Sudoku Players' Forums

Rating rules / Puzzles. Ordering the rules

Goto page Previous 1, 2, 3 ... 28, 29, 30

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

View previous topic :: View next topic

Author	Message
denis_berthier Joined: 19 Jun 2007 Posts: 1116 Location: Paris, France	Posted: Sun Nov 22, 2009 10:34 pm Post subject: quote **Red Ed**, I'm not interested in discussing who has the biggest or the fastest one. The only nrczt program I've ever seen is Paul's and he said himself that it can't compute braids with the correct length. Paul knows that if he needs some test data, he'll get them by PM. Back to top profile pm www
Red Ed Joined: 06 Jun 2005 Posts: 1000	Posted: Sun Nov 22, 2009 11:37 pm Post subject: quote What a strange attitude. The "Algorithms ..." thread is to do with breaking new ground in making harder cases your whips/braids readily computable. I would've thought you'd be pleased that your invention wasn't going to be implemented only in SudoRules, and that by coming up with fast 'C' (probably) programs we'd also be able to bring the NRCZT ratings more into the mainstream. Do you make Paul promise not to share when you give him test data? Back to top profile pm
denis_berthier Joined: 19 Jun 2007 Posts: 1116 Location: Paris, France	Posted: Mon Nov 23, 2009 12:35 am Post subject: quote **Red Ed**, I really appreciate that different people try to implement whips/braids. Paul was the first to show the way and to come with a correct implementation of whips. And I also appreciate that, thanks to him, we now have an algorithm faster than SudoRules (with fewer possibilities). I appreciate that Allan and you also try to implement whips/braids (although I've seen none of your code). I appreciate all this ... even though it comes a little too late for me: - at a moment when I'll have no time for Sudoku for a while; - just after I have obtained major classification results, all the computations of which relied on SudoRules, for the most part much before such faster programs

were available.
With faster nrczt solvers, you may be able to add some degree of precision to these results (in particular the least precise one: the 0.3% difference between pB-NRCZT and pNRCZT), but, for all the classification results (which is, after all, the main reason for wanting a fast nrczt solver), my samples were already so large that I don't think it will make much difference.

As for "breaking new ground", I'll take it as an example of your natural modesty. What would be interesting, as I said to Paul long ago, is the full application of the zt-ing principle to the construction of braids(FP) for a family FP of patterns.

What I'm not directly interested in:
- programmers discussions about a 10% increase in speed; if such discussions were about the understanding of whips/braids, instead of implementation, I guess you would have them in the appropriate threads;
- talks about code that I can't run (such as yours).
If there happens to be a final available version of braids that runs correctly, let me know.

My interest in having a fast implementation doesn't entail that I have to entertain a confusion between the level of mathematics (where I have given and where I can discuss the definitions and proofs) and the level of implementation in C or C++ (in which I have little competency, as I already said). I fear you're often confusing these two levels.

By definition, a private communication is private. I trust Paul to know this.

I know you'd like to have all my data, but as you've already stated that you won't make yours available (your generator of complete grids*, your 3322 tests, ...), you're not really in a position to ask anything.

* not that I'm any longer interested. Thanks to gsf, I've found a way to circumvent your lack of cooperation in the controlled-bias generator. And, for me, this topic is closed.

Back to top profile pm www

ronk

Posted: Mon Nov 23, 2009 2:38 am Post subject: quote

Joined: 02 Nov 2005
Posts: 2700
Location: Southeastern USA

denis_berthier wrote:

> I appreciate that Allan and you also try to implement whips/braids (although I've seen none of your code).
> ...
> What I'm not directly interested in:
> ...
> - talks about code that I can't run (such as yours).
> ...
> I know you'd like to have all my data, but as you've already stated that you won't make yours available (your generator of complete grids*, your 3322 tests, ...), you're not really in a position to ask anything.

These comments beg the question: Have you made an *executable* and the *source code* for SudoRules available -- or whatever the correct terms are for CLIPS -- to anyone? If publicly available, links would be appreciated.

Also, for over a year now I've been wondering what a ratings thread is doing on this *Advanced solving techniques* forum. Would you please explain it to me?

