Sudoku Players' Forums

? FAQ **Q** Search **▤** Memberlist **▤** Usergroups **☑** Register
👤 Profile **▣** Log in to check your private messages **◉** Log in

Fully supersymmetric chains
Goto page **Previous** **1**, **2**, **3** ... , **18**, 19, **20** **Next**

▤ newtopic **▤ postreply**   **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: Mon Nov 30, 2009 9:21 pm Post subject: **🗨 quote** **Paul**, The nrc notation system: - relies on a consistent view of rc, rn, cn and bn spaces and on the idea of r/c/n symmetries, - is designed to be readable by a human reader - not by any specific program. In the computer world, the normal way for a software to be made compatible with another notation is to provide input for this other notation, not to change it. If you want to use Allan's notation, use it, but please don't introduce confusion by calling it the nrc notation.
Back to top	**👤 profile** **👥 pm** **🌐 www**
PIsaacson Joined: 02 Jul 2008 Posts: 355 Location: Campbell, CA	**▫** Posted: Tue Dec 01, 2009 12:59 am Post subject: **🗨 quote** Denis, Point taken. Since I can automatically generate *.sud files, the nrczt solution logs don't have to reflect Allan's set/link-set notation. I changed my notation to conform with the new notation (mostly). I still would like to achieve alignment so I dropped the "R" "C" within the braces for BN space notation as shown in my prior posting with the updated output. I think the alignment makes it even easier to study/review long lists of nrc chains. Cheers, Paul
Back to top	**👤 profile** **👥 pm**
David P Bird Joined: 16 Sep 2008	**▫** Posted: Tue Dec 01, 2009 1:14 am Post subject: **🗨 quote** **Denis_Berthier wrote:** If you want to use Allan's notation, use it, but please don't introduce

Posts: 215
Location: Middle
England

confusion by calling it the nrc notation.

Denis, does that apply to any other terms as well or is it still OK to hijack them and bend them whichever way we want to suit our own ends?

Back to top   (profile) (pm)

---

denis_berthier   Posted: Tue Dec 01, 2009 1:31 am   Post subject: (quote)

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

> **PIsaacson wrote:**
> I changed my notation to conform with the new notation (mostly). I still would like to achieve alignment so I dropped the "R" "C" within the braces for BN space notation as shown in my prior posting with the updated output.

Hi **Paul**,
Dropping the rc in the bn spaces makes it more difficult to read and disrupts the homogeneity of the notation.

Back to top   (profile) (pm) (www)

---

denis_berthier   Posted: Tue Dec 01, 2009 1:31 am   Post subject: (quote)

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

> **David P Bird wrote:**
>> **Denis_Berthier wrote:**
>> If you want to use Allan's notation, use it, but please don't introduce confusion by calling it the nrc notation.
>
> Denis, does that apply to any other terms as well or is it still OK to hijack them and bend them whichever way we want to suit our own ends?

I can't imagine what you're speaking of.

Back to top   (profile) (pm) (www)

---

denis_berthier   Posted: Sun Dec 06, 2009 10:27 pm   Post subject: (quote)

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

> **denis_berthier wrote:**
>> **PIsaacson wrote:**
>> I'm still validating my tests against the sudogen0_1m collection, but preliminary indications are that braids produced a smaller score for at least 300k puzzles.
>
> Utterly unlikely. You should take the first such examples and check all their "braids".

> **denis_berthier wrote:**
>> **PIsaacson wrote:**
>> There are 464 cases in which braids scored higher than my nrczt chains/whips scores

Impossible, of course, for a correct implementation.

I've been wondering why you had such a discrepancy. But I haven't yet found any possible reason (except of course an error in the way you compute the total length).

At one time, I wondered whether you allowed repetitions of llc's in braids, but disallowing them could obviously not explain the first type of discrepancies. Neither could it explain the second, because loops, and therefore llc repetitions, are not allowed in whips.

Anyway, that led me to an interesting (though not really surprising) result. I tried disallowing repetitions of llc's in my implementation of braids.
From a theoretical POV, it is not a good idea because it destroys some good theoretical properties of braids.
But, from a practical POV, I couldn't find an example in which it entailed a difference in the pB-NRCZT rating.

