xllist_c__xcadadr	3	2	1	1	1	0
176	xllist_c__xcaddar	3	2	1	1	1	0
177	xllist_c__xcadddr	3	2	1	1	1	0
178	xllist_c__xcdaaar	3	2	1	1	1	0
179	xllist_c__xcdaadr	3	2	1	1	1	0
180	xllist_c__xcdadar	3	2	1	1	1	0
181	xllist_c__xcdaddr	3	2	1	1	1	0
182	xllist_c__xcddaar	3	2	1	1	1	0
183	xllist_c__xcddadr	3	2	1	1	1	0
184	xllist_c__xcdddar	3	2	1	1	1	0
185	xllist_c__xcddddr	3	2	1	1	1	0
186	xllist_c__cxr	15	23	21	25	23	8
187	xllist_c__xcons	3	2	1	1	1	0
188	xllist_c__xlist	8	10	3	5	5	0
189	xllist_c__xappend	12	18	7	17	15	11.8

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	xllist_c__xreverse	7	10	4	5	5	0
191	xllist_c__xlast	9	14	6	7	7	0
192	xllist_c__xmember	9	13	5	6	6	0
193	xllist_c__xassoc	11	17	9	12	12	0
194	xllist_c__xsubst	3	2	1	1	1	0
195	xllist_c__subst	8	10	4	4	4	0
196	xllist_c__xsublis	3	2	1	1	1	0
197	xllist_c__sublis	9	11	4	4	4	0
198	xllist_c__assoc	11	17	9	15	12	20
199	xllist_c__xremove	12	17	8	11	11	0
200	xllist_c__dotest	3	2	1	1	1	0
201	xllist_c__xnth	3	2	1	1	1	0
202	xllist_c__xnthcdr	3	2	1	1	1	0
203	xllist_c__nth	18	28	96	97	97	0
204	xllist_c__xlength	17	26	13	14	14	0
205	xllist_c__xmapc	3	2	1	1	1	0
206	xllist_c__xmapcar	3	2	1	1	1	0
207	xllist_c__xmapl	3	2	1	1	1	0
208	xllist_c__xmaplist	3	2	1	1	1	0
209	xllist_c__map	27	41	72	106	102	3.8
210	xllist_c__xrplca	5	5	2	2	2	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	xllist_c__xrplcd	5	5	2	2	2	0
212	xllist_c__xnconc	14	21	10	21	20	4.8
213	xllist_c__xdelete	17	28	40	43	43	0
214	xllist_c__xatom	7	8	3	3	3	0
215	xllist_c__xsymbolp	7	8	3	3	3	0
216	xllist_c__xnumberp	8	10	4	4	4	0
217	xllist_c__xboundp	6	6	2	2	2	0
218	xllist_c__xnull	5	5	2	2	2	0
219	xllist_c__xlistp	7	8	3	3	3	0
220	xllist_c__xconsp	7	8	3	3	3	0
221	xllist_c__xeq	3	2	1	1	1	0
222	xllist_c__xeql	3	2	1	1	1	0
223	xllist_c__xequal	3	2	1	1	1	0
224	xllist_c__cequal	5	5	2	2	2	0
225	xlmath_c__xadd	3	2	1	1	1	0
226	xlmath_c__xsub	3	2	1	1	1	0
227	xlmath_c__xmul	3	2	1	1	1	0
228	xlmath_c__xdiv	3	2	1	1	1	0
229	xlmath_c__xrem	3	2	1	1	1	0
230	xlmath_c__xmin	3	2	1	1	1	0
231	xlmath_c__xmax	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	xlmath_c__xexpt	3	2	1	1	1	0
233	xlmath_c__xbitand	3	2	1	1	1	0
234	xlmath_c__xbitior	3	2	1	1	1	0
235	xlmath_c__xbitxor	3	2	1	1	1	0
236	xlmath_c__binary	60	99	7_030	7_205	7_205	0
237	xlmath_c__checkizero	5	5	2	2	2	0
238	xlmath_c__checkfzero	5	5	2	2	2	0
239	xlmath_c__checkfneg	5	5	2	2	2	0
240	xlmath_c__xbitnot	3	2	1	1	1	0
241	xlmath_c__xabs	3	2	1	1	1	0
242	xlmath_c__xadd1	3	2	1	1	1	0
243	xlmath_c__xsub1	3	2	1	1	1	0
244	xlmath_c__xsin	3	2	1	1	1	0
245	xlmath_c__xcos	3	2	1	1	1	0
246	xlmath_c__xtan	3	2	1	1	1	0
247	xlmath_c__xexp	3	2	1	1	1	0
248	xlmath_c__xsqrt	3	2	1	1	1	0
249	xlmath_c__xfix	3	2	1	1	1	0
250	xlmath_c__xfloat	3	2	1	1	1	0
251	xlmath_c__xrand	3	2	1	1	1	0
252	xlmath_c__unary	48	74	28	28	28	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
253	xlmath_c__xminusp	3	2	1	1	1	0
254	xlmath_c__xzerop	3	2	1	1	1	0
255	xlmath_c__xplusp	3	2	1	1	1	0
256	xlmath_c__xevenp	3	2	1	1	1	0
257	xlmath_c__xoddp	3	2	1	1	1	0
258	xlmath_c__predicate	30	44	30	30	30	0
259	xlmath_c__xlss	3	2	1	1	1	0
260	xlmath_c__xleq	3	2	1	1	1	0
261	xlmath_c__xequ	3	2	1	1	1	0
262	xlmath_c__xneq	3	2	1	1	1	0
263	xlmath_c__xgeq	3	2	1	1	1	0
264	xlmath_c__xgtr	3	2	1	1	1	0
265	xlmath_c__compare	55	90	15_520	15_520	15_520	0
266	xlmath_c__badiop	3	2	1	1	1	0
267	xlmath_c__badfop	3	2	1	1	1	0
268	xlobj_c_xlclass	3	2	1	1	1	0
269	xlobj_c_xladdivar	3	2	1	1	1	0
270	xlobj_c_xladdmsg	3	2	1	1	1	0
271	xlobj_c_xlsend	5	5	2	2	2	0
272	xlobj_c_xlobgetvalue	24	39	25	50	35	30
273	xlobj_c_xlobsetvalue	24	39	25	50	35	30

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
274	xlobj_c__obisnew	3	2	1	1	1	0
275	xlobj_c__obclass	3	2	1	1	1	0
276	xlobj_c__obshow	10	14	6	10	9	10
277	xlobj_c__obsendsuper	5	5	2	2	2	0
278	xlobj_c__clnew	3	2	1	1	1	0
279	xlobj_c__clisnew	9	10	4	4	4	0
280	xlobj_c__clanswer	3	2	1	1	1	0
281	xlobj_c__entermsg	7	9	3	5	4	20
282	xlobj_c__findmsg	10	15	5	19	10	47.4
283	xlobj_c__sendmsg	18	27	80	81	81	0
284	xlobj_c__getivcnt	6	7	3	3	3	0
285	xlobj_c__listlength	7	10	4	5	5	0
286	xlobj_c__xloinit	3	2	1	1	1	0
287	xlprin_c__xlprint	34	53	19	23	23	0
288	xlprin_c__xlterpri	3	2	1	1	1	0
289	xlprin_c__xlputstr	5	6	2	3	3	0
290	xlprin_c__putstring	23	32	16	31	31	0
291	xlprin_c__putatm	3	2	1	1	1	0
292	xlprin_c__putdec	3	2	1	1	1	0
293	xlprin_c__putfloat	3	2	1	1	1	0
294	xlprin_c__putoct	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
295	xlread_c__xlload	16	23	13	22	20	9.1
296	xlread_c__xlread	6	7	2	3	3	0
297	xlread_c__readone	15	21	8	8	8	0
298	xlread_c__rmhash	17	22	7	7	7	0
299	xlread_c__rmquote	3	2	1	1	1	0
300	xlread_c__rmdquote	21	30	2	29	20	31
301	xlread_c__rmbquote	3	2	1	1	1	0
302	xlread_c__rmcomma	6	6	2	2	2	0
303	xlread_c__rmlpar	3	2	1	1	1	0
304	xlread_c__rmrpar	3	2	1	1	1	0
305	xlread_c__rmsemi	6	7	2	3	3	0
306	xlread_c__phexnumber	14	20	4	34	22	35.3
307	xlread_c__plist	20	28	1	64	43	32.8
308	xlread_c__pvector	18	25	2	33	23	30.3
309	xlread_c__pquote	5	5	2	2	2	0
310	xlread_c__pname	20	28	32	40	40	0
311	xlread_c__tentry	9	12	5	5	5	0
312	xlread_c__nextch	7	9	2	5	4	20
313	xlread_c__checkeof	5	5	2	2	2	0
314	xlread_c__badeof	3	2	1	1	1	0
315	xlread_c__isnumber	34	56	1_558	1_636	1_625	0.7

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
316	xlread_c__defmacro	3	2	1	1	1	0
317	xlread_c__callmacro	3	2	1	1	1	0
318	xlread_c__needsextension	7	9	3	5	4	20
319	xlread_c__xlrinit	9	14	8	11	11	0
320	xlstr_c__xstrcat	9	12	4	6	6	0
321	xlstr_c__xsubstr	12	16	16	17	17	0
322	xlstr_c__xstring	3	2	1	1	1	0
323	xlstr_c__xchar	6	7	3	3	3	0
324	xlsubr_c__xlsubr	3	2	1	1	1	0
325	xlsubr_c__xlarg	6	7	3	3	3	0
326	xlsubr_c__xlmatch	10	13	6	6	6	0
327	xlsubr_c__xlevarg	3	2	1	1	1	0
328	xlsubr_c__xlevmatch	10	13	6	6	6	0
329	xlsubr_c__xltest	18	25	29	29	29	0
330	xlsubr_c__xlgetfile	11	15	6	6	6	0
331	xlsubr_c__xllastarg	5	5	2	2	2	0
332	xlsubr_c__eq	3	2	1	1	1	0
333	xlsubr_c__eql	22	34	54	54	54	0
334	xlsubr_c__equal	13	19	8	8	8	0
335	xlsym_c__xlenter	9	12	4	6	5	16.7
336	xlsym_c__xlsenter	3	2	1	1	1	0

Table 8: SPEC CINT95 — 130.li

no.	function	nodes	edges	npp	ncp	loncp	rd.
337	xlsym_c_xlmakesym	9	10	4	4	4	0
338	xlsym_c_xlframe	3	2	1	1	1	0
339	xlsym_c_xlbind	3	2	1	1	1	0
340	xlsym_c_xlgetvalue	6	7	1	4	3	25
341	xlsym_c_xlxgetvalue	13	19	9	17	12	29.4
342	xlsym_c_xlygetvalue	10	14	4	12	7	41.7
343	xlsym_c_xlsetvalue	12	18	9	17	12	29.4
344	xlsym_c_xlgetprop	5	5	2	2	2	0
345	xlsym_c_xlputprop	5	5	2	2	2	0
346	xlsym_c_xlremprop	15	24	16	19	19	0
347	xlsym_c_findprop	14	22	9	11	10	9.1
348	xlsym_c_hash	7	9	4	5	5	0
349	xlsym_c_xlsinit	3	2	1	1	1	0
350	xlsys_c_xload	18	24	40	40	40	0
351	xlsys_c_xgc	3	2	1	1	1	0
352	xlsys_c_xexpand	9	12	6	7	7	0
353	xlsys_c_xalloc	3	2	1	1	1	0
354	xlsys_c_xmem	3	2	1	1	1	0
355	xlsys_c_xtype	17	28	13	13	13	0
356	xlsys_c_xbaktrace	6	6	2	2	2	0
357	xlsys_c_xexit	3	2	1	2	2	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	jcapi_c__jpeg_create_compress	7	8	1	3	3	0
2	jcapi_c__jpeg_destroy_compress	3	2	1	1	1	0
3	jcapi_c__jpeg_suppress_tables	13	17	8	14	14	0
4	jcapi_c__jpeg_start_compress	10	12	8	8	8	0
5	jcapi_c__jpeg_write_scanlines	13	17	32	32	32	0
6	jcapi_c__jpeg_write_raw_data	16	21	34	34	34	0
7	jcapi_c__jpeg_finish_compress	15	22	24	37	33	10.8
8	jcapi_c__jpeg_write_marker	6	7	3	3	3	0
9	jcapi_c__jpeg_write_tables	5	5	2	2	2	0
10	jcapi_c__jpeg_abort_compress	3	2	1	1	1	0
11	jccoefct_c__start_pass_coef	19	25	8	8	8	0
12	jccoefct_c__compress_data	24	39	58	264	105	60.2
13	jccoefct_c__compress_first_pass	24	40	121	273	251	8.1
14	jccoefct_c__compress_output	22	36	60	130	84	35.4
15	jccoefct_c__jinit_c_coef_controller	10	13	3	5	5	0
16	jccolor_c__rgb_ycc_start	5	5	1	2	2	0
17	jccolor_c__rgb_ycc_convert	7	10	3	7	6	14.3
18	jccolor_c__rgb_gray_convert	7	10	3	7	6	14.3
19	jccolor_c__cmyk_ycck_convert	7	10	3	7	6	14.3
20	jccolor_c__grayscale_convert	7	10	3	7	6	14.3
21	jccolor_c__null_convert	9	14	4	15	10	33.3

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	jccolor_c__null_method	3	2	1	1	1	0
23	jccolor_c__jinit_color_converter	43	67	340	340	340	0
24	jcdctmgr_c__start_pass_fdctmgr	28	40	22	49	46	6.1
25	jcdctmgr_c__forward_DCT	17	23	5	19	14	26.3
26	jcdctmgr_c__forward_DCT_float	9	12	2	7	5	28.6
27	jcdctmgr_c__jinit_forward_dct	13	16	4	5	5	0
28	jchuff_c__start_pass_huff	23	34	154	230	230	0
29	jchuff_c__fix_huff_tbl	15	24	24	33	31	6.1
30	jchuff_c__dump_buffer	5	5	2	2	2	0
31	jchuff_c__emit_bits	13	20	20	32	26	18.8
32	jchuff_c__flush_bits	5	5	2	2	2	0
33	jchuff_c__encode_one_block	29	46	360	379	372	1.8
34	jchuff_c__emit_restart	11	17	12	13	13	0
35	jchuff_c__encode_mcu_huff	16	23	22	24	23	4.2
36	jchuff_c__finish_pass_huff	5	5	2	2	2	0
37	jchuff_c__htest_one_block	22	32	72	86	84	2.3
38	jchuff_c__encode_mcu_gather	11	16	8	10	10	0
39	jchuff_c__gen_huff_coding	43	67	432	494	480	2.8
40	jchuff_c__finish_pass_gather	13	18	10	19	19	0
41	jchuff_c__jinit_huff_encoder	5	5	1	2	2	0
42	jcmainct_c__start_pass_main	7	8	3	3	3	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	jcmainct_c__process_data_simple_main	13	19	11	15	15	0
44	jcmainct_c__jinit_c_main_controller	8	11	4	5	5	0
45	jcmarker_c__emit_byte	6	7	3	3	3	0
46	jcmarker_c__emit_marker	3	2	1	1	1	0
47	jcmarker_c__emit_2bytes	3	2	1	1	1	0
48	jcmarker_c__emit_dqt	15	20	12	16	16	0
49	jcmarker_c__emit_dht	16	22	12	15	15	0
50	jcmarker_c__emit_dac	3	2	1	1	1	0
51	jcmarker_c__emit_dri	3	2	1	1	1	0
52	jcmarker_c__emit_sof	8	11	6	7	7	0
53	jcmarker_c__emit_sos	5	6	2	3	3	0
54	jcmarker_c__emit_jfif_app0	3	2	1	1	1	0
55	jcmarker_c__emit_adobe_app14	9	10	3	3	3	0
56	jcmarker_c__write_any_marker	6	8	3	4	4	0
57	jcmarker_c__write_file_header	7	8	4	4	4	0
58	jcmarker_c__write_frame_header	21	32	216	220	220	0
59	jcmarker_c__write_scan_header	9	12	6	7	7	0
60	jcmarker_c__write_file_trailer	3	2	1	1	1	0
61	jcmarker_c__write_tables_only	14	19	10	16	16	0
62	jcmarker_c__jinit_marker_writer	3	2	1	1	1	0
63	jcmaster_c__initial_setup	27	43	1_980	1_997	1_997	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	jcmaster_c__per_scan_setup	23	33	105	123	122	0.8
65	jcmaster_c__master_selection	8	9	4	4	4	0
66	jcmaster_c__prepare_for_pass	23	31	48	49	49	0
67	jcmaster_c__pass_startup	3	2	1	1	1	0
68	jcmaster_c__finish_pass_master	3	2	1	1	1	0
69	jcmaster_c__jinit_master_compress	3	2	1	1	1	0
70	jcomapi_c__jpeg_abort	8	9	2	3	3	0
71	jcomapi_c__jpeg_destroy	5	5	2	2	2	0
72	jcomapi_c__jpeg_alloc_quant_table	3	2	1	1	1	0
73	jcomapi_c__jpeg_alloc_huff_table	3	2	1	1	1	0
74	jcparam_c__jpeg_add_quant_table	16	22	48	60	60	0
75	jcparam_c__jpeg_set_linear_quality	3	2	1	1	1	0
76	jcparam_c__jpeg_quality_scaling	10	12	8	8	8	0
77	jcparam_c__jpeg_set_quality	3	2	1	1	1	0
78	jcparam_c__add_huff_table	5	5	2	2	2	0
79	jcparam_c__std_huff_tables	3	2	1	1	1	0
80	jcparam_c__jpeg_set_defaults	11	14	8	9	9	0
81	jcparam_c__jpeg_default_colorspace	12	16	6	6	6	0
82	jcparam_c__jpeg_set_colorspace	20	29	20	21	21	0
83	jcprepct_c__start_pass_prep	5	5	2	2	2	0
84	jcprepct_c__expand_bottom_edge	6	7	2	3	3	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	jcprepct_c__pre_process_data	21	34	98	133	132	0.8
86	jcprepct_c__pre_process_context	32	52	702	896	882	1.6
87	jcprepct_c__create_context_buffer	13	18	9	19	18	5.3
88	jcprepct_c__jinit_c_prep_controller	9	12	6	7	7	0
89	jcsample_c__start_pass_downsample	3	2	1	1	1	0
90	jcsample_c__expand_right_edge	8	12	4	8	7	12.5
91	jcsample_c__sep_downsample	6	7	2	3	3	0
92	jcsample_c__int_downsample	11	18	5	31	15	51.6
93	jcsample_c__fullsize_downsample	3	2	1	1	1	0
94	jcsample_c__h2v1_downsample	7	10	3	7	6	14.3
95	jcsample_c__h2v2_downsample	7	10	3	7	6	14.3
96	jcsample_c__h2v2_smooth_downsample	7	10	3	7	6	14.3
97	jcsample_c__fullsize_smooth_downsample	7	10	3	7	6	14.3
98	jcsample_c__jinit_downsampler	28	42	222	258	258	0
99	jdapi_c__jpeg_create_decompress	7	8	1	3	3	0
100	jdapi_c__jpeg_destroy_decompress	3	2	1	1	1	0
101	jdapi_c__jpeg_set_marker_processor	8	9	3	3	3	0
102	jdapi_c__default_decompress_parms	33	49	24	24	24	0
103	jdapi_c__jpeg_read_header	17	23	21	21	21	0
104	jdapi_c__jpeg_start_decompress	17	23	4	13	13	0
105	jdapi_c__jpeg_read_scanlines	9	11	8	8	8	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	jdapi_c__jpeg_read_raw_data	13	17	18	18	18	0
107	jdapi_c__jpeg_finish_decompress	19	26	24	24	24	0
108	jdapi_c__jpeg_abort_decompress	3	2	1	1	1	0
109	jdatadst_c__init_destination	3	2	1	1	1	0
110	jdatadst_c__empty_output_buffer	5	5	2	2	2	0
111	jdatadst_c__term_destination	8	10	6	6	6	0
112	jdatadst_c__jpeg_stdio_dest	5	5	2	2	2	0
113	jdatasrc_c_init_source	3	2	1	1	1	0
114	jdatasrc_c_fill_input_buffer	7	8	3	3	3	0
115	jdatasrc_c_skip_input_data	7	9	3	4	4	0
116	jdatasrc_c_term_source	3	2	1	1	1	0
117	jdatasrc_c_jpeg_stdio_src	5	5	2	2	2	0
118	jdcoefct_c_start_pass_coef	19	25	8	8	8	0
119	jdcoefct_c_decompress_data	21	33	16	125	49	60.8
120	jdcoefct_c_decompress_read	25	41	90	163	116	28.8
121	jdcoefct_c_decompress_output	14	22	11	29	24	17.2
122	jdcoefct_c_jinit_d_coef_controller	10	13	3	5	5	0
123	jdcolor_c_ycc_rgb_start	5	5	1	2	2	0
124	jdcolor_c_ycc_rgb_convert	7	10	3	7	6	14.3
125	jdcolor_c_null_convert	9	14	4	15	10	33.3
126	jdcolor_c_grayscale_convert	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	jdcolor_c_ycck_cmyk_convert	7	10	3	7	6	14.3
128	jdcolor_c_null_method	3	2	1	1	1	0
129	jdcolor_c_jinit_color_deconverter	38	56	300	301	301	0
130	jddctmgr_c_start_input_pass	25	36	16	41	35	14.6
131	jddctmgr_c_start_output_pass	8	11	4	7	7	0
132	jddctmgr_c_jinit_inverse_dct	26	35	10	19	19	0
133	jdhuff_c_start_pass_huff_decoder	11	16	10	19	19	0
134	jdhuff_c_fix_huff_tbl	25	39	72	93	86	7.5
135	jdhuff_c_fill_bit_buffer	22	33	25	73	43	41.1
136	jdhuff_c_slow_DECODE	13	19	15	19	17	10.5
137	jdhuff_c_process_restart	9	12	6	7	7	0
138	jdhuff_c_decode_mcu	62	100	7_306	11_006	9_096	17.4
139	jdhuff_c_jinit_huff_decoder	5	5	1	2	2	0
140	jdmainct_c_make_funny_pointers	13	20	9	23	20	13
141	jdmainct_c_set_wraparound_pointers	8	11	3	7	6	14.3
142	jdmainct_c_set_bottom_pointers	12	17	9	19	18	5.3
143	jdmainct_c_start_pass_main	15	19	6	6	6	0
144	jdmainct_c_process_data_simple_main	8	10	5	5	5	0
145	jdmainct_c_process_data_context_main	20	30	39	39	39	0
146	jdmainct_c_process_data_input_only	5	5	2	2	2	0
147	jdmainct_c_process_data_crank_post	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	jdmainct_c__jinit_d_main_controller	13	18	14	15	15	0
149	jdmarker_c__get_soi	7	8	2	3	3	0
150	jdmarker_c__get_sof	58	91	1_573_119	1_573_135	1_573_127	0
151	jdmarker_c__get_sos	46	73	32_654	32_684	32_667	0.1
152	jdmarker_c__get_app0	36	53	351	359	353	1.7
153	jdmarker_c__get_app14	28	41	111	117	113	3.4
154	jdmarker_c__get_dac	26	38	115	163	139	14.7
155	jdmarker_c__get_dht	37	56	1_219	1_811	1_511	16.6
156	jdmarker_c__get_dqt	39	58	523	742	626	15.6
157	jdmarker_c__get_dri	18	26	59	59	59	0
158	jdmarker_c__skip_variable	10	13	7	7	7	0
159	jdmarker_c__next_marker	21	31	33	55	49	10.9
160	jdmarker_c__first_marker	13	18	15	15	15	0
161	jdmarker_c__read_markers	57	82	41	101	101	0
162	jdmarker_c__read_restart_marker	9	12	7	7	7	0
163	jdmarker_c__jpeg_resync_to_restart	21	31	21	42	42	0
164	jdmarker_c__reset_marker_reader	3	2	1	1	1	0
165	jdmarker_c__jinit_marker_reader	7	8	2	3	3	0
166	jdmaster_c__use_merged_upsample	20	34	16	16	16	0
167	jdmaster_c__jpeg_calc_output_dimensions	43	68	19_040	19_065	19_064	0
168	jdmaster_c__per_scan_setup	19	27	35	53	52	1.9

