

## ⑩ Towards an Efficient Compile-Time Granularity Analysis Algorithm

X.Zhong, E.Tick\*, S.Duvvuru, L.Hansen, A.V.S.Sastry, R.Sundararajan  
(Univ. of Oregon,米国)

### 発表要旨

並行ロジックプログラムに対し、新しい粒度解析スキームを提案する。あらゆるゴールのコストを詳細に評価するのではなく、実行時のゴールのコストを一つ調べ、活性化ゴールの相対コストを効率的に、良い近似で調べられる。プログラムの呼び出しグラフから活性化ゴールとそのサブゴールのコストの関係を評価し、帰納的手続きには試行回数パラメータを導入した。この結果、簡単なベンチマークプログラムのコストの見積りに対して9割程の正確さが実証された。これはあらゆるプログラムに適用可能で、静的解析方法より詳細なコスト見積りができる。そしてコストの絶対値を解析する方法より実行時間のオーバーヘッドを少なくできる。

### 質疑応答

質問：議論中のプログラムで、付加パラメータ `qsort` というのは普通の `qsort` と同じですね。では、初期重みを与えるメカニズムはあるのでしょうか？

回答：はい、大変良い質問です。初期重みは議論中の問題です。オリジナルのアイデアは、できる限り大きな整数を与えてしまうことです。でも、また2つ問題が生じてしまいます。単調減少しているの でいつか0になってしまうこと、そして大問題は、評価は初期値に大きく依存した近似になってしまうことです。

質問：プログラムサイズを計算するのに重みを渡すことにした理由がわかりません。

回答：数値的重みをプログラム中の子に渡して個別に計算できるからです。

質問：そのままある場所からある場所へのサイズを計算した方が簡単そうですが。

回答：これは単にプログラムサイズだけを計算しているのではなく、構造を持った複雑なプログラムの実行時間を意味的に解釈して求めているのです。

質問：無限ストリーム等では、正確に計算できなく、少なく見積りそうですが。

回答：そういうプログラムでは、サイズよりも他に問題が起こりそうですね。