Back to top   [profile] [pm] [www]

denis_berthier   Posted: Sat Dec 12, 2009 11:03 pm   Post subject: [quote]

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

—————————————————————————————————————

Extending the nrc notation to inner patterns in zt-whips(FP)

—————————————————————————————————————

The nrc notation, even in its slightly improved strict version (http://www.sudoku.com/boards/viewtopic.php?t=5591&start=263), deals only with simple patterns and with 2D/3D (z)(t) chains/whips/braids in which only candidates can be used as right-linking objects.

In zt-whips(FP) or zt-braids(FP) (http://www.sudoku.com/boards/viewtopic.php?t=5591&postdays=0&postorder=asc&start=204 and subsequent posts), where FP is a family of (simple) patterns, elements of FP, instead of mere candidates, can appear as right-linking objects.
It may be because I haven't given many examples, but (considering some recent attempts at defining extremely complex pseudo-linear patterns), it seems to me that the strength of zt-whips(FP) has been largely under-estimated. Whips(FP) or braids(FP) allow some very local forms of branching - branching limited to the inside of a right-linking pattern, which is generally not considered as branching.

Notice that nothing in theory disallows complex patterns, such as chains, to be present in the FP families.
Indeed, the T&E vs braids theorem implies that all the known puzzles can be solved with braids(braids).
But all the known chain patterns (ALS/AHS/AFish chains, AICs) are subsumed or statistically almost subsumed by nrczt whips.
Moreover, the statistical results here http://www.sudoku.com/boards/viewtopic.php?t=6390&postdays=0&postorder=asc&start=44 show that taking such complex FP family is like taking the exponential of a hammer to kill a fly: only simple FP patterns are needed to solve all the known puzzles with whips(FP).

Notice also that almost-somethings never appear in zt-whips(FP) or zt-braids(FP). What appears is the "somethings" themselves. This makes it easier to extend the nrc notation.

As the nrc notation is already defined for Basic Interactions and Subsets - Naked, Hidden or Super-Hidden (Fish) - one could merely write these included patterns in nrc notation within the chain structure. I have already done it. But, unless these patterns are very simple, this rapidly makes things difficult to read.

Another option is to design a simplified notation for the simple patterns. I have also already used this possibility, e.g. I wrote something like NT(r1c125n123); but it would be more difficult for a NT in a block: NT(b1r1c1r2c2r3c3n123) is not really appealing.

The third possibility is simpler in complex cases (as the NT above): each of the inner patterns is written as a mere capital letter within the whip/braid and each of these patterns appears in successive lines starting with the corresponding capital letter and written in the usual nrc notation.

Example: (x is the length)
1st option:
zt-whip(NT)[x] r1n1{c1 c8} - b3{n1r1c7 NT({r1c7 r2c8 r3c9}{n2 n3 n4}) - c9n4{r1 r7} - ...

2nd option:
zt-whip(NT)[x] r1n1{c1 c8} - b3{n1r1c7 NT(r1c7r2c8r3c9n234)} - c9n4{r1 r7} - ...

3rd option:
zt-whip(NT)[x] r1n1{c1 c8} - b3{n1r1c7 A} - c9n4{r1 r7} - ...
...A = Naked-triplet-in-a-block b3{r1c7 r2c8 r3c9}{n2 n3 n4}

In this example, notice that, if you consider the 3 cells r1c7 r2c8 r3c9 in block b3, they don't make a NT in the block (because n1 is present in r1c7). They make a NT only modulo the first rlc (n1r1c8).

Edit 12/28/09: it seemed the nrc notation wasn't defined for Basic Interactions (BI), but it was indeed defined implicitly by their equivalence with whips[1]. To make things more explicit, when appearing as a right-linking pattern within a generalised whip or braid, BI can be written in abbreviated form:
row r interaction with block b for number n : rn{b} or rnb
block b interaction with row r for number n : bn{r} or bnr
column c interaction with block b for number n : cn{b} or cnb
block b interaction with column c for number n : bn{c} or bnc
where "X interaction with Y" means that the eliminations occur in Y.
Here, (considering the preceding rn, cn or bn space), the symbol between the curly brackets defines a segment instead of a candidate - which is all that there is in Basic Interactions and in whips(BI).

