Comparing Conservative Coalescing Criteria

MAX HAILPERIN
Gustavus Adolphus College

Graph-coloring register allocators can eliminate copy instructions from a program by coalescing the interference graph nodes corresponding to the source and destination. Briggs showed that by limiting coalescing to those situations that he dubbed "conservative," it could be prevented from causing spilling, i.e., a situation where the allocator fails to assign a register to each live range. George and Appel adopted Briggs's conservativeness criterion in general, but provided an alternative criterion (the George test) to use in those cases where one of the nodes has been "precolored," i.e., pre-assigned a specific register. They motivated this alternative criterion by efficiency considerations, and provided no indication of the relative power of the two criteria. Thus it remained an open question whether the efficiency had been bought at the expense of reduced coalescing. Their implementation also used a limited version of the Briggs test, in place of the original, full version, without any comment on the impact of this substitution. In this paper we also present an analogously limited version of the George test.

Thus we are now confronted with four different criteria for conservative coalescing: the full and limited Briggs tests and the full and limited George tests. We present a number of theorems characterizing the relative power of these different criteria, and a number of theorems characterizing the form of safety that each achieves. For example, we show that for coalescing with precolored nodes, the full George criterion is strictly more powerful than the full Briggs criterion, while offering an equally strong safety guarantee. Thus no coalesces are lost through George and Appel's introduction of the George test, and some can be gained without sacrificing safety.

We also show that George and Appel's limited version of the Briggs test is probably undesirable. Although a slightly stronger safety result applies to it than to the full Briggs test, this comes at the expense of eliminating all coalesces that can reduce spilling.

Categories and Subject Descriptors: D.3.4 [**Programming Languages**]: Processors—*code generation; optimization*; G.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms*

General Terms: Algorithms, Languages

Additional Key Words and Phrases: Copy propagation, graph coloring, register allocation, register coalescing

1. INTRODUCTION

Chaitin [1981; 1982] introduced a graph-coloring technique for assigning registers to hold a program's values. Each live range in the program is represented as a node in an interference graph, with an edge connecting any two that may need to hold different values simultaneously, and hence cannot be assigned the same register. A clique of k special precolored nodes represents the registers; there can also be edges between precolored and non-precolored nodes. The graph is then colored with the k available registers using the following heuristic technique. A non-precolored node with degree less than k is removed and the remaining subgraph recursively colored.

Author's address: Gustavus Adolphus College, 800 West College Avenue, Saint Peter, MN 56082, USA; email: max@gustavus.edu.

The removed node is then added back in and assigned a color (register number) different from that of its neighbors. If at any stage the graph has non-precolored nodes, but none of degree less than k, the heuristic is said to spill. Spilling involves at least the potential for rewriting the program to have more memory references (and smaller live ranges) and hence is undesirable.

Chaitin also suggested coalescing the nodes corresponding to the source and destination of a copy instruction, provided they didn't interfere. This would ensure that they were assigned the same register, allowing the copy to be eliminated.

Briggs [1994; 1992] identified a subset of coalescing opportunities that could be safely taken without causing the coloring heuristic to spill. Namely, if the coalesced node would have fewer than k neighbors of degree k or higher, the coalesce can be safely done. (Saftey in this case does not refer to semantic correctness, but only to avoiding extra spills. We will be more precise about the safety guarantee later.) Briggs terms these coalesces "conservative." In his register allocator, conservative coalesces were used in conjunction with live-range splitting and rematerialization, not in the general case.

George and Appel [1996] developed an iterated coalescing heuristic, in which conservative coalescing steps are interleaved with the steps of the coloring heuristic. The coloring heuristic only removes nodes that are of degree less than k, not precolored, and not eligible for coalescing. When no such nodes remain, the next priority is to do a conservative coalesce. If that is also impossible, the heuristic's third choice is to mark a pair of nodes as ineligible for coalescing, which may allow one of them to be removed. If all these techniques leave any non-precolored nodes in the graph, the heuristic spills.

One key novelty George and Appel introduced was the handling of precolored nodes, that is, nodes from the special k-clique used to represent machine registers. They noted that the precolored nodes in their programs typically had very high degrees, which would make their adjacency lists expensive to store and to search in order to straightforwardly implement Briggs's conservativeness test. Therefore, they provided an alternative criterion, which Appel [2002] later called the George test, for testing the safety of coalesces with a precolored node. The George test allows a non-precolored node to be coalesced into a precolored node if all the neighbors of the non-precolored node that have degree k or more are also neighbors of the precolored node.

