Factorization with SAT – classical propositional calculus as a programming environment

April 20, 2004

Mateusz Srebrny
<mati@students.mimuw.edu.pl>

Plan

- SAT
- RSA
- goal – translation
- details of the translation
- experimental results
- conclusions, questions

SAT

- satisfiability of classical propositional formulas
- *Is a given formula satisfiable?*
  - *Find a satisfying valuation for a given formula.*
  - *Find all satisfying valuations.*
- NP-complete (Cook 1971)
- exponential time required in general

SAT solvers

- many SAT instances can be tested efficiently
- many competing algorithms/implementations
- Davies-Putnam Procedure, 1960
- CNF: conjunctive normal form, conjunction of disjunctions of literals

Well-known SAT solvers

- zChaff (http://www.ee.princeton.edu/ chaff/zchaff.php)
- BerkMin (http://eigold.tripod.com/BerMin.html)
- Stalmarck's PROVER
- GSAT, SATO, MAXSAT, WalkSat, GRASP

RSA (1977) – Factorization

- two factors: $n = pq$, p, q prime
- historical anecdote
- p, q are cryptografically strong (detailed requirements)
  - p, q are odd
  - n large, p, q are of similar length
- *"Handbook of Applied Cryptography"*, Menezes, et al.

Complexity of factorization

- noone knows any polynomial time algotithm in general (asympthotically)
- all non-polynomial time algorithms are not feasible
- starting point for complexity theory – non-polynomial implies infeasible
- today – about 2^{60} instructions is an upper bound

RSA challenge

http://www.rsasecurity.com/rsalabs/challenge/factoring/numbers.html

- **512** – Factored, 1999
  7.5 month, about 300 computers (including one Cray)
  general number field sieve
- **576** – Factored, December 3, 2003

  188198812920607963838697239461650439807163563379417382700763356422988859715234665485319060606504743045317388013033967161996923212057340318795506569962213051687593076502570591
- **640** – Not factored, reward $10K

  3107418240490043721350750035888567930037346022842754572016194882320644051808150455634682967172328678243791627283803341547107310850191954852900733772482278352574238645401469173660247765234660 9
- 1000-bit is safe for now

Programming paradigm

- to solve a problem:
  1. translate the problem to SAT
  2. let SAT checker solve its satisfiability
- for a given n,
  generate such a propositional formula
  that its satisfying valuation encodes two integer factors p and q of n

Represention of integers

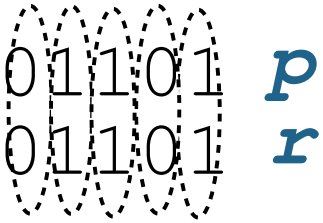
- l-bit integer $p \longrightarrow l$ propositional variables

$$P_0, ..., P_{l-1}$$

- e.g. $13 = (1101)_2$ is represented by formula $P = 13$:

$$\neg P_4 \wedge P_3 \wedge P_2 \wedge \neg P_1 \wedge P_0$$

$R = P$ **represents** $r = p$

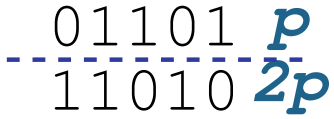


$$\bigwedge_{i=0}^{l-1}(R_i \wedge P_i) \vee (\neg R_i \wedge \neg P_i)$$

(CNF):

$$\bigwedge_{i=0}^{l-1}(R_i \vee \neg P_i) \wedge (P_i \vee \neg R_i)$$

$R = 2P$ **represents** $r = 2p$



$$\neg R_0 \wedge \bigwedge_{i=1}^{l-1} (R_i \wedge P_{i-1}) \vee (\neg R_i \wedge \neg P_{i-1})$$

(CNF):

$$\neg R_0 \wedge \bigwedge_{i=1}^{l-1} (R_i \vee \neg P_{i-1}) \wedge (\neg R_i \vee P_{i-1})$$

$R = BP$ **represents** $r = bp$

```
  01101 p
      1 b
-------
  01101 bp
```

$$\bigwedge_{i=0}^{l-1}((R_i \wedge (B \wedge P_i)) \vee (\neg R_i \wedge \neg(B \wedge P_i)))$$

