Fully supersymmetric chains

Goto page Previous 1, 2, 3 ... 14, 15, 16

new topic post reply **Sudoku Players' Forums Forum Index -> Advanced solving techniques**

View previous topic :: View next topic

Author	Message
PIsaacson Joined: 02 Jul 2008 Posts: 311 Location: Campbell, CA	Posted: Thu Oct 29, 2009 7:53 pm Post subject: quote

Denis,

The structure involved in retaining all the path information is path_t which contains a level counter (highest level used) and up to 127 edges which equate to a given BFS/DFS depth. It takes 2 BFS/DFS edges/levels to form a lh/rh conjugate nrc pair. The level 0 entry is the z-target. Level 1 would be the first child of the z-target, which is then used as a parent that requires a strong-linked child at level 2. All even level parents only require weak linked children (any nrc linked candidate). All odd level parents mandate a strong linked child - either native or via zt-promotion. The path structure is carried in each BFS queue element, and each path is developed independantly. DFS requires only a single copy, this being one of the distinguishing features between BFS and DFS.

As the tree is traversed, there is an associated vt_parent structure that is used to provide a "where-used" or parent pointer for each child node encountered. This can be used to trace-back (there's code in the nrc_braids_print that accomplishes this) from any given child back to either the z-target or any given parent within that path. I only use this during the one-of-N mutually exclusive z-targets conflict testing, and then only to build the path that conflicted, which does not include any of the associated partials.

Normal chains/whips are just displayed using the path_t structure and this is absolutely accurate. For braids, I display the current path_t structure as in normal chains, but even though I could use the vt_parent to re-construct all partials used, I currently have not tracked which partials are involved. I've confessed to this in the other nrczt chain construction postings, and it's on my to-do list. My current priority is working on standard chains/whips to fix the problem that you found recently in my accidental use of braids conflict testing within standard chains/whips.

Cheers,
Paul

Back to top profile pm

denis_berthier

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

Posted: Sat Nov 07, 2009 7:29 am Post subject: quote edit

PIsaacson wrote:

> For braids, I display the current path_t structure as in normal chains, but even though I could use the vt_parent to re-construct all partials used, I currently have not tracked which partials are involved.

The problem may be that you are seeing braids as nets. But, what's fundamental in braids is, they have a sequential structure. The only difference with whips is that an llc can be linked to any previous rlc or to the target.
For braids, there are therefore two different parenthoods: one associated with the sequential structure and one associated with the nrc-link structure. The most important is the first.

PIsaacson wrote:

> My current priority is working on standard chains/whips to fix the problem that you found recently in my accidental use of braids conflict testing within standard chains/whips.

The corrected version you sent me some time ago seems to have solved this problem.
I ran it on the Sudogen0_1M collection (1,000,000 puzzles) and I found that your rating is higher 681 times (by 1 in most cases, by 2 in 73 cases). This is not much, but I'm quite sure you can still improve this and you're very close to a correct and very fast implementation. Unfortunately, I haven't had much time to investigate the origin of the discrepancy (apart from the example I PM'd you).

Back to top profile pm www

denis_berthier

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

Posted: Sun Nov 08, 2009 5:03 am Post subject: quote edit

denis_berthier wrote:

> I ran it on the Sudogen0_1M collection (1,000,000 puzzles) and I found that your rating is higher 681 times (by 1 in most cases, by 2 in 73 cases). This is not much, but I'm quite sure you can still improve this and you're very close to a correct and very fast implementation. Unfortunately, I haven't had much time to investigate the origin of the discrepancy (apart from the example I PM'd you).

PIsaacson wrote:

> I think we don't have agreement on what are the exact permitted techniques for scoring. When I looked at your solution log in detail, there were several steps involving triplets/pairs. That sounds like locked sets???