George and Appel leave open the question of how their conservative coalescing criterion compares in power with Briggs's, that is, in the number of safe coalesces it can identify. Instead, they motivate it solely on efficiency grounds. The power question is important for two reasons. One is that if Briggs's test is more powerful, it would be worth finding a way to solve its efficiency problems with regard to high-degree precolored nodes. The other is that if the George criterion is more powerful, then it would be worth using even when there is no efficiency motivation for doing so.

The question of the relative power and safety of conservative coalescing criteria actually extends beyond the original criteria of Briggs and of George. It turns out that George and Appel's pseudocode implements not Briggs's original criterion, but rather a modified form that sanctions fewer coalesces.[1] We term the original the FullBriggs test, and the modified version the LimitedBriggs test. The essential difference is that LimitedBriggs does not take into account that a degree k mutual neighbor of the coalesced nodes will only have degree $k-1$ after the coalesce. In this paper we will also show an analogous modification to the George test; again we call the original FullGeorge, with the new modification being LimitedGeorge. Thus we have four criteria to compare the power and safety of.

We will start by providing a formalization of the graph coloring process, coalescing, the coalescing criteria, and the safety standards that we later show they satisfy. Next we provide our results on the relative power of the criteria: for coalesces with precolored nodes, FullGeorge subsumes FullBriggs, while for coalescing pairs of non-precolored nodes, the two criteria are of incomparable power. Similar results hold for the two Limited forms. We next take up the question of safety. The power result for the precolored case allows us to plug a hole in the previously published proof of the FullBriggs criterion's safety. Additionally, we show that there are varying degrees of safety that the criteria satisfy. Finally, we connect these varying safety guarantees to the recent observation that coalescing can sometimes actually reduce spilling.

2. FORMALIZATION

We will assume we have a finite set of non-precolored nodes, N, representing the live ranges in a program, and disjoint from it a set, P, of k precolored nodes representing the k registers. We will use V to represent the full set of vertices, $N \cup P$. The graph we are to color is $G = \langle V, E\rangle$, where the set of edges E obeys the inequality $P \times P \subseteq E \subseteq V \times V$. That is, the precolored nodes, P, must form a clique (complete subgraph). The goal is to assign each of the nodes in N one of the k colors (registers) present in P, such that no two nodes (precolored or otherwise) that are adjacent in G have the same color.

The graphs that arise in the course of the coloring and coalescing process will all have vertex sets that are subsets of V, but some may have edges not in E. The graphs we are considering, then, are those graphs H that obey $\langle P, P \times P\rangle \sqsubseteq H \sqsubseteq \langle V, V \times V\rangle$, where $\sqsubseteq$ indicates the subgraph relation. These graphs form a finite lattice under the $\sqsubseteq$ relation. The particular lattice depends on the graph G being colored, since G supplies the sets V and P. To avoid confusion, we will reserve the name G for the initial graph derived directly from the program under consideration, which induces the lattice we are working in, and instead use the letter H to indicate some arbitrary graph drawn from that lattice.

We'll consider a number of functions on this lattice. A function f is said to be monotonic if $H \sqsubseteq H' \implies f(H) \sqsubseteq f(H')$, and we will call it nonincreasing if for all H, $f(H) \sqsubseteq H$. Several of our functions are defined in terms of the degree or adjacency of nodes in a graph. Since we'll be considering multiple graphs containing the same nodes, it is important to use a notation that makes the graph explicit;

[1] The modified version also shows up in the pseudocode of Appel's book, even in the second edition [2002], though the text describes the original version. Interestingly, even Briggs seems to have used the modified version, at least in the code that he contributed to the DIMACS coloring challenge [1993].

```
function R(H):
    let R_0 = H
    for i from 0 until exited do
        if ∃S ∈ {S_n | n ∈ N} such that S(R_i) ≠ R_i then
            choose any such S
                let R_{i+1} = S(R_i)
        else
            return R_i
```

Fig. 1. Algorithm for computing the residue, $R(H)$

we'll write $\deg_H(v)$ for the degree of v in H, and $\mathrm{adj}_H(v)$ for the set of all vertices adjacent to v in H. We will adopt the convention that $\deg_H(v) = 0$ if $v \notin H$. Another notational convention is that we'll use the name of a graph to refer as well to its set of vertices or edges where the context disambiguates, and will use subtraction of a set of vertices from a graph to indicate the subgraph that results from removing the specified vertices and all edges adjacent to them.

