Sudoku Players' Forums

Abominable TRIAL-and-ERROR and lovely BRAIDS

Goto page Previous 1, 2, 3 ... 9, 10, 11, 12 Next

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

View previous topic :: View next topic

Author	Message
ronk Joined: 02 Nov 2005 Posts: 2748 Location: Southeastern USA Back to top	Posted: Fri Jan 08, 2010 9:49 pm Post subject: quote [Withdrawn, since I was determining length incorrectly.] Last edited by ronk on Sat Jan 09, 2010 11:12 am; edited 1 time in total profile pm
denis_berthier Joined: 19 Jun 2007 Posts: 1164 Location: Paris, France	Posted: Sat Jan 09, 2010 6:53 am Post subject: quote edit **Mauricio**, Some precision: only part of your post can have been developed before braids; part of it is the result of the exchange of several PMs. Some problems I can see with Mbraids: - what you now define as an Mbraid is the output of some form of a breadth-first procedure; as I mentioned in a PM, it contains lots of unnecessary information and it is much more complex than a braid (a sequence of sets of candidates instead of a sequence of candidates); - your definition of length (say Mlength) doesn't take into account the breadth of the Mbraid, which is generally very large; - the size of Mbraids (number of candidates included in your definition) increases exponentially with Mlength (the size of a braid increases linearly with length), due to all the useless information it contains; - the Mlength is not consistent with the notion of length for known chain or subset structures; - the Mlength is obviously smaller than the length of any braid that could be extracted from an Mbraid; it is smaller in unknown proportions, which doesn't make it a very useful tool in general; - but it may be useful when it is very large, as an oracle telling us that it will be difficult to find a braids solution, as in your example; - it can therefore also be used as a filter for hard puzzles. **Mauricio wrote:** I don't think this procedure can be easily turned into a pBraids rating calculator. You're right. I think Paul started with a similar algorithm; he has now modified it to have a better approximation of length. The problem of extracting the smallest braid from an Mbraid is not easy. Moreover, the problem is more complex than this, as it consists of extracting this for all possible Z targets at the same time. And it has to be redone each time a candidate is deleted (*). **Paul**, can you give us some updated information on what you're now doing about braids length? * In SudoRules, my implementation of braids uses the natural possibilities of an inference engine to automatically update the current set of partially computed braids. This makes it slow and greedy for memory but it spares lots of programming work. Back to top profile pm www
Mauricio Joined: 22 Mar 2006 Posts: 1084	Posted: Thu Jan 14, 2010 4:03 am Post subject: quote I have recently finished my own implementation of braids, I tried to solve the puzzle with Mlegth 20 I posted recently **Code:** 000001002000030040005600700002800003040050060900007400007003004030020080600400300

This puzzle has so few braids, that solving it is actually not that difficult.

Every braid is the shortest one possible (modulo bugs in my code).