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	jdmaster_c__prepare_range_limit_table	7	8	1	3	3	0
170	jdmaster_c__master_selection	27	36	228	228	228	0
171	jdmaster_c__prepare_for_pass	34	46	42	42	42	0
172	jdmaster_c__finish_pass_master	21	28	10	10	10	0
173	jdmaster_c__jinit_master_decompress	3	2	1	1	1	0
174	jdmerge_c__start_pass_merged_upsample	5	5	1	2	2	0
175	jdmerge_c__merged_2v_upsample	15	19	18	18	18	0
176	jdmerge_c__merged_1v_upsample	3	2	1	1	1	0
177	jdmerge_c__h2v1_merged_upsample	7	9	4	5	5	0
178	jdmerge_c__h2v2_merged_upsample	7	9	4	5	5	0
179	jdmerge_c__jinit_merged_upsampler	6	6	2	2	2	0
180	jdpostct_c__start_pass_dpost	19	25	8	8	8	0
181	jdpostct_c__post_process_1pass	5	5	2	2	2	0
182	jdpostct_c__post_process_prepass	9	11	8	8	8	0
183	jdpostct_c__post_process_2pass	11	14	16	16	16	0
184	jdpostct_c__jinit_d_post_controller	7	8	3	3	3	0
185	jdsample_c__start_pass_upsample	3	2	1	1	1	0
186	jdsample_c__sep_upsample	13	18	24	25	25	0
187	jdsample_c__fullsize_upsample	3	2	1	1	1	0
188	jdsample_c__noop_upsample	3	2	1	1	1	0
189	jdsample_c__int_upsample	11	17	7	21	16	23.8

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	jdsample_c__h2v1_upsample	7	10	3	7	6	14.3
191	jdsample_c__h2v2_upsample	7	10	3	7	6	14.3
192	jdsample_c__h2v1_fancy_upsample	7	10	3	7	6	14.3
193	jdsample_c__h2v2_fancy_upsample	12	17	5	21	14	33.3
194	jdsample_c__jinit_upsampler	34	52	534	622	622	0
195	jerror_c__error_exit	3	2	1	2	2	0
196	jerror_c__output_message	3	2	1	1	1	0
197	jerror_c__emit_message	10	13	5	5	5	0
198	jerror_c__format_message	20	30	144	145	145	0
199	jerror_c__reset_error_mgr	3	2	1	1	1	0
200	jerror_c__jpeg_std_error	3	2	1	1	1	0
201	jfdctflt_c__jpeg_fdct_float	7	8	1	3	3	0
202	jfdctfst_c__jpeg_fdct_ifast	7	8	1	3	3	0
203	jfdctint_c__jpeg_fdct_islow	7	8	1	3	3	0
204	jidctflt_c__jpeg_idct_float	10	12	2	5	5	0
205	jidctfst_c__jpeg_idct_ifast	13	16	4	8	8	0
206	jidctint_c__jpeg_idct_islow	13	16	4	8	8	0
207	jidctred_c__jpeg_idct_4x4	14	18	6	11	11	0
208	jidctred_c__jpeg_idct_2x2	17	23	10	17	17	0
209	jidctred_c__jpeg_idct_1x1	3	2	1	1	1	0
210	jmemmgr_c__out_of_memory	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	jmemmgr_c__alloc_small	25	36	200	204	203	0.5
212	jmemmgr_c__alloc_large	11	14	16	16	16	0
213	jmemmgr_c__alloc_sarray	13	19	20	26	25	3.8
214	jmemmgr_c__alloc_barray	14	20	20	26	25	3.8
215	jmemmgr_c__request_virt_sarray	5	5	2	2	2	0
216	jmemmgr_c__request_virt_barray	5	5	2	2	2	0
217	jmemmgr_c__realize_virt_arrays	29	45	441	451	451	0
218	jmemmgr_c__do_sarray_io	13	18	13	21	21	0
219	jmemmgr_c__do_barray_io	13	18	13	21	21	0
220	jmemmgr_c__access_virt_sarray	19	27	150	150	150	0
221	jmemmgr_c__access_virt_barray	19	27	150	150	150	0
222	jmemmgr_c__free_pool	19	30	80	86	86	0
223	jmemmgr_c__self_destruct	5	5	1	2	2	0
224	jmemmgr_c__jinit_memory_mgr	13	18	10	11	11	0
225	jmemnobs_c__jpeg_get_small	3	2	1	1	1	0
226	jmemnobs_c__jpeg_free_small	3	2	1	1	1	0
227	jmemnobs_c__jpeg_get_large	3	2	1	1	1	0
228	jmemnobs_c__jpeg_free_large	3	2	1	1	1	0
229	jmemnobs_c__jpeg_mem_available	3	2	1	1	1	0
230	jmemnobs_c__jpeg_open_backing_store	3	2	1	1	1	0
231	jmemnobs_c__jpeg_mem_init	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	jmemnobs_c__jpeg_mem_term	3	2	1	1	1	0
233	jquant1_c_select_ncolors	17	26	24	31	30	3.2
234	jquant1_c_output_value	3	2	1	1	1	0
235	jquant1_c_largest_input_value	3	2	1	1	1	0
236	jquant1_c_create_colormap	30	48	396	461	439	4.8
237	jquant1_c_color_quantize	9	14	4	15	10	33.3
238	jquant1_c_color_quantize3	7	10	3	7	6	14.3
239	jquant1_c_quantize_ord_dither	9	14	4	15	10	33.3
240	jquant1_c_quantize3_ord_dither	7	10	3	7	6	14.3
241	jquant1_c_quantize_fs_dither	14	20	6	23	16	30.4
242	jquant1_c_start_pass_1_quant	3	2	1	1	1	0
243	jquant1_c_finish_pass_1_quant	3	2	1	1	1	0
244	jquant1_c_jinit_1pass_quantizer	22	30	28	29	29	0
245	jquant2_c_prescan_quantize	9	13	4	11	9	18.2
246	jquant2_c_find_biggest_color_pop	8	11	4	7	7	0
247	jquant2_c_find_biggest_volume	7	9	3	5	5	0
248	jquant2_c_update_box	70	121	87_880	87_988	87_925	0.1
249	jquant2_c_median_cut	22	31	43	83	83	0
250	jquant2_c_compute_color	13	19	5	23	14	39.1
251	jquant2_c_select_colors	5	6	2	3	3	0
252	jquant2_c_find_nearby_colors	32	46	387	517	517	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
253	jquant2_c__find_best_colors	15	21	3	34	12	64.7
254	jquant2_c__fill_inverse_cmap	9	11	1	8	4	50
255	jquant2_c__pass2_no_dither	9	13	4	11	9	18.2
256	jquant2_c__pass2_fs_dither	13	18	7	17	15	11.8
257	jquant2_c__init_error_limit	10	14	4	7	7	0
258	jquant2_c__finish_pass1	3	2	1	1	1	0
259	jquant2_c__finish_pass2	3	2	1	1	1	0
260	jquant2_c__start_pass_2_quant	11	13	3	4	4	0
261	jquant2_c__jinit_2pass_quantizer	20	27	128	129	129	0
262	jutils_c__jdiv_round_up	3	2	1	1	1	0
263	jutils_c__jround_up	3	2	1	1	1	0
264	jutils_c__jcopy_sample_rows	5	6	2	3	3	0
265	jutils_c__jcopy_block_row	3	2	1	1	1	0
266	jutils_c__jzero_far	3	2	1	1	1	0
267	libpbm1_c__pm_allocrow	5	5	2	2	2	0
268	libpbm1_c__pm_freerow	3	2	1	1	1	0
269	libpbm1_c__pm_allocarray	9	12	8	9	9	0
270	libpbm1_c__pm_freearray	3	2	1	1	1	0
271	libpbm1_c__pm_keymatch	14	20	11	23	15	34.8
272	libpbm1_c__pm_maxvaltobits	20	35	17	17	17	0
273	libpbm1_c__pm_bitstomaxval	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
274	libpbm1_c__pm_init	15	22	10	36	21	41.7
275	libpbm1_c__pbm_init	3	2	1	1	1	0
276	libpbm1_c__pm_usage	3	2	1	2	2	0
277	libpbm1_c__pm_perror	7	8	3	3	3	0
278	libpbm1_c__pm_message	5	5	2	2	2	0
279	libpbm1_c__pm_error	3	2	1	2	2	0
280	libpbm1_c__pm_openr	9	11	4	6	6	0
281	libpbm1_c__pm_openw	5	5	1	2	2	0
282	libpbm1_c__pm_close	8	10	6	6	6	0
283	libpbm1_c__pm_readbigshort	6	7	3	3	3	0
284	libpbm1_c__pm_writebigshort	3	2	1	1	1	0
285	libpbm1_c__pm_readbiglong	8	11	5	5	5	0
286	libpbm1_c__pm_writebiglong	3	2	1	1	1	0
287	libpbm1_c__pm_readlittleshort	6	7	3	3	3	0
288	libpbm1_c__pm_writelittleshort	3	2	1	1	1	0
289	libpbm1_c__pm_readlittlelong	8	11	5	5	5	0
290	libpbm1_c__pm_writelittlelong	3	2	1	1	1	0
291	libpbm2_c__pbm_getbit	10	14	2	6	6	0
292	libpbm2_c__pbm_readmagicnumber	7	8	4	4	4	0
293	libpbm2_c__pbm_readpbminitrest	3	2	1	1	1	0
294	libpbm2_c__pbm_readpbminit	10	12	6	6	6	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
295	libpbm2_c__pbm_readpbmrow	15	21	6	9	9	0
296	libpbm2_c__pbm_readpbm	5	6	2	3	3	0
297	libpbm3_c__pbm_writepbminit	6	6	2	2	2	0
298	libpbm3_c__pbm_writepbmrowraw	11	15	10	14	14	0
299	libpbm3_c__pbm_writepbmrowplain	10	13	5	9	9	0
300	libpbm3_c__pbm_writepbmrow	6	6	2	2	2	0
301	libpbm3_c__pbm_writepbm	5	6	2	3	3	0
302	libpbm4_c__pbm_getc	10	14	10	12	12	0
303	libpbm4_c__pbm_getrawbyte	5	5	2	2	2	0
304	libpbm4_c__pbm_getint	12	17	2	7	7	0
305	libpbm5_c__pbm_defaultfont	16	22	12	56	38	32.1
306	libpbm5_c__pbm_dissectfont	29	44	240	272	257	5.5
307	libpbm5_c__pbm_dumpfont	21	32	267	471	379	19.5
308	libpgm1_c__pgm_init	3	2	1	1	1	0
309	libpgm1_c__pgm_readpgminitrest	5	5	2	2	2	0
310	libpgm1_c__pgm_readpgminit	15	20	15	15	15	0
311	libpgm1_c__pgm_readpgmrow	24	35	12	16	16	0
312	libpgm1_c__pgm_readpgm	5	6	2	3	3	0
313	libpgm2_c__pgm_writepgminit	6	6	2	2	2	0
314	libpgm2_c__putus	5	5	2	2	2	0
315	libpgm2_c__pgm_writepgmrowraw	5	6	2	3	3	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
316	libpgm2_c__pgm_writepgmrowplain	12	16	8	11	11	0
317	libpgm2_c__pgm_writepgmrow	6	6	2	2	2	0
318	libpgm2_c__pgm_writepgm	6	7	2	3	3	0
319	libppm1_c__ppm_init	3	2	1	1	1	0
320	libppm1_c__ppm_readppminitrest	5	5	2	2	2	0
321	libppm1_c__ppm_readppminit	22	31	35	35	35	0
322	libppm1_c__ppm_readppmrow	23	35	10	15	15	0
323	libppm1_c__ppm_readppm	5	6	2	3	3	0
324	libppm2_c__ppm_writeppminit	6	6	2	2	2	0
325	libppm2_c__putus	5	5	2	2	2	0
326	libppm2_c__ppm_writeppmrowraw	5	6	2	3	3	0
327	libppm2_c__ppm_writeppmrowplain	12	16	8	11	11	0
328	libppm2_c__ppm_writeppmrow	6	6	2	2	2	0
329	libppm2_c__ppm_writeppm	6	7	2	3	3	0
330	libppm3_c__ppm_computecolorhist	5	5	2	2	2	0
331	libppm3_c__ppm_addtocolorhist	20	33	17	26	23	11.5
332	libppm3_c__ppm_computecolorhash	21	32	18	131	44	66.4
333	libppm3_c__ppm_alloccolorhash	7	8	2	3	3	0
334	libppm3_c__ppm_addtocolorhash	5	5	2	2	2	0
335	libppm3_c__ppm_colorhashtocolorhist	9	12	4	8	7	12.5
336	libppm3_c__ppm_colorhisttocolorhash	13	20	11	29	25	13.8

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
337	libppm3_c__ppm_lookupcolor	10	14	5	11	8	27.3
338	libppm3_c__ppm_freecolorhist	3	2	1	1	1	0
339	libppm3_c__ppm_freecolorhash	7	9	2	6	5	16.7
340	libppm4_c__canonstr	10	13	4	7	7	0
341	libppm4_c__rgbnorm	20	28	10	10	10	0
342	libppm4_c__ppm_parsecolor	66	107	141	151	145	4
343	libppm4_c__ppm_colorname	8	9	4	4	4	0
344	libppm5_c__ppmd_point_drawproc	8	11	5	5	5	0
345	libppm5_c__ppmd_filledrectangle	18	26	64	71	69	2.8
346	libppm5_c__ppmd_setlinetype	3	2	1	1	1	0
347	libppm5_c__ppmd_setlineclip	3	2	1	1	1	0
348	libppm5_c__ppmd_line	79	119	42_076	42_108	42_108	0
349	libppm5_c__ppmd_spline3	17	22	64	64	64	0
350	libppm5_c__ppmd_polyspline	5	6	2	3	3	0
351	libppm5_c__ppmd_circle	23	33	160	400	400	0
352	libppm5_c__ppmd_fill_init	7	8	4	4	4	0
353	libppm5_c__ppmd_fill_drawproc	28	44	181	182	182	0
354	libppm5_c__yx_compare	10	13	5	5	5	0
355	libppm5_c__ppmd_fill	35	56	905	941	941	0
356	rdbmp_c_read_byte	5	5	2	2	2	0
357	rdbmp_c_read_colormap	13	18	5	7	7	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
358	rdbmp_c_get_8bit_row	5	6	2	3	3	0
359	rdbmp_c_get_24bit_row	5	6	2	3	3	0
360	rdbmp_c_preload_image	19	27	42	52	50	3.8
361	rdbmp_c_start_input_bmp	57	81	81_920	81_921	81_921	0
362	rdbmp_c_finish_input_bmp	3	2	1	1	1	0
363	rdbmp_c_jinit_read_bmp	3	2	1	1	1	0
364	rdgif_c_ReadByte	5	5	2	2	2	0
365	rdgif_c_GetDataBlock	6	7	3	3	3	0
366	rdgif_c_SkipDataBlocks	5	5	1	2	2	0
367	rdgif_c_ReInitLZW	3	2	1	1	1	0
368	rdgif_c_InitLZWCode	3	2	1	1	1	0
369	rdgif_c_GetCode	10	13	4	6	5	16.7
370	rdgif_c_LZWReadByte	29	42	57	59	59	0
371	rdgif_c_ReadColorMap	5	6	2	3	3	0
372	rdgif_c_DoExtension	3	2	1	1	1	0
373	rdgif_c_start_input_gif	39	58	82_944	82_948	82_948	0
374	rdgif_c_get_pixel_rows	5	6	2	3	3	0
375	rdgif_c_load_interlaced_image	11	16	10	16	15	6.2
376	rdgif_c_get_interlaced_row	16	22	12	13	13	0
377	rdgif_c_finish_input_gif	3	2	1	1	1	0
378	rdgif_c_jinit_read_gif	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
379	rdppm_c__pbm_getc	6	8	3	4	4	0
380	rdppm_c__read_pbm_integer	16	25	10	33	24	27.3
381	rdppm_c__get_text_gray_row	5	6	2	3	3	0
382	rdppm_c__get_text_rgb_row	5	6	2	3	3	0
383	rdppm_c__get_scaled_gray_row	7	9	4	5	5	0
384	rdppm_c__get_scaled_rgb_row	7	9	4	5	5	0
385	rdppm_c__get_raw_row	5	5	2	2	2	0
386	rdppm_c__start_input_ppm	36	50	384	385	385	0
387	rdppm_c__finish_input_ppm	3	2	1	1	1	0
388	rdppm_c__jinit_read_ppm	3	2	1	1	1	0
389	rdtarga_c__read_byte	5	5	2	2	2	0
390	rdtarga_c__read_colormap	7	9	4	5	5	0
391	rdtarga_c__read_non_rle_pixel	5	6	2	3	3	0
392	rdtarga_c__read_rle_pixel	11	15	7	8	8	0
393	rdtarga_c__get_8bit_gray_row	5	6	2	3	3	0
394	rdtarga_c__get_8bit_row	5	6	2	3	3	0
395	rdtarga_c__get_16bit_row	5	6	2	3	3	0
396	rdtarga_c__get_24bit_row	5	6	2	3	3	0
397	rdtarga_c__get_memory_row	3	2	1	1	1	0
398	rdtarga_c__preload_image	9	12	6	8	8	0
399	rdtarga_c__start_input_tga	59	85	17_280	17_281	17_281	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
400	rdtarga_c__finish_input_tga	3	2	1	1	1	0
401	rdtarga_c__jinit_read_targa	3	2	1	1	1	0
402	spec_image_c__spec_view_image	3	2	1	1	1	0
403	spec_image_c__spec_read_original_image	10	12	2	5	5	0
404	spec_image_c__spec_allocate_similar_image	5	5	1	2	2	0
405	spec_image_c__spec_free_image	5	5	2	2	2	0
406	spec_image_c__spec_define_subimage_fp	16	26	11	22	22	0
407	spec_image_c__spec_define_subimage_int	21	34	22	45	45	0
408	spec_image_c__spec_free_subimage	5	5	2	2	2	0
409	spec_image_c__spec_write_image	10	12	2	5	5	0
410	spec_image_c__zero_histogram	5	5	1	2	2	0
411	spec_image_c__print_histogram	5	5	1	2	2	0
412	spec_image_c__spec_difference_images	47	73	3_480	6_514	4_785	26.5
413	spec_image_c__spec_checksum_image	10	15	4	15	10	33.3
414	spec_jmemdst_c__jmd_init_destination	3	2	1	1	1	0
415	spec_jmemdst_c__jmd_empty_output_buffer	3	2	1	2	2	0
416	spec_jmemdst_c__jmd_term_destination	3	2	1	1	1	0
417	spec_jmemdst_c__spec_jpeg_mem_dest	6	7	2	3	3	0
418	spec_jmemsrc_c__jms_init_source	3	2	1	1	1	0
419	spec_jmemsrc_c__jms_fill_input_buffer	3	2	1	2	2	0
420	spec_jmemsrc_c__jms_skip_input_data	3	2	1	1	1	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
421	spec_jmemsrc_c__jms_term_source	3	2	1	1	1	0
422	spec_jmemsrc_c__spec_jpeg_mem_src	5	5	2	2	2	0
423	spec_main_c__compress	5	5	1	2	2	0
424	spec_main_c__decompress	5	6	2	3	3	0
425	spec_main_c__print_parameters	3	2	1	1	1	0
426	spec_main_c__go_execute_compression	7	8	2	3	3	0
427	spec_main_c__go_execute_decompression	5	5	2	2	2	0
428	spec_main_c__go_execute_compression_and_decompression	5	5	2	2	2	0
429	spec_main_c__Usage	3	2	1	1	1	0
430	spec_main_c__parse_args	88	133	66	155	111	28.4
431	spec_main_c__main	3	2	1	2	2	0
432	wrbmp_c__put_pixel_rows	7	10	4	6	6	0
433	wrbmp_c__put_gray_rows	7	10	4	6	6	0
434	wrbmp_c__start_output_bmp	3	2	1	1	1	0
435	wrbmp_c__write_bmp_header	15	20	48	48	48	0
436	wrbmp_c__write_os2_header	13	17	24	24	24	0
437	wrbmp_c__write_colormap	25	38	48	56	56	0
438	wrbmp_c__finish_output_bmp	16	23	40	46	45	2.2
439	wrbmp_c__jinit_write_bmp	13	18	16	17	17	0
440	wrgif_c__flush_packet	7	8	3	3	3	0
441	wrgif_c__output	11	15	9	11	11	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
442	wrgif_c__clear_hash	3	2	1	1	1	0
443	wrgif_c__clear_block	3	2	1	1	1	0
444	wrgif_c__compress_init	3	2	1	1	1	0
445	wrgif_c__compress_byte	21	29	31	35	33	5.7
446	wrgif_c__compress_term	8	10	6	6	6	0
447	wrgif_c__put_word	3	2	1	1	1	0
448	wrgif_c__put_3bytes	3	2	1	1	1	0
449	wrgif_c__emit_header	22	30	40	45	45	0
450	wrgif_c__start_output_gif	6	6	2	2	2	0
451	wrgif_c__put_pixel_rows	5	6	2	3	3	0
452	wrgif_c__finish_output_gif	5	5	2	2	2	0
453	wrgif_c__jinit_write_gif	11	15	20	20	20	0
454	wrppm_c__put_pixel_rows	3	2	1	1	1	0
455	wrppm_c__copy_pixel_rows	5	6	2	3	3	0
456	wrppm_c__put_demapped_rgb	5	6	2	3	3	0
457	wrppm_c__put_demapped_gray	5	6	2	3	3	0
458	wrppm_c__start_output_ppm	10	11	3	3	3	0
459	wrppm_c__finish_output_ppm	5	5	2	2	2	0
460	wrppm_c__jinit_write_ppm	10	12	4	4	4	0
461	wrtarga_c__write_header	13	16	12	12	12	0
462	wrtarga_c__put_pixel_rows	5	6	2	3	3	0