The first collection of functions is the simplification functions, S_n for each $n \in N$. Their definition is

$$S_n(H) = \begin{cases} H - \{n\} & \text{if } \deg_H(n) < k \\ H & \text{otherwise} \end{cases}$$

We can see by inspection that these functions are monotonic and nonincreasing. This means we can define another function, $R(H)$, the residue of H, as the greatest common fixed point bounded by H of the functions S_n, and can compute it by an algorithm that applies the functions in any fair order:

THEOREM 1. *For any graph H, the functions $\{S_n \mid n \in N\}$ have a greatest common fixed point bounded by H, $R(H)$. That is, $R(H) \sqsubseteq H$, $S_n(R(H)) = R(H)$ for each $n \in N$, and we have $F \sqsubseteq R(H)$ for any fixed point, F, that also has the properties $F \sqsubseteq H$ and $S_n(F) = F$ for all $n \in N$. Moreover, the algorithm shown in Figure 1 computes $R(H)$, independent of the choices made at each step.*

PROOF. Because the functions S_n are monotonic and nonincreasing, and the subgraph partial order is finite, and hence necessarily well-founded (i.e., lacking infinite chains), the theorem directly follows from the corresponding general theorem about monotonic nonincreasing functions on well-founded partial orders. That more general theorem can be proved by concentrating on the algorithm. Because the functions are nonincreasing, the iterates, R_i, form a descending chain. Because the partial order is well-founded, this chain must terminate at a fixed point. Because the functions are monotonic, all fixed points bounded by the starting value of the iteration, H, inductively remain bounded by all iterates, and hence in particular by the terminating fixed point, $R(H)$. Thus $R(H)$ is the greatest common fixed point bounded by H. Since this is necessarily unique, the choices made at each step of the iteration can have no impact on the result. □

The significance of the residue function, R, is that if $R(G) = \langle P, P \times P \rangle$ (in which case, we say that G is reducible), then G is k-colorable, and if not, spilling can be limited to $R(G)$. It is worth noting for later reference that the function R is itself monotonic and nonincreasing. We'll also need a closely related function, σ, which finds the "significant degree" subgraph of a graph: $\sigma(H) = H - \{n \in N \mid \deg_H(n) < k\}$. Notice that σ acts on any particular graph as some composition of S_n functions, so by Theorem 1, $R \circ \sigma = R$.

Continuing with the other key transformation, coalescing, we'll define a collection of coalescing functions, $C_{n,v}$, for each $n \in N$ and $v \in V$ with $n \neq v$. These functions operate on an arbitrary graph $H = \langle V', E' \rangle$ with $\langle P, P \times P \rangle \sqsubseteq H \sqsubseteq \langle V, V \times V \rangle$. Their definition is

$$C_{n,v}(\langle V', E' \rangle) = \begin{cases} \langle V', E' \rangle & \text{if } (n, v) \in E' \\ \langle V' - \{n\}, E' - \{(t, n) \in E'\} \cup \{(t, v) \mid (t, n) \in E'\} \rangle & \text{otherwise} \end{cases}$$

In considering the impact of these coalesces, we use four different definitions of the safety of coalescing n into v in the graph H; we call these $\text{WeakSubSafe}_H(n, v)$, $\text{StrongSubSafe}_H(n, v)$, $\text{WeakEquiSafe}_H(n, v)$, and $\text{StrongEquiSafe}_H(n, v)$:

$$\begin{aligned} \text{WeakSubSafe}_H(n, v) &\iff R(C_{n,v}(H)) \sqsubseteq R(H) \\ \text{StrongSubSafe}_H(n, v) &\iff R(C_{n,v}(H)) \sqsubseteq R(H) \wedge (v \notin R(C_{n,v}(H)) - P) \\ \text{WeakEquiSafe}_H(n, v) &\iff R(C_{n,v}(H)) = R(H) \\ \text{StrongEquiSafe}_H(n, v) &\iff R(C_{n,v}(H)) = R(H) \wedge (v \notin R(H) - P) \end{aligned}$$

All four forms of safety are sufficient to show that if H is reducible, and hence k-colorable without spilling by the coloring heuristic, then the same is true even after the coalesce. Where the four forms of safety guarantee differ is in the assurances they provide with regard to graphs that may cause the heuristic to spill.