Code:

```
000001002000030040005600700002800003040050060900007400007003004030020080600400300
r1c1<>7,  braid[3]  r2n7{ c2   c4 } - r5n7{ c1   c9 } - r8n7{ c9    . }
r4c6<>9,  braid[4]  r5c6{ n9   n2 } - c6n6{ r4   r8 } - r7n6{ c5   c7 } - c7n2{
r7    . }
r6c4<>1,  braid[4]  r6c5{ n1   n6 } - r6n2{ c4   c8 } - r7n6{ c5   c7 } - c7n2{
r7    . }
r3c6<>9,  braid[5]  r5c6{ n9   n2 } - c6n4{ r3   r4 } - c6n6{ r4   r8 } - r7n6{
c5   c7 } - c7n2{ r7    . }
r4c5<>9,  braid[5]  r5c6{ n9   n2 } - r4n4{ c5   c6 } - c6n6{ r4   r8 } - r7n6{
c5   c7 } - c7n2{ r7    . }
r5c7<>9,   whip[1]  r4n9{ c8    . }
r5c9<>9,   whip[1]  r4n9{ c8    . }
r5c4<>1,   whip[3]  r6c5{ n1   n6 } - r4c5{ n6   n4 } - r4c6{ n4    . }
r7c5<>1,   whip[1]  c4n1{ r8    . }
r9c5<>1,   whip[1]  c4n1{ r8    . }
r6c3<>1,  braid[5]  r6c5{ n1   n6 } - r6n3{ c3   c4 } - r6n2{ c4   c8 } - c7n2{
r5   r7 } - r7n6{ c7    . }
r2c6<>9, braid[11]  r5c6{ n9   n2 } - r5n9{ c6   c4 } - c4n2{ r6   r2 } - c7n2{
r5   r7 } - b2n5{r2c4 r1c4} - r7n6{ c7   c5 } - c1n2{ r7   r3 } - r3n3{ c1   c8 }
- r1c8{ n3   n9 } - r3n9{ c9   c2 } - r7n9{ c2    . }
r7c4<>9, braid[10]  c4n1{ r7   r8 } - r8n7{ c4   c9 } - c8n7{ r9   r4 } - c9n6{
r8   r2 } - r4n9{ c8   c7 } - r8n9{ c7   c3 } - r2n9{ c3   c2 } - c2n7{ r2   r1 }
- r1n6{ c2   c3 } - c3n4{ r1    . }
r7c7<>1,  braid[5]  r7c4{ n1   n5 } - r7n6{ c7   c5 } - r8c6{ n6   n9 } - r5c6{
n9   n2 } - c7n2{ r5    . }
r4c7<>1, braid[11]  r4n9{ c7   c8 } - c5n1{ r4   r6 } - c8n7{ r4   r9 } - r8n7{
c9   c4 } - r9n2{ c8   c2 } - c4n1{ r8   r7 } - c8n1{ r7   r3 } - r3n3{ c8   c1 }
- c1n2{ r3   r2 } - r2n7{ c1   c2 } - c2n1{ r2    . }
r7c7<>9,  braid[5]  r4c7{ n9   n5 } - r7n6{ c7   c5 } - r6c5{ n6   n1 } - r6c8{
n1   n2 } - c7n2{ r5    . }
r6c2<>1, braid[10]  r6c5{ n1   n6 } - r4n6{ c6   c2 } - r7n6{ c5   c7 } -
b4n5{r4c2 r4c1} - r1n6{ c7   c3 } - c3n4{ r1   r8 } - r8c1{ n4   n1 } - c3n1{ r9
 r2 } - c7n1{ r2   r5 } - c7n2{ r5    . }
r1c7<>9,  whip[11]  r4c7{ n9   n5 } - r4n9{ c7   c8 } - c8n7{ r4   r9 } - r8n7{
c9   c4 } - r1c4{ n7   n5 } - r2n5{ c6   c9 } - b3n6{r2c9 r2c7} - r7n6{ c7   c5 }
- r6c5{ n6   n1 } - r6c9{ n1   n8 } - c7n8{ r5    . }
r7c1<>1,  whip[11]  r7c4{ n1   n5 } - c4n1{ r7   r8 } - r8n7{ c4   c9 } - r5n7{
c9   c1 } - r4c1{ n7   n5 } - c2n5{ r6   r9 } - b7n2{r9c2 r7c2} - r7n8{ c2   c5 }
- r9c6{ n8   n9 } - r5c6{ n9   n2 } - c7n2{ r5    . }
r6c9<>1, braid[13]  r6c5{ n1   n6 } - r4n6{ c6   c2 } - r7n6{ c5   c7 } - r1n6{
c7   c3 } - c7n2{ r7   r5 } - c3n4{ r1   r8 } - r6c8{ n2   n5 } - r4c7{ n5   n9 }
- r4n5{ c7   c1 } - r8c1{ n5   n1 } - r8c7{ n1   n5 } - c9n5{ r9   r2 } - r2n6{
c9    . }
r2c7<>9, braid[13]  r4c7{ n9   n5 } - r6c9{ n5   n8 } - r3c9{ n8   n1 } - r5c9{
n1   n7 } - r8n7{ c9   c4 } - r9n7{ c5   c8 } - r9n2{ c8   c2 } - r9n1{ c2   c3 }
- r8n1{ c1   c7 } - r5n1{ c7   c1 } - r4c1{ n1   n7 } - r2n1{ c1   c2 } - r2n7{
c2    . }
r9c6<>9, braid[13]  r5c6{ n9   n2 } - c4n2{ r6   r2 } - c7n2{ r5   r7 } - r7n6{
c7   c5 } - r9n2{ c8   c2 } - r3n2{ c2   c1 } - r8c6{ n6   n5 } - r3n3{ c1   c8 }
- r7c4{ n5   n1 } - c4n5{ r7   r1 } - r1c8{ n5   n9 } - r7c8{ n9   n5 } - r9n5{
c9    . }
r7c2<>1, braid[13]  r7c4{ n1   n5 } - r9c6{ n5   n8 } - r9c3{ n8   n9 } - r9c5{
n9   n7 } - r8n7{ c4   c9 } - c8n7{ r9   r4 } - c9n6{ r8   r2 } - c9n9{ r2   r3 }
- c8n9{ r3   r7 } - c5n9{ r7   r1 } - r1c4{ n9   n7 } - r2n9{ c4   c2 } - c2n7{
r2    . }