Table 9: SPEC CINT95 — 132.ijpeg

no.	function	nodes	edges	npp	ncp	loncp	rd.
463	wrtarga_c__put_gray_rows	5	6	2	3	3	0
464	wrtarga_c__put_demapped_gray	5	6	2	3	3	0
465	wrtarga_c__start_output_tga	16	22	8	9	9	0
466	wrtarga_c__finish_output_tga	5	5	2	2	2	0
467	wrtarga_c__jinit_write_targa	3	2	1	1	1	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	array_c_afetch	17	23	11	11	11	0
2	array_c_astore	24	35	55	57	57	0
3	array_c_anew	3	2	1	1	1	0
4	array_c_afake	11	15	12	14	14	0
5	array_c_aclear	11	16	7	8	8	0
6	array_c_afree	10	14	7	8	8	0
7	array_c_apush	3	2	1	1	1	0
8	array_c_apop	6	6	2	2	2	0
9	array_c_aunshift	11	16	5	7	7	0
10	array_c_ashift	6	6	2	2	2	0
11	array_c_alen	3	2	1	1	1	0
12	array_c_afill	6	6	2	2	2	0
13	cmd_c_cmd_exec	402	635	317	3.44415×10^{12}	3.44194×10^{12}	0.1
14	cmd_c_copyopt	3	2	1	1	1	0
15	cmd_c_saveary	7	8	4	4	4	0
16	cmd_c_savehash	7	8	4	4	4	0
17	cmd_c_saveitem	3	2	1	1	1	0
18	cmd_c_saveint	5	5	2	2	2	0
19	cmd_c_savelong	5	5	2	2	2	0
20	cmd_c_savesptr	5	5	2	2	2	0
21	cmd_c_savenostab	3	2	1	1	1	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	cmd_c__savehptr	5	5	2	2	2	0
23	cmd_c__saveaptr	5	5	2	2	2	0
24	cmd_c__savelist	5	6	2	3	3	0
25	cmd_c__restorelist	25	36	22	32	32	0
26	consarg_c__make_split	13	17	18	18	18	0
27	consarg_c__mod_match	28	40	78	78	78	0
28	consarg_c__make_op	46	76	49_500	49_504	49_504	0
29	consarg_c__evalstatic	382	619	8_018	8_018	8_018	0
30	consarg_c__l	120	188	15_685	15_731	15_708	0.1
31	consarg_c__fixl	9	13	8	8	8	0
32	consarg_c__dehoist	5	5	2	2	2	0
33	consarg_c__addflags	3	2	1	1	1	0
34	consarg_c__hide_ary	6	7	3	3	3	0
35	consarg_c__jmaybe	6	7	3	3	3	0
36	consarg_c__make_list	20	27	16	18	18	0
37	consarg_c__listish	6	7	3	3	3	0
38	consarg_c__maybelistish	15	23	22	22	22	0
39	consarg_c__localize	3	2	1	1	1	0
40	consarg_c__rcatmaybe	8	11	5	5	5	0
41	consarg_c__stab2arg	3	2	1	1	1	0
42	consarg_c__cval_to_arg	3	2	1	1	1	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	consarg_c__op_new	3	2	1	1	1	0
44	consarg_c__free_arg	3	2	1	1	1	0
45	consarg_c__make_match	7	8	3	3	3	0
46	consarg_c__cmd_to_arg	3	2	1	1	1	0
47	consarg_c__nothing_in_common	5	5	2	2	2	0
48	consarg_c__arg_common	43	71	150	266	173	35
49	consarg_c__spat_common	8	11	7	7	7	0
50	cons_c__make_sub	16	22	40	40	40	0
51	cons_c__make_usub	12	16	8	8	8	0
52	cons_c__make_form	12	17	10	18	18	0
53	cons_c__block_head	53	78	35_572	47_428	47_428	0
54	cons_c__make_cswitch	26	40	72	97	91	6.2
55	cons_c__make_nswitch	45	72	2_354	2_442	2_414	1.1
56	cons_c__append_line	15	22	21	23	23	0
57	cons_c__dodb	10	13	8	8	8	0
58	cons_c__make_acmd	10	12	8	8	8	0
59	cons_c__make_ccmd	11	14	12	12	12	0
60	cons_c__make_icmd	24	36	432	439	437	0.5
61	cons_c__opt_arg	157	260	13_480_364	13_480_372	13_480_368	0
62	cons_c__add_label	5	5	2	2	2	0
63	cons_c__addcond	3	2	1	1	1	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	cons_c__addloop	14	21	28	28	28	0
65	cons_c__invert	7	8	4	4	4	0
66	cons_c__cpy7bit	5	6	2	3	3	0
67	cons_c__yyerror	35	55	804	815	813	0.2
68	cons_c__while_io	21	31	27	27	27	0
69	cons_c__wopt	58	94	24_195	24_338	24_332	0
70	cons_c__over	3	2	1	1	1	0
71	cons_c__cmd_free	29	44	651	759	759	0
72	cons_c__arg_free	22	35	14	26	26	0
73	cons_c__spat_free	27	43	305	327	324	0.9
74	cons_c__cmd_tosave	30	51	151	201	201	0
75	cons_c__arg_tosave	22	32	90	95	95	0
76	cons_c__spat_tosave	7	8	4	4	4	0
77	doarg_c__do_subst	124	189	48_582_270	48_582_307	48_582_307	0
78	doarg_c__do_trans	28	42	120	128	128	0
79	doarg_c__do_join	28	41	448	452	452	0
80	doarg_c__do_pack	253	391	245_336_590	245_414_329	245_350_230	0
81	doarg_c__doencodes	9	13	6	9	9	0
82	doarg_c__do_sprintf	84	131	48	25_415	25_413	0
83	doarg_c__do_push	7	9	3	5	5	0
84	doarg_c__do_unshift	5	6	2	3	3	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	doarg_c__do_subr	34	49	2_016	2_017	2_017	0
86	doarg_c__do_assign	75	114	58_400	58_585	58_566	0
87	doarg_c__do_study	29	41	244	249	249	0
88	doarg_c__do_defined	30	45	60	60	60	0
89	doarg_c__do_undef	30	42	52	53	53	0
90	doarg_c__do_vec	36	51	1_328	1_328	1_328	0
91	doarg_c__do_vecset	14	18	10	10	10	0
92	doarg_c__do_chop	26	37	15	17	17	0
93	doarg_c__do_vop	38	56	1_024	1_027	1_027	0
94	doarg_c__do_syscall	3	2	1	1	1	0
95	doio_c__do_open	122	194	2_014_942	2_016_641	2_016_558	0
96	doio_c__nextargv	47	66	150	710	630	11.3
97	doio_c__do_close	24	34	31	31	31	0
98	doio_c__do_eof	16	23	13	15	14	6.7
99	doio_c__do_tell	10	13	7	7	7	0
100	doio_c__do_seek	10	13	7	7	7	0
101	doio_c__do_ctl	24	34	28	28	28	0
102	doio_c__do_stat	34	49	180	180	180	0
103	doio_c__do_truncate	5	5	2	2	2	0
104	doio_c__looks_like_number	36	62	139	393	366	6.9
105	doio_c__do_print	32	51	400	400	400	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	doio_c_do_aprint	23	36	36	42	39	7.1
107	doio_c_mystat	25	35	31	31	31	0
108	doio_c_mylstat	22	30	50	50	50	0
109	doio_c_do_fttext	50	76	232	250	241	3.6
110	doio_c_do_aexec	20	27	21	24	24	0
111	doio_c_do_execfree	7	8	4	4	4	0
112	doio_c_do_exec	32	55	282	456	337	26.1
113	doio_c_do_gpwent	3	2	1	1	1	0
114	doio_c_do_ggrent	3	2	1	1	1	0
115	doio_c_do_dirop	3	2	1	1	1	0
116	doio_c_apply	47	71	51	75	75	0
117	doio_c_cando	21	29	32	32	32	0
118	doio_c_ingroup	9	10	4	4	4	0
119	dolist_c_do_match	139	219	7_389_891_690	7_390_257_297	7_389_943_917	0
120	dolist_c_do_split	142	242	650_927_520	651_856_143	651_855_984	0
121	dolist_c_do_unpack	264	438	8.27261×10^{11}	8.27264×10^{11}	8.27262×10^{11}	0
122	dolist_c_do_slice	50	74	209	217	217	0
123	dolist_c_do_splice	99	155	1_496_880	1_496_897	1_496_890	0
124	dolist_c_do_grep	24	36	102	118	118	0
125	dolist_c_do_reverse	7	9	3	5	5	0
126	dolist_c_do_sreverse	11	15	12	13	13	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	dolist_c__do_sort	39	54	266	269	269	0
128	dolist_c__sortsub	6	6	2	2	2	0
129	dolist_c__sortcmp	9	11	4	4	4	0
130	dolist_c__do_range	30	44	144	146	146	0
131	dolist_c__do_repeatary	13	18	18	19	19	0
132	dolist_c__do_caller	34	49	486	490	490	0
133	dolist_c__do_tms	8	9	3	3	3	0
134	dolist_c__do_time	9	11	5	5	5	0
135	dolist_c__do_kv	25	39	30	93	86	7.5
136	dolist_c__do_each	13	17	14	14	14	0
137	eval_c__eval	1376	2151	528_025	338_291_296	18_305_971	94.6
138	form_c__form_parseargs	17	25	39	51	51	0
139	form_c__format	218	371	3.39713×10^{33}	9.04017×10^{40}	9.04017×10^{40}	0
140	form_c__countlines	7	9	3	5	5	0
141	form_c__do_write	21	30	39	39	39	0
142	hash_c__hfetch	23	35	92	103	96	6.8
143	hash_c__hstore	25	38	157	164	161	1.8
144	hash_c__hdelete	21	33	37	45	41	8.9
145	hash_c__hsplit	17	25	10	30	21	30
146	hash_c__hnew	6	6	2	2	2	0
147	hash_c__hentfree	5	5	2	2	2	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	hash_c__hentdelayfree	5	5	2	2	2	0
149	hash_c__hclear	6	7	3	3	3	0
150	hash_c__hfreeentries	9	11	3	4	4	0
151	hash_c__hfree	5	5	2	2	2	0
152	hash_c__hiterinit	3	2	1	1	1	0
153	hash_c__hiternext	10	14	10	11	11	0
154	hash_c__hiterkey	3	2	1	1	1	0
155	hash_c__hiterval	3	2	1	1	1	0
156	perl_c__main	195	309	3.01145×10^{19}	6.0229×10^{19}	6.0229×10^{19}	0
157	perl_c__magicalize	6	7	2	3	3	0
158	perl_c__magicname	5	5	2	2	2	0
159	perl_c__incpush	14	21	10	16	15	6.2
160	perl_c__savelines	10	14	6	8	8	0
161	perl_c__do_eval	119	186	1_357_987_968	1_357_987_977	1_357_987_971	0
162	perl_c__do_try	11	14	12	12	12	0
163	perl_c__moreswitches	46	75	27	36	34	5.6
164	perl_c__my_unexec	3	2	1	2	2	0
165	perly_c__yyparse	331	560	248_056_337	531_509_471	529_143_300	0.4
166	regcomp_c__regcomp	101	160	53_258_634	54_324_323	54_324_081	0
167	regcomp_c__reg	32	47	778	783	781	0.3
168	regcomp_c__regbranch	16	24	19	23	21	8.7

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	regcomp_c__regpiece	77	120	32_805	32_826	32_814	0
170	regcomp_c__regatom	94	159	32_499	32_618	32_617	0
171	regcomp_c__regset	7	8	3	3	3	0
172	regcomp_c__regclass	82	133	6_472	6_915	6_896	0.3
173	regcomp_c__regnode	10	12	4	4	4	0
174	regcomp_c__reganode	10	12	4	4	4	0
175	regcomp_c__regc	6	6	2	2	2	0
176	regcomp_c__reginsert	10	14	5	7	7	0
177	regcomp_c__regtail	6	7	2	3	3	0
178	regcomp_c__regoptail	7	9	4	4	4	0
179	regcomp_c__regcurly	16	25	12	24	21	12.5
180	regcomp_c__regfree	11	14	16	16	16	0
181	regexec_c__regexec	228	380	810_914_314	810_925_677	810_925_574	0
182	regexec_c__regtry	10	14	9	10	10	0
183	regexec_c__regmatch	118	211	296	436	366	16.1
184	regexec_c__regrepeat	52	103	98	149	141	5.4
185	regexec_c__regnext	5	5	2	2	2	0
186	stab_c__stab_str	77	132	64	65	65	0
187	stab_c__stab_len	29	47	23	23	23	0
188	stab_c__stabset	251	384	511	520	516	0.8
189	stab_c__whichsig	14	20	11	13	12	7.7

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	stab_c_sighandler	29	41	256	256	256	0
191	stab_c_aadd	5	5	2	2	2	0
192	stab_c_hadd	5	5	2	2	2	0
193	stab_c_fstab	5	5	2	2	2	0
194	stab_c_stabent	64	101	158_760	158_763	158_763	0
195	stab_c_stab_fullname	5	5	2	2	2	0
196	stab_c_stab_efullname	5	5	2	2	2	0
197	stab_c_stio_new	3	2	1	1	1	0
198	stab_c_stab_check	9	13	4	11	9	18.2
199	stab_c_genstab	3	2	1	1	1	0
200	stab_c_stab_clear	10	13	6	6	6	0
201	str_c_str_grow	11	14	7	7	7	0
202	str_c_str_numset	6	7	3	3	3	0
203	str_c_str_2ptr	18	25	10	11	11	0
204	str_c_str_2num	15	20	15	15	15	0
205	str_c_str_sset	34	51	60	60	60	0
206	str_c_str_nset	9	11	5	5	5	0
207	str_c_str_set	9	11	5	5	5	0
208	str_c_str_chop	9	11	4	4	4	0
209	str_c_str_ncat	9	11	5	5	5	0
210	str_c_str_scat	8	10	5	5	5	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	str_c__str_cat	10	13	6	6	6	0
212	str_c__str_append_till	22	34	28	37	37	0
213	str_c__str_new	9	11	6	6	6	0
214	str_c__str_magic	7	9	4	4	4	0
215	str_c__str_insert	34	50	51	57	53	7
216	str_c__str_replace	13	17	17	17	17	0
217	str_c__str_free	19	28	57	57	57	0
218	str_c__str_len	9	11	5	5	5	0
219	str_c__str_eq	20	29	18	18	18	0
220	str_c__str_cmp	24	36	30	30	30	0
221	str_c__str_gets	30	47	433	455	454	0.2
222	str_c__parselist	19	27	168	168	168	0
223	str_c__intrpcompile	153	251	5.37996×10^{54}	1.08461×10^{60}	1.08461×10^{60}	0
224	str_c__interp	48	72	1_321	1_485	1_485	0
225	str_c__ucase	8	10	3	5	5	0
226	str_c__lcase	8	10	3	5	5	0
227	str_c__str_inc	34	54	86	101	96	5
228	str_c__str_dec	10	13	5	5	5	0
229	str_c__str_mortal	11	14	8	8	8	0
230	str_c__str_2mortal	13	18	10	10	10	0
231	str_c__str_make	5	5	2	2	2	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	str_c__str_nmake	3	2	1	1	1	0
233	str_c__str_smake	13	17	10	10	10	0
234	str_c__str_reset	23	37	32	135	86	36.3
235	toke_c__skipspace	7	9	2	5	4	20
236	toke_c__check_uni	14	23	4	29	23	20.7
237	toke_c__yylex	1357	2275	31.9649	*none*	*none*	9.2245×10^{31}
238	toke_c__checkcomma	37	67	1_390	2_343	2_316	1.2
239	toke_c__scanident	45	78	3_968	4_018	4_013	0.1
240	toke_c__scanconst	38	63	2_361	2_478	2_478	0
241	toke_c__scanpat	74	118	30_755	31_387	31_382	0
242	toke_c__scansubst	91	152	331_512_987	331_514_559	331_513_939	0
243	toke_c__hoistmust	22	34	39	39	39	0
244	toke_c__scantrans	48	73	437	456	456	0
245	toke_c__scanstr	261	439	2.76822×10^{14}	2.76822×10^{14}	2.76822×10^{14}	0
246	toke_c__load_format	137	232	3.3199×10^{18}	8.63171×10^{18}	8.63171×10^{18}	0
247	toke_c__set_csh	3	2	1	1	1	0
248	usersub_c__userinit	3	2	1	1	1	0
249	util_c__safemalloc	9	11	4	6	6	0
250	util_c__saferealloc	11	14	8	12	12	0
251	util_c__safefree	5	5	2	2	2	0
252	util_c__cpytill	11	16	6	10	10	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
253	util_c_instr	16	25	9	18	12	33.3
254	util_c_ninstr	16	25	15	23	18	21.7
255	util_c_rninstr	15	23	13	21	16	23.8
256	util_c_fbmcompile	23	36	32	42	42	0
257	util_c_fbminstr	46	75	76	96	83	13.5
258	util_c_screaminstr	32	52	56	100	73	27
259	util_c_savestr	3	2	1	1	1	0
260	util_c_nsavestr	3	2	1	1	1	0
261	util_c_growstr	8	9	3	3	3	0
262	util_c_mess	24	34	192	192	192	0
263	util_c_Myfatal	19	28	6	26	24	7.7
264	util_c_warn	3	2	1	1	1	0
265	util_c_my_setenv	17	25	20	23	23	0
266	util_c_envix	8	11	4	6	6	0
267	util_c_my_bcopy	9	13	4	6	6	0
268	util_c_mypopen	3	2	1	1	1	0
269	util_c_dup2	10	15	5	10	9	10
270	util_c_mypclose	3	2	1	1	1	0
271	util_c_pidgone	3	2	1	1	1	0
272	util_c_repeatcpy	11	17	5	10	9	10
273	util_c_castulong	8	9	3	3	3	0