A StrongEquiSafe coalesce will leave completely unchanged the residual graph that the heuristic has left when it is first faced with the possibility of spilling. Thus, if the heuristic proceeds to spill, we can conclude, without needing to know anything about the spill-choice heuristic used, that the exact same nodes will be spilled as without the coalesce. Moreover, there may be some way (perhaps outside the basic coloring heuristic) to color the residual graph without spilling; for example, if the residual graph is small enough a backtracking search may be feasible. The key point is that whatever mechanism exists for coping with the residual graph, it will be presented with the exact same problem after a StrongEquiSafe coalesce as without that coalesce.

A StrongSubSafe coalesce can't provide this absolute of an assurance, but as the residual graph it leaves is no worse than without the coalesce, many reasonable strategies for dealing with the residual graph can be expected to perform no worse than without the coalesce.

For the weak forms of the safety guarantees, the same is nearly true—the residual graph is still a subgraph of what it would be without the coalesce, and in the case of WeakEquiSafe is even the identical residual graph. However, since the coalesced node, v, is non-precolored and remains in the residual graph, it could be chosen to spill. (We assume precolored nodes are never spilled.) If v is spilled, the cost of that spill will almost certainly be worse than it would have been without the coalesce. Thus we have less reason than we would for strongly safe coalesces to assume that the total spill cost won't have increased.

We can now define the four predicates representing the four criteria for when it is conservative (in varying senses) to coalesce n into v in the graph H:

$$
\begin{aligned}
\mathrm{FullBriggs}_H(n,v) &\iff \deg_{\sigma(C_{n,v}(H))}(v) < k \\
\mathrm{FullGeorge}_H(n,v) &\iff \{t \in \mathrm{adj}_H(n) \mid \deg_H(t) \geq k\} \subseteq \mathrm{adj}_H(v) \\
\mathrm{LimitedBriggs}_H(n,v) &\iff |\{t \in \mathrm{adj}_H(n) \cup \mathrm{adj}_H(v) \mid \deg_H(t) \geq k \vee t \in P\}| < k \\
\mathrm{LimitedGeorge}_H(n,v) &\iff \mathrm{FullGeorge}_H(n,v) \wedge \deg_{\sigma(H)}(n) < k
\end{aligned}
$$

These formal definitions may be easier to understand in conjunction with the informal versions given for FullBriggs, FullGeorge, and LimitedBriggs in Section 1. The fourth conservative coalescing criterion, LimitedGeorge, is new here; the analogy between it and LimitedBriggs is made most clear by the theorems in the subsequent sections.

We now have eight predicates that can hold for a pair of nodes (n, v) in a graph H, namely the four conservative coalescing criteria and the four safety standards. Thus there are 64 implications of the form $(\forall n \in N, v \in V)\mathrm{Predicate}(n, v) \implies \mathrm{Predicate}'(n, v)$. In this paper, we will prove or disprove each of these, at least indirectly. In fact, we will actually prove or disprove each of 128 implications, because it can matter whether v is in N or P, and so we will address these cases separately when necessary. Luckily nowhere near this many proofs are actually needed, because many implications follow by transitivity from others (as well as because in some cases it is irrelevant whether v is in N or P).

We divide the possible implications into four categories, depending on whether each of the predicates Predicate and Predicate$'$ is a conservativeness criterion or a safety standard. We term an implication of one conservativeness criterion by another a power theorem, and in the next section we prove all of these that hold. In the section after that one, we prove all the safety theorems that hold, which is our term for an implication of a safety standard by a conservativeness criterion. Two categories remain: implications of conservativeness criteria by safety standards, and implications of one safety standard by another. The former of these categories we call completeness theorems, but none of these hold (which is why the criteria are all deemed "conservative," i.e., safe but not complete). The latter category remains nameless, because these theorems are sufficiently obvious that we don't bother to prove them. (Each strong form of safety implies the corresponding weak form, and for coalesces with precolored nodes, vice versa. Each equigraph form of safety implies the corresponding subgraph form.)