r2c9<>8, braid[14]  r6c9{ n8   n5 } - c7n8{ r1   r5 } - c9n6{ r2   r8 } - c7n2{
r5   r7 } - c6n6{ r8   r4 } - r9n2{ c8   c2 } - c6n4{ r4   r3 } - r3n2{ c6   c1 }
- c6n8{ r3   r9 } - r3n3{ c1   c8 } - r9n5{ c6   c8 } - r1c8{ n5   n9 } - c9n9{
r3   r9 } - b9n7{r9c9   . }
r9c2<>8, braid[14]  r9c6{ n8   n5 } - r7n8{ c1   c5 } - r9n2{ c2   c8 } - r7n6{
c5   c7 } - r6n2{ c8   c4 } - c9n6{ r8   r2 } - r6n3{ c4   c3 } - r6n8{ c3   c9 }
- c3n6{ r6   r1 } - c9n5{ r6   r8 } - c3n4{ r1   r8 } - r8c1{ n4   n1 } - r9n1{
c3   c9 } - b9n7{r9c9   . }
r2c2<>2, braid[17]  r3n2{ c1   c6 } - r9n2{ c2   c8 } - r5c6{ n2   n9 } - c8n7{
r9   r4 } - r4n9{ c8   c7 } - r5n7{ c9   c1 } - r2n7{ c1   c4 } - r8n7{ c4   c9 }
- c9n6{ r8   r2 } - r2n9{ c9   c3 } - r8n9{ c3   c4 } - r1c4{ n9   n5 } - r2c6{
n5   n8 } - r2c1{ n8   n1 } - r3c2{ n1   n8 } - c1n8{ r1   r7 } - r7n2{ c1    . }
r2c2<>6, braid[17]  r1n6{ c3   c7 } - c3n6{ r1   r6 } - c9n6{ r2   r8 } - r6c5{
n6   n1 } - r8n7{ c9   c4 } - c4n1{ r8   r7 } - c5n7{ r9   r1 } - c2n7{ r1   r4 }
- c8n7{ r4   r9 } - r9n2{ c8   c2 } - c2n1{ r9   r3 } - c8n1{ r3   r4 } - r4c1{
n1   n5 } - r6c2{ n5   n8 } - r1c2{ n8   n9 } - r2c3{ n9   n8 } - r1n8{ c1    . }
r1c5<>4, braid[18]  r4n4{ c5   c6 } - c3n4{ r1   r8 } - c5n7{ r1   r9 } - c6n6{
r4   r8 } - r8n7{ c4   c9 } - c8n7{ r9   r4 } - c9n6{ r8   r2 } - r4n9{ c8   c7 }
- r8n9{ c7   c4 } - r7c5{ n9   n8 } - r3c5{ n8   n9 } - r9n8{ c6   c3 } -
b3n9{r3c9 r1c8} - c3n9{ r1   r2 } - c3n1{ r2   r5 } - r4c1{ n1   n5 } - r8n5{ c1
 c7 } - b3n5{r2c7   . }
r3c1<>4,   whip[1]  r1n4{ c3    . }
r5c1<>3, braid[18]  r3n3{ c1   c8 } - r5n7{ c1   c9 } - r6n3{ c3   c4 } - r8n7{
c9   c4 } - r9n7{ c5   c8 } - r6n2{ c4   c8 } - c4n1{ r8   r7 } - r9n2{ c8   c2 }
- c8n1{ r7   r4 } - r5c7{ n1   n8 } - r5c3{ n8   n1 } - c9n8{ r6   r3 } -
```

```
b7n1{r9c3 r8c1} - r3n1{ c1   c2 } - r3n9{ c2   c5 } - r1c4{ n9   n5 } - c8n5{ r1
 r7 } - b7n5{r7c2   . }
r1c3<>3,   whip[1]  c1n3{ r3    . }
r3c6<>8, braid[26]  r9c6{ n8   n5 } - r2c6{ n5   n2 } - r5c6{ n2   n9 } - r8c6{
n9   n6 } - r7n6{ c5   c7 } - c9n6{ r8   r2 } - c7n2{ r7   r5 } - r5c4{ n2   n3 }
- r6n3{ c4   c3 } - c3n6{ r6   r1 } - r1n4{ c3   c1 } - r1n3{ c1   c8 } - r3n3{
c8   c1 } - c1n2{ r3   r7 } - r9n2{ c2   c8 } - c8n7{ r9   r4 } - r5n7{ c9   c1 }
- c1n8{ r5   r2 } - r1n8{ c2   c7 } - b3n5{r1c7 r2c7} - c7n1{ r2   r8 } - r8c1{
n1   n5 } - r4c1{ n5   n1 } - r5c3{ n1   n8 } - r6n8{ c2   c9 } - c9n5{ r6    . }
r7c5<>8, braid[15]  r7n6{ c5   c7 } - r9n8{ c6   c3 } - c6n8{ r9   r2 } - c7n2{
r7   r5 } - c9n6{ r8   r2 } - r6n2{ c8   c4 } - c7n8{ r5   r1 } - r6n3{ c4   c3 }
- r5c3{ n3   n1 } - c3n6{ r6   r1 } - r2c3{ n1   n9 } - r1c2{ n9   n7 } - r2c2{
n7   n1 } - c1n1{ r3   r8 } - c7n1{ r8    . }
r9c3<>8,   whip[1]  r7n8{ c2    . }
r7c2<>9,   braid[9]  r7c5{ n9   n6 } - r7n8{ c2   c1 } - c6n6{ r8   r4 } -
b7n2{r7c1 r9c2} - c6n4{ r4   r3 } - r3n2{ c6   c1 } - c1n3{ r3   r1 } - c1n4{ r1
 r8 } - b7n5{r8c1   . }
r7c4<>5,   whip[8]  r7n1{ c4   c8 } - r7n9{ c8   c5 } - r7n6{ c5   c7 } - c7n2{
r7   r5 } - r6c8{ n2   n5 } - r4c7{ n5   n9 } - r8c7{ n9   n5 } - r1n5{ c7    . }
r7c4=1
r1c8<>9,   braid[9]  r1n3{ c8   c1 } - r7n9{ c8   c5 } - r1n4{ c1   c3 } - r7n6{
c5   c7 } - r1n6{ c7   c2 } - c3n6{ r2   r6 } - r6n3{ c3   c4 } - r6n2{ c4   c8 }
- c7n2{ r5    . }
r2c4<>2,   whip[9]  r6n2{ c4   c8 } - c7n2{ r5   r7 } - c1n2{ r7   r3 } - r3n3{
c1   c8 } - r1c8{ n3   n5 } - r7c8{ n5   n9 } - r4n9{ c8   c7 } - c7n5{ r4   r8 }