Back to top   [profile] [pm] [www]

Allan Barker   Posted: Sun Dec 27, 2009 11:23 pm   Post subject: [quote]

Joined: 20 Feb 2008
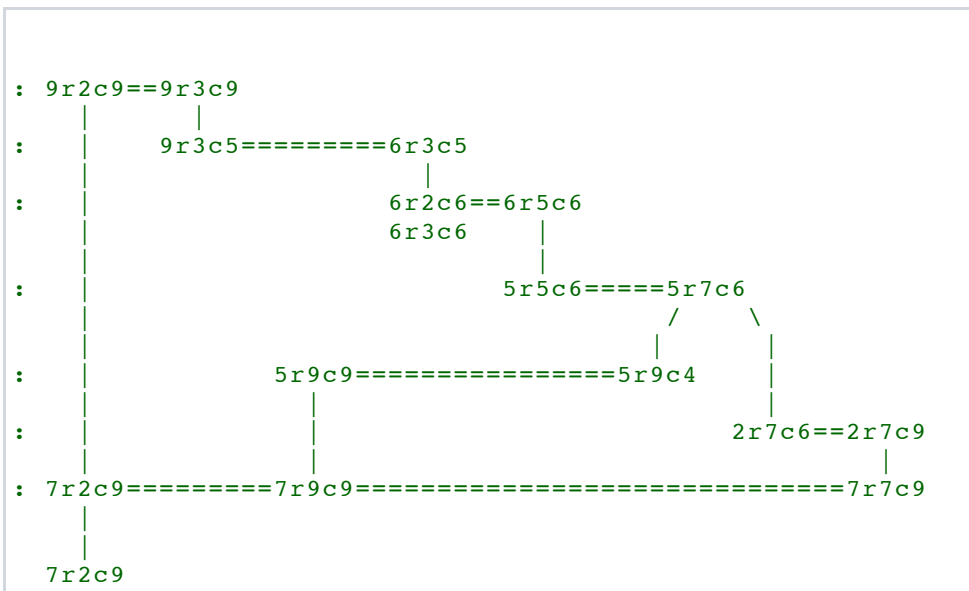Posts: 478
Location: Bangkok

Denis,

I was scanning through some puzzles when I came across the following logic that Xsudo identified as a "Looped Chain". Problem was I was running the NRCZT solver at the time. When I looked at the logic, sure enough, the logic seemed to be fully branched, i.e., there are two independent paths through the chain. I am aware of the internal loops but I did not think they would lead to this kind of full branching.

Is this a correct nrczt chain or did I find something else?

chain[7] r2c9 <> 8 = n9c9{r2,-,r3} - r3c5{n9,-,n6} - n6c6{r23,-,r5} - n5c6{r5,-,r7} - n2r7{c6,-,c9} - n7c9{r7,r2,r9} - n5r9{c9,c4,-}

Logic Diagram

Code:

```
: 9r2c9==9r3c9
     |         |
:    |      9r3c5========6r3c5
     |                     |
:    |                  6r2c6==6r5c6
     |                  6r3c6    |
     |                           |
:    |                        5r5c6=====5r7c6
     |                                /     \
     |                               |       |
:    |           5r9c9==============5r9c4    |
     |               |                       |
:    |               |                    2r7c6==2r7c9
     |               |                          |
: 7r2c9========7r9c9============================7r7c9
     |
     |
   7r2c9
```

Grid Logic Diagram Branch in r7c6 and in r7c9

.
.