Table 10: SPEC CINT95 — 134.perl

no.	function	nodes	edges	npp	ncp	loncp	rd.
274	util_c__same_dirent	22	30	76	76	76	0
275	util_c__scanoct	9	14	6	7	7	0
276	util_c__scanhex	8	11	3	6	5	16.7

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	bitvec_c__BitVec_Create	13	17	18	18	18	0
2	bitvec_c__BitVec_PutBit	16	21	36	36	36	0
3	bitvec_c__BitVec_GetBit	12	15	12	12	12	0
4	bmt01_c_BMT_CreateParts	90	151	61_901_840	61_939_982	61_939_898	0
5	bmt01_c_BMT_Validate	97	158	74_656_068	74_656_073	74_656_073	0
6	bmt01_c_BMT_ValidateNamedDrawObjs	18	26	49	73	61	16.4
7	bmt01_c_BMT_QueryOn	55	80	902	902	902	0
8	bmt01_c_BMT_DbTransaction	20	27	42	42	42	0
9	bmt01_c_BMT_CommitPartDrawObj	28	42	202	202	202	0
10	bmt01_c_BMT_DeletePartDrawObj	47	74	1_096	1_096	1_096	0
11	bmt01_c_BMT_TraverseSets	24	37	250	254	253	0.4
12	bmt01_c_BMT_Iter1	11	13	8	8	8	0
13	bmt01_c_BMT_Iter2	22	31	108	108	108	0
14	bmt01_c_BMT_Iter3	19	26	80	80	80	0
15	bmt01_c_BMT_Iter4	18	24	60	60	60	0
16	bmt01_c_BMT_ExportPart	9	11	6	6	6	0
17	bmt01_c_BMT_ImportPart	9	11	6	6	6	0
18	bmt01_c_BMT_BuildPersonLib	26	41	48	78	69	11.5
19	bmt01_c_BMT_InitQuerys	97	183	2_636_102	2_636_102	2_636_102	0
20	bmt0_c_main	65	107	97	261	199	23.8
21	bmt10_c_BMT_LookUpParts	24	40	400	499	499	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	bmt10_c_BMT_CommitParts	60	96	2_351_112	2_351_624	2_351_624	0
23	bmt10_c_BMT_DeleteParts	22	35	264	296	296	0
24	bmt10_c_BMT_DeleteAllObjects	48	79	417_314	417_485	417_480	0
25	bmt10_c_BMT_DeleteTestObjs	35	53	2_382	2_425	2_425	0
26	bmt_c_BMT_Test	147	249	2.96341×10^{15}	2.96444×10^{15}	2.96444×10^{15}	0
27	bmt_c_BMT_Init	32	47	4_752	4_752	4_752	0
28	bmt_c_BMT_ConfigAllClasses	51	79	107_496	107_496	107_496	0
29	bmtlib_c_PartLib_Init	20	31	74	74	74	0
30	bmtlib_c_PartLib_Create	45	74	5_784	5_784	5_784	0
31	bmtlib_c_PartLib_ActivateDb	24	38	540	540	540	0
32	bmtlib_c_PartLib_Topology	3	2	1	1	1	0
33	bmtobj_c_DrawPartPair_InitClass	12	16	16	16	16	0
34	bmtobj_c_Part_InitClass	75	126	17_203_352	17_203_352	17_203_352	0
35	bmtobj_c_Part_Create	17	25	117	117	117	0
36	bmtobj_c_Part_GetToken	8	9	4	4	4	0
37	bmtobj_c_Part_Connect	28	43	1_446	1_574	1_570	0.3
38	bmtobj_c_Part_Traverse	35	54	3_220	3_274	3_238	1.1
39	bmtobj_c_Part_Reverse	43	66	11_904	12_000	11_936	0.5
40	bmtobj_c_Part_ExportTo	6	6	2	2	2	0
41	bmtobj_c_Part_ImportFrom	6	6	2	2	2	0
42	bmtobj_c_Part_DisConnect	44	73	43_944	44_865	44_862	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	bmtobj_c__Part_Delete	79	132	543_096_000	543_096_411	543_096_409	0
44	bmtobj_c__Part_Topology	3	2	1	1	1	0
45	core01_c_Core_SetPageSize	3	2	1	1	1	0
46	core01_c_Void_ExtendCore	40	58	1_307	1_308	1_308	0
47	core01_c_Void_FreeCore	13	17	18	18	18	0
48	core01_c_Void_SysFarCoreLeft	22	33	21	27	26	3.7
49	core01_c_Core_MoreCore	13	17	20	20	20	0
50	core01_c_Core_FreeCoreSpace	11	15	14	14	14	0
51	core01_c_Core0_MoreCore	14	19	28	28	28	0
52	core01_c_Core0_FreeCoreSpace	11	15	14	14	14	0
53	core01_c_Core0_AllocString	11	15	14	14	14	0
54	core01_c_Core0_FreeString	11	15	14	14	14	0
55	core01_c_Core_AssignChunks	3	2	1	1	1	0
56	core01_c_Core_MoveBytes	14	19	36	36	36	0
57	core01_c_Core_ShowStats	7	9	5	5	5	0
58	core01_c_Ut_MoveBytes	18	24	13	13	13	0
59	core01_c_Ut_CompareBytes	12	15	12	12	12	0
60	dba_c_DbmCreateDb	158	286	5.77957×10^{14}	5.77957×10^{14}	5.77957×10^{14}	0
61	dba_c_Dbm_DeCacheDbs	34	57	1_836	1_884	1_880	0.2
62	dbm0_c_Dbm_CommitDb	3	2	1	1	1	0
63	dbm0_c_Dbm_SetDefaults	3	2	1	1	1	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	dbm0_c__Dbm_GetDefaults	3	2	1	1	1	0
65	dbm1_c__Dbm_LoadDb	29	44	172	192	176	8.3
66	dbm1_c__Dbm_LoadDbHdr	67	110	9_147_102	9_147_102	9_147_102	0
67	dbm1_c__Dbm_FileInBlkHdr	42	73	52_675	52_675	52_675	0
68	dbm1_c__Dbm_FileInDbHdr	101	180	1.9071×10^{11}	1.9071×10^{11}	1.9071×10^{11}	0
69	dbm2_c__Dbm_FreeDb	3	2	1	1	1	0
70	domain_c__Domain_ImplodeMemory	9	10	4	4	4	0
71	domain_c__Domain_BeHereNow	16	22	24	24	24	0
72	domain_c__Domain_MmiInvoke	13	16	2	5	5	0
73	domain_c__Domain_Omi0Invoke	6	6	2	2	2	0
74	domain_c__Domain_OmiInvoke	20	27	10	13	13	0
75	domain_c__Domain_Enter	18	25	28	28	28	0
76	domain_c__Domain_Exit	17	23	48	49	49	0
77	draw07_c__Draw7_Init	10	12	8	8	8	0
78	draw07_c__TestObj_InitClass	66	121	18_712	18_712	18_712	0
79	draw07_c__TestObj_new0	14	20	42	42	42	0
80	draw07_c__TestObj_new1	15	22	56	56	56	0
81	draw07_c__TestObj_show	5	5	2	2	2	0
82	draw07_c__TestObj_Topology	3	2	1	1	1	0
83	draw07_c__TestObj_delete	20	28	80	80	80	0
84	draw7_c__Draw7	217	386	2.56338×10^{21}	2.56338×10^{21}	2.56338×10^{21}	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	drawlib_c__DrawLib_InitLibrary	19	30	42	42	42	0
86	drawlib_c__DrawLib_InitClass	18	27	75	75	75	0
87	drawlib_c__DrawLib_InitDb	13	17	11	11	11	0
88	drawlib_c__DrawLib_QueryDb	7	9	5	5	5	0
89	drawlib_c__DrawLib_CreateDb	39	62	28_782	28_782	28_782	0
90	drawlib_c__DrawLib_ActivateDbByName	22	34	261	261	261	0
91	drawlib_c__DrawLib_ActivateDbByToken	23	35	360	360	360	0
92	drawlib_c__DrawLib_DeleteDb	7	9	5	5	5	0
93	drawlib_c__DrawLib_NewId	7	9	5	5	5	0
94	drawobj_c__DrawObj_InitClass	16	24	42	42	42	0
95	drawobj_c__DrawObj_new0	17	25	84	84	84	0
96	drawobj_c__DrawObj_new1	17	25	84	84	84	0
97	drawobj_c__DrawObj_new2	13	18	28	28	28	0
98	drawobj_c__DrawObj_import	11	15	14	14	14	0
99	drawobj_c__DrawObj_export	11	15	14	14	14	0
100	drawobj_c__DrawObj_draw	9	10	3	3	3	0
101	drawobj_c__DrawObj_area	9	10	3	3	3	0
102	drawobj_c__DrawObj_delete	23	32	81	81	81	0
103	drawobj_c__DrawObj_Create0	15	21	56	56	56	0
104	drawobj_c__DrawObj_Create1	14	20	42	42	42	0
105	drawobj_c__DrawObj_Import	11	15	14	14	14	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	drawobj_c_DrawObj_Export	11	15	14	14	14	0
107	drawobj_c_DrawObjs_InitClass	12	16	16	16	16	0
108	drawobj_c_OwnerOfDrawObjs	6	6	2	2	2	0
109	drawobj_c_DrawObjs_AddInto	6	6	2	2	2	0
110	drawobj_c_DrawObjs_FindIn	6	6	2	2	2	0
111	drawobj_c_DrawObjs_IterateOn	15	19	16	16	16	0
112	drawobj_c_NamedDrawObj_InitClass	33	52	1_548	1_548	1_548	0
113	drawobj_c_NamedDrawObj_new0	19	28	140	140	140	0
114	drawobj_c_NamedDrawObj_new1	16	23	56	56	56	0
115	drawobj_c_NamedDrawObj_new2	13	18	28	28	28	0
116	drawobj_c_NamedDrawObj_Create0	19	28	210	210	210	0
117	drawobj_c_NamedDrawObj_Create1	19	29	154	154	154	0
118	drawobj_c_NamedDrawObj_ExportFunc	13	18	24	24	24	0
119	drawobj_c_NamedDrawObj_delete	6	6	2	2	2	0
120	drawobj_c_NamedDrawObjs_InitClass	12	16	16	16	16	0
121	drawobj_c_NamedDrawObjs_AddInto	6	6	2	2	2	0
122	drawobj_c_NamedDrawObjs_FindIn	6	6	2	2	2	0
123	drawobj_c_NamedDrawObjs_IterateOn	8	9	4	4	4	0
124	emplib_c_PersonLib_InitLibrary	10	13	8	8	8	0
125	emplib_c_PersonLib_InitClass	18	27	66	66	66	0
126	emplib_c_Person_InitDb	13	17	11	11	11	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
127	emplib_c__PersonLib_QueryDb	7	9	5	5	5	0
128	emplib_c__PersonLib_CreateDb	28	43	972	972	972	0
129	emplib_c__PersonLib_ActivateDbByName	17	25	63	63	63	0
130	emplib_c__PersonLib_ActivateDbByToken	16	22	50	50	50	0
131	emplib_c__PersonLib_DeleteDb	10	13	10	10	10	0
132	emplib_c__PersonObjs_InitClass	12	16	16	16	16	0
133	emplib_c__PersonObjs_AddInto	6	6	2	2	2	0
134	emplib_c__PersonObjs_FindIn	6	6	2	2	2	0
135	emplib_c__PersonObjs_IterateOn	8	9	4	4	4	0
136	emplib_c__PersonNames_InitClass	12	16	16	16	16	0
137	emplib_c__PersonNames_AddInto	6	6	2	2	2	0
138	emplib_c__PersonNames_FindIn	6	6	2	2	2	0
139	emplib_c__PersonNames_IterateOn	8	9	4	4	4	0
140	empobj_c__Address_InitClass	23	38	78	78	78	0
141	empobj_c__Person_InitClass	73	121	10_223_642	10_223_642	10_223_642	0
142	empobj_c__Person_new0	15	22	63	63	63	0
143	empobj_c__Person_new1	15	22	63	63	63	0
144	empobj_c__Person_new2	11	15	14	14	14	0
145	empobj_c__Person_Create0	15	22	63	63	63	0
146	empobj_c__Person_Create1	15	22	63	63	63	0
147	empobj_c__Person_Import	74	117	5_099_136	5_099_700	5_099_412	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
148	empobj_c__Person_Export	11	15	14	14	14	0
149	empobj_c__Person_delete	20	28	80	80	80	0
150	empobj_c__PersonParts_InitClass	12	16	16	16	16	0
151	empobj_c__PersonParts_OwnerOf	6	6	2	2	2	0
152	empobj_c__PersonParts_AddInto	6	6	2	2	2	0
153	empobj_c__PersonParts_FindIn	6	6	2	2	2	0
154	empobj_c__PersonParts_IterateOn	15	19	16	16	16	0
155	env01_c__Env_GetTypeId	21	31	180	180	180	0
156	env01_c__Env_GetClassId	21	31	180	180	180	0
157	env01_c__Env_GetAttrId	21	31	180	180	180	0
158	env01_c__Env_NewFieldStruc	10	13	8	8	8	0
159	env01_c__Env_GetFieldStruc	16	23	33	33	33	0
160	env01_c__Env_DeleteFieldStruc	8	9	4	4	4	0
161	env01_c__Env_GetSetId	14	19	36	36	36	0
162	env01_c__Env_GetTupleId	14	19	36	36	36	0
163	env01_c__Env_GetMatrixId	14	19	36	36	36	0
164	env01_c__Env_LoadCreateCode	9	11	6	6	6	0
165	env01_c__Env_LoadGetTknCode	9	11	6	6	6	0
166	env01_c__Env_InitClassMap	9	11	6	6	6	0
167	env01_c__Env_AppendToMap	13	18	27	27	27	0
168	env01_c__Env_InvokeMap	8	9	4	4	4	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
169	env01_c__Env_GetClassMap	9	11	6	6	6	0
170	env01_c__Env_GetAttrInfo	12	16	18	18	18	0
171	env01_c__Env_IsValidToken	11	14	12	12	12	0
172	env01_c__Env_TokenIsEquiv	7	8	3	3	3	0
173	env01_c__Env_ReclaimHandles	11	14	12	12	12	0
174	env01_c__Env_GenerateRandomNumbers	24	36	71	73	72	1.4
175	env01_c__Env_Random	8	9	3	3	3	0
176	env0_c__VORTEx	19	28	80	80	80	0
177	env0_c__CreateImage	12	16	15	15	15	0
178	env0_c__ImplodeTheOry	8	9	4	4	4	0
179	env0_c__CreateKernel	10	12	6	6	6	0
180	env0_c__LoadKernel	21	33	107	107	107	0
181	env0_c__LoadEnv0	8	9	4	4	4	0
182	env0_c__LoadObj0	6	6	2	2	2	0
183	env1_c__EnvNewObjDesc	3	2	1	1	1	0
184	env1_c__EnvFetchObjNum	8	9	4	4	4	0
185	env1_c__EnvFetchObjHandle	6	6	2	2	2	0
186	env1_c__EnvPairObjHandle	6	6	2	2	2	0
187	env1_c__EnvFetchObjSize	14	18	24	24	24	0
188	env1_c__EnvFetchObjName	8	9	4	4	4	0
189	env1_c__EnvFetchClassSize	11	14	10	10	10	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
190	env1_c_EnvFetchLastObjCount	6	6	2	2	2	0
191	env1_c_EnvFetchObjCestrs	9	11	6	6	6	0
192	env1_c_Env_FetchObjCendents	9	11	6	6	6	0
193	env1_c_EnvFetchSchemaObjs	9	11	6	6	6	0
194	env1_c_DeleteObjDesc	6	6	2	2	2	0
195	env1_c_EnvBldObjHdr	25	35	148	148	148	0
196	env1_c_EnvInstallObjHdr	37	57	4_664	4_664	4_664	0
197	env1_c_InstallObjChunks	131	224	2.98598×10^{11}	3.01397×10^{11}	2.99531×10^{11}	0.6
198	env1_c_EnvBldObjImage	47	81	85_140	85_908	85_396	0.6
199	env1_c_EnvNewObjHdr	16	21	20	20	20	0
200	env1_c_EnvInstallAttr	13	19	14	14	14	0
201	env1_c_EnvFetchLocalAttrNum	14	18	15	15	15	0
202	env1_c_EnvFetchAttrHandle	6	6	2	2	2	0
203	env1_c_EnvFetchAttrSize	6	6	2	2	2	0
204	env1_c_CppEnvGetThatTkn	26	40	164	170	166	2.4
205	env1_c_EnvFetchAttrOffset	16	23	28	28	28	0
206	env1_c_Env_FetchObj0AttrOffset	26	38	116	122	118	3.3
207	env1_c_Env_FetchObjAttrOffset	27	41	408	412	410	0.5
208	env1_c_Env_FetchObjAttrSpec	35	56	1_248	1_266	1_257	0.7
209	env1_c_Env_FetchFieldOffset	20	31	44	44	44	0
210	env1_c_Env_FetchObjFieldSpec	54	89	113_457	165_719	123_901	25.2