- c4n5{ r8    . }
r5c6<>2,   whip[1]  c4n2{ r6    . }
r5c6=9
r3c5<>9,   whip[4]  c4n9{ r2   r8 } - r8n7{ c4   c9 } - c9n6{ r8   r2 } -
b3n9{r2c9   . }
r3c9<>8,   whip[6]  r3c5{ n8   n4 } - r4n4{ c5   c6 } - c6n6{ r4   r8 } - r7n6{
c5   c7 } - c7n2{ r7   r5 } - c7n8{ r5    . }
r5c7<>8,   whip[1]  c9n8{ r6    . }
r1c3<>9,   whip[8]  r9c3{ n9   n1 } - r8c3{ n1   n4 } - r8c1{ n4   n5 } - r8c6{
n5   n6 } - c9n6{ r8   r2 } - r2c3{ n6   n8 } - c6n8{ r2   r9 } - b8n5{r9c6   . }
r8c1<>5,   whip[8]  r8c6{ n5   n6 } - r7n6{ c5   c7 } - r7n5{ c7   c8 } -
b9n2{r7c8 r9c8} - r6c8{ n2   n1 } - c5n1{ r6   r4 } - r4c1{ n1   n7 } - c8n7{ r4
  . }
r2c3<>9,   whip[3]  r9c3{ n9   n1 } - r8c1{ n1   n4 } - r8c3{ n4    . }
r9c2<>9,   whip[1]  c3n9{ r8    . }
r9c2<>1,   whip[3]  r8c1{ n1   n4 } - r8c3{ n4   n9 } - r9c3{ n9    . }
r5c3<>1,  braid[6]  b7n1{r9c3 r8c1} - c1n4{ r8   r1 } - c7n1{ r8   r2 } - r3c9{
n1   n9 } - r3c8{ n9   n3 } - r1n3{ c8    . }
r2c7<>1,   whip[6]  r5c7{ n1   n2 } - r5c4{ n2   n3 } - r5c3{ n3   n8 } - r2c3{
n8   n6 } - r1n6{ c2   c7 } - c7n8{ r1    . }
r8c7<>5,   whip[4]  r8c6{ n5   n6 } - r7n6{ c5   c7 } - r1c7{ n6   n8 } - r2c7{
n8    . }
r8c7<>9,   whip[4]  r7n9{ c8   c5 } - r7n6{ c5   c7 } - c7n2{ r7   r5 } - c7n1{
r5    . }
r4c7=9
r3c1<>1,   whip[4]  r2n1{ c3   c9 } - r5n1{ c9   c7 } - r8c7{ n1   n6 } - c9n6{
r8    . }
r8c9<>9,  braid[4]  r7n9{ c8   c5 } - r8n7{ c9   c4 } - r8n5{ c4   c6 } -
b8n6{r8c6   . }
r1c8<>5,   whip[5]  c7n5{ r2   r7 } - r7n6{ c7   c5 } - r6c5{ n6   n1 } - r6c8{
n1   n2 } - c7n2{ r5    . }
r1c8=3
r3c1=3
r2c9<>1,   whip[2]  r3c8{ n1   n9 } - r3c9{ n9    . }
r3c2<>1,   whip[1]  r2n1{ c3    . }
r2c9<>9,   whip[2]  r3c8{ n9   n1 } - r3c9{ n1    . }
r3c2<>9,   whip[1]  b3n9{r3c9   . }
r1c2<>6,   whip[3]  c2n9{ r1   r2 } - c2n1{ r2   r4 } - c2n7{ r4    . }
r6c3<>6,   whip[1]  c2n6{ r4    . }
r1c3<>8,   whip[2]  r5c3{ n8   n3 } - r6c3{ n3    . }
r2c3<>8,   whip[2]  r5c3{ n8   n3 } - r6c3{ n3    . }
r5c1<>8,   whip[1]  c3n8{ r6    . }
r6c2<>8,   whip[1]  c3n8{ r5    . }
r1c2<>8,   whip[3]  r3c2{ n8   n2 } - r7c2{ n2   n5 } - r9c2{ n5    . }
r2c2<>8,   whip[3]  r3c2{ n8   n2 } - r7c2{ n2   n5 } - r9c2{ n5    . }
r2c4<>5,   whip[3]  r2n9{ c4   c2 } - r1c2{ n9   n7 } - r2n7{ c1    . }
r2c7<>6,  braid[3]  r2c3{ n6   n1 } - r8c7{ n6   n1 } - r9n1{ c9    . }
r4c2<>5,   whip[3]  r9c2{ n5   n2 } - r3c2{ n2   n8 } - r7c2{ n8    . }
r4c1<>1,  braid[3]  r5c1{ n1   n7 } - r4n5{ c1   c8 } - r4n7{ c8    . }
r4c2<>6,   whip[3]  c2n1{ r4   r2 } - c2n7{ r2   r1 } - c2n9{ r1    . }
r6c2=6
r6c5=1
r4c1=5
r7c8<>5,   whip[3]  r6c8{ n5   n2 } - r9n2{ c8   c2 } - c2n5{ r9    . }
r2c1<>8,   whip[3]  r2c7{ n8   n5 } - r7n5{ c7   c2 } - r7n8{ c2    . }
r8c3<>1,  braid[3]  r2c3{ n1   n6 } - r8c7{ n1   n6 } - r1n6{ c7    . }
r8c9<>1,   whip[3]  r9n1{ c8   c3 } - r2c3{ n1   n6 } - c9n6{ r2    . }
r9c8<>1,  braid[3]  r3c8{ n1   n9 } - r9c3{ n1   n9 } - c9n9{ r9    . }
r9c9<>5,   whip[3]  r2c9{ n5   n6 } - r2c3{ n6   n1 } - r9n1{ c3    . }
r1c4<>9,   whip[4]  r1n5{ c4   c7 } - r1n6{ c7   c3 } - c3n4{ r1   r8 } - r8n9{
c3    . }
r1c5<>8,   whip[4]  r1c1{ n8   n4 } - r8n4{ c1   c3 } - r8n9{ c3   c4 } - c5n9{
```

```
r9    . }
r1c4<>7,   whip[2]  r1c2{ n7   n9 } - r1c5{ n9    . }
r1c4=5
r9c8<>5,   whip[3]  r9c6{ n5   n8 } - r2n8{ c6   c7 } - c7n5{ r2    . }
Singles to the end
Most difficult rule: NRCZT Braid[26]
```

