まわりみちの威力

千代 英一郎　(株)日立製作所

〔受賞論文〕
・ Deductive System による C プログラムのポインタ解析
・ 千代英一郎((株)日立製作所)
・ 情報処理学会論文誌，Vol.47, No.SIG2 (PRO28), pp.1-17 (2006)

本論文は「実用的な（大規模な C プログラムを扱えるような)ポインタ解析を簡単に作る方法」について述べたものである．ポインタ解析とは，ポインタの指示先を静的に求める解析で，コンパイラがポインタを利用するプログラムを最適化する際には欠かせない処理である．また，C プログラムの信頼性や脆弱性の検証にも有用であり，検証時にポインタ解析をあわせて行うことで，検証精度を大きく改善することができる．

ポインタ解析の実用上の課題に，大規模なプログラムを解析する際の解析時間の爆発がある．特にプログラム全体にまたがって解析を行う場合，素朴な実装ではプログラムの規模が大きくなるとすぐに解析が終わらなくなってしまう．そのため，解析効率を向上させる手法が特に 1990 年代後半から盛んに研究されてきている．我々がポインタ解析の研究に着手した頃は，解の計算をポインタ指示先集合に関する制約グラフ上で行う方法が注目されており，計算を効率化するためのさまざまな工夫が提案されていた．

当初，我々も同じアプローチを取っていたが，処理系依存な部分も含めた C の言語機能のすべてを制約グラフ上の処理に綺麗に対応させるのは難しく，そこにさらにさまざまな効率化を加えていった結果，実装はきわめて複雑なものになってしまった．

経験された方なら賛同していただけると思うが，大規模なプログラムに対する解析結果の正しさの確認は至難の業である．我々は，このままでは解析の実装品質を製品レベルにまで高めることが難しいと判断し，原点に立ち戻り再検討を行った結果，ポインタ解析を deductive system を媒介として関係代数演算に帰着させる，という提案手法を着想するに至った．

関係代数演算が大規模データの処理に適していることは関係データベース（RDB）の分野で実証されているが，その適用にあたっては，解析対象プログラムの表現を従来とは大きく変えなければならない．特に，プログラムの内包するすべての参照関係を，人工的な識別子の導入等により「表」として表現する必要がある．

たとえば，C の式は木構造データとして，親から子(部分式）への物理的な参照（アドレス参照）によって表現するのが自然である．ポインタ解析では，式の値を求めるために，その部分式を再帰的にたどる（たとえば **p の値を求めるために，その部分式 *p および p をたどる）という処理を何度も行うが，これは部分式への物理的な参照を用いて効率的に行える．一方，関係代数演算を用いる場合，式ごとに式識別子を割り当てた上で，その親子関係を式識別子の表によって表現する必要がある．式をたどる処理はこの表を引きながら行うため，物理的な参照に比べ，1 回の参照あたりの効率は当然悪くなる．

このような表現は業務情報システムなどでは日常的に利用されているが，ポインタ解析では大量の式を何度も繰り返したどる必要があるため，果たして性能が出るのだろうか，という不安が当初は常につきまとっていた．ある意味，1970 年に Codd が RDB の概念を提案したときに当時の技術者たちが感じていた実用性に対する疑念や不信を追体験していたと言えるかもしれない．

慣れ親しんだ手法からの脱却には大きな心理的抵抗があったが，30 年以上 RDB の分野で積み重ねられてきた理論や技術の効果は絶大であり，初歩的な indexing と join に関する手法を適用しただけの簡単な実装で，従来手法を上回る性能を得ることができた．本論文の結果が，多くの(？)ポインタ解析の研究者や開発者の一助となれば幸いである．

なお，実際に C のポインタ解析を開発する必要に迫られた方は，我々が情報処理学会で以前に発表した論文「型安全でない C プログラムのポインタ解析」を合わせて参照されると，待ち受けているであろうデバッグ地獄を未然に回避できるかもしれない．読み通すためには無限の忍耐心が要請されるため，とてもご一読くださいとは言えないが，ざっと眺めていただければ「C 固有の落とし穴」は把握可能かと思われる．

最後に，荒削りな論文からその本質的な内容を読み解き評価してくださった査読者ならびにプログラミング研究会の皆様に深く感謝する．

（平成 19 年 4 月 27 日受付）

千代英一郎(正会員)　eiichiro.chishiro.qp@hitachi.com
1999 年東京大学大学院工学系研究科情報工学専攻修士課程修了．同年日立製作所(株)入社．システム開発研究所にてコンパイラの研究開発に従事．