After the two sections in which implications are proven (power and safety theorems), we use counterexamples to show that no additional implications hold beyond those transitively implied (this includes showing the absence of any completeness theorems). We also present a summary, in diagrammatic form, showing all the implications that hold.

3. POWER THEOREMS

It is clear by inspection that for both the Briggs criteria and the George criteria, the limited form of the criterion implies the full form; we will omit the proofs of these trivial power theorems. This leaves the relationship between the Briggs criteria and the George criteria. For coalescing a pair of non-precolored nodes, the Briggs criteria are of incomparable power with the George criteria, as we'll show in the section on counterexamples. Therefore, we are left with the problem of establishing their relative power in the case of coalescing with a precolored node; the following theorems accomplish this.

THEOREM 2. *For all* $n \in N$ *and* $p \in P - adj_H(n)$*,* $FullBriggs_H(n,p) \Longrightarrow FullGeorge_H(n,p)$*.*

PROOF. Because the precolored nodes, P, form a k-clique, we can conclude from the premise $\deg_{\sigma(C_{n,p}(H))}(p) < k$ that p is not adjacent to any non-precolored nodes of degree $\geq k$ in $C_{n,p}(H)$. This in turn implies that in the graph H, the only non-precolored nodes of degree $\geq k$ that n can be adjacent to are ones that are of degree k and also adjacent to p, so that they have degree $k-1$ after the coalesce. Another consequence of the fact that P forms a clique is that any precolored node adjacent to n must necessarily be adjacent to p as well. Thus we have shown that $\{t \in \mathrm{adj}_H(n) \mid \deg_H(t) \geq k\} \subseteq \mathrm{adj}_H(p)$, i.e., that $\mathrm{FullGeorge}_H(n,p)$ holds as well. □

THEOREM 3. *For all* $n \in N$ *and* $p \in P - adj_H(n)$*,* $LimitedBriggs_H(n,p) \Longrightarrow LimitedGeorge_H(n,p)$*.*

PROOF. Because LimitedBriggs implies FullBriggs, by Theorem 2 we know it also implies FullGeorge. Thus we only need to additionally show that $deg_{\sigma(H)}(n) < k$. Because P forms a k-clique, in order for $\mathrm{LimitedBriggs}_H(n,p)$ to hold, neither n nor p may have any non-precolored neighbors in H of degree $\geq k$ (not even degree k mutual neighbors); further, n and p must not themselves be adjacent. Thus, we conclude that $\mathrm{adj}_{\sigma(H)}(n) \subseteq P - \{p\}$, establishing our result. □

4. SAFETY THEOREMS

The above power theorems provide some motivation for considering safety theorems. In particular, we showed that for coalesces with precolored nodes, the George criteria are more powerful than the respective Briggs criteria; this raises the question of whether they achieve equally strong safety results. The below theorems (together with the counterexamples in the next section) answer this in the affirmative, as well as characterizing the situation with regard to coalescing pairs of non-precolored nodes.

THEOREM 4. *For all* $n \in N$ *and* $v \in V - adj_H(n)$*,* $FullGeorge_H(n,v) \Longrightarrow WeakSubSafe_H(n,v)$*.*

PROOF. Clearly $C_{n,v}(H)$ has no nodes that aren't in H, and the only edges it has that aren't in H are the ones added in the coalesce, connecting v to each of n's neighbors in H. By the FullGeorge premise, the only added edges connect v to non-precolored nodes that have degree $< k$. Thus $\sigma(C_{n,v}(H)) \sqsubseteq H$, and so by the monotonicity of R, $R(\sigma(C_{n,v}(H))) \sqsubseteq R(H)$. But since $R \circ \sigma = R$, this says that $R(C_{n,v}(H)) \sqsubseteq R(H)$, as desired. □

THEOREM 5. *For all* $n \in N$ *and* $v \in V - adj_H(n)$*,* $LimitedGeorge_H(n,v) \Longrightarrow WeakEquiSafe_H(n,v)$*.*