Edit: Solution given in rigorous notation.
Note that I now list whips, if the given braid is also whip, I list it as a whip.

Last edited by Mauricio on Sat Jan 16, 2010 8:27 am; edited 3 times in total

Back to top profile pm

denis_berthier

Posted: Thu Jan 14, 2010 5:14 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 1164
Location: Paris, France

Hi **Mauricio**,

It's great to have a second implementation of braids.
Your example illustrates the difference between Mlength (20) and real length (26) and confirms that Mlength is not a sufficiently tight lower bound to be very useful.

I haven't seen your code, but let me mention that you have checked it against the SudoRules implementation of braids, using:
- a few detailed solutions I gave here and on my website,
- the 10,000 puzzles of the sudogen0 collection (pB-NRCZT ratings here http://www.carva.org/denis.berthier/HLS/Classification/sudogen0-pB-NRCZT-1-10000.txt)
and that you reported 100% compatibility for the pB-NRCZT ratings, which leaves no doubt on the correctness of your implementation.

Considering how difficult it has been for other tentative implementations of the pB-NRCZT rating, this is worth noticing.

Could you say a few words on your implementation?

[Edit: it may be the place to recall that **Paul** has a correct implementation of whips and the associated pNRCZT rating, tested on the same sudogen0 collection]

Back to top profile pm www

Red Ed

Posted: Thu Jan 14, 2010 8:29 am Post subject: quote

Joined: 06 Jun 2005
Posts: 1041

> **denis_berthier wrote:**
> Could you say a few words on your implementation?

I'd be interested in that, too. How about on the algorithms thread.

Back to top profile pm

denis_berthier

Posted: Thu Jan 14, 2010 8:58 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 1164
Location: Paris, France

> **Red Ed wrote:**
> > **denis_berthier wrote:**
> > Could you say a few words on your implementation?
>
> I'd be interested in that, too. How about on the algorithms thread.

As this discussion about nrczt-braids has started here, in the thread where nrczt-braids were defined, and it is strictly about standard nrczt-braids instead of misleading variants, I can't see any reason to move it elsewhere (especially in a thread that hasn't produced any useable result) as long as it doesn't become overly technical.