Back to top   [profile] [pm] [www]

denis_berthier   Posted: Sun Dec 27, 2009 11:56 pm   Post subject: [quote]

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

Allan Barker wrote:

I came across the following logic that Xsudo identified as a "Looped Chain". Problem
was I was running the NRCZT solver at the time. When I looked at the logic, sure
enough, the logic seemed to be fully branched, i.e., there are two independent paths
through the chain. I am aware of the internal loops but I did not think they would
lead to this kind of full branching.
Is this a correct nrczt chain or did I find something else?
chain[7] r2c9 <> 8 = n9c9{r2,-,r3} - r3c5{n9,-,n6} - n6c6{r23,-,r5} - n5c6{r5,-
,r7} - n2r7{c6,-,c9} - n7c9{r7,r2,r9} - n5r9{c9,c4,-}

Your chain is not correctly written as a whip, but it is one. In whips, there is always only one left-linking candidate - even in generalised whip(FP).
From the POV of whips, there is no branching in this example.
In standard nrc notation, this is merely
whip[7] c9n9{r2 r3} - r3c5{n9 n6} - c6n6{r3 r5 r2#2} - c6n5{r5 r7} - r7n2{c6 c9} - c9n7{r7 r9 r2*} - r9n5{c9 . c4#4} ==> r2c9 <> 8

What may have troubled you is that one can also identify a different whip in which, in cell 3, the role of the llc is interchanged with that of the t candidate.
These whips are equivalent in the sense that they have the same *sequence* of right-linking candidates.

In SudoRules, once a partial whip has been found, no other partial whip equivalent to it is considered. But this is merely for efficiency purposes, it can't change anything for eliminations based on longer whips extending any of these two equivalent versions.
Said otherwise, one needs only consider one member of each equivalence class.
This is conceptually much simpler than allowing several llc's - an option that would give the false impression that there is branching.

Back to top   (profile) (pm) (www)

Allan Barker   Posted: Mon Dec 28, 2009 12:25 am   Post subject: (quote)

Joined: 20 Feb 2008
Posts: 478
Location: Bangkok

> **denis_berthier wrote:**
>
> > **Allan Barker wrote:**
> >
> > > Is this a correct nrczt chain or did I find something else?

> .... it is one. In whips, there is always only one left-linking candidate.
> From the POV of whips, there is no branching in this example.

But surely, from the constraint diagram this logic is badly branched. What is the difference?

Edit: are you suggesting that chains can't be viewed as constraints?

Back to top   [profile] [pm] [www]

denis_berthier   Posted: Mon Dec 28, 2009 1:10 am   Post subject: [quote]

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

> **Allan Barker wrote:**
>
> > **denis_berthier wrote:**
> >
> > > **Allan Barker wrote:**
> > >
> > > > Is this a correct nrczt chain or did I find something else?
> > >
> > > .... it is one. In whips, there is always only one left-linking candidate.
> > > From the POV of whips, there is no branching in this example.
> >
> > But surely, from the constraint diagram this logic is badly branched. What is the difference
> > Edit: are you suggesting that chains can't be viewed as constraints

I don't know exactly what you mean by "constraints".
In Sudoku, I know only 4 constraints (the defining 4 constraints).
These 4 constraints don't in and by themselves define any resolution method.
Resolution rules and the way they are used define resolution methods.

Various types of resolution rules can be devised.
Our two approaches have pluses and minuses.

Apart from the basic rules, *my* preferred type is chains/whips - with an accent on nrc-continuity and look-back (to borrow this word from you) which implies (most of the time) non reversibility. Your "constraints diagrams" are irrelevant to this approach. Trying to view chains as "constraints", i.e. as in your diagrams, is forgetting their main quality: sequentiality.

Your preferred type (I mean before you started speaking of ribbons) is some reversible nets - so reversible that any ordering is irrelevant. Trying to make them appear as linear is IMO irrelevant to your approach. It is forgetting their main quality: full reversibility (and the multiple eliminations it allows).

I think our approaches can nevertheless be compared - in a way I'll describe (hopefully) soon. Unfortunately, I don't have much time for Sudoku.