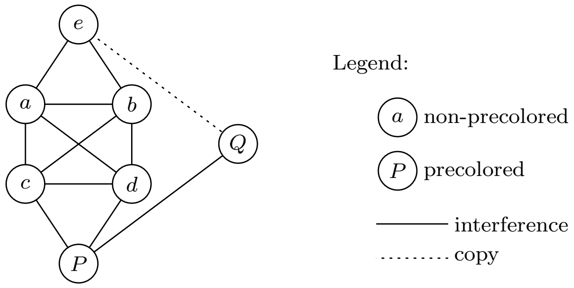

Fig. 2. With four colors but only two precolored nodes, coalescing e and Q satisfies both Briggs criteria but no safety standard; in fact, it makes a reducible graph irreducible.

Proof. By Theorem 4, $R(C_{n,v}(H)) \sqsubseteq R(H)$; thus it only remains to show that $R(H) \sqsubseteq R(C_{n,v}(H))$. By the premise, $\deg_{\sigma(H)}(n) < k$, so $n \notin S_n(\sigma(H))$. Thus $S_n(\sigma(H)) \sqsubseteq C_{n,v}(H)$, and by the monotonicity of R, $R(S_n(\sigma(H))) \sqsubseteq R(C_{n,v}(H))$. Since $R \circ S_n = R$ and $R \circ \sigma = R$, we have $R(H) \sqsubseteq R(C_{n,v}(H))$, completing the proof that $R(H) = R(C_{n,v}(H))$. □

Theorem 6. *For all* $n \in N$ *and* $v \in V - adj_H(n)$, $\mathit{FullBriggs}_H(n,v) \implies \mathit{StrongSubSafe}_H(n,v)$.

We will divide the proof into two cases, $v \in N$ and $v \in P$. The former case was given by Briggs, but to our knowledge the latter case has been overlooked until now. (It was irrelevant to Briggs, since he used conservative coalescing only to undo splitting of non-precolored live ranges.) We plug this hole in the proof using Theorems 2 and 4; this relies upon the often overlooked assumption that there are k precolored nodes. Figure 2 shows that this assumption is necessary.

Proof. When v is in P, Theorems 2 and 4 directly imply our result. For $v \in N$, the following reasoning (due to Briggs [1994; 1992]) applies. Because $\deg_{\sigma(C_{n,v}(H))}(v) < k$, we can conclude that $v \notin S_v(\sigma(C_{n,v}(H)))$. Thus, nor are any of the edges added by the coalesce, and so $S_v(\sigma(C_{n,v}(H))) \sqsubseteq H$. By the monotonicity of R, this implies that $R(S_v(\sigma(C_{n,v}(H)))) \sqsubseteq R(H)$, and so since $R \circ S_v = R$ and $R \circ \sigma = R$, we have $R(C_{n,v}(H)) \sqsubseteq R(H)$, as desired; we also have $v \notin R(C_{n,v}(H))$. □

Theorem 7. *For all* $n \in N$ *and* $v \in V - adj_H(n)$, $\mathit{LimitedBriggs}_H(n,v) \implies \mathit{StrongEquiSafe}_H(n,v)$.

Proof. All we need to add to Theorem 6 is that $R(H) \sqsubseteq R(C_{n,v}(H))$. From the definition of LimitedBriggs, we can see that $\deg_{\sigma(H)}(n) < k$. Therefore the same reasoning used in the proof of Theorem 5 again allows us to conclude that $R(H) \sqsubseteq R(C_{n,v}(H))$, establishing our result. □

5. COUNTEREXAMPLES

In order to show that Theorems 2–7, together with the trivial implications previously stated, cover all the bases, we can use counterexamples to show that no other implications between our predicates are generally true (other, of course, than those