Last edited by denis_berthier on Thu Jan 14, 2010 10:27 am; edited 1 time in total

Back to top profile pm www

ronk

Posted: Thu Jan 14, 2010 10:22 am Post subject: quote

> **denis_berthier wrote:**

Joined: 02 Nov 2005
Posts: 2748
Location: Southeastern USA

> As this discussion about braids has started here, in the thread where braids were defined, and it is strictly about standard braids ...

As I feared, you are now forgetting that you agreed to use a qualifier to distinguish between your braids and prior usage of the term. I took that to mean a qualifier would appear at least once in every post using the term.

BTW you never put a qualifier into this thread's title.

Back to top profile pm

denis_berthier Posted: Thu Jan 14, 2010 10:27 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 1164
Location: Paris, France

ronk wrote:

> **denis_berthier wrote:**
>
> > As this discussion about braids has started here, in the thread where braids were defined, and it is strictly about standard braids ...
>
> As I feared, you are now forgetting that you agreed to use a qualifier to distinguish between your braids and prior usage of the term. I took that to mean a qualifier would appear at least once in every post using the term.
>
> BTW you never put a qualifier into this thread's title.

For once, I agree with you. Preaching for non ambiguity as often as I can, I should be more careful about this. I have corrected my previous post.

Back to top profile pm www

Red Ed Posted: Thu Jan 14, 2010 8:22 pm Post subject: quote

Joined: 06 Jun 2005
Posts: 1041

denis_berthier wrote:

> **Red Ed wrote:**
>
> > **denis_berthier wrote:**
> >
> > > Could you say a few words on your implementation?
> >
> > I'd be interested in that, too. How about on the algorithms thread.
>
> As this discussion about nrczt-braids has started here, in the thread where nrczt-braids were defined, and it is strictly about standard nrczt-braids instead of misleading variants, I can't see any reason to move it elsewhere (especially in a thread that hasn't produced any useable result) as long as it doesn't become overly technical.

As the discussion about algorithms has started there, in the thread where several of us with implementations gathered to compare notes, and it is strictly about standard nrczt-stuff when not being misled by your unclear exposition, I can't see any reason to keep it here (especially in a thread that hasn't produced any useable algorithm) as long as it has any non-trivial algorithmic content.

Back to top profile pm

denis_berthier Posted: Thu Jan 14, 2010 9:11 pm Post subject: quote edit

Joined: 19 Jun 2007
Posts: 1164
Location: Paris, France

Poor old **Red Ed**,
Still as counter productive as ever.
You should take Mauricio as an example; instead of spending his time in sterile confrontational posts, he has done a real work and the result is very interesting. Implementing braids with the correct length computation (i.e. with a correct pB-NRCZT rating) is difficult (you should know: you tried and you failed) and you'd better congratulate him for this than try to draw attention on you.

Apart from SudoRules, whose outputs have been widely published in this forum even before you had any interest in all my nrczt stuff, there is currently:
- one and only one correct implementation of nrczt-whips: Paul's,
- one and only one correct implementation of nrczt-braids: Mauricio's.
Both of them have been developed much before (Paul) or independently (Mauricio) of your sudden conversion to nrczt. Your contribution to the whole nrczt stuff, be it theory or implementation, has been inexistent. I defy you to mention any theorem, any additional definition, any implementation idea that would be yours.

As for my exposition, it has been published in peer reviewed international conferences. It isn't my fault if you're unable to read a mathematical definition. Because I had excluded inner loops from whips, you thought that "llc reuse" was excluded from braids. But these are 2 different topics, in relation to two different patterns.

Back to top profile pm www

Red Ed

Posted: Thu Jan 14, 2010 10:31 pm Post subject: quote

Joined: 06 Jun 2005
Posts: 1041

Your LPU papers don't impress me, but if droning on about them (and your book! - don't forget your book!) helps with your self esteem issues then I'm happy to be this evening's excuse.

As for my "conversion" to nrczt - ha ha, you idiot. My interest is only academic, just as yours is only marketing. As it happens (and you would know, if you were competent to understand the algorithms thread), I have a perfectly good whips finding algorithm. And, outside of that thread, I have a braids finder which I am now very keen to compare against Mauricio's. So why don't you get back in your box and give Mauricio some space to show us his **algorithm**. There's a good boy.

Back to top profile pm

Mauricio

Posted: Thu Jan 14, 2010 11:09 pm Post subject: quote

Joined: 22 Mar 2006
Posts: 1084

Now calm down, the algorithm is here. There is where it is more appropiate to post.

Back to top profile pm

denis_berthier

Posted: Fri Jan 15, 2010 5:22 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 1164
Location: Paris, France

Red Ed wrote:

> ... you idiot
> ... if you were competent
> ... why don't you get back in your box
> ... There's a good boy
>
> My interest is only academic

Obviously, we don't have the same notion of "academic".

Back to top profile pm www

PIsaacson

Posted: Fri Jan 15, 2010 10:05 am Post subject: quote

Joined: 02 Jul 2008
Posts: 360
Location: Campbell, CA

Hello all!

