Open Pattern Matching for C++

Parasol
Smarter computing.
Texas A&M University

Yuriy Solodkyy · Gabriel Dos Reis · Bjarne Stroustrup

λ-calculus in C++

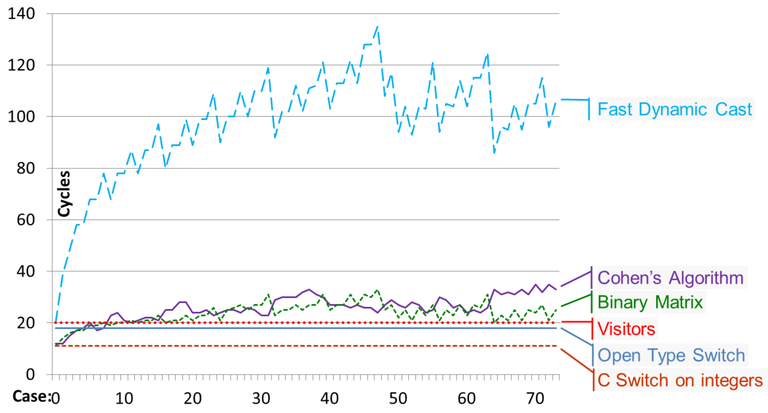
```
struct Term       { virtual ~Term() {} };
struct Var : Term { std::string name; };
struct Abs : Term { Var*  var;  Term* body; };
struct App : Term { Term* func; Term* arg;  };

Term* eval(Term* t)
{
  var<const Var&>  v;
  var<const Term&> t1,t2;

  Match(t)
  {
    Case(C<Var>())                  return &match0;
    Case(C<Abs>())                  return &match0;
    Case(C<App>(C<Abs>(&v,&t1),&t2)) return eval(subs(t1,v,t2));
    Otherwise()                     std::cerr << "Invalid term";
  }
  EndMatch

  return nullptr;
}

bool operator==(const Term& left, const Term& right)
{
  var<std::string> s;
  var<const Term&> v,t,f;

  Match( left          , right          )
  {
    Case(C<Var>(s)     , C<Var>(+s)     ) return true;
    Case(C<Abs>(&v,&t) , C<Abs>(&+v,&+t)) return true;
    Case(C<App>(&f,&t) , C<App>(&+f,&+t)) return true;
    Otherwise()                           return false;
  }
  EndMatch
}
```

Generalized n+k Patterns

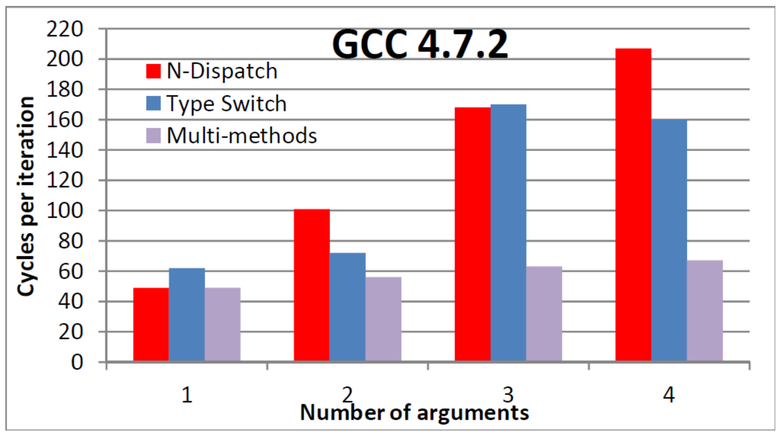
```
double power(double x, int n)
{
    var<int> m;

    Match(n)
    {
      Case(0)      return 1.0;
      Case(1)      return x;
      Case(2*m)    return   sqr(power(x,m));
      Case(2*m+1) return x*sqr(power(x,m));
    }
    EndMatch
}
```

$$x^n = \begin{cases} 1 & n = 0 \\ x & n = 1 \\ (x^m)^2 & n = 2m \\ x(x^m)^2 & n = 2m+1 \end{cases}$$