Right. The above discrepancy may be due to the different ratings used.
In SudoRules, I use the NRCZT rating, with naked/hidden/super-hidden rules for pairs/triplets /quads included at respective levels 2/3/4. No rating system that

would exclude these rules has any practical interest, as they are the most widely used.
I've also introduced the pNRCZT rating (for NRCZT integrists), without these rules, whose interest is mainly for evaluating the impact of the subset rules. I've shown this impact on classification is small: most cases of subsets are subsumed by nrczt chains. This justifies the integration of levels for nrczt whips and subsets. Missing this integration would be missing an important part of the theory.

In the computation I've done with your program, I unfortunately activated only the option -Xprn, which corresponds to the pNRCZT rating.
Before I try again with subsets on (-Xprln ???), do you classify the subsets at the correct 2/3/4 levels?

Back to top profile pm www

PIsaacson

Posted: Sun Nov 08, 2009 5:20 am Post subject: quote

Joined: 02 Jul 2008
Posts: 311
Location: Campbell, CA

Denis,

The version you have does not correctly score locked sets. I'm modifying the code and I'll e-mail you an updated version when I've run tests to ensure that I haven't broken anything else.

Cheers,
Paul

Back to top profile pm

denis_berthier

Posted: Mon Nov 09, 2009 1:54 pm Post subject: quote edit

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

Paul,
In order to make comparisons between comparable things, I've computed the pNRCZT rating for the 10,000 puzzles in sudogen0.
Result: identical to those I obtained with your version V2.

This version V2 is therefore very likely to be a correct implementation of nrczt-whips and it is terribly fast - faster than I imagined any implementation could be.
The bit representation is a great idea.
Again, congratulations for your work.

Back to top profile pm www

denis_berthier

Posted: Tue Nov 10, 2009 7:01 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

Paul,

Considering your difficulty with Subsets, I think a simple modification of the end of your sudoku_solve function would do the job. The way I write it may not be very smart but it is correct. In SudoRules, I don't have to write such things explicitly; this is dealt with by the inference engine.

Here, I use the fact that the final rule to solve a puzzle can only be NS or HS. (If other such rules were ever found and you added them to your solver, you should just apply Try to them instead of try.)

Code:

```
For any pattern X but NS or HS, let try(X) be the
(meta) procedure:
   look for an X;
   if you find one, apply it once immediately and GOTO
1a, else continue.

If X = NS or HS, let Try(X) be the procedure:
   look for an X;
   if you find one, apply it once immediately;
        if the puzzle is solved, report it with
current nrczt value;
        otherwise, apply all the elementary
constraints propagation rules until quiescence and
GOTO 1a;
   otherwise, continue.

A little copy-paste could easily insert these meta
procedures explicitly in the following algorithm.


0: n=0; nrczt=0

1a: n=0.9; nrczt=max(n, nrczt); Try NS;
1b: Try HS;

1c: n=1; nrczt=max(n, nrczt); try BI (basic
interactions; equivalent no NRCZT1);

2a: n=2; nrczt=max(n, nrczt); try NP;
2b: try HP;
2c: try SHP (X-Wing);
Note that NP, HP and SHP are subsumed by NRCZT2, but
if you have triplets and quads, it's nicer to have
also explicit Pairs.
2d: try nrczt-whips[2]

3a: n=3; nrczt=max(n, nrczt); try NT;
3b: try HT;
3c: try SHT (Swordfish);
3d: try nrczt-whips[3];

4a: n=4; nrczt=max(n, nrczt); try NQ;
4b: try HQ;
4c: try SHQ (Jellyfish);
4d: try nrczt-whips[4];

The sequel is simpler:
5a: n=5;
5b: nrczt=max(n, nrczt);
5c: try nrczt-whips[n];
5d: n=n+1; GOTO 5b;
```

Back to top profile pm www

PIsaacson Posted: Tue Nov 10, 2009 9:02 am Post subject: quote

Denis,

Joined: 02 Jul 2008
Posts: 311
Location: Campbell, CA

Thanks for the insight. I need to break my subsets processing into stages as you have indicated. Currently, when my subset engine runs, it will look for any and all naked/hidden locked sets and simple fish starting from size 2 and then running to size 4. But this is independent of the nrczt processing and occurs before any nrczt expanding depth searches. So I can find triples/quads before I even look for my first nrczt whip, and this really throws scoring off when I enable subsets.