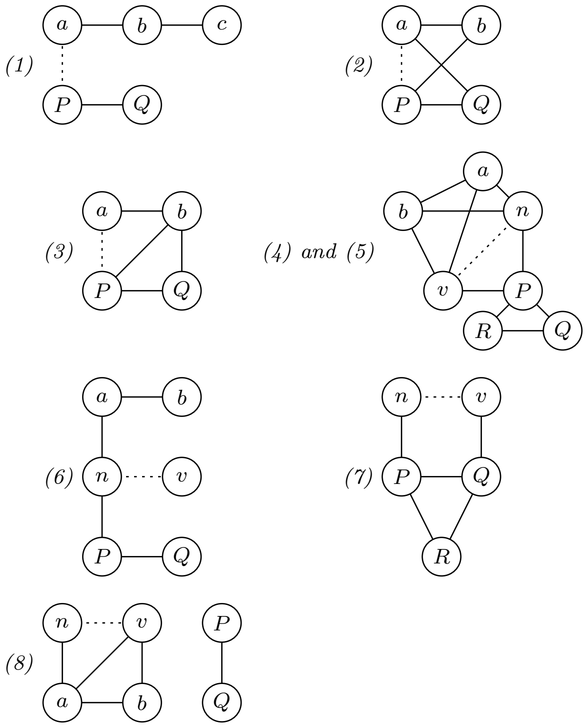

Fig. 3. Examples showing that our theorems capture all implications between the eight predicates under consideration.

implied by reflexivity and transitivity). This is needed to substantiate a number of claims. For example, we claimed that the George criteria are *strictly* more powerful than the corresponding Briggs criteria for coalesces with precolored nodes. Only now will we finish substantiating that claim, by ruling out the possibility that the criteria are equivalent.

Theorem 8. *For each of the following statements, there exist a graph H and nodes $n \in N$ and $v \in V - adj_H(n)$ that make the statement true:*

(1) $v \in P \wedge \mathit{StrongEquiSafe}_H(n, v) \wedge \overline{\mathit{FullGeorge}_H(n, v)}$
(2) $v \in P \wedge \mathit{FullBriggs}_H(n, v) \wedge \overline{\mathit{WeakEquiSafe}_H(n, v)}$
(3) $v \in P \wedge \mathit{LimitedGeorge}_H(n, v) \wedge \overline{\mathit{FullBriggs}_H(n, v)}$
(4) $v \in N \wedge \mathit{FullGeorge}_H(n, v) \wedge \overline{\mathit{WeakEquiSafe}_H(n, v)}$
(5) $v \in N \wedge \mathit{FullBriggs}_H(n, v) \wedge \overline{\mathit{WeakEquiSafe}_H(n, v)}$
(6) $v \in N \wedge \mathit{StrongEquiSafe}_H(n, v) \wedge \overline{\mathit{FullBriggs}_H(n, v)}$
(7) $v \in N \wedge \mathit{LimitedBriggs}_H(n, v) \wedge \overline{\mathit{FullGeorge}_H(n, v)}$
(8) $v \in N \wedge \mathit{LimitedGeorge}_H(n, v) \wedge \overline{\mathit{StrongSubSafe}_H(n, v)}$

Proof. Figure 3 shows an example for each of the statements listed above, labeled correspondingly. In each example, nodes with uppercase labels are precolored while those with lowercase labels are non-precolored. The solid lines indicate the

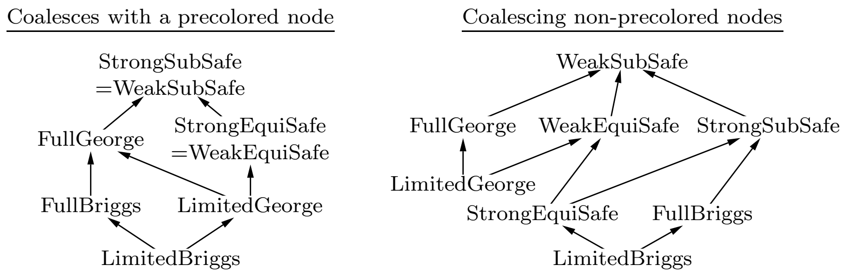

Fig. 4. A summary of all implications between conservativeness and safety predicates

graph's interference edges, while the dotted line indicates the coalesce under consideration (i.e., it identifies the n and v of the theorem). The number of colors, k, is in each case implicit in the number of precolored nodes. □

One can check that all possible implications between pairs of predicates drawn from the eight we are considering have been either proven or disproven by the theorems presented above, both in the case of coalescing with a precolored node and in the case of coalescing two non-precolored nodes.[2] We are therefore now ready to summarize the full structure of implications between the predicates. The diagrams in Figure 4 provide this summary; one predicate implies another if and only if there is a path between them.