Back to top   [profile] [pm] [www]

ronk

Posted: Mon Dec 28, 2009 1:31 am   Post subject: [quote]

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

Allan Barker wrote:

> **denis_berthier wrote:**
>
> > **Allan Barker wrote:**
> >
> > > Is this a correct nrczt chain or did I find something else?
> >
> > .... it is one. In whips, there is always only one left-linking
> > candidate.
> > From the POV of whips, there is no branching in this example.
>
> But surely, from the constraint diagram this logic is badly branched. What is
> the difference?

Allan, you obviously understand very little about sudoku. ;-) The logic diagram for this whip looks like this ...

Code:

```
 8r2c9
   |
 9r2c9==9r3c9
          |
       9r3c5=========6r3c5
                        |
                     6r3c6==6r5c6
                             |
                          5r5c6=====5r7c6
                                     |
                                  2r7c6==2r7c9
                                          |

 7r7c9====7r9c9
                                                        |

  5r9c9====SPLAT
```

... and that almightly "SPLAT" at the end simply vaporizes some links ... and allows us to redefine the traditional meaning of 'branching.' 😃

Back to top   [profile] [pm]

David P Bird

Posted: Mon Dec 28, 2009 3:25 am   Post subject: [quote]

Ronk, this is how the exclusion translates into a branched AIC:

Joined: 16 Sep 2008
Posts: 215
Location: Middle
England

Code:

```
    / (2)r7c6 = (2-7)r7c9 \
(9)r2c9 = (9)r3c9 – (9=6)r3c5 – (6)r23c6 = (6-5)r5c6 =
(5)r7c6 –                            = (7)r2c9

    \ (5)r9c4 = (5-7)r9c9 /
```

Using this would you expand on your thinking about redefining the traditional meaning of branching please.

[Edit] Ahh! It was meant as a joke, and a bit of a jibe. In which case it's the same violin that I've been playing for ages.

Back to top

| **Allan Barker** | 🗋 Posted: Mon Dec 28, 2009 9:13 am   Post subject: |

Denis

Joined: 20 Feb 2008
Posts: 478
Location: Bangkok

BTW, I am not anti-branching. The question was mostly about my solver. You might say I'm pro-choice when it comes to logic, each to their own.

> **Denis Berthier wrote:**
> I don't know exactly what you mean by "constraints".

The 324 native constraints.

> **Denis Berthier wrote:**
> Various types of resolution rules can be devised.
> Our two approaches have pluses and minuses.

Um, of course. One difference that might not be apparent is that I don't have a fully documented perfectly integrated system to stack against yours. On the other hand, that provides a lot of freedom, if I want to branch today, I branch. My goal is mostly finding interesting logic, it need not fit certain specs.

> **Denis Berthier wrote:**
> Apart from the basic rules, my preferred type is chains/whips - with an accent on nrc-continuity and look-back (to borrow this word from you) which implies (most of the time) non reversibility. Your "constraints diagrams" are irrelevant to this approach. Trying to view chains as "constraints", i.e. as in your diagrams, is forgetting their main quality: sequentiality.

What you're saying (your view) is that the quality, the essence of the logic is in the definition, the resolution rule, correct?

> **Denis Berthier wrote:**
> Your preferred type (I mean before you started speaking of ribbons) is some reversible nets - so reversible that any ordering is irrelevant. Trying to make them appear as linear is IMO irrelevant to your approach. It is forgetting their main quality: full reversibility (and the multiple eliminations it allows).

But, but, but, I have been doing that since day one, making a sequence of truths by one means or another. I have no idea how to do it otherwise. The real difference is I always work with entire sets (native constraints). Ribbons are the same, they have been in Xsudo for a long time. Nrczt chains may pop up, but I didn't look for them as such.

> **Denis Berthier wrote:**
>> I think our approaches can nevertheless be compared - in a way I'll describe (hopefully) soon. Unfortunately, I don't have much time for Sudoku.

I think you might be at a disadvantage as my "approach" is not as pinned down and not fully documented. Hmm, perhaps I could help, it would be fun.