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
211	env1_c__EnvFetchOffsetMap	19	28	52	52	52	0
212	env1_c__EnvInitCodes	17	24	16	16	16	0
213	env1_c__EnvMakeCodeChunks	8	9	4	4	4	0
214	env1_c__EnvPairCode	6	6	2	2	2	0
215	env1_c__EnvFetchCode	6	6	2	2	2	0
216	env1_c__EnvAlignMember	34	57	2_295	2_295	2_295	0
217	env1_c__EnvAlignStruc	16	23	30	30	30	0
218	env1_c__TestEnv0	6	6	2	2	2	0
219	env1_c__CreateObjDesc	22	32	85	85	85	0
220	fm_c__File_GetSomeBytes	12	15	16	16	16	0
221	fm_c__File_InBlk	20	29	94	94	94	0
222	fm_c__File_InChunk	20	30	105	105	105	0
223	fm_c__File_GetZeroedChunk	19	27	125	125	125	0
224	gdbm_c__DbmLoadGrpHdr	36	58	5_262	5_262	5_262	0
225	gdbm_c__DbmFileInGrpHdr	48	84	9_606	9_606	9_606	0
226	gdbm_c__DbmFileInGrpRgnChunk	28	46	389	392	390	0.5
227	gdbm_c__DbmDeleteGrpHdr	6	6	2	2	2	0
228	grp0_c__Grp_MakeHdrChunks	29	50	84	84	84	0
229	grp0_c__Grp_GetPacket	12	16	14	14	14	0
230	grp0_c__Grp_NewRegion	57	95	693_090	693_090	693_090	0
231	grp0_c__Grp_GetRegion	38	59	1_632	1_632	1_632	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
232	grp0_c__Grp_CleanRgnChunks	34	56	606	610	608	0.3
233	grp0_c__Grp_DeleteRgn	38	61	5_544	5_548	5_548	0
234	grp0_c__Grp_RgnSwapTopAndPop	45	75	22_846	22_859	22_859	0
235	grp1_c__Grp_NewPacket	35	54	3_392	3_392	3_392	0
236	grp1_c__Grp_SetPacketCache	17	24	34	34	34	0
237	grp1_c__Grp_SetPacketAccess	17	24	34	34	34	0
238	grp1_c__Grp_FreezePacket	17	24	34	34	34	0
239	grp1_c__Grp_ThawPacket	13	17	14	14	14	0
240	grp1_c__Grp_DefrostPacket	13	17	14	14	14	0
241	grp1_c__Grp_FreePacket	37	62	484	508	490	3.5
242	grp1_c__Grp_CommitPacket	9	11	6	6	6	0
243	grp1_c__Grp_DeletePacket	17	25	36	36	36	0
244	grp1_c__Grp_ShowStats	10	13	7	7	7	0
245	grp1_c__Grp_DumpStruct	6	6	2	2	2	0
246	grp1_c__Grp_DumpEntrys	6	6	2	2	2	0
247	grp2_c__Grp_EntryCount	21	32	480	480	480	0
248	grp2_c__Grp_CreateEntry	54	80	35_840	35_840	35_840	0
249	grp2_c__Grp_NewEntry	55	82	53_760	53_760	53_760	0
250	grp2_c__Grp_PutNewEntry	62	92	100_800	100_800	100_800	0
251	grp2_c__Grp_PutEntry	44	65	2_135	2_135	2_135	0
252	grp2_c__Grp_GetBaseRegion	14	19	18	18	18	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
253	grp2_c_Grp_GetEntry	26	36	145	145	145	0
254	grp2_c_Grp_GetInLineEntry	20	28	65	65	65	0
255	grp2_c_Grp_GetFrozenEntry	21	30	95	95	95	0
256	grp2_c_Grp_FreezeEntry	15	21	35	35	35	0
257	grp2_c_Grp_DefrostEntry	18	25	80	80	80	0
258	grp2_c_Grp_EntryIsFrosted	17	24	60	60	60	0
259	grp2_c_Grp_FrostStatus	15	22	35	39	37	5.1
260	grp2_c_Grp_DeleteEntry	33	50	590	590	590	0
261	hm_c_Hm_SetDRIswi	3	2	1	1	1	0
262	hm_c_Hm_DRIswi	3	2	1	1	1	0
263	hm_c_Hm_SetDefaults	3	2	1	1	1	0
264	hm_c_Hm_GetDefaults	3	2	1	1	1	0
265	hm_c_Hm_MakeDbHdr	84	149	1_802_243_520	1_802_243_520	1_802_243_520	0
266	hm_c_Hm_GetObjNum	8	9	4	4	4	0
267	hm_c_Hm_IncrementMemRef	8	9	4	4	4	0
268	hm_c_Hm_DecrementMemRef	8	9	4	4	4	0
269	hm_c_Hm_NewHandle	21	32	35	35	35	0
270	hm_c_Hm_PairDbObject	11	15	10	10	10	0
271	hm_c_Hm_FetchDbObject	10	13	7	7	7	0
272	hm_c_Hm_ClearObject	10	13	8	8	8	0
273	hm_c_Hm_PointToInnerRealm	8	9	4	4	4	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
274	hm_c__Hm_FreeHandle	10	12	6	6	6	0
275	hm_c__Hm_NextFreeHandle	11	14	8	8	8	0
276	hm_c__HmReclaimHandles	6	6	2	2	2	0
277	hm_c__Hm_DumpDbHdr	6	6	2	2	2	0
278	iam_c__ImagePutAttrValue	35	56	238	238	238	0
279	iam_c__ImageGetAttrValue	69	112	5_787	5_795	5_789	0.1
280	iam_c__ImageCompareAttr	121	197	1_484_676	1_484_684	1_484_678	0
281	iam_c__ImagePutObjTkn	32	51	847	847	847	0
282	iam_c__ImageGetObjTkn	32	52	1_001	1_001	1_001	0
283	iam_c__ImageRefTknPut	20	31	33	33	33	0
284	iam_c__ImageTokenToRef	36	58	363	363	363	0
285	iam_c__ImageNewString	25	39	150	150	150	0
286	iam_c__ImageGetString	44	70	6_825	6_825	6_825	0
287	iam_c__ImageGetAttrUnitSize	13	18	12	12	12	0
288	iam_c__ImageCreateArray	40	64	10_710	10_710	10_710	0
289	iam_c__ImageCreateSubArray	33	54	3_570	3_570	3_570	0
290	iam_c__ImageArrayActivate	39	63	645	651	647	0.6
291	iam_c__ImageArrayDeActivate	36	60	345	351	347	1.1
292	iam_c__ImageArrayGetSize	41	70	5_995	5_995	5_995	0
293	iam_c__ImageSubArrayGetSize	37	62	646	646	646	0
294	iam_c__ImageArrayAssertSize	71	120	1_476_475	1_476_505	1_476_481	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
295	iam_c__ImageSubArrayAssertSize	51	84	11_275	11_275	11_275	0
296	iam_c__ImageArrayDelete	59	98	24_827	24_833	24_829	0
297	iam_c__ImageSubArrayDelete	45	75	5_159	5_159	5_159	0
298	iam_c__ImageCreateVarray	21	33	182	182	182	0
299	iam_c__ImageGetVarray	19	29	294	294	294	0
300	iam_c__ImagePutVarrayStackPtr	19	29	294	294	294	0
301	iam_c__ImageGetVarrayStackPtr	11	15	14	14	14	0
302	ifm_c__ImagePutFieldValue	41	68	6_083	6_083	6_083	0
303	ifm_c__ImageGetFieldValue	67	108	4_824	4_832	4_826	0.1
304	im_c__ImageCreateDb	9	11	5	5	5	0
305	im_c__ImageConfigClass	6	6	2	2	2	0
306	im_c__ImageDbCreate	6	6	2	2	2	0
307	im_c__ImageNewObject	10	13	8	8	8	0
308	im_c__ImageGetObject	13	18	16	16	16	0
309	im_c__ImageCreateObject	14	20	40	40	40	0
310	im_c__Image_IsActive	12	16	12	12	12	0
311	im_c__ImageCommitObject	16	21	24	24	24	0
312	im_c__ImageDeleteObject	6	6	2	2	2	0
313	im_c__ImageDumpObject	3	2	1	1	1	0
314	im_c__ImageDumpPseudo	3	2	1	1	1	0
315	im_c__ImageInitClassMap	12	17	11	11	11	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
316	im_c__ImageAppendToMap	8	9	4	4	4	0
317	im_c__ImageInvokeMap	14	18	10	10	10	0
318	im_c__ImageGetClassMap	9	10	4	4	4	0
319	im_c__ImageFreeClassById	12	14	6	6	6	0
320	im_c__ImageFreeClass	15	20	18	18	18	0
321	im_c__ImageCommitClass	12	14	6	6	6	0
322	im_c__Image_GetClassObjectCount	6	6	2	2	2	0
323	im_c__ImageFaxToThis	12	14	6	6	6	0
324	im_c__ImageAssertToThis	18	26	38	38	38	0
325	im_c__ImageGetActiveObject	22	33	28	28	28	0
326	im_c__ImageReFaxToDb	12	14	6	6	6	0
327	im_c__ImageFreeDbObject	12	14	6	6	6	0
328	im_c__ImageFreeCppObject	6	6	2	2	2	0
329	im_c__ImageRevokeObject	9	10	4	4	4	0
330	km_c_KernelCreateDb	20	30	124	124	124	0
331	km_c_KernelLoadDbHdr	6	6	2	2	2	0
332	km_c_KernelFreezeObjClass	6	6	2	2	2	0
333	km_c_Kernel_GetClassObjectCount	16	25	20	20	20	0
334	km_c_KernelNew	8	9	4	4	4	0
335	km_c_KernelCreateObject	6	6	2	2	2	0
336	km_c_KernelNewObject	8	9	4	4	4	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
337	km_c__KernelCreateArray	6	6	2	2	2	0
338	km_c__KernelCreateVarray	6	6	2	2	2	0
339	km_c__KernelIamA	20	30	180	180	180	0
340	km_c__KernelWhatAmI	10	12	6	6	6	0
341	km_c__KernelGet	6	6	2	2	2	0
342	km_c__KernelGetObject	8	9	4	4	4	0
343	km_c__KernelPutObject	8	9	4	4	4	0
344	km_c__KernelGetAttrInfo	11	15	10	10	10	0
345	km_c__Kernel_GetFieldStruc	39	62	21_759	28_965	24_160	16.6
346	km_c__KernelGetFieldInfo	6	6	2	2	2	0
347	km_c__KernelPutAttr	6	6	2	2	2	0
348	km_c__KernelGetAttr	6	6	2	2	2	0
349	km_c__KernelPutField	6	6	2	2	2	0
350	km_c__KernelGetField	6	6	2	2	2	0
351	km_c__KernelCompare	13	16	5	5	5	0
352	km_c__KernelFieldCompare	6	6	2	2	2	0
353	km_c__KernelCreateBitField	10	13	8	8	8	0
354	km_c__KernelDeleteBitField	6	6	2	2	2	0
355	km_c__KernelPutBit	17	24	90	90	90	0
356	km_c__KernelGetBit	16	23	80	80	80	0
357	km_c__KernelFirstBit	20	30	100	102	102	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
358	km_c_KernelNextBit	13	18	12	14	14	0
359	km_c_KernelOwnerOf	15	21	28	28	28	0
360	km_c_KernelFirstOf	6	6	2	2	2	0
361	km_c_KernelNextOf	6	6	2	2	2	0
362	km_c_KernelAddInto	6	6	2	2	2	0
363	km_c_KernelFindIn	6	6	2	2	2	0
364	list01_c_List01_Init	13	18	18	18	18	0
365	list01_c_List01_Create	9	11	6	6	6	0
366	list01_c_List01_FindListHead	16	23	16	23	20	13
367	list01_c_List01_CreateNode	10	13	8	8	8	0
368	list01_c_Is_List01Head	6	6	2	2	2	0
369	list01_c_Is_List01Node	6	6	2	2	2	0
370	list01_c_Is_List01Member	27	43	295	301	298	1
371	list01_c_List01_GetCount	8	9	4	4	4	0
372	list01_c_List01_Reset	8	9	4	4	4	0
373	list01_c_List01_SetCurrent	8	9	4	4	4	0
374	list01_c_List01_GetCurrent	8	9	4	4	4	0
375	list01_c_List01_IsEmpty	10	12	5	5	5	0
376	list01_c_List01_Append	19	28	63	63	63	0
377	list01_c_List01_FirstIn	23	35	162	162	162	0
378	list01_c_List01_NextIn	27	42	702	702	702	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
379	list01_c__List01_IterateOn	21	33	210	213	213	0
380	mem00_c__Mem_Init	3	2	1	1	1	0
381	mem00_c__Mem_MakeOry	8	9	4	4	4	0
382	mem00_c__Mem_NewOry	27	44	346	355	349	1.7
383	mem00_c__Mem_MakeOryChunks	13	18	27	27	27	0
384	mem00_c__MemMakeKrnlChunk	28	41	819	819	819	0
385	mem00_c__Mem_TestTheOry	6	6	2	2	2	0
386	mem00_c__Mem_NewRegion	6	6	2	2	2	0
387	mem00_c__Mem_InitBlocks	5	5	2	2	2	0
388	mem00_c__Mem_BlockCount	5	5	2	2	2	0
389	mem00_c__Mem_NewChunkChunk	13	17	10	11	11	0
390	mem00_c__Mem_FreeChunkChunk	8	9	4	4	4	0
391	mem00_c__Mem_ExpandChunkTables	11	15	6	9	7	22.2
392	mem00_c__Mem_DumpChunkChunk	3	2	1	1	1	0
393	mem00_c__Chunk_MakeChunk	18	26	108	108	108	0
394	mem00_c__Chunk_NewChunk	12	15	12	12	12	0
395	mem00_c__Chunk_InitChunk	17	21	32	32	32	0
396	mem00_c__Mem_NewChunkBlk	9	11	6	6	6	0
397	mem00_c__Mem_PutChunkStruc	9	10	4	4	4	0
398	mem00_c__Chunk_GrowNumericChunk	6	6	2	2	2	0
399	mem00_c__Chunk_GrowTextChunk	17	26	112	112	112	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
400	mem00_c__Chunk_ExpandChunk	38	53	760	760	760	0
401	mem00_c__Mem_ExpandKrnlChunk	18	25	120	120	120	0
402	mem00_c__ChkPushChunk	13	17	10	10	10	0
403	mem00_c__Chunk_ChkPutChunk	14	18	10	10	10	0
404	mem00_c__Chunk_ChkPopChunk	10	13	7	7	7	0
405	mem00_c__Chunk_ChkGetChunk	17	22	14	14	14	0
406	mem00_c__Mem_ShowStats	3	2	1	1	1	0
407	mem00_c__Mem_DumpStats	8	11	5	7	7	0
408	mem01_c__Mem_MakeChunk	12	16	15	15	15	0
409	mem01_c__Mem_MakeCppChunk	6	6	2	2	2	0
410	mem01_c__Mem_MakeStrChunk	12	16	14	14	14	0
411	mem01_c__Mem_DumpChunk	9	10	4	4	4	0
412	mem01_c__Mem_DumpChunkPart	6	6	2	2	2	0
413	mem01_c__Mem_FreeChunk	10	13	8	8	8	0
414	mem01_c__Mem_FreeChunkNum	8	9	4	4	4	0
415	mem01_c__Mem_ClearChunkSpace	6	6	2	2	2	0
416	mem01_c__Mem_FreeChunkSpace	7	8	3	3	3	0
417	mem01_c__Mem_SwapOutChunk	6	6	2	2	2	0
418	mem01_c__Mem_SwapInChunk	6	6	2	2	2	0
419	mem01_c__Mem_CopyOutChunk	6	6	2	2	2	0
420	mem01_c__Mem_CopyInChunk	6	6	2	2	2	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
421	mem01_c__Mem_MoveChunk	19	28	123	123	123	0
422	mem01_c__Mem_MakeXmemChunks	15	22	28	28	28	0
423	mem01_c__Mem_AssignXmemFile	10	13	5	5	5	0
424	mem01_c__Mem_CloseXmemFile	3	2	1	1	1	0
425	mem01_c__Mem_GetFileBlk	14	20	14	14	14	0
426	mem01_c__Mem_FreeFileBlk	3	2	1	1	1	0
427	mem01_c__Mem_NewXmemBlk	3	2	1	1	1	0
428	mem01_c__Mem_FreeXmemBlk	3	2	1	1	1	0
429	mem01_c__Mem_ShowXmemStats	3	2	1	1	1	0
430	mem10_c__Mem_FreezeChunk	6	6	2	2	2	0
431	mem10_c__Mem_DefrostChunk	6	6	2	2	2	0
432	mem10_c__Mem_GetChunkSize	6	6	2	2	2	0
433	mem10_c__Mem_GetChunkStruc	6	6	2	2	2	0
434	mem10_c__Mem_DumpChunkStruc	6	6	2	2	2	0
435	mem10_c__Mem_PutChunkAddr	10	13	9	9	9	0
436	mem10_c__Mem_GetChunkAddr	10	13	9	9	9	0
437	mem10_c__Mem_PutStackPtr	16	21	21	21	21	0
438	mem10_c__Mem_AssertStackPtr	8	9	4	4	4	0
439	mem10_c__Mem_GetStackPtr	8	9	4	4	4	0
440	mem10_c__Mem_PushEntity	12	15	8	8	8	0
441	mem10_c__Mem_PutEntity	11	13	6	6	6	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
442	mem10_c__Mem_ExtractEntity	10	13	6	7	7	0
443	mem10_c__Mem_GetEntity	13	17	11	11	11	0
444	mem10_c__Mem_NewBitChunk	10	13	8	8	8	0
445	mem10_c__Mem_PushBit	3	2	1	1	1	0
446	mem10_c__Mem_PutBit	12	15	12	12	12	0
447	mem10_c__Mem_PopBit	3	2	1	1	1	0
448	mem10_c__Mem_GetBit	11	15	12	12	12	0
449	mem10_c__Mem_PushShort	9	11	6	6	6	0
450	mem10_c__Mem_PutShort	11	14	8	8	8	0
451	mem10_c__Mem_PopShort	9	11	6	6	6	0
452	mem10_c__Mem_GetShort	10	13	8	8	8	0
453	mem10_c__Mem_PushWord	9	11	6	6	6	0
454	mem10_c__Mem_PutWord	11	14	8	8	8	0
455	mem10_c__Mem_PopWord	9	11	6	6	6	0
456	mem10_c__Mem_GetWord	10	13	8	8	8	0
457	mem10_c__Mem_PushLong	9	11	6	6	6	0
458	mem10_c__Mem_PutLong	11	14	8	8	8	0
459	mem10_c__Mem_PopLong	9	11	6	6	6	0
460	mem10_c__Mem_GetLong	10	13	8	8	8	0
461	mem10_c__Mem_PushAddr	8	9	4	4	4	0
462	mem10_c__Mem_PutAddr	10	12	6	6	6	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
463	mem10_c__Mem_PopAddr	8	9	4	4	4	0
464	mem10_c__Mem_GetAddr	9	11	6	6	6	0
465	mem10_c__Mem_PushToken	8	9	4	4	4	0
466	mem10_c__Mem_PutToken	10	12	6	6	6	0
467	mem10_c__Mem_PopToken	8	9	4	4	4	0
468	mem10_c__Mem_GetToken	9	11	6	6	6	0
469	mem10_c__Mem_PushSomeBytes	8	9	4	4	4	0
470	mem10_c__Mem_PutSomeBytes	10	12	6	6	6	0
471	mem10_c__Mem_PopSomeBytes	8	9	4	4	4	0
472	mem10_c__Mem_GetSomeBytes	9	11	6	6	6	0
473	mem10_c__Mem_PushFuncPtr	8	9	4	4	4	0
474	mem10_c__Mem_PutFuncPtr	10	12	6	6	6	0
475	mem10_c__Mem_PopFuncPtr	8	9	4	4	4	0
476	mem10_c__Mem_GetFuncPtr	9	11	6	6	6	0
477	mem10_c__Mem_NewString	11	13	8	8	8	0
478	mem10_c__Mem_DeleteString	12	17	10	12	12	0
479	mem10_c__Mem_PushString	13	17	14	14	14	0
480	mem10_c__Mem_PutString	18	27	28	30	30	0
481	mem10_c__Mem_PopString	9	11	5	5	5	0
482	mem10_c__Mem_GetString	17	25	20	20	20	0
483	oa0_c_OaCreateDb	9	10	4	4	4	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
484	oa0_c_OaInstallObjHdr	25	40	402	402	402	0
485	oa0_c_OaGetObjClass	13	19	22	22	22	0
486	oa0_c_OaConfigClass	6	6	2	2	2	0
487	oa0_c_OaFreezeObjClass	8	9	4	4	4	0
488	oa0_c_OaThawObjClass	15	22	26	26	26	0
489	oa0_c_OaCreateObject	45	74	73_402	73_402	73_402	0
490	oa0_c_OaNewObject	27	41	875	875	875	0
491	oa0_c_OaGetObject	19	29	126	126	126	0
492	oa0_c_OaRevokeCppObject	10	13	10	10	10	0
493	oa0_c_OaInvokeCppObject	10	13	10	10	10	0
494	oa0_c_OaUpdateObject	27	42	828	828	828	0
495	oa0_c_OaChkImage	23	38	308	308	308	0
496	oa0_c_OaFreeObject	17	26	32	32	32	0
497	oa0_c_OaDeleteObject	15	23	28	28	28	0
498	oa0_c_OaGetObjHandles	11	15	10	10	10	0
499	oa0_c_OaReclaimHandles	32	52	616	629	626	0.5
500	oa1_c_OaIamA	16	23	33	33	33	0
501	oa1_c_ForeignCreate	16	23	38	38	38	0
502	oa1_c_OaInitObject	3	2	1	1	1	0
503	oa1_c_OaCopy	3	2	1	1	1	0
504	oa1_c_OaGetDbObjNums	40	64	3_380	3_800	3_800	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
505	oa1_c_OaDelete	22	35	546	546	546	0
506	oa1_c_OaDeleteFields	149	274	389_650	428_935	428_879	0
507	oa1_c_OaDeleteEmbedded	8	9	4	4	4	0
508	oa1_c_OaCreateArray	55	93	188_092	188_092	188_092	0
509	oa1_c_OaCreateVarray	19	29	200	200	200	0
510	oa1_c_OaPut	86	140	13_349_880	13_349_880	13_349_880	0
511	oa1_c_OaPutString	16	20	12	12	12	0
512	oa1_c_OaPutToEmbedded	60	102	14_493_690	14_493_690	14_493_690	0
513	oa1_c_OaPutField	25	41	1_560	1_560	1_560	0
514	oa1_c_OaGet	61	102	2_107_620	2_107_620	2_107_620	0
515	oa1_c_OaGetString	12	16	15	15	15	0
516	oa1_c_OaGetField	43	74	177_480	177_481	177_481	0
517	oa1_c_OaChkForAttr	25	39	1_260	1_260	1_260	0
518	oa1_c_OaGetAttrSize	13	19	22	22	22	0
519	oa1_c_OaCompare	63	98	63_680	63_680	63_680	0
520	oa1_c_OaCompareField	56	87	21_336	21_336	21_336	0
521	oadmp_c__OaDumpObjHdr	6	6	2	2	2	0
522	oadmp_c__OaDumpObjHndls	6	6	2	2	2	0
523	oadmp_c__OaDumpObject	6	6	2	2	2	0
524	oadmp_c__OaDumpEmbedded	6	6	2	2	2	0
525	obj01_c__Object_Create	14	19	16	16	16	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
526	obj01_c__Object_Dump	7	9	4	4	4	0
527	obj01_c__Object_IsA	6	6	2	2	2	0
528	obj01_c__Object_IsKindOf	9	10	3	3	3	0
529	obj01_c__Object_Delete	14	19	16	16	16	0
530	obj01_c__Object_NewImage	6	6	2	2	2	0
531	obj01_c__Object_GetImage	6	6	2	2	2	0
532	obj01_c__Object_ImageIsActive	9	10	3	3	3	0
533	obj01_c__Object_PutImage	11	14	10	10	10	0
534	obj01_c__Object_FreeImage	6	6	2	2	2	0
535	obj01_c__Object_CommitImage	6	6	2	2	2	0
536	obj01_c__Image_DumpPseudo	3	2	1	1	1	0
537	obj01_c__Attr_ValuePut	6	6	2	2	2	0
538	obj01_c__Attr_ValueGet	6	6	2	2	2	0
539	obj01_c__Attr_ValueCompare	6	6	2	2	2	0
540	obj01_c__Attr_ObjTknPut	6	6	2	2	2	0
541	obj01_c__Attr_ObjTknGet	6	6	2	2	2	0
542	obj01_c__Attr_RefTknPut	6	6	2	2	2	0
543	obj01_c__Attr_TokenToRef	6	6	2	2	2	0
544	obj01_c__Attr_StringCreate	6	6	2	2	2	0
545	obj01_c__Attr_StringGet	6	6	2	2	2	0
546	obj01_c__Attr_StrCpyGet	12	16	10	10	10	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
547	obj01_c__Attr_StrCpyFree	8	9	4	4	4	0
548	obj01_c__Attr_ArrayCreate	6	6	2	2	2	0
549	obj01_c__Attr_SubArrayCreate	6	6	2	2	2	0
550	obj01_c__Attr0_ArrayActivate	12	16	14	14	14	0
551	obj01_c__Attr0_ArrayDeActivate	9	10	4	4	4	0
552	obj01_c__Attr_ArrayGetSize	6	6	2	2	2	0
553	obj01_c__Attr_SubArrayGetSize	6	6	2	2	2	0
554	obj01_c__Attr_ArrayAssertSize	6	6	2	2	2	0
555	obj01_c__Attr_SubArrayAssertSize	6	6	2	2	2	0
556	obj01_c__Attr_ArrayDelete	6	6	2	2	2	0
557	obj01_c__Attr_SubArrayDelete	6	6	2	2	2	0
558	obj01_c__Field_ValuePut	6	6	2	2	2	0
559	obj01_c__Field_ValueGet	6	6	2	2	2	0
560	obj01_c__Owner_SetIterateOn	13	16	16	16	16	0
561	obj01_c__Owner_SetFirstOf	6	6	2	2	2	0
562	obj01_c__Owner_KeySetFindIn	6	6	2	2	2	0
563	obj01_c__Owner_KeySetFindInBy2aryKey	3	2	1	1	1	0
564	obj01_c__Member_SetOwnerOf	6	6	2	2	2	0
565	obj01_c__Member_SetNextOf	6	6	2	2	2	0
566	obj01_c__Member_KeySetAddInto	6	6	2	2	2	0
567	odbm_c__DbmLoadObjHdr	46	75	128_376	128_376	128_376	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
568	odbm_c_DbmFileInObjHdr	60	103	4_928_852	4_928_852	4_928_852	0
569	odbm_c_DbmFileInRegionChunk	3	2	1	1	1	0
570	odbm_c_DbmPairRgnObjects	28	44	758	788	788	0
571	odbm_c_DbmUnPairRgnObjects	3	2	1	1	1	0
572	odbm_c_DbmDumpObjHdr	6	6	2	2	2	0
573	ogrp_c_Grp_NewObject	53	81	49_536	49_536	49_536	0
574	ogrp_c_Grp_GetObject	25	37	236	236	236	0
575	ogrp_c_Grp_DeleteObject	56	84	76_336	76_336	76_336	0
576	om_c_OmSetConfig	3	2	1	1	1	0
577	om_c_OmConfigClass	3	2	1	1	1	0
578	om_c_OmNewObjHdr	63	104	2_947_828	2_947_828	2_947_828	0
579	om_c_OmMakeObjChunks	33	53	1_122	1_122	1_122	0
580	om_c_OmPutHandleOffset	14	20	42	42	42	0
581	om_c_OmGetObjHdr	17	25	30	30	30	0
582	om_c_OmNextAvailObject	11	13	6	6	6	0
583	om_c_OmNewObject	13	19	20	20	20	0
584	om_c_OmNewObjRegion	32	53	947	947	947	0
585	om_c_OmPairDbObject	16	23	84	84	84	0
586	om_c_OmUnPairRgnObjects	25	39	278	288	288	0
587	om_c_OmPairRgnObjects	25	39	278	288	288	0
588	om_c_OmGetRegion	3	2	1	1	1	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
589	om_c__OmGetObject	25	39	784	784	784	0
590	om_c__OmDeleteObject	11	15	14	14	14	0
591	om_c__OmChkImage	23	38	231	231	231	0
592	om_c__OmGetObjHandles	45	75	33_286	35_385	35_375	0
593	point_c__Point_InitClass	10	13	8	8	8	0
594	point_c__Point_GetToken	3	2	1	1	1	0
595	point_c__Point_x	3	2	1	1	1	0
596	point_c__Point_y	3	2	1	1	1	0
597	point_c__Point_Theta	3	2	1	1	1	0
598	point_c__Point_Radius	3	2	1	1	1	0
599	point_c__CartesianPoint_InitClass	24	37	264	264	264	0
600	point_c__CartesianPoint_New	10	14	14	14	14	0
601	point_c__CartesianPoint_new0	13	18	28	28	28	0
602	point_c__CartesianPoint_new1	13	18	28	28	28	0
603	point_c__CartesianPoint_x	3	2	1	1	1	0
604	point_c__CartesianPoint_y	3	2	1	1	1	0
605	point_c__CartesianPoint_Radius	3	2	1	1	1	0
606	point_c__CartesianPoint_Theta	12	15	5	5	5	0
607	point_c__CartesianPoint_show	3	2	1	1	1	0
608	point_c__CartesianPoint_delete	20	28	80	80	80	0
609	point_c__PolarPoint_InitClass	24	37	264	264	264	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
610	point_c__PolarPoint_New	10	14	14	14	14	0
611	point_c__PolarPoint_new0	14	20	42	42	42	0
612	point_c__PolarPoint_new1	13	18	28	28	28	0
613	point_c__PolarPoint_x	3	2	1	1	1	0
614	point_c__PolarPoint_y	3	2	1	1	1	0
615	point_c__PolarPoint_Radius	3	2	1	1	1	0
616	point_c__PolarPoint_Theta	3	2	1	1	1	0
617	point_c__PolarPoint_show	3	2	1	1	1	0
618	point_c__PolarPoint_delete	20	28	80	80	80	0
619	primal_c__Primal_CreateDb	16	21	28	28	28	0
620	primal_c__Primal_ActivateDbByName	3	2	1	1	1	0
621	primal_c__Primal_ActivateDbByToken	3	2	1	1	1	0
622	primal_c__Primal_FreeDb	3	2	1	1	1	0
623	primal_c__Primal_CommitDb	3	2	1	1	1	0
624	primal_c__Primal_DeleteDb	3	2	1	1	1	0
625	primal_c__Primal_ConfigYourClass	6	6	2	2	2	0
626	primal_c__Primal_GetClassObjectCount	6	6	2	2	2	0
627	primal_c__Primal_IterateOnClassObjects	30	47	540	543	543	0
628	primal_c__Primal_FreezeClass	6	6	2	2	2	0
629	primal_c__Primal_FreeClass	6	6	2	2	2	0
630	pstub_c__CppCreateObject	11	15	14	14	14	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
631	pstub_c_CppRefToTkn	8	11	7	7	7	0
632	pstub_c_CppCastObject	11	15	14	14	14	0
633	pstub_c_Image01_GetFreeStoreAddr	3	2	1	1	1	0
634	query_c_Query_Create	21	34	60	60	60	0
635	query_c_Query_BeginBuild	13	19	22	22	22	0
636	query_c_Query_ScopeOn	8	9	4	4	4	0
637	query_c_Query1_ScopeOn	23	36	63	69	66	4.3
638	query_c_Query_SortBy	8	9	4	4	4	0
639	query_c_Query1_SortBy	10	12	6	6	6	0
640	query_c_Query_OpenParen	6	6	2	2	2	0
641	query_c_Query_AffixCompare	15	20	18	18	18	0
642	query_c_Query1_AffixCompare	29	44	148	148	148	0
643	query_c_Query2_AffixCompare	16	22	20	20	20	0
644	query_c_Query_AffixRefQuery	13	17	14	14	14	0
645	query_c_Query1_AffixRefQuery	26	39	200	200	200	0
646	query_c_Query_AffixBoolOp	10	13	10	10	10	0
647	query_c_Query_CloseParen	10	13	10	10	10	0
648	query_c_Query_EndBuild	21	34	100	109	109	0
649	query_c_Query_AssertOnObject	142	238	2.79387×10^{12}	2.87076×10^{12}	2.87076×10^{12}	0
650	query_c_Query_AssertOnDb	38	63	14_688	14_711	14_710	0
651	query_c_Query_CompareValue	66	105	1_518	1_518	1_518	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
652	query_c__Query_CompareWild	48	76	1_254	1_265	1_263	0.2
653	query_c__Query_CheckType	22	35	245	245	245	0
654	query_c__Query_GetClassList	38	62	2_310	2_346	2_317	1.2
655	query_c__Query_Dump	3	2	1	1	1	0
656	query_c__Query_DumpComdObj	3	2	1	1	1	0
657	query_c__Query_DumpSortObj	3	2	1	1	1	0
658	query_c__Query_DumpFieldObj	3	2	1	1	1	0
659	rect_c__Rectangle_InitClass	38	61	5_383	5_383	5_383	0
660	rect_c__Rectangle_new0	17	25	84	84	84	0
661	rect_c__Rectangle_new1	19	28	196	196	196	0
662	rect_c__Rectangle_new2	17	25	84	84	84	0
663	rect_c__Rectangle_new3	19	29	112	112	112	0
664	rect_c__Rectangle_new4	25	36	308	308	308	0
665	rect_c__Rectangle_area	12	16	20	20	20	0
666	rect_c__Rectangle_draw	14	19	24	24	24	0
667	rect_c__Rectangle_show	3	2	1	1	1	0
668	rect_c__Rectangle_error	3	2	1	2	2	0
669	rect_c__Rectangle_delete	20	28	80	80	80	0
670	rect_c__LibRectangles_InitClass	12	16	16	16	16	0
671	rect_c__OwnerOfLibRectangles	6	6	2	2	2	0
672	rect_c__LibRectangles_AddInto	6	6	2	2	2	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
673	rect_c__LibRectangles_FindIn	6	6	2	2	2	0
674	rect_c__LibRectangles_IterateOn	8	9	4	4	4	0
675	rects_c__Rects_InitLibrary	17	26	36	36	36	0
676	rects_c__XyRect_InitClass	30	46	508	508	508	0
677	rects_c__XyRect_new0	16	23	56	56	56	0
678	rects_c__XyRect_new1	13	18	28	28	28	0
679	rects_c__XyRect_area	14	20	36	36	36	0
680	rects_c__XyRect_draw	15	22	40	40	40	0
681	rects_c__XyRect_dump	6	6	2	2	2	0
682	rects_c__XyRect_error	3	2	1	2	2	0
683	rects_c__XyRect_show	3	2	1	1	1	0
684	rects_c__XyRect_delete	20	28	80	80	80	0
685	rects_c__NamedXyRect_InitClass	38	59	14_434	14_434	14_434	0
686	rects_c__NamedXyRect_new0	17	26	90	90	90	0
687	rects_c__NamedXyRect_new1	13	18	28	28	28	0
688	rects_c__NamedXyRect_area	14	20	36	36	36	0
689	rects_c__NamedXyRect_draw	15	22	40	40	40	0
690	rects_c__NamedXyRect_dump	6	6	2	2	2	0
691	rects_c__NamedXyRect_error	3	2	1	2	2	0
692	rects_c__NamedXyRect_show	3	2	1	1	1	0
693	rects_c__NamedXyRect_delete	20	28	80	80	80	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
694	rects_c_ArrayRect_InitClass	25	37	272	272	272	0
695	rects_c_ArrayRect_new0	22	35	336	336	336	0
696	rects_c_ArrayRect_new1	13	18	28	28	28	0
697	rects_c_ArrayRect_area	14	20	36	36	36	0
698	rects_c_ArrayRect_draw	15	22	40	40	40	0
699	rects_c_ArrayRect_dump	6	6	2	2	2	0
700	rects_c_ArrayRect_error	3	2	1	2	2	0
701	rects_c_ArrayRect_show	3	2	1	1	1	0
702	rects_c_ArrayRect_delete	20	28	80	80	80	0
703	rects_c_DblPtrRect_InitClass	22	32	140	140	140	0
704	rects_c_DblPtrRect_new0	29	47	2_079	2_079	2_079	0
705	rects_c_DblPtrRect_new1	13	18	28	28	28	0
706	rects_c_DblPtrRect_area	14	20	36	36	36	0
707	rects_c_DblPtrRect_draw	15	22	40	40	40	0
708	rects_c_DblPtrRect_dump	6	6	2	2	2	0
709	rects_c_DblPtrRect_error	3	2	1	2	2	0
710	rects_c_DblPtrRect_show	3	2	1	1	1	0
711	rects_c_DblPtrRect_delete	20	28	80	80	80	0
712	rects_c_VarrayRect_InitClass	25	37	272	272	272	0
713	rects_c_VarrayRect_new0	28	45	1_848	1_848	1_848	0
714	rects_c_VarrayRect_new1	13	18	28	28	28	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
715	rects_c_VarrayRect_area	14	20	36	36	36	0
716	rects_c_VarrayRect_draw	15	22	40	40	40	0
717	rects_c_VarrayRect_dump	6	6	2	2	2	0
718	rects_c_VarrayRect_error	3	2	1	2	2	0
719	rects_c_VarrayRect_show	3	2	1	1	1	0
720	rects_c_VarrayRect_delete	20	28	80	80	80	0
721	rects_c_IntChunkRect_InitClass	25	37	400	400	400	0
722	rects_c_IntChunkRect_new0	28	45	1_848	1_848	1_848	0
723	rects_c_IntChunkRect_new1	13	18	28	28	28	0
724	rects_c_IntChunkRect_area	14	20	36	36	36	0
725	rects_c_IntChunkRect_draw	15	22	40	40	40	0
726	rects_c_IntChunkRect_dump	6	6	2	2	2	0
727	rects_c_IntChunkRect_error	3	2	1	2	2	0
728	rects_c_IntChunkRect_show	3	2	1	1	1	0
729	rects_c_IntChunkRect_delete	20	28	80	80	80	0
730	rects_c_VchunkRect_InitClass	22	32	140	140	140	0
731	rects_c_VchunkRect_new0	24	38	287	287	287	0
732	rects_c_VchunkRect_new1	13	18	28	28	28	0
733	rects_c_VchunkRect_area	15	22	54	54	54	0
734	rects_c_VchunkRect_draw	16	24	48	48	48	0
735	rects_c_VchunkRect_dump	6	6	2	2	2	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
736	rects_c_VchunkRect_error	3	2	1	2	2	0
737	rects_c_VchunkRect_show	3	2	1	1	1	0
738	rects_c_VchunkRect_delete	20	28	80	80	80	0
739	rects_c_RefRect_InitClass	25	37	272	272	272	0
740	rects_c_RefRect_new0	16	23	56	56	56	0
741	rects_c_RefRect_new1	17	25	70	70	70	0
742	rects_c_RefRect_new2	18	27	84	84	84	0
743	rects_c_RefRect_new3	33	50	2_828	2_828	2_828	0
744	rects_c_RefRect_area	14	20	36	36	36	0
745	rects_c_RefRect_draw	16	23	40	40	40	0
746	rects_c_RefRect_dump	6	6	2	2	2	0
747	rects_c_RefRect_error	3	2	1	2	2	0
748	rects_c_RefRect_show	3	2	1	1	1	0
749	rects_c_RefRect_delete	20	28	80	80	80	0
750	rects_c_PortRect_InitClass	34	52	2_076	2_076	2_076	0
751	sa_c_SetInitSetHeads	14	20	16	19	19	0
752	sa_c_SaAddInto	190	319	4.41213×10^{17}	4.41213×10^{17}	4.41213×10^{17}	0
753	sa_c_SaFindIn	127	210	6_911_883_132	7_577_598_904	7_577_597_248	0
754	sa_c_SaPutRootNode	82	130	763_674	763_675	763_675	0
755	sa_c_SaGetRootNode	72	116	57_570	57_578	57_572	0
756	sa_c_SaDeleteNode	67	114	3_818_276	3_818_280	3_818_280	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
757	sa_c_SaAdjustRootNode	42	66	3_590	3_590	3_590	0
758	sa_c_SaDeleteSetHead	44	73	19_040	19_088	19_052	0.2
759	shell_c_ShellLoadObjCode	6	6	2	2	2	0
760	shell_c_ShellGetObjCode	14	19	40	40	40	0
761	shell_c_ShellLoadTokenCode	19	28	256	256	256	0
762	shell_c_ShellGetTokenCode	16	22	80	80	80	0
763	sm_c_SetInitSet	35	56	8_228	8_228	8_228	0
764	sm_c_SetAddInto	13	16	8	8	8	0
765	sm_c_SetFindIn	13	16	8	8	8	0
766	sm_c_SetIterateOn	19	29	120	123	123	0
767	sm_c_SetDeleteFrom	16	21	20	20	20	0
768	sm_c_SetDeleteSet	3	2	1	1	1	0
769	sm_c_SetFirstOf	50	83	119_004	119_004	119_004	0
770	sm_c_SetNextOf	29	44	2_016	2_016	2_016	0
771	testobj_c_Draw701	442	752	5.461×10^{38}	5.461×10^{38}	5.461×10^{38}	0
772	tm_c_TmNewCoreDb	8	9	4	4	4	0
773	tm_c_TmRenvToken	6	6	2	2	2	0
774	tm_c_TmWenvToken	6	6	2	2	2	0
775	tm_c_TmFetchCoreDb	18	25	70	70	70	0
776	tm_c_TmMakeToken	6	6	2	2	2	0
777	tm_c_TmIsValid	13	18	9	9	9	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
778	tm_c__TmGetObject	11	14	12	12	12	0
779	tm_c__TmFreeToken	11	14	12	12	12	0
780	tm_c__TmReclaimHandles	8	9	4	4	4	0
781	trans00_c_TransInitMap	48	77	20_118	20_118	20_118	0
782	trans00_c_TransAppendToMap	13	17	11	11	11	0
783	trans00_c_TransInvokeMap	69	117	1_573_931	1_575_507	1_574_325	0.1
784	trans00_c_TransBuildMap	45	71	24_478	24_478	24_478	0
785	trans00_c_TransGetMap	12	16	9	9	9	0
786	trans00_c_Trans_DumpMap	6	6	2	2	2	0
787	trans00_c_Trans_MapIsActive	11	14	6	6	6	0
788	trans00_c_TransCreateMapDirs	28	46	316	316	316	0
789	trans00_c_TransNewImageMap	55	98	17_500	17_500	17_500	0
790	trans00_c_TransBuildFields	94	155	6_911_561_712	6_920_115_630	6_920_115_627	0
791	trans00_c_TransBuildMapField	13	17	12	12	12	0
792	trans00_c_Trans_FetchAttrOffset	13	17	16	16	16	0
793	trans00_c_Trans_FetchAttrOffsets	20	29	72	76	74	2.6
794	trans00_c_Trans_FetchFieldOffsets	6	6	2	2	2	0
795	trans00_c_Trans_FetchFieldOffset	9	11	6	6	6	0
796	trans00_c_Trans_FetchObjFieldSpec	41	64	5_570	5_802	5_795	0.1
797	trans01_c_C_GetObjectImage	23	34	254	254	254	0
798	trans01_c_C_ObjectNewImage	24	35	276	276	276	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
799	trans01_c_C_InvokeAtThis	22	33	207	207	207	0
800	trans01_c_C_FaxToThis	51	83	30_746	30_746	30_746	0
801	trans01_c_C_FaxToClassFields	62	102	696	1_947	1_005	48.4
802	trans01_c_C_MapRefToAddr	25	41	150	153	151	1.3
803	trans01_c_C_RefToAddr	24	37	98	98	98	0
804	trans01_c_C_CreateArray	64	102	1_580_481	1_580_489	1_580_483	0
805	trans01_c_C_CreateSubArray	45	74	24_201	24_209	24_203	0
806	trans01_c_C_CreateFieldArray	44	72	30_213	30_213	30_213	0
807	trans01_c_C_CreateFieldSubArray	34	56	2_763	2_763	2_763	0
808	trans01_c_C_CreateObject	10	12	6	6	6	0
809	trans10_c_C_ReFaxToDb	128	215	10_009_752	10_021_920	10_011_490	0.1
810	trans10_c_C_GroupArrays	6	6	2	2	2	0
811	trans10_c_C_MapRefToDb	63	105	6_364	7_053	6_468	8.3
812	trans10_c_C_RefToTkn	22	30	37	37	37	0
813	trans10_c_C_CommitClass	18	28	54	69	59	14.5
814	trans20_c_C_FreeClass	3	2	1	1	1	0
815	trans20_c_C_FreeObject	3	2	1	1	1	0
816	trans20_c_C_DeleteObject	26	41	360	360	360	0
817	tree00_c_Tree_AddInto	64	108	104_079	104_099	104_086	0
818	tree00_c_Tree_FindIn	20	28	126	126	126	0
819	tree00_c_Tree_Traverse	37	59	2_468	2_486	2_474	0.5