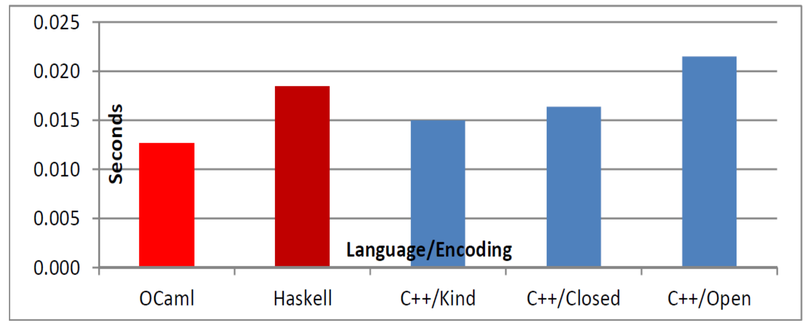
Comparison to Alternatives



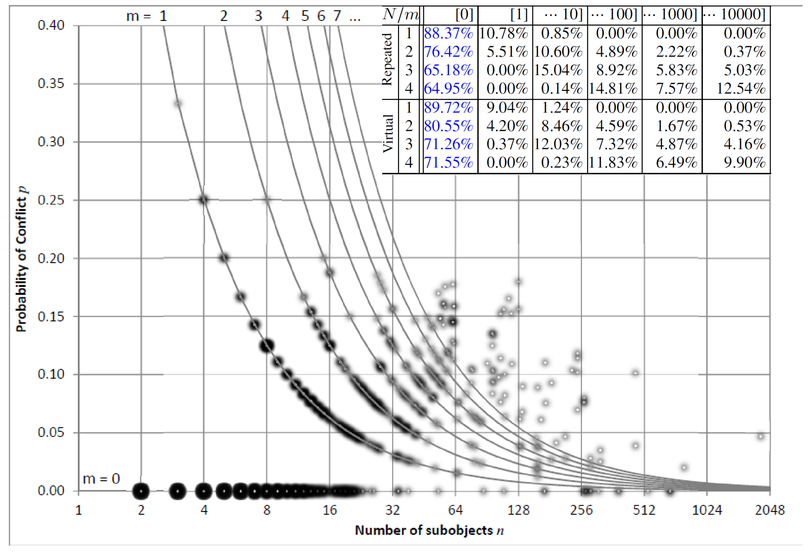
Comparison to Multiple Dispatch



Comparison with OCaml & Haskell



Efficiency of Hashing



	N/m	[0]	[1]	··· 10]	··· 100]	··· 1000]	··· 10000]
Repeated	1	88.37%	10.78%	0.85%	0.00%	0.00%	0.00%
	2	76.42%	5.51%	10.60%	4.89%	2.22%	0.37%
	3	65.18%	0.00%	15.04%	8.92%	5.83%	5.03%
	4	64.95%	0.00%	0.14%	14.81%	7.57%	12.54%
Virtual	1	89.72%	9.04%	1.24%	0.00%	0.00%	0.00%
	2	80.55%	4.20%	8.46%	4.59%	1.67%	0.53%
	3	71.26%	0.37%	12.03%	7.32%	4.87%	4.16%
	4	71.55%	0.00%	0.23%	11.83%	6.49%	9.90%

Performance Evaluation

42% pattern matching is faster than visitors.
42% visitors are faster than pattern matching

	Open						Closed					
	G++		MS Visual C++				G++		MS Visual C++			
	Lnx	Win	PGO		w/o PGO		Lnx	Win	PGO		w/o PGO	
	x86-32	x86-32	x86-32	x86-64	x86-32	x86-64	x86-32	x86-32	x86-32	x86-64	x86-32	x86-64
REP	16%	14%	1%	18%	2%	37%	124%	122%	100%	41%	76%	37%
SEQ	56%	12%	48%	22%	2%	46%	640%	467%	29%	15%	30%	10%
RND	56%	0%	9%	19%	5%	46%	603%	470%	35%	20%	32%	6%
Forwarding REP	33%	22%	8%	17%	24%	36%	53%	49%	24%	11%	20%	36%
Forwarding SEQ	55%	233%	135%	135%	193%	32%	86%	290%	48%	139%	12%	24%
Forwarding RND	78%	25%	3%	4%	13%	23%	88%	33%	8%	1%	18%	16%

Y.Solodkyy, G.Dos Reis, B.Stroustrup. "Open and Efficient Type Switch for C++" Proceedings OOPSLA'12, October 19-26, 2012, Tucson, Arizona, USA

Partially supported by NSF grants CCF-0702765, CCF-1043084, and CCF-1150055