6. COALESCES THAT REDUCE SPILLING

We have shown that there are SubSafe coalesces (whether Strong or Weak) that aren't EquiSafe; this means that we can use such a coalesce to shrink the residual subgraph, and perhaps even convert an irreducible graph into a reducible one. This observation, that coalescing can actually reduce spilling rather than increasing it, was made explicit independently by Vegdahl [1999] and by Park and Moon [1998].[3] For example, the graph shown in Figure 3 as an illustration of cases 4 and 5 is irreducible. Using Chaitin's original pessimistic coloring heuristic, spill code would necessarily be introduced. Using Briggs's optimistic coloring heuristic, the spill code might or might not be avoided, all depending on the arbitrary choices made. On the other hand, coalescing nodes n and v renders the graph reducible. These nodes don't interfere, and coalescing them is conservative relative to both the FullBriggs and the FullGeorge criteria. Therefore, coalescing them might be a smart move, even if there is no copy instruction connecting them.

Note that by definition, an EquiSafe coalesce will never have this favorable consequence. As such, although in principle EquiSafety is slightly safer than SubSafety, in practice the dominant effect is likely to be a loss of favorable coalesces. This suggests that Appel and George's limited version of the Briggs test is a mistake; the original, full version should be used, so as not to rule out shrinking the residual subgraph and thus reducing spilling.

[2]A prolog program to do this checking is available from the author. The author and a reviewer each checked by hand as well.

[3]Linda Torczon and Keith Cooper report in private correspondence that this fact was "well known in the field prior to the work by Vegdahl and by Park and Moon."

7. CONCLUSIONS

Although implementation concerns originally motivated the substitution of the FullGeorge criterion for the FullBriggs criterion when testing the conservativeness of coalesces with precolored nodes, we have shown that in fact this substitution has a stronger motivation as well. Namely the FullGeorge criterion identifies a superset of the coalesces that the FullBriggs criterion does, while maintaining an equally strong safety guarantee. We've also clarified the impact of the switch from FullBriggs to LimitedBriggs, and showed how a similar alteration can also be made to the George test. We also showed that for coalescing two non-precolored nodes, the George and Briggs tests are of incomparable power, suggesting that some leverage could be gained by using both. However, we showed that there would be some cost in safety for doing so, in that the George tests do not assure that the coalesced node will not itself be a spill candidate.

ACKNOWLEDGMENTS

Andrew Appel, Lal George, Linda Torczon, Keith Cooper, and the anonymous reviewers provided helpful comments on earlier drafts of this paper.

REFERENCES

Appel, A. W. 2002. *Modern Compiler Implementation in Java*, 2nd ed. Cambridge University Press, Cambridge, UK.

Briggs, P. 1992. Register allocation via graph coloring. Tech. Rep. CRPC-TR92218, Center for Research on Parallel Computation, Rice University. Ph.D. Dissertation.

Briggs, P. 1993. ILOC register allocator contributed to the DIMACS graph challenge. ftp://dimacs.rutgers.edu/pub/challenge/graph/contributed/lewandowski/. Edited by Gary Lewandowski.

Briggs, P., Cooper, K. D., and Torczon, L. 1994. Improvements to graph coloring register allocation. *ACM Trans. Program. Lang. Syst. 16,* 3 (May), 428–455.

Chaitin, G. J. 1982. Register allocation and spilling via graph coloring. In *Proceedings of the ACM SIGPLAN 82 Symposium on Compiler Construction.* Association for Computing Machinery, New York, 98–105.

Chaitin, G. J., Auslander, M. A., Chandra, A. K., Cocke, J., Hopkins, M. E., and Markstein, P. W. 1981. Register allocation via coloring. *Comput. Lang. 6,* 1, 47–57.

George, L. and Appel, A. W. 1996. Iterated register coalescing. *ACM Trans. Program. Lang. Syst. 18,* 3 (May), 300–324.

Park, J. and Moon, S.-M. 1998. Optimistic register coalescing. In *Proceedings of the 1998 International Conference on Parallel Architectures and Compilation Techniques.* IEEE Computer Society, Washington, DC, 196–204.

Vegdahl, S. R. 1999. Using node merging to enhance graph coloring. In *Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI).* Association for Computing Machinery, New York, 150–154.

Received date will go here in published version.