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
820	tree00_c_Tree_IterateOn	23	35	92	92	92	0
821	tree00_c_Tree_Validate	48	78	12_744	12_765	12_751	0.1
822	tree00_c_Tree_Delete	24	37	1_170	1_170	1_170	0
823	tree00_c_Tree_PromoteRootNode	15	21	54	54	54	0
824	tree00_c_Tree_PromoteInternalNode	71	120	7_926_110	7_926_155	7_926_125	0
825	tree00_c_Tree_RecurseSearch	41	62	6_516	6_696	6_696	0
826	tree00_c_Tree_GetRecursePos	34	48	294	299	299	0
827	tree00_c_Tree_CreateNode	9	11	6	6	6	0
828	tree00_c_Tree_GetFrozenNode	8	9	4	4	4	0
829	tree00_c_Tree_FreezeNode	6	6	2	2	2	0
830	tree00_c_Tree_DefrostNode	6	6	2	2	2	0
831	tree00_c_Tree_NodeIsFrosted	3	2	1	1	1	0
832	tree00_c_Tree_DumpNode	19	29	55	61	61	0
833	tree00_c_Tree_NewKey	9	10	4	4	4	0
834	tree00_c_Tree_GetFrozenKey	9	10	4	4	4	0
835	tree00_c_Tree_DefrostKey	8	9	4	4	4	0
836	tree00_c_Tree_GetNodeInsertPos	46	71	2_484	2_493	2_493	0
837	tree00_c_Tree_CompareKey	77	125	52_000	52_000	52_000	0
838	tree01_c_SpclAddIntoTree	63	103	10_767_330	10_767_330	10_767_330	0
839	tree01_c_SpclFindIn2aryTree	6	6	2	2	2	0
840	tree01_c_SpclFindInTree	66	111	49_719_600	49_719_600	49_719_600	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
841	tree01_c_SpclTraverseTree	26	41	490	490	490	0
842	tree01_c_Spcl_TreeIterateOn	29	46	914	914	914	0
843	tree01_c_SpclDeleteFromTree	46	72	90_540	90_540	90_540	0
844	tree01_c_SpclDeleteTree	19	29	160	160	160	0
845	tree01_c_Tree_Create	40	66	17_220	17_220	17_220	0
846	tree01_c_Tree_Compare2aryKey	17	23	22	22	22	0
847	tree0_c_Tree_DeleteFrom	63	103	543_270	543_295	543_274	0
848	tree0_c_Tree_Adjust	61	94	98_427	98_430	98_428	0
849	tree0_c_Tree_Redistribute	37	57	448	452	452	0
850	tree0_c_Tree_Concatenate	41	64	2_728	2_739	2_733	0.2
851	tree0_c_Tree_DeleteNode	10	12	8	8	8	0
852	tree0_c_Test_Iterate	9	11	6	6	6	0
853	ut_c_pow2Round	5	6	2	3	3	0
854	ut_c_Ut_PrintErr	21	35	17	17	17	0
855	ut_c_Ut_SetWatch	3	2	1	1	1	0
856	ut_c_Ut_StopWatch	3	2	1	1	1	0
857	ut_c_Ut_SetBreak	3	2	1	1	1	0
858	ut_c_Ut_VoidTrack	18	26	36	36	36	0
859	ut_c_Ut_PrintTrack	8	10	4	4	4	0
860	ut_c_Ut_StackTrack	9	11	4	4	4	0
861	ut_c_Ut_TraceMsg	14	20	15	15	15	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
862	ut_c_Ut_SendMsg	11	15	14	14	14	0
863	ut_c_Ut_TraceBytes	24	35	12	15	15	0
864	ut_c_Ut_TraceField	47	75	132	144	138	4.2
865	ut_c_Ut_TraceValue	42	68	63	87	69	20.7
866	ut_c_Ut_AlignMember	25	40	207	207	207	0
867	ut_c_Ut_AlignStruc	9	12	10	10	10	0
868	ut_c_Ut_StrToLower	6	7	1	4	3	25
869	ut_c_Ut_StrToUpper	6	7	1	4	3	25
870	ut_c_Ut_ReverseStr	5	6	2	3	3	0
871	ut_c_Ut_IntToStr	9	11	4	5	5	0
872	ut_c_Ut_FindInList	13	18	9	12	11	8.3
873	ut_c_Ut_CompareWild	52	83	3_861	3_872	3_870	0.1
874	ut_c_Ut_CompareString	24	38	420	426	423	0.7
875	ut_c_Ut_Random	3	2	1	1	1	0
876	vchunk_c_Vchunk_Create	6	6	2	2	2	0
877	vchunk_c_Vchunk_IsValidToken	28	42	228	228	228	0
878	vchunk_c_Vchunk_GetVstruc	8	9	4	4	4	0
879	vchunk_c_Vchunk_DumpVstruc	8	9	4	4	4	0
880	vchunk_c_Vchunk_Freeze	8	9	4	4	4	0
881	vchunk_c_Vchunk_GetAddress	8	9	4	4	4	0
882	vchunk_c_Vchunk_PutAddress	8	9	4	4	4	0