TIA

Back to top profile pm

denis_berthier

Posted: Mon Nov 23, 2009 3:21 am Post subject: quote

Joined: 19 Jun 2007
Posts: 1116
Location: Paris, France

> **ronk wrote:**
>
> Have you made an *executable* and the *source code* for SudoRules available -- or whatever the correct terms are for CLIPS -- to anyone? If publicly available, links would be appreciated.

CLIPS is a free software (inference engine) that anyone can download from the CLIPS website or from sourceforege: http://clipsrules.sourceforge.net/
SudoRules is just the transcription in the CLIPS syntax of the rules I've defined in purely logical terms.
My position has always been clear: I don't want to discuss implementation details on this forum and I therefore don't make SudoRules public.
If the consequence for you is that you are not interested in nrczt-chains, don't worry, I've never counted you among their defendors.

> **ronk wrote:**
>
> Also, for over a year now I've been wondering what a ratings thread is doing on this *Advanced solving techniques* forum. Would you please explain it to me

It should be clear if you read the full title of this thread: ordering the rules, i.e. setting priorities between the rules.
Also:
- evaluating the efficiency of the rules, as made concrete by the classification results,
- comparing different sets of rules.

All this is obviously about rules and not about puzzles.

Back to top profile pm www

Red Ed

Posted: Mon Nov 23, 2009 9:19 am Post subject: quote

Joined: 06 Jun 2005
Posts: 1000

> **denis_berthier wrote:**
>
> What I'm not directly interested in:
> - programmers discussions about a 10% increase in speed; if such discussions were about the understanding of whips/braids, instead of implementation, I guess you would have them in the appropriate threads;

- talks about code that I can't run (such as yours).
If there happens to be a final available version of braids that runs correctly, let me know.

My interest in having a fast implementation doesn't entail that I have to entertain a confusion between the level of mathematics (where I have given and where I can discuss the definitions and proofs) and the level of implementation in C or C++ (in which I have little competency, as I already said). I fear you're often confusing these two levels.

An algorithm can be - and is being - discussed in a way that requires no skill in any particular programming language, merely some competence in algorithm design in general. Perhaps you lack that, too, in which case fair enough for being scared off by the "Algorithms ..." thread. But you should stick with it, be brave! There's plenty of stuff in there concerning real examples where the technique author's judgement would've been welcome.

Back to top profile pm

denis_berthier

Posted: Mon Nov 23, 2009 10:15 am Post subject: quote

Sorry, red Ed, but your tone in general doesn't give me any desire of discussing anything with you.

Joined: 19 Jun 2007
Posts: 1116
Location: Paris, France

Back to top profile pm www

ronk

Posted: Wed Nov 25, 2009 10:00 pm Post subject: quote

Joined: 02 Nov 2005
Posts: 2700
Location: Southeastern USA

denis_berthier wrote:

ronk wrote:

denis_berthier wrote:

I appreciate that Allan and you also try to implement whips/braids (although **I've seen none of your code**).
...
What **I'm not directly interested in:**
...
- talks about code that I can't run (such as yours).
...
I know you'd like to have all my data, but as you've already stated that **you won't make yours available** (your generator of complete grids*, your 3322 tests, ...), you're not really in a position to ask anything.

Have you made an *executable* and the *source code* for SudoRules available -- or whatever the correct terms are for CLIPS -- to anyone? If publicly available, links would be appreciated.

My position has always been clear: I don't want to discuss implementation details on this forum and I therefore don't make SudoRules public.

Whoosh! I'm picturing an Exocet flying directly over your head.

Back to top profile pm

denis_berthier

Posted: Wed Nov 25, 2009 10:09 pm Post subject: quote

Joined: 19 Jun 2007
Posts: 1116
Location: Paris, France

ronk wrote:

> Whoosh! I'm picturing an Exocet flying directly over your head.