I'm still recovering from a PC failure, so excuse my late entrance here. I've attached my braids solution log for comparison to Mauricio's. There are some interesting similarities and differences. I've released my code and described my pseudo-braids implementation elsewhere, but I'm running somewhat crippled until a new PC arrives, so I'll post the exact references later.

From some hasty reading (and forgive me in advance for errors), it looks like Mauricio and Ed both use a technique of starting with a z-target and then propagating the rh truths as if the rh was actually assigned, then going back and reconstructing the chain by finding the associated lh candidates. This is vastly different from the technique that I use which is based on building an adjacency matrix of parent-child nrc relationships, then using standard BFS/DFS tree walk algorithms to produce linear chains of lh-rh conjugate pairs while applying the zt testing/promoting at every even level.

Code:

```
000001002000030040005600700002800003040050060900007400007003004030020080600400300

 3478       6789       34689     |579        4789       1          |5689       359
2
 1278       126789     1689      |2579       3          2589       |15689      4
15689
 12348      1289       5         |6          489        2489       |7          139
189
 ---------- ---------- ----------+---------- ---------- ----------+---------- ----------
----------
```

```
 157       1567      2         |8         1469      469       |159       1579
3
 1378      4         138       |1239      5         29        |1289      6
 1789
 9         1568      1368      |123       16        7         |4         125
 158
 --------- --------- ---------+--------- --------- ---------+--------- ---------
---------
 1258      12589     7         |159       1689      3         |12569     1259
4
 145       3         149       |1579      2         569       |1569      8
 15679
 6         12589     189       |4         1789      589       |3         12579
 1579

  1) r1c1 <> 7 nrc[2x2]-whip  c4n7{r1 r2} - r8n7{c4 c9}
  2) r4c6 <> 9 nrc[3x3]-whip  r5c6{d9 d2} - c7n2{r5 r7} - {6r7c7 .}
  3) r5c7 <> 9 nrc[3x3]-whip  b5n9{54 45} - r4n4{c5 c6} - c6n6{r4 r8}
  4) r6c4 <> 1 nrc[3x3]-whip  r6n2{c4 c8} - b9n2{98 77} - {6r7c7 .}
  5) r7c5 <> 1 nrc[3x3]-whip  c4n1{r8 r5} - c4n3{r5 r6} - b5n2{64 56}
  6) r4c5 <> 9 nrc[3x3]-whip  r4n4{c5 c6} - b5n6{46 65} - r7n6{c5 c7}
  7) r5c4 <> 1 nrc[2x3]-braid r5n9{c4 c6} - r5n2{c6 c4}
                  [1]-strand   c4n3{r5 r6}
  8) r5c9 <> 9 nrc[1x1]-whip  b5n9{54 45}
  9) r9c5 <> 1 nrc[1x1]-whip  c4n1{r7 r8}
 10) r3c6 <> 9 nrc[3x3]-whip  c6n4{r3 r4} - c6n6{r4 r8} - r7n6{c5 c7}
 11) r6c3 <> 1 nrc[3x3]-whip  r6n3{c3 c4} - r6n2{c4 c8} - b9n2{98 77}
 12) r6c2 <> 1 nrc[7x9]-braid r6c5{d1 d6} - b4n6{63 42} - r1n6{c2 c3} - c3n4{r1
r8} - r8c1{d4 d1} - b1n1{31 23} - {1r2c7 .}
                  [3]-strand   r6c5{d1 d6} - b4n6{63 42} - r4n5{c2 c1}
                  [2]-strand   r6c5{d1 d6} - r7n6{c5 c7}
 13) r7c4 <> 9 nrc[8x9]-braid c4n1{r7 r8} - r8n7{c4 c9} - b6n7{59 48} - r4n9{c8
c7} - r8n9{c7 c3} - c3n4{r8 r1} - r1n6{c3 c2} - {7r1c2 .}
                  [3]-strand   c4n1{r7 r8} - r8n7{c4 c9} - c9n6{r8 r2}