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
883	vchunk_c_Vchunk_GetStackPtr	8	9	4	4	4	0
884	vchunk_c_Vchunk_PutStackPtr	9	11	6	6	6	0
885	vchunk_c_Vchunk_GetChunkSize	8	9	4	4	4	0
886	vchunk_c_Vchunk_Copy	16	24	18	18	18	0
887	vchunk_c_Vchunk_Dump	10	13	8	8	8	0
888	vchunk_c_Vchunk_DumpPartial	10	13	8	8	8	0
889	vchunk_c_Vchunk_Free	6	6	2	2	2	0
890	vchunk_c_Vchunk_Commit	6	6	2	2	2	0
891	vchunk_c_Vchunk_Delete	6	6	2	2	2	0
892	vchunk_c_Vchunk_PushUnit	9	11	6	6	6	0
893	vchunk_c_Vchunk_PutUnit	9	11	6	6	6	0
894	vchunk_c_Vchunk_GetUnit	9	11	6	6	6	0
895	vchunk_c_BitField_Create	9	11	6	6	6	0
896	vchunk_c_BitField_Put	13	17	16	16	16	0
897	vchunk_c_BitField_Get	11	15	12	12	12	0
898	vdbm_c_DbmNewVchunk	27	44	699	699	699	0
899	vdbm_c_DbmPutVchunkTkn	13	18	30	30	30	0
900	vdbm_c_DbmFreeVchunk	20	31	102	102	102	0
901	vdbm_c_DbmDeleteVchunk	22	34	308	308	308	0
902	vdbm_c_DbmCommitVchunk	14	21	21	21	21	0
903	vdbm_c_DbmInvokeVchunk	38	61	3_032	3_112	3_072	1.3

Table 11: SPEC CINT95 — 147.vortex

no.	function	nodes	edges	npp	ncp	loncp	rd.
904	vdbm_c_DbmGetVchunkTkn	36	56	8_494	8_494	8_494	0
905	vdbm_c_DbmDumpVchunkVchunk	17	25	68	68	68	0
906	vdbm_c_DbmDumpVchunk	48	77	7_360	7_396	7_372	0.3
907	voa_c_OaCreateVchunk	10	13	10	10	10	0
908	voa_c_OaFreezeVchunk	8	9	4	4	4	0
909	voa_c_OaDumpVchunkVchunk	11	15	10	10	10	0
910	voa_c_OaGetVchunkAddr	12	16	20	20	20	0
911	voa_c_OaPutVchunkAddr	12	16	20	20	20	0
912	voa_c_OaGetVchunkAllocQty	12	16	20	20	20	0
913	voa_c_OaGetVchunkStackPtr	12	16	20	20	20	0
914	voa_c_OaPutVchunkStackPtr	12	16	20	20	20	0
915	voa_c_OaPushUnit	12	16	15	15	15	0
916	voa_c_OaPutUnit	9	11	6	6	6	0
917	voa_c_OaGetUnit	9	11	6	6	6	0
918	voa_c_OaDumpVchunk	10	13	8	8	8	0
919	voa_c_OaDeleteVchunk	3	2	1	1	1	0
920	vom_c_OmNewVchunk	3	2	1	1	1	0
921	vom_c_OmGetVchunkToken	20	29	76	76	76	0
922	vom_c_OmGetVchunkStruc	8	9	4	4	4	0
923	vom_c_OmDeleteVchunk	3	2	1	1	1	0

Table 12: SPEC CFP95 — 101.tomcatv

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	tomcatv_f__MAIN__	76	118	35_005	77_928	73_925	5.1

Table 13: SPEC CFP95 — 102.swim

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	swim_f__MAIN__	20	27	8	48	27	43.8
2	swim_f__inital_	26	41	108	122	119	2.5
3	swim_f__calc1_	15	22	12	18	17	5.6
4	swim_f__calc2_	15	22	12	18	17	5.6
5	swim_f__calc3z_	9	12	3	7	6	14.3
6	swim_f__calc3_	15	22	12	18	17	5.6

Table 14: SPEC CFP95 — 103.su2cor

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	su2cor_data_f__trinit_	3	2	1	1	1	0
2	su2cor_f__MAIN__	81	132	149_952	225_368	187_635	16.7
3	su2cor_f__eval_	16	23	34	51	51	0
4	su2cor_f__sweep_	47	75	2_604	3_975	3_486	12.3
5	su2cor_f__geom_	43	69	1_344	1_376	1_368	0.6
6	su2cor_f___g77_masterfun_matmat	36	59	18	26	26	0
7	su2cor_f__matmat_	3	2	1	1	1	0
8	su2cor_f__addmm_	3	2	1	1	1	0
9	su2cor_f__adjmat_	3	2	1	1	1	0
10	su2cor_f__addam_	3	2	1	1	1	0
11	su2cor_f__matadj_	3	2	1	1	1	0
12	su2cor_f__addma_	3	2	1	1	1	0
13	su2cor_f__adjadj_	3	2	1	1	1	0
14	su2cor_f__addaa_	3	2	1	1	1	0
15	su2cor_f__perm_	6	7	2	3	3	0
16	su2cor_f__intact_	21	26	8	8	8	0
17	su2cor_f__bestab_	25	35	1	20	12	40
18	su2cor_f__bespol_	6	7	2	3	3	0
19	su2cor_f__int4v_	16	23	12	19	18	5.3
20	su2cor_f__int2v_	16	23	12	19	18	5.3
21	su2cor_f__corr_	21	33	48	60	58	3.3

Table 14: SPEC CFP95 — 103.su2cor

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	su2cor_f__loops_	231	362	7.60406×10^{16}	7.63927×10^{16}	7.63927×10^{16}	0
23	su2cor_f__clear2_	13	18	3	9	8	11.1
24	su2cor_f__accum2_	13	19	10	12	12	0
25	su2cor_f__aver2_	7	9	4	4	4	0
26	su2cor_f__stat2_	49	77	78_374	78_395	78_386	0
27	su2cor_f__clear4_	15	22	4	17	12	29.4
28	su2cor_f__accum4_	11	16	5	9	8	11.1
29	su2cor_f__aver4_	10	14	6	6	6	0
30	su2cor_f__sigma4_	12	17	7	7	7	0
31	su2cor_f__covar4_	13	19	24	24	24	0
32	su2cor_f___g77_masterfun_trngv	28	41	20	32	30	6.2
33	su2cor_f__trngv_	3	2	1	1	1	0
34	su2cor_f__trnget_	3	2	1	1	1	0
35	su2cor_f__trnset_	3	2	1	1	1	0
36	su2cor_f__init_	19	29	13	35	29	17.1
37	su2cor_f__trngv1_	17	24	9	18	16	11.1

Table 15: SPEC CFP95 — 104.hydro2d

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	hydro2d_f__MAIN__	5	6	2	6	5	16.7
2	hydro2d_f__prepar_	44	67	331_776	331_896	331_896	0
3	hydro2d_f__inival_	3	2	1	1	1	0
4	hydro2d_f__inimod_	26	38	4	24	17	29.2
5	hydro2d_f__bbh_	12	17	4	9	8	11.1
6	hydro2d_f__bbf_	12	17	4	9	8	11.1
7	hydro2d_f__gridco_	57	86	16_384	16_414	16_410	0
8	hydro2d_f__tistep_	59	93	176_256	176_373	176_337	0
9	hydro2d_f__advnce_	3	2	1	1	1	0
10	hydro2d_f__adlen_	24	34	18	20	20	0
11	hydro2d_f__corih_	11	15	4	11	9	18.2
12	hydro2d_f__stagh1_	3	2	1	1	1	0
13	hydro2d_f__stagh2_	3	2	1	1	1	0
14	hydro2d_f__corif_	11	15	4	11	9	18.2
15	hydro2d_f__stagf1_	27	42	81	97	93	4.1
16	hydro2d_f__stagf2_	3	2	1	1	1	0
17	hydro2d_f__s1_	10	14	6	10	9	10
18	hydro2d_f__s2_	10	14	6	10	9	10
19	hydro2d_f__b1_	22	32	24	28	28	0
20	hydro2d_f__b2_	21	30	20	24	24	0
21	hydro2d_f__trans1_	35	55	324	347	341	1.7

Table 15: SPEC CFP95 — 104.hydro2d

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	hydro2d_f__trans2_	41	65	972	999	992	0.7
23	hydro2d_f__t1_	8	11	3	7	6	14.3
24	hydro2d_f__t2_	8	11	3	7	6	14.3
25	hydro2d_f__artdif_	21	32	36	46	44	4.3
26	hydro2d_f__fct_	37	60	1_296	1_316	1_312	0.3
27	hydro2d_f__filter_	177	280	2.88587×10^{17}	2.88587×10^{17}	2.88587×10^{17}	0
28	hydro2d_f__check_	6	7	3	3	3	0
29	hydro2d_f__cut_	29	43	432	485	468	3.5
30	hydro2d_f__vpr_	28	44	144	160	156	2.5
31	hydro2d_f__vps_	18	26	8	18	15	16.7
32	hydro2d_f__con_	7	8	1	4	3	25
33	hydro2d_f__output_	22	47	3_063	8_600	3_087	64.1
34	hydro2d_f__term_	6	6	2	4	4	0
35	hydro2d_f__sdot_	10	13	8	9	9	0
36	hydro2d_f__ismin_	10	13	6	8	8	0
37	hydro2d_f__i0max_	10	13	6	8	8	0
38	hydro2d_f__isamax_	10	13	6	8	8	0
39	hydro2d_f__wnfle_	10	13	6	8	8	0
40	hydro2d_f__wnfgt_	10	13	6	8	8	0
41	hydro2d_f__isfne_	9	12	6	7	7	0
42	inpdta_f__inpdta_	3	2	1	1	1	0

Table 16: SPEC CFP95 — 107.mgrid

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	mgrid_f__MAIN__	18	27	18	48	41	14.6
2	mgrid_f__mg3p_	10	14	6	8	8	0
3	mgrid_f__psinv_	19	30	16	38	28	26.3
4	mgrid_f__resid_	12	17	4	15	10	33.3
5	mgrid_f__rprj3_	12	17	4	15	10	33.3
6	mgrid_f__interp_	45	72	676	790	750	5.1
7	mgrid_f__comm3_	19	30	27	39	36	7.7
8	mgrid_f__norm2u3_	14	20	5	23	14	39.1
9	mgrid_f__setup_	17	26	24	28	28	0
10	mgrid_f__zero3_	10	15	4	15	10	33.3
11	mgrid_f__zran3_	20	30	4	20	15	25
12	mgrid_f__bubble_	10	13	6	7	7	0

Table 17: SPEC CFP95 — 110.applu

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	applu_f__MAIN__	14	20	11	22	22	0
2	applu_f__setbv_	18	29	27	39	36	7.7
3	applu_f__setiv_	14	20	4	23	11	52.2
4	applu_f__blts_	32	49	9	223	43	80.7
5	applu_f__buts_	33	51	9	231	44	81
6	applu_f__erhs_	107	170	157_216	157_682	157_448	0.1
7	applu_f__exact_	5	5	1	2	2	0
8	applu_f__error_	17	26	4	26	14	46.2
9	applu_f__jacld_	12	17	4	15	10	33.3
10	applu_f__jacu_	12	17	4	15	10	33.3
11	applu_f__pintgr_	39	62	729	753	747	0.8
12	applu_f__rhs_	101	161	157_216	157_658	157_445	0.1
13	applu_f__ssor_	63	102	38_018	53_943	53_877	0.1
14	applu_f__maxnorm_	16	24	5	40	17	57.5
15	applu_f__l2norm_	18	26	4	25	13	48
16	applu_f__verify_	55	78	131	137	137	0

Table 18: SPEC CFP95 — 125.turb3d

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	turb3d_f__MAIN__	6	6	2	4	4	0
2	turb3d_f__turb3d_	51	76	9_600	13_143	11_529	12.3
3	turb3d_f__param_	7	8	4	4	4	0
4	turb3d_f__wavnum_	37	58	1_792	1_804	1_804	0
5	turb3d_f__tgvel_	12	17	4	15	10	33.3
6	turb3d_f__wcal_	11	16	4	15	10	33.3
7	turb3d_f__dcopy_	11	16	6	8	8	0
8	turb3d_f__enr_	16	23	6	46	22	52.2
9	turb3d_f__zfft_	29	41	128	128	128	0
10	turb3d_f__dcft_	12	17	5	13	11	15.4
11	turb3d_f__cfft_	66	106	6_066	6_087	6_081	0.1
12	turb3d_f__fftz1_	42	64	1_296	1_306	1_305	0.1
13	turb3d_f__trans_	15	23	6	14	12	14.3
14	turb3d_f__fftz2_	42	64	1_296	1_306	1_305	0.1
15	turb3d_f__xyfft_	28	40	128	128	128	0
16	turb3d_f__drcft_	11	16	6	11	10	9.1
17	turb3d_f__dcrft_	10	15	6	11	10	9.1
18	turb3d_f__uxw_	10	15	4	15	10	33.3
19	turb3d_f__linavg_	14	20	5	23	14	39.1
20	turb3d_f__mixavg_	11	16	4	15	10	33.3
21	turb3d_f__lin_	14	20	5	23	14	39.1

Table 18: SPEC CFP95 — 125.turb3d

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	turb3d_f__verifytr_	140	207	1.47574×10^{20}	1.47574×10^{20}	1.47574×10^{20}	0
23	turb3d_f__verify_	140	207	1.47574×10^{20}	1.47574×10^{20}	1.47574×10^{20}	0

Table 19: SPEC CFP95 — 141.apsi

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	apsi_f__MAIN__	52	80	58_330	116_682	116_669	0
2	apsi_f__pset_	13	17	32	32	32	0
3	apsi_f__run_	55	88	2_073_600	2_177_304	2_177_292	0
4	apsi_f__advc_	17	27	7	22	16	27.3
5	apsi_f__dcdtz_	20	29	18	75	53	29.3
6	apsi_f__advt_	17	27	7	22	16	27.3
7	apsi_f__dtdtz_	24	35	66	267	197	26.2
8	apsi_f__hyd_	11	16	4	15	10	33.3
9	apsi_f__advu_	17	27	7	22	16	27.3
10	apsi_f__dudtz_	19	28	18	75	53	29.3
11	apsi_f__advv_	17	27	7	22	16	27.3
12	apsi_f__dvdtz_	22	32	26	111	78	29.7
13	apsi_f__wcont_	16	24	8	39	24	38.5
14	apsi_f__hordfc_	8	11	4	6	6	0
15	apsi_f__dctdx_	16	24	17	39	36	7.7
16	apsi_f__dctdxd_	17	27	25	57	53	7
17	apsi_f__dctdy_	14	22	17	39	36	7.7
18	apsi_f__dctdyd_	14	24	25	57	53	7
19	apsi_f__dpdx_	13	20	9	23	20	13
20	apsi_f__dpdy_	12	19	9	23	20	13
21	apsi_f__dftdx_	13	20	9	23	20	13

Table 19: SPEC CFP95 — 141.apsi

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	apsi_f__dftdy_	11	18	9	23	20	13
23	apsi_f__cpade_	17	23	18	22	22	0
24	apsi_f__padec_	22	30	54	64	64	0
25	apsi_f__ccrank_	11	14	6	8	8	0
26	apsi_f__crankc_	11	14	6	8	8	0
27	apsi_f__upade_	11	15	8	10	10	0
28	apsi_f__ucrank_	8	10	4	5	5	0
29	apsi_f__tpade_	15	21	18	22	22	0
30	apsi_f__tcrank_	10	13	6	8	8	0
31	apsi_f__tmpade_	21	29	50	58	58	0
32	apsi_f__tmcrnk_	13	17	10	14	14	0
33	apsi_f__dwdz_	8	10	4	5	5	0
34	apsi_f__trid_	8	11	4	6	6	0
35	apsi_f__tridc_	17	23	18	22	22	0
36	apsi_f__leapfr_	11	15	4	6	6	0
37	apsi_f__dctdxf_	9	12	3	7	6	14.3
38	apsi_f__dctdyf_	9	12	3	7	6	14.3
39	apsi_f__strech_	18	26	30	36	36	0
40	apsi_f__smth_	6	7	2	3	3	0
41	apsi_f__smthf_	6	7	2	3	3	0
42	apsi_f__csmth_	12	15	5	9	9	0