What a nicely refreshing breeze

Back to top profile pm www

denis_berthier

Posted: Wed Dec 30, 2009 12:49 am Post subject: quote

Joined: 19 Jun 2007
Posts: 1116
Location: Paris, France

FURTHER QUESTIONS ABOUT RATING AND CLASSIFICATION

REFERENCES
If you don't know what the following words mean in my approach, this is not the place to ask or to discuss them. See my website (recently reorganised) or the relevant threads of this forum:

nrczt-whip
nrczt-braid
NRCZT-rating
classification results

zt-ing principle
whip(FP)
braid(FP)
scope of braid(FP)
T&E(FP) vs braid(FP) theorem

PRELIMINARIES
A resolution rule is written as: pattern ==> elimination (or assertion)
Given a family FP of patterns associated with resolution rules, I have defined zt-whips(FP) and zt-braids(FP).
They are conceptually simple generalisations of ordinary whips and braids, in which a pattern from the FP family can be used as a right-linking object instead of a mere candidate.
For instance, if FP = Basic Interactions (BI), whips(BI) and braids(BI) are merely grouped-whips or grouped-braids, in the sense that their right-linking objects can be the elements of a segment instead of mere candidates.
(segment = intersection block with row or intersection block with column).

A whip, a braid, a whip(FP) or a braid(FP) is a linear oriented structure.

Moreover, a whip or a whip(FP) satisfies the nrc-continuity condition. Whips can therefore be called nrc-continuous-sequential and they can be preferred to braids. Braids still have to keep some sequentiality but may have some branching on their right-linking candidates/patterns.
But braids have a theoretical advantage: the braid(FP) vs T&E(FP) theorem, which allows easier computations of their scope.

Given a family FP of patterns, it is called the generating family of the whips(FP) and the braids(FP).

INCREASING SEQUENCES OF FAMILIES OF PATTERNS
Consider an increasing sequence of families of patterns FP0, FP1, FP2,...
Here, increasing is to be understood in the set theoretic sense: FP0 included in FP1, FP1 included in FP2, ...
We have increasing families of whips whips(FP0), whips(FP1), whips(FP2),... and of braids whips(FP0), whips(FP1), whips(FP2),... It means that we have whips with more and more complex right-linking objects.
If each of the FPi families is closed under symmetry and super-symmetry, then this is also the case for each of the whips(FPi) and braids(FPi).

Here are natural examples.

Example 1:
FP0 = Singles (Naked + Hidden)
FP1 = FP0 + Basic Interactions = FP0 + NRCZT[1]
FP2 = FP1 + Subsets2, where Subsets2 = Pairs (Naked, Hidden and Super-Hidden - i.e. XWing)
FP3 = FP2 + Subsets3, where Subsets3 = Triplets (Naked, Hidden and Super-Hidden - i.e. Swordfish)
FP4 = FP3 + Subsets4, where Subsets4 = Quads (Naked, Hidden and Super-Hidden - i.e. Jellyfish)
...

Example 2:
GP0 = FP0
GP1 = FP1
GP2 = FP2 + finned-XWing
GP3 = GP2 + finned-Swordfish
GP4 = GP3 + finned-Jellyfish
...

Example 3:
HP0 = FP0
HP1 = FP1
HP2 = HP1 + whips[2]
HP3 = HP2 + whips[3]
HP4 = HP3 + whips[4]
...
HPn+1 = HPn + whips[n]

Example 4: same as 3, but with braids instead of whips

Example 5:
KP0 = FP0
KP1 = FP1
KP2 = KP1 + whips[2] + Subset2
KP3 = KP2 + whips[3] + Subset3
KP4 = KP3 + whips[4] + Subset4
KP5 = KP4 + whips[5]
...
KPn+1 = KPn + whips[n]

Example 6: same as 5, but with braids instead of whips

Notice that in all these examples:
- whips(FP0) or braids(FP0) are mere nrczt-whips or nrczt-braids
- whips(FP1) or braids(FP1) are mere grouped-whips or grouped-braids.

Given an increasing sequence FPi, whips(FPi) or braids(FPi) can obviously solve increasing sets of puzzles.