From you pseudo-code logic, I think I now understand how SudoRules is interpreting the order of execution in searching for patterns. You don't look for triples until you've exhausted anything that could be found at depth 2, you don't look for quads until you've exhausted anything that coule be found at level 3...

Wheh!!! I think I can make this happen now, but I need to figure out some way of sync'ing the depths of the subsets and the nrczt chains.

Cheers,
Paul

Back to top profile pm

denis_berthier

Posted: Wed Nov 11, 2009 7:43 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

ABOUT LOOPS

I've already stated many times that I didn't allow loops in "my" chains, whips, braids. But some seem not to have understood why.

Before talking of loops, it should be recalled that a chain is a sequence of candidates (and is therefore linearly ordered) each nrc-linked to the previous one (let's call this the nrc-continuity condition).

If inner loops (i.e. the identity of two candidates) were allowed in a chain, that should obviously not change this basic defining condition.

I've proven in my book that allowing inner loops in chains that don't use the t-extension is useless (i.e. doesn't lead to more eliminations). This proof is obvious.

In whips or braids, some kinds of pseudo-loops are taken care of by the lassos: what would have given rise to an llc-to-rlc or to an rlc-to-llc loop leads indeed to an elimination.

For nrczt- whips or braids, there remains the question of re-using a right-linking or a left-linking candidate (with the same status left or right - and still with the same nrc continuity condition) for the sake of accumulating rlc's for later contradictions.

For braids, this is useless, as one can always accumulate rlc's along several branches.

For nrczt-whips, it is a different matter. I don't consider inner loops for two reasons:
- one theoretical: they obviously introduce an additional level of complexity (and both Paul and I have checked this in our very different implementations); they should therefore entail a penalty in the rating;
- one experimental: in ~ 10,000,000 randomly generated puzzles, I've found none that couldn't be solved with standard nrczt-whips (without inner loops); in conformance with my "least general pattern" approach, I've no reason to introduce loops.

Back to top profile pm www

Red Ed

Posted: Wed Nov 11, 2009 8:02 am Post subject: quote

Joined: 06 Jun 2005
Posts: 906

What a shame that you cannot bring yourself to post in the "Algorithm for NRCZT chains" topic.

Where have you "stated many times that I didn't allow loops"? I ask for the sake of completeness, whilst noting that this appears to be a change from your original stance:

denis_berthier wrote:

> Definition: a 3D-chain is a sequence of candidates, such that the first and the last candidates in the sequence are different (there is no global loop) and any two consecutive candidates are nrc-linked.
>
> Notice that global loops are excluded by definition, but not internal loops. Internal loops can be shown to be useless only for some types of chains (e.g. in 2D: xy-, hxy-, c-).

Back to top profile pm

PIsaacson

Posted: Wed Nov 11, 2009 9:08 am Post subject: quote

Joined: 02 Jul 2008
Posts: 311
Location: Campbell, CA

Denis,

I strongly believe that allowing inner-loops is tantamount to braids. Even if there is no penalty, an inner-loop provides accumlating non-linear truths by back-stepping to pick up an alternate path. Isn't that the domain of braids as opposed to whips?

Here's a crude diagram of a whip with 2 inner-loops re-drawn as a braid:

Code:

```
  @@@@@
 /     |
/      |
@      |
\      |
 \     |
  @@@@@++++++++++++++++
       #
```

```
                    #
==================#**********
                   |          |
                   |          |
                    \        /
                     \      /
                      *****

 whip with 3 segments "=" "#" and "+" and the two
inner-loops "@" and "*"


                     @@@@@@@@@@@@@@@@
                     |
                     |
==================###+++++++++++++++
                   |
                   |
                   ********************

 identical braid with 3 strands "=#+" and "=*" and
"=#@"
```

Cheers,
Paul

Last edited by PIsaacson on Wed Nov 11, 2009 9:58 am; edited 2 times in total

Back to top profile pm

denis_berthier

Posted: Wed Nov 11, 2009 9:15 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

PIsaacson wrote:

> I strongly believe that allowing inner-loops is tantamount to braids. Even if there is no penalty, an inner-loop provides accumlating non-linear truths by back-stepping to pick up an alternate path. Isn't that the domain of braids as opposed to whips?