(CNF):

$$\bigwedge_{i=0}^{l-1}((B \vee \neg R_i) \wedge (P_i \vee \neg R_i) \wedge (R_i \vee \neg B \vee \neg P_i))$$

$R = P + Q$ represents $r = p + q$ (carry)

```
  011010 c
   01101 p
+  00101 q
-----------
```

$(C_0, C_1, \ldots, C_l)$ – carry bits

for $i > 0$:

$$(C_i \wedge ((C_{i-1} \wedge P_i) \vee (C_{i-1} \wedge Q_i) \vee (P_i \wedge Q_i)))$$
$$\vee$$
$$(\neg C_i \wedge ((\neg C_{i-1} \wedge \neg P_i) \vee (\neg C_{i-1} \wedge \neg Q_i) \vee (\neg P_i \wedge \neg Q_i)))$$

(CNF):

$$(\neg C_i \vee P_i \vee C_{i-1}) \wedge (\neg C_i \vee P_i \vee Q_i) \wedge$$
$$(\neg C_i \vee Q_i \vee C_{i-1}) \wedge (C_i \vee \neg P_i \vee \neg C_{i-1}) \wedge$$
$$(C_i \vee \neg P_i \vee \neg Q_i) \wedge (C_i \vee \neg Q_i \vee \neg C_{i-1})$$

$R = P + Q$ (result)

```
  011010  c
   01101  p
+  00101  q
  ------------
  10010 p+q
```

$$(R_i \wedge ((C_{i-1} \wedge \neg P_i \wedge \neg Q_i) \vee (\neg C_{i-1} \wedge P_i \wedge \neg Q_i) \vee$$
$$(\neg C_{i-1} \wedge \neg P_i \wedge Q_i) \vee (C_{i-1} \wedge P_i \wedge Q_i)))$$
$$\vee$$
$$(\neg R_i \wedge ((C_{i-1} \wedge P_i \wedge \neg Q_i) \vee (\neg C_{i-1} \wedge P_i \wedge Q_i) \vee$$
$$(C_{i-1} \wedge \neg P_i \wedge Q_i) \vee (\neg C_{i-1} \wedge \neg P_i \wedge \neg Q_i)))$$

(CNF):

$$(R_i \vee Q_i \vee P_i \vee \neg C_{i-1}) \wedge (R_i \vee Q_i \vee \neg P_i \vee C_{i-1}) \wedge$$
$$(R_i \vee \neg Q_i \vee P_i \vee C_{i-1}) \wedge (R_i \vee \neg Q_i \vee \neg P_i \vee \neg C_{i-1}) \wedge$$
$$(\neg R_i \vee Q_i \vee P_i \vee C_{i-1}) \wedge (\neg R_i \vee Q_i \vee \neg P_i \vee \neg C_{i-1}) \wedge$$
$$(\neg R_i \vee \neg Q_i \vee P_i \vee \neg C_{i-1}) \wedge (\neg R_i \vee \neg Q_i \vee \neg P_i \vee C_{i-1})$$

$R = P + Q$ (the whole)

$$
\begin{aligned}
&\neg C_0 \wedge \neg C_l \wedge \\
&\textstyle\bigwedge_{i=1}^{l} (\\
&(\neg C_i \vee P_i \vee C_{i-1}) \wedge (\neg C_i \vee P_i \vee Q_i) \wedge \\
&(\neg C_i \vee Q_i \vee C_{i-1}) \wedge (C_i \vee \neg P_i \vee \neg C_{i-1}) \wedge \\
&(C_i \vee \neg P_i \vee \neg Q_i) \wedge (C_i \vee \neg Q_i \vee \neg C_{i-1}) \\
&) \\
&\wedge \\
&\textstyle\bigwedge_{i=0}^{l-1} (\\
&(R_i \vee Q_i \vee P_i \vee \neg C_i) \wedge (R_i \vee Q_i \vee \neg P_i \vee C_i) \wedge \\
&(R_i \vee \neg Q_i \vee P_i \vee C_i) \wedge (R_i \vee \neg Q_i \vee \neg P_i \vee \neg C_i) \wedge \\
&(\neg R_i \vee Q_i \vee P_i \vee C_i) \wedge (\neg R_i \vee Q_i \vee \neg P_i \vee \neg C_i) \wedge \\
&(\neg R_i \vee \neg Q_i \vee P_i \vee \neg C_i) \wedge (\neg R_i \vee \neg Q_i \vee \neg P_i \vee C_i) \\
&)
\end{aligned}
$$