Allan

Back to top   [profile] [pm] [www]

denis_berthier   Posted: Mon Dec 28, 2009 10:28 am   Post subject: [quote]

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

> **Allan Barker wrote:**
>> **Denis Berthier wrote:**
>>> Apart from the basic rules, my preferred type is chains/whips - with an accent on nrc-continuity and look-back (to borrow this word from you) which implies (most of the time) non reversibility. Your "constraints diagrams" are irrelevant to this approach. Trying to view chains as "constraints", i.e. as in your diagrams, is forgetting their main quality: sequentiality.
>>
>> What you're saying (your view) is that the quality, the essence of the logic is in the definition, the resolution rule, correct?

I generally avoid speaking of essence (you certainly know the anâtman doctrine).
There are 2 ≠ things in what I'm saying here:
- my approach is based on looking for *predefined patterns* associated to resolution rules - be they fish or whips.
- my chains/whips are sequential and nrc-continuous; this is an additional structure wrt your constraint diagrams. Remember, we already had this discussion: "*sequence = set + linear order*" and not "sequence = set - something".

I don't know if it'd be easy to do, but if you want to use XSudo to display oriented chains without depriving them of their chain structure, you should:
- provide an input format for the adequate syntax (nrc)
- draw *arrows instead of simple lines* between consecutive rlc's and llc's
- display the additional lines justifying the z- and t- candidates as light dotted lines (need not be oriented) / don't display them at all; the best being the possibility of switching from one mode to the other: highlight only the main structure (the whip) vs show also the details.

> **Allan Barker wrote:**
>> **Denis Berthier wrote:**
>>> Your preferred type (I mean before you started speaking of ribbons) is some reversible nets - so reversible that any ordering is irrelevant. Trying to make them appear as linear is IMO irrelevant to your approach. It is forgetting their main quality: full reversibility (and the multiple eliminations it allows).

> But, but, but, I have been doing that since day one, making a sequence of truths by one means or another. I have no idea how to do it otherwise. The real difference is I always work with entire sets (native constraints). Ribbons are the same, they have been in Xsudo for a long time.

What you have shown from the beginning and you continue to show is 2D diagrams. Most of the time, you have multiple eliminations for each of your diagrams. These eliminations can't correspond to the same linearisations of the diagram.
In your approach, *the pattern is the unordered diagram*, not the ribbon. At least, that's the best way of considering it I can imagine. The various linearisations of the diagram (various "ribbons"?) correspond to the conditions on various targets.

> **Denis Berthier wrote:**
>
> Nrczt chains may pop up, but I didn't look for them as such.

That's the difference with "looking for predefined patterns".
I don't think nrczt-chains or any other pattern "pop up". You interpret them as such but your solver doesn't know what an nrczt-chain is. It has no possibility of looking only for them.
You shouldn't see any value judgement in this. As I already said, this is a plus or a minus, depending on the goals.

> **Allan Barker wrote:**
>
> > **Denis Berthier wrote:**
> >
> > I think our approaches can nevertheless be compared - in a way I'll describe (hopefully) soon. Unfortunately, I don't have much time for Sudoku.
>
> I think you might be at a disadvantage as my "approach" is not as pinned down and not fully documented. Hmm, perhaps I could help, it would be fun.

One thing you could do is develop a classification of puzzles based on the minimum number of 2D-cells (truths in your vocabulary) in diagrams necessary to solve it (what you once called the "absolute rating"). If you can control the size (number of 2D-cells) of your diagrams (can you?), that should be easy.
It would be interesting to see how many times and how far it is below the nrczt-rating. Sudogen0_1M would be a good first test case.

Back to top   profile   pm   www

Display posts from previous: All Posts ▲▼   Oldest First ▲▼   Go

Sudoku Players' Forums   All times are GMT - 8 Hours
new topic   post reply   **Forum Index ->**   Goto page **Previous** **1**, **2**, **3** ... , **18**, 19, **20** **Next**
                          Advanced solving techniques

Page 19 of 20

Jump to: Advanced solving techniques   ▲▼   Go

You **cannot** post new topics in this forum
You **cannot** reply to topics in this forum

You **cannot** edit your posts in this forum
You **cannot** delete your posts in this forum
You **cannot** vote in polls in this forum

Powered by phpBB © 2001, 2005 phpBB Group