LENGTH OF A WHIP(FP) OR A BRAID(FP)
If all the patterns in FP have a well defined size*, then one can define in an obvious way the length of a whip(FP) or a braid(FP) as the sum of the sizes of its right-linking objects.
* which is the case for all the examples above:
1 for NS,HS, BI, 2 for patterns in Subset2, 3 for patterns in Subset3, 4 for patterns in Subset4
size = length for an ordinary whip or braid
Remember that these sizes have been defined in order to respect the subsumption theorems and quasi-subsumption results, so that two interpretations of the same structure (e.g. as a Triplet or as a whip[3]) have the same size.

SEQUENCE OF RATINGS ASSOCIATED WITH AN INCREASING SEQUENCE OF FAMILIES OF PATTERNS
Given any family of patterns FP, it is easy to define the whip(FP) rating - resp. the braid(FP) rating - of any puzzle P wrt to FP as the smallest n such that P can be solved with whips(FP) - resp. braids(FP) - no longer than n. If the puzzle can't be solved by whip(FP) - reps. braids(FP) - the corresponding rating is set to + infinity.
Given any increasing sequence FPi, for any puzzle P, one gets a decreasing sequence of ratings whip(FP0)-rating > whip(FP1)-rating > whip(FP2)-rating,... and similarly for braids

Notice also that all these ratings:
- are defined in purely logical terms, independent of any implementation,
- are stable under symmetry and super-symmetry whenever the FPi families are.

ABSOLUTE RATING
This is inspired by **Allan**'s ideas about an absolute rating and may have to be revised if he proposes a better definition.
Define AR(P) - A for "absolute" or for "Allan" - as the smallest n such that P can be solved using only Allan's diagrams with at most n 2D-cells (n Truths).
Obviously, AR(P) is smaller than any of the previously defined ratings.

KNOWN RESULTS
My previous results have shown that there are two complementary aspects of the complexity of a puzzle: the length of the whips/braids necessary to solve it and the type of these whips/braids (i.e. of their inner right-linking patterns).

Mere nrczt-whips have solved all the puzzles in various randomly generated samples totaling ~ 10,000,000 puzzles. They fail to solve only exceptionally hard puzzles. This is important as it shows that the NRCZT-rating is finite for almost all the puzzles.

Remember that (taking FP = braids themselves) braids(braids) can solve all the known puzzles. But the results here (http://www.sudoku.com/boards/viewtopic.php?t=6390&postdays=0&postorder=asc&start=43 and next post) show that a family FP much simpler than braids is enough.

OPEN QUESTIONS

Here are three types of (hard) questions corresponding to different goals.

Question type 1: Given an increasing sequence FPi, how does the whip(FPi)-rating or braid(FPi)-rating of a puzzle P vary with i?
We know that this rating is bounded from above by nrczt-rating(P) and from below by AR(P).
I think that:
- most of the time (say for more than 99% of the puzzles), for random samples, nrczt-rating(P) = AR(P), which would completely solve the question for these puzzles; this can easily be tested (e.g. on the Sudogen0_1M collection) if **Allan** can compute the AR rating;
- for some hard puzzles, there may be a noticeable difference between e.g. the nrczt-rating and the whip(FP4)-rating (FP4 defined above).

Question type 2: Given a collection of puzzles, how does the number of puzzles solved by whips(FPi) or braids(FPi) increase with i?
This question is independent of ratings. It deals only with inner patterns necessary to solve a puzzle.
Examples have been given here: http://www.sudoku.com/boards/viewtopic.php?t=6390&postdays=0&postorder=asc&start=44 for gsf's collection.
Collections of reference should include exceptionally hard puzzles and can't

therefore be random. As of this writing, I know only 2: gsf's and Tarek's Pearly6000 (recently signaled to me by Paul). I plan to do with the Pearly the same thing as I did with gsf's.

The question about braids can be dealt with thanks to the T&E theorem, as in the above mentioned post.
The question about whips can be dealt with thanks to **Paul**'s program, for some families FP. (The exact length of the whips(FP) doesn't have to be known).

Question type 3: Find a family FP of simple patterns such that whips(FP) or braids(FP) solve all the known puzzles.
Here the goal is to find the simplest family FP. The already mentioned results suggest that it shouldn't be too hard (at least for braids).

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 - 8 Hours
Goto page Previous 1, 2, 3 ... 28, 29, 30

Page 30 of 30

Jump to: Advanced solving techniques Go