$N = PQ$ represents $n = pq$

$$pq = q_0 p + q_1 2p + q_2 2^2 p + \ldots + q_{l-1} 2^{l-1} p$$

$$(S^0 = P) \wedge (\bigwedge_{i=1}^{l-1} S^i = 2S^{i-1}) \wedge$$

$$(\bigwedge_{i=0}^{l-1} M^i = Q_i S^i) \wedge$$

$$(R^0 = M^0) \wedge (\bigwedge_{i=1}^{l-1} R^i = R^{i-1} + M^i) \wedge$$

$$(R^{l-1} = N)$$

Complexity of the formula $N = PQ$ – variables

In the formula there are this many l-bit numbers represented:

- P, Q – 2
- shifts of P (S^i) – l
- products $Q_i S^i$ (M^i) – l
- sums $R^{i-1} + M^i$ (R^i) – l
- implicit carry – l

$4l + 2$ numbers $\longrightarrow$ $4l^2 + 2l$ propositional variables

Complexity of the formula – clauses

- $l-1$ shifts ($S^i = 2S^{i-1}$) – $2l-1$ clauses each
- l bit multiplications ($M^i = Q_i S^i$) – $3l-3$ each
- $l-1$ sums ($R^i = R^{i-1} + M^i$) – $14l+2$ each
- 2 equalities ($S^0 = P$, $R^0 = M^0$) – $2l$ each
- 1 equality ($R^{l-1} = n$) – l

total: $19l^2 - 13l - 1$ clauses

Optimizations of the formula

- the formula can have $l^2 + O(l)$ variables
- e.g. p, q – about $\frac{l}{2}$-bit
- e.g. S^i – i first bits are zeros

Experimental results

- RSA is out of reach of this exact method :(
- Task: to test what can be done on computers available today
- Experiments on 2GHz, 2GB RAM computer:
  - 32-bit – 15 seconds
  - 46-bit – 3 hours
  - 47-bit – 3 days wasn't enough
  - 48-bit – 33 hours
  - 49-bit – 3 days wasn't enough
  - 212-bit (random) – 10 seconds (one of the factors: 11)

The limits of what currently can be done

- What can be done with the fastest computers ($35 \cdot 10^{12}$ ops)?
- How long time is needed for 50-bit and 640-bit with the fastest computers?
- How many bits is the limit of the fastest computers?
- How to estimate the limits of large nets (e.g. the Internet)?

SAT solver dedicated for factorization?

- zChaff has neither specific structural nor number theoretic info about the formula
- probably brute force is used
- How to teach/optimize zChaff about/for the formula?

Ideas:

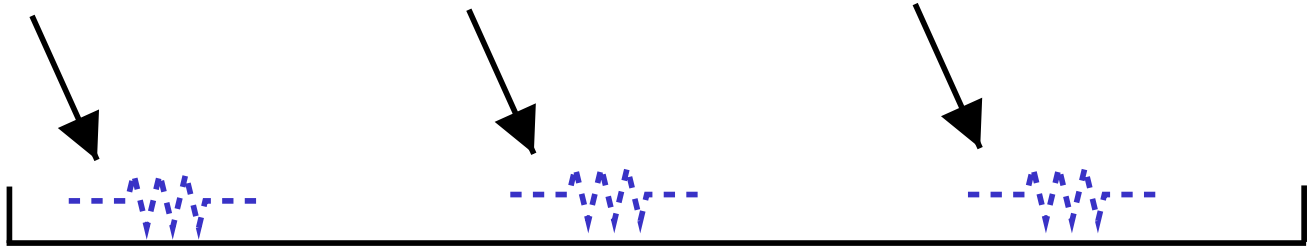
- grouping of clauses, sorting of clauses and variables
- parallelization

SAT solver dedicated for factorization?

- zChaff has neither specific structural nor number theoretic info about the formula
- probably brute force is used
- How to teach/optimize zChaff about/for the formula?

Ideas:

- grouping of clauses, sorting of clauses and variables
- parallelization



Thank you for your attention