The braid condition allows nrc-discontinuities. This is a priori more general that allowing inner loops in whips.
But what's true is that, if you want to allow loops, with all the additional complexity they entail, you'd better switch directly to braids.

I haven't excluded inner loops from whips *a priori*, but I've finally done it, based on the 2 reasons mentioned in my previous post.

[Edit] I've just seen your added diagram. It shows how a whip with inner loops can be considered a braid, OK. But the converse need not be true.

Back to top profile pm www

Red Ed

Posted: Wed Nov 11, 2009 6:32 pm Post subject: quote

Joined: 06 Jun 2005

denis_berthier wrote:

> I haven't excluded inner loops from whips *a priori*, but I've finally done

Posts: 906

> it, based on the 2 reasons mentioned in my previous post.

As in: finally done it right now? Well, OK, that's your prerogative.

How about editing the first post in the thread, and your web page, to make this explicit. Until you do that, NRCZT-chains should be considered undefined.

Back to top profile pm

denis_berthier

Posted: Thu Nov 12, 2009 7:06 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

> **Red Ed wrote:**
>
> > **denis_berthier wrote:**
> >
> > > I haven't excluded inner loops from whips *a priori*, but I've finally done it, based on the 2 reasons mentioned in my previous post.
> >
> > As in: finally done it right now? Well, OK, that's your prerogative.How about editing the first post in the thread, and your web page, to make this explicit.

I have used the no-loops condition from the beginning and it has been present in all the versions of SudoRules.
See **p. 336 of my book: definition: a 3D-chain is a sequence of candidates, ALL DIFFERENT, such that any two consecutive candidates are nrc-linked.**
It has been discussed at length on Eureka (in the lovely Eureka style!). At that time, no one here has reacted to it.
Included in the definition, as it was in the book, it seems to be an arbitrary *a priori* condition. In this forum, it is justified by the two reasons mentioned in my previous posts.
Let's say that "standard" nrczt-whips don't have inner loops. You can play with variants with loops but, for the same two reasons, it is very unlikely to be fruitful - and whatever you may find will be subsumed by shorter (<=) braids.

> **Red Head wrote:**
>
> > NRCZT-chains should be considered undefined.

Jeez, I didn't know you could be so funny when you make an effort!
Thanks. Yesterday evening, after seeing "2012" for 10 euros, this was real comfort - and for free!

Back to top profile pm www

Red Ed

Posted: Thu Nov 12, 2009 7:45 am Post subject: quote

Joined: 06 Jun 2005
Posts: 906

OK, but when you've recovered from your fit of giggles, Denis, don't you think that an update to the first post in this thread and to your web page might bring a little more clarity to the presentation for the benefit of future readers? Or must they buy your book?

Back to top profile pm

denis_berthier

Posted: Thu Nov 12, 2009 8:29 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 991
Location: Paris, France

Red Ed wrote:

> don't you think that an update to the first post in this thread and to your web page might bring a little more clarity to the presentation for the benefit of future readers?

Done.

Red Ed wrote:

> Or must they buy your book?

I'm reasonably well paid as a full professor and I need not sell books.
Still, buying my book is better investment than seeing "2012"

Back to top profile pm www

Display posts from previous: All Posts Oldest First Go

newtopic postreply **Sudoku Players' Forums Forum Index -> Advanced solving techniques**

All times are GMT
Goto page Previous 1, 2, 3 ... 14, 15, **16**

Page 16 of 16

Stop watching this topic

Jump to: Advanced solving techniques Go

You **can** post new topics in this forum
You **can** reply to topics in this forum
You **can** edit your posts in this forum
You **can** delete your posts in this forum
You **can** vote in polls in this forum