Table 19: SPEC CFP95 — 141.apsi

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	apsi_f__smooth_	5	6	2	3	3	0
44	apsi_f__smim_	5	6	2	3	3	0
45	apsi_f__horims_	18	30	30	46	42	8.7
46	apsi_f__smimp_	10	13	5	9	9	0
47	apsi_f__horsmt_	16	24	12	20	18	10
48	apsi_f__printr_	27	40	1_120	1_120	1_120	0
49	apsi_f__graph_	20	32	44	79	70	11.4
50	apsi_f__stats_	3	2	1	1	1	0
51	apsi_f__setall_	66	102	583_736	803_171	729_729	9.1
52	apsi_f__horbc_	26	42	43	117	97	17.1
53	apsi_f__stab_	5	5	2	2	2	0
54	apsi_f__synset_	3	2	1	1	1	0
55	apsi_f__uvset_	10	12	4	6	6	0
56	apsi_f__dinitu_	5	6	2	3	3	0
57	apsi_f__final_	5	6	2	3	3	0
58	apsi_f__test_	10	13	6	7	7	0
59	apsi_f__topo_	32	50	2_341	2_729	2_633	3.5
60	apsi_f__topplt_	20	31	69	173	155	10.4
61	apsi_f__filtpr_	12	18	12	16	16	0
62	apsi_f__prn_	6	7	2	3	3	0
63	apsi_f__rffti_	5	5	2	2	2	0

Table 19: SPEC CFP95 — 141.apsi

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	apsi_f__costi_	8	10	3	4	4	0
65	apsi_f__cost_	17	25	19	21	21	0
66	apsi_f__rffti1_	24	37	40	64	53	17.2
67	apsi_f__rfftb_	5	5	2	2	2	0
68	apsi_f__rfftb1_	35	49	39	52	52	0
69	apsi_f__radb2_	18	28	24	34	29	14.7
70	apsi_f__radb3_	13	19	8	13	12	7.7
71	apsi_f__radb4_	18	28	24	34	29	14.7
72	apsi_f__radb5_	13	19	8	13	12	7.7
73	apsi_f__radbg_	105	181	4_010_958	4_011_065	4_011_023	0
74	apsi_f__rfftf_	5	5	2	2	2	0
75	apsi_f__rfftf1_	36	50	39	52	52	0
76	apsi_f__radf2_	18	28	24	34	29	14.7
77	apsi_f__radf3_	13	19	8	13	12	7.7
78	apsi_f__radf4_	18	28	24	34	29	14.7
79	apsi_f__radf5_	13	19	8	13	12	7.7
80	apsi_f__radfg_	107	184	3_939_516	3_939_624	3_939_582	0
81	apsi_f__dkzmh_	42	64	1_600	1_876	1_768	5.8
82	apsi_f__topbl_	20	30	52	129	102	20.9
83	apsi_f__initbl_	8	11	3	7	6	14.3
84	apsi_f__sfcpar_	18	26	192	193	193	0

Table 19: SPEC CFP95 — 141.apsi

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	apsi_f__surfac_	10	12	5	5	5	0
86	apsi_f__blsolv_	9	13	6	8	8	0
87	apsi_f__srflay_	20	28	72	72	72	0
88	apsi_f__ekmlay_	9	11	5	5	5	0
89	apsi_f__dkzmn_	3	2	1	1	1	0
90	apsi_f__dkzp_	31	47	302	1_243	913	26.5
91	apsi_f__blm_	26	38	264	285	283	0.7
92	apsi_f__mixhgt_	22	32	54	59	57	3.4
93	apsi_f__klass_	13	17	32	32	32	0
94	apsi_f__ovl_	20	26	14	14	14	0
95	apsi_f__srfclr_	29	42	264	264	264	0
96	apsi_f__ekmnlr_	7	8	3	3	3	0

Table 20: SPEC CFP95 — 145.fpppp

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	aclear_f__aclear_	5	6	2	3	3	0
2	d2esp_f__d2esp_	31	48	3_462	3_463	3_463	0
3	efill_f__efill_	62	100	710_174	3_088_243	2_191_047	29.1
4	fmtgen_f__fmtgen_	42	63	368	381	376	1.3
5	fmtset_f__fmtset_	35	51	992	999	999	0
6	fpppp_f__fpppp_	3	2	1	1	1	0
7	gabs_f__gabs_	3	2	1	1	1	0
8	gacos_f__gacos_	3	2	1	1	1	0
9	gamgen_f__gamgen_	12	15	2	9	7	22.2
10	gasin_f__gasin_	3	2	1	1	1	0
11	gatan2_f__gatan2_	3	2	1	1	1	0
12	gatan_f__gatan_	3	2	1	1	1	0
13	gcabs_f__gcabs_	3	2	1	1	1	0
14	gcexp_f__gcexp_	3	2	1	1	1	0
15	gcmplx_f__gcmplx_	3	2	1	1	1	0
16	gconjg_f__gconjg_	3	2	1	1	1	0
17	gcos_f__gcos_	3	2	1	1	1	0
18	gexp_f__gexp_	3	2	1	1	1	0
19	gfloat_f__gfloat_	3	2	1	1	1	0
20	gimag_f__gimag_	3	2	1	1	1	0
21	gint_f__gint_	6	6	2	2	2	0

Table 20: SPEC CFP95 — 145.fpppp

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	glog10_f__glog10_	3	2	1	1	1	0
23	glog_f__glog_	3	2	1	1	1	0
24	gmax1_f__gmax1_	5	5	2	2	2	0
25	gmin1_f__gmin1_	5	5	2	2	2	0
26	gmod_f__gmod_	3	2	1	1	1	0
27	greal_f__greal_	3	2	1	1	1	0
28	gsign_f__gsign_	5	5	2	2	2	0
29	gsin_f__gsin_	3	2	1	1	1	0
30	gsqrt_f__gsqrt_	3	2	1	1	1	0
31	gtan_f__gtan_	3	2	1	1	1	0
32	igfix_f__igfix_	3	2	1	1	1	0
33	ilsw_f__ilsw_	3	2	1	1	1	0
34	intowp_f__intowp_	3	2	1	1	1	0
35	lclear_f__lclear_	5	6	2	3	3	0
36	main000_f__MAIN__	15	22	12	32	28	12.5
37	nprio_f__nprio_	12	16	16	16	16	0
38	twldrv_f__twldrv_	163	271	6.42169×10^{14}	8.69604×10^{14}	7.40278×10^{14}	14.9

Table 21: SPEC CFP95 — 146.wave5

no.	function	nodes	edges	npp	ncp	loncp	rd.
1	wave5_data_f__inidat_	3	2	1	1	1	0
2	wave5_data_f__genprb_	3	2	1	1	1	0
3	wave5_data_f__init_	136	214	4_645_056	9_685_741	7_166_258	26
4	wave5_data_f__ip2_	3	2	1	1	1	0
5	wave5_data_f__dens_	3	2	1	1	1	0
6	wave5_data_f__densx_	3	2	1	1	1	0
7	wave5_data_f__densy_	3	2	1	1	1	0
8	wave5_data_f__denpt_	22	33	96	103	103	0
9	wave5_data_f__trans_	48	73	55_304	86_568	71_008	18
10	wave5_data_f__celbnd_	58	91	245_760	245_768	245_768	0
11	wave5_data_f__bcnd_	40	62	670	736	736	0
12	wave5_data_f___g77_masterfun_bcnds	81	118	61	61	61	0
13	wave5_data_f__bcnds_	3	2	1	1	1	0
14	wave5_data_f__bcndr_	3	2	1	1	1	0
15	wave5_data_f__bcndl_	3	2	1	1	1	0
16	wave5_data_f__bcndt_	3	2	1	1	1	0
17	wave5_data_f__bcndb_	3	2	1	1	1	0
18	wave5_data_f__ibin_	5	5	2	2	2	0
19	wave5_data_f__recre_	16	22	39	39	39	0
20	wave5_data_f__setinj_	11	14	9	9	9	0
21	wave5_data_f__vavg_	5	5	2	2	2	0

Table 21: SPEC CFP95 — 146.wave5

no.	function	nodes	edges	npp	ncp	loncp	rd.
22	wave5_data_f_tpart_	10	12	4	4	4	0
23	wave5_data_f_tcomp_	21	29	10	18	18	0
24	wave5_data_f_denitl_	9	12	5	6	6	0
25	wave5_data_f_denitr_	9	12	5	6	6	0
26	wave5_data_f_linj_	14	19	24	28	28	0
27	wave5_data_f_rinj_	14	19	24	28	28	0
28	wave5_data_f_vnewl_	7	8	2	3	3	0
29	wave5_data_f_injbat_	56	87	6_736	6_844	6_832	0.2
30	wave5_data_f_injall_	34	54	107	135	133	1.5
31	wave5_data_f_injcon_	20	29	70	78	78	0
32	wave5_data_f_injchk_	23	36	418	455	455	0
33	wave5_data_f_erf_	3	2	1	1	1	0
34	wave5_data_f_field_	217	343	8.06819×10^{13}	8.06819×10^{13}	8.06819×10^{13}	0
35	wave5_data_f__g77_masterfun_advbnd	46	72	1_516	1_543	1_537	0.4
36	wave5_data_f_advbnd_	3	2	1	1	1	0
37	wave5_data_f_inibnd_	3	2	1	1	1	0
38	wave5_data_f__g77_masterfun_laser	35	57	25	37	36	2.7
39	wave5_data_f_laser_	3	2	1	1	1	0
40	wave5_data_f_lasden_	3	2	1	1	1	0
41	wave5_data_f_laspow_	3	2	1	1	1	0
42	wave5_data_f_pdiag_	57	89	483_667	485_985	485_984	0

Table 21: SPEC CFP95 — 146.wave5

no.	function	nodes	edges	npp	ncp	loncp	rd.
43	wave5_data_f__diagns_	19	29	90	94	93	1.1
44	wave5_data_f__energy_	22	33	72	78	77	1.3
45	wave5_data_f__solv2p_	3	2	1	1	1	0
46	wave5_data_f__slv2pd_	8	10	3	5	5	0
47	wave5_data_f__slv2xp_	33	50	1_488	1_509	1_499	0.7
48	wave5_data_f__vslv1p_	46	75	10_368	10_433	10_414	0.2
49	wave5_data_f__vslv1x_	39	64	1_728	1_748	1_744	0.2
50	wave5_data_f__gen_	22	32	50	59	54	8.5
51	wave5_data_f__vffa_	16	25	16	43	27	37.2
52	wave5_data_f__vffs_	17	26	16	43	27	37.2
53	wave5_data_f__abrt_	5	6	2	6	5	16.7
54	wave5_data_f__solv2y_	3	2	1	1	1	0
55	wave5_data_f__slv2xd_	8	10	3	5	5	0
56	wave5_data_f__slv2xy_	41	65	12_240	12_245	12_245	0
57	wave5_data_f__genb_	13	17	6	6	6	0
58	wave5_data_f__fftf_	16	24	33	69	67	2.9
59	wave5_data_f__fftb_	18	27	65	133	131	1.5
60	wave5_data_f__smooth_	58	98	476_288	476_454	476_428	0
61	wave5_data_f___g77_masterfun_ecrd	9	13	4	6	6	0
62	wave5_data_f__ecrd_	3	2	1	1	1	0
63	wave5_data_f__ecwr_	3	2	1	1	1	0

Table 21: SPEC CFP95 — 146.wave5

no.	function	nodes	edges	npp	ncp	loncp	rd.
64	wave5_data_f__ranf_	9	11	8	8	8	0
65	wave5_data_f__parmvr_	74	116	15_863_040	15_863_079	15_863_079	0
66	wave5_data_f__parmov_	68	106	3_965_760	3_965_797	3_965_797	0
67	wave5_data_f__parmve_	55	85	198_288	198_319	198_319	0
68	wave5_data_f__jobtim_	3	2	1	1	1	0
69	wave5_data_f___g77_masterfun_partbl	103	158	9_848_646	9_848_664	9_848_656	0
70	wave5_data_f__partbl_	3	2	1	1	1	0
71	wave5_data_f__setb_	3	2	1	1	1	0
72	wave5_data_f__getb_	3	2	1	1	1	0
73	wave5_data_f__putb_	3	2	1	1	1	0
74	wave5_data_f__numb_	3	2	1	1	1	0
75	wave5_data_f__sudtbl_	6	7	3	3	3	0
76	wave5_data_f__putdt_	18	27	20	24	22	8.3
77	wave5_data_f__getdt_	3	2	1	2	2	0
78	wave5_data_f__endrun_	3	2	1	2	2	0
79	wave5_data_f__clrdt_	3	2	1	1	1	0
80	wave5_data_f__rewdt_	3	2	1	1	1	0
81	wave5_data_f__rfftb_	5	5	2	2	2	0
82	wave5_data_f__rfftb1_	36	50	39	52	52	0
83	wave5_data_f__rfftf_	5	5	2	2	2	0
84	wave5_data_f__rfftf1_	37	51	39	52	52	0

Table 21: SPEC CFP95 — 146.wave5

no.	function	nodes	edges	npp	ncp	loncp	rd.
85	wave5_data_f__rffti_	5	5	2	2	2	0
86	wave5_data_f__rffti1_	24	37	40	64	53	17.2
87	wave5_data_f__cosqb_	8	9	3	3	3	0
88	wave5_data_f__cosqb1_	13	20	32	35	35	0
89	wave5_data_f__cosqf_	7	8	3	3	3	0
90	wave5_data_f__cosqf1_	13	20	32	35	35	0
91	wave5_data_f__cosqi_	5	6	2	3	3	0
92	wave5_data_f__cost_	17	25	19	21	21	0
93	wave5_data_f__costi_	8	10	3	4	4	0
94	wave5_data_f__sinqb_	10	14	5	7	7	0
95	wave5_data_f__sinqf_	9	13	5	7	7	0
96	wave5_data_f__sinqi_	3	2	1	1	1	0
97	wave5_data_f__sint_	17	24	18	20	20	0
98	wave5_data_f__sinti_	7	9	3	4	4	0
99	wave5_data_f__radb2_	26	41	42	60	50	16.7
100	wave5_data_f__radb3_	21	32	14	23	21	8.7
101	wave5_data_f__radb4_	26	41	42	60	50	16.7
102	wave5_data_f__radb5_	21	32	14	23	21	8.7
103	wave5_data_f__radbg_	104	179	3_929_310	3_929_417	3_929_375	0
104	wave5_data_f__radf2_	26	41	42	60	50	16.7
105	wave5_data_f__radf3_	21	32	14	23	21	8.7

Table 21: SPEC CFP95 — 146.wave5

no.	function	nodes	edges	npp	ncp	loncp	rd.
106	wave5_data_f__radf4_	26	41	42	60	50	16.7
107	wave5_data_f__radf5_	21	32	14	23	21	8.7
108	wave5_data_f__radfg_	107	184	3_939_516	3_939_624	3_939_582	0
109	wave5_data_f__rand_	18	24	13	13	13	0
110	wave5_f__MAIN__	32	45	55	126	118	6.3

8 Conclusion

In this paper we have dissected the entire SPEC95 benchmark suite in order to derive the corresponding resource requirements for symbolic evaluation. For our measurements we have developed a series of metrics on path expressions that capture the control flow information contained in the underlying control flow graphs.

We have set up a data-flow problem for path expression generation, along with a proof that for reducible flowgraphs the generated path expressions are minimal with respect to our metrics. We have furthermore presented an adversary argument showing that for irreducible flowgraphs this does not hold in general.

Path expressions as well as the metrics data itself were computed by a data-flow framework. We have applied several data analysis methods from [Cle93] to evaluate the data collected in our experiment.

Our measurements show that the largest part of the surveyed SPEC95 procedures constitute very small resource requirements with respect to symbolic evaluation.

Appendix

Regular Expressions and Path Expressions

It is shown in [Tar81] how program paths π can be represented as regular expressions: Let Σ be a finite alphabet disjoint from $\{\Lambda, \emptyset, (,)\}$. A *regular expression* is any expression built by applying the following rules:

(1a) "Λ" and "$\emptyset$" are *atomic* regular expressions; for any $a \in \Sigma$, "a" is an atomic regular expression.

(1b) If R_1 and R_2 are regular expressions, then $(R_1 + R_2)$, $(R_1 \cdot R_2)$, and $(R_1)^*$ are *compound* regular expressions.

In a regular expression, Λ denotes the empty string, $\emptyset$ denotes the empty set, $+$ denotes set union, $\cdot$ denotes concatenation, and * denotes reflexive, transitive closure under concatenation. Thus each regular expression R over Σ defines a set $\sigma(R)$ of strings over Σ as follows:

(2a) $\sigma(\Lambda) = \{\Lambda\}$; $\sigma(\emptyset) = \emptyset$; $\sigma(a) = \{a\}$ for $a \in \Sigma$.

(2b) $\sigma(R_1 + R_2) = \sigma(R_1) + \sigma(R_2) = \{w \mid w \in \sigma(R_1) \text{ or } w \in \sigma(R_2)\}$;

(2c) $\sigma(R_1 \cdot R_2) = \sigma(R_1) \cdot \sigma(R_2) = \{w_1 w_2 \mid w_1 \in \sigma(R_1) \text{ and } w_2 \in \sigma(R_2)\}$;

(2d) $\sigma(R_1^*) = \bigcup_{k=0}^{\infty} \sigma(R_1)^k$, where $\sigma(R_1)^0 = \{\Lambda\}$, and $\sigma(R_1)^i = \sigma(R_1)^{i-1} \cdot \sigma(R_1)$.

Two regular expressions R_1 and R_2 are said to be *equivalent*, denoted by $R_1 \sim R_2$, if $\sigma(R_1) = \sigma(R_2)$. A regular expression R is *simple* if $R = \emptyset$ or R does not contain $\emptyset$ as a subexpression. According to the definition given in [Sal66, p. 159] two regular expressions are *identical*, denoted by $R_1 \equiv R_2$, if they contain the same symbols in the same order.

Given a CFG $G = \langle N, E, n_e, n_x \rangle$, we can regard any path π in G as a string over E, but not all strings over E are paths in G. A *path expression* P of *type* (v, w) is a simple regular expression over E such that every string in $\sigma(P)$ is a program path from v to w.

References

[ASU86] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers—Principles, Techniques, and Tools*. Addison-Wesley, 1986.

[BEGO71] Ronald Book, Shimon Even, Sheila Greibach, and Gene Ott. Ambiguity in Graphs and Expressions. *IEEE Transactions on Computers*, 20(2):149–153, February 1971.

[BL96] Thomas Ball and James R. Larus. Efficient Path Profiling. In *Proceedings of the 29th Annual ACM/IEEE International Symposium on Microarchitecture*, pages 46–57. IEEE Computer Society, 1996.

[Bli02] Johann Blieberger. Data-Flow Frameworks for Worst-Case Execution Time Analysis. *Real-Time Systems*, 22:183–227, 2002.

[Bur04] Bernd Burgstaller. *Symbolic Evaluation of Imperative Programming Languages (to be submitted)*. PhD thesis, Institute of Computer Aided Automation, Vienna University of Technology, Vienna, Austria, 2004.

[Cle93] William S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[CPU95] SPEC CPU95 Benchmark Suite, Version 1.10, August 1995.

[Gin67] A. Ginzburg. A procedure for checking equality of regular expressions. *J. ACM*, 14(2):355–362, 1967.

[HMU01] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, N. Reading, MA, 2nd edition edition, 2001.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, N. Reading, MA, 1979.

[Pau88] Marvin C. Paull. *Algorithm Design: A Recursion Transformation Framework*. Wiley-Interscience, 1988.

[Ram99] G. Ramalingam. Identifying loops in almost linear time. *ACM Transactions on Programming Languages and Systems*, 21(2):175–188, 1999.

[Ros95] Kenneth H. Rosen. *Discrete Mathematics And Its Applications (3rd ed.)*. McGraw-Hill, Inc., 1995.

[RW94] Darrell Raymond and Derick Wood. Grail: A C++ Library for Automata and Expressions. *Journal of Symbolic Computation*, 17(4):341–350, 1994.

[Sal66] Arto Salomaa. Two Complete Axiom Systems for the Algebra of Regular Events. *J. ACM*, 13(1):158–169, 1966.

[Sre95] V. C. Sreedhar. *Efficient Program Analysis Using DJ Graphs*. PhD thesis, School of Computer Science, McGill University, Montréal, Québec, Canada, 1995.

[Tar81] Robert Endre Tarjan. A Unified Approach to Path Problems. *Journal of the ACM (JACM)*, 28(3):577–593, 1981.

[ZC91] Hans Zima and Barbara Chapman. *Supercompilers for Parallel and Vector Computers*. ACM Press, New York, 1991.