 14) r7c7 <> 1 nrc[3x3]-whip  c7n2{r7 r5} - r5c6{d2 d9} - {9r8c6 .}
 15) r4c7 <> 1 nrc[8x10]-braid r4n9{c7 c8} - b6n7{48 59} - r8n7{c9 c4} - c4n1{r8
r7} - c8n1{r7 r3} - r3n3{c8 c1} - c1n2{r3 r2} - {7r2c1 .}
                  [3]-strand   r4n9{c7 c8} - c8n7{r4 r9} - r9n2{c8 c2}
 16) r7c7 <> 9 nrc[3x3]-whip  r7n6{c7 c5} - r6c5{d6 d1} - {1r6c8 .}
 17) r2c6 <> 9 nrc[7x11]-braid r5c6{d9 d2} - c4n2{r5 r2} - c1n2{r2 r3} - r3n3{c1
c8} - r1c8{d3 d9} - b1n9{13 32} - {9r7c2 .}
                  [4]-strand   r5c6{d9 d2} - c7n2{r5 r7} - r7n6{c7 c5} - c5n9{r7
r9}
                  [3]-strand   r5c6{d9 d2} - c4n2{r5 r2} - c4n5{r2 r1}
 18) r1c7 <> 9 nrc[8x10]-braid r4n9{c7 c8} - c8n7{r4 r9} - c5n7{r9 r1} - r1c4{d7
d5} - r1c8{d5 d3} - c1n3{r1 r3} - r3n2{c1 c6} - c6n4{r3 r4}
                  [3]-strand   r4n9{c7 c8} - c8n7{r4 r9} - r9n2{c8 c2}
                  [1]-strand   r4c7{d9 d5}
 19) r7c1 <> 1 nrc[8x10]-braid c4n1{r7 r8} - r8n7{c4 c9} - c8n7{r9 r4} - r4c1{d7
d5} - r8c1{d5 d4} - c3n4{r8 r1} - c3n6{r1 r6} - r6n3{c3 c4}
                  [3]-strand   c4n1{r7 r8} - r8n7{c4 c9} - c9n6{r8 r2}
                  [3]-strand   c4n1{r7 r8} - r8n7{c4 c9} - r5n7{c9 c1}
 20) r9c6 <> 9 nrc[8x12]-braid r5c6{d9 d2} - c7n2{r5 r7} - r7n6{c7 c5} - b5n6{65
46} - c6n4{r4 r3} - r3c5{d4 d9} - c9n9{r3 r8} - {9r8c3 .}
                  [6]-strand   r5c6{d9 d2} - c7n2{r5 r7} - r7n6{c7 c5} - r8c6{d6
d5} - b2n5{26 14} - r1c8{d5 d9}
                  [4]-strand   r5c6{d9 d2} - c7n2{r5 r7} - r7n6{c7 c5} - c5n8{r7
r9}
 21) r6c9 <> 1 nrc[8x12]-braid r6c5{d1 d6} - r7n6{c5 c7} - c7n2{r7 r5} - r5n8{c7
c9} - r3c9{d8 d9} - r1c8{d9 d3} - r3c8{d3 d1} - {1r2c7 .}
                  [5]-strand   r6c5{d1 d6} - r7n6{c5 c7} - c7n2{r7 r5} - r5n8{c7
c9} - b6n7{59 48}
                  [4]-strand   r6c5{d1 d6} - r7n6{c5 c7} - c7n2{r7 r5} - r6c8{d2
d5}
                  [3]-strand   r6c5{d1 d6} - r4n6{c6 c2} - r1n6{c2 c3}
 22) r7c2 <> 1 nrc[7x9]-braid r7c4{d1 d5} - r9c6{d5 d8} - b7n8{93 71} - b7n2{71
92} - r3c2{d2 d8} - c5n8{r3 r1} - {9r1c5 .}
                  [4]-strand   r7c4{d1 d5} - r9c6{d5 d8} - r9c3{d8 d9} - c9n9{r9
r3}
 23) r9c2 <> 8 nrc[7x12]-braid c5n8{r9 r7} - b8n6{75 86} - c9n6{r8 r2} - c3n6{r2
r1} - c3n4{r1 r8} - r8c1{d4 d1} - {1r8c7 .}
                  [5]-strand   r9n2{c2 c8} - r6n2{c8 c4} - r6n3{c4 c3} - r6n8{c3
c9} - c9n5{r6 r8}
 24) r2c7 <> 9 nrc[7x9]-braid r4c7{d9 d5} - r6c9{d5 d8} - b3n8{39 17} - b3n6{17
29} - r2c3{d6 d8} - r5n8{c3 c1} - {1r5c1 .}
                  [4]-strand   r4c7{d9 d5} - r6c9{d5 d8} - r3c9{d8 d1} - r9n1{c9
c3}
 25) r2c9 <> 8 nrc[7x12]-braid b6n8{69 57} - b6n2{57 68} - r9n2{c8 c2} - r3n2{c2
c1} - r3n3{c1 c8} - r1c8{d3 d9} - {9r3c9 .}
                  [5]-strand   c9n6{r2 r8} - c6n6{r8 r4} - c6n4{r4 r3} - c6n8{r3
r9} - r9n5{c6 c8}
 26) r5c1 <> 3 nrc[9x14]-braid c4n3{r5 r6} - r6n2{c4 c8} - b9n2{98 77} - r7n6{c7
c5} - r4c5{d6 d4} - r3n4{c5 c6} - r3n2{c6 c1} - r2n2{c1 c4} - {5r2c4 .}
                  [7]-strand   c4n3{r5 r6} - r6n2{c4 c8} - b9n2{98 77} - r7n6{c7
c5} - r4c5{d6 d4} - r3n4{c5 c6} - c6n8{r3 r2}
                  [3]-strand   c4n3{r5 r6} - r6n2{c4 c8} - r6n1{c8 c5}
                  [3]-strand   c4n3{r5 r6} - r6n2{c4 c8} - r9n2{c8 c2}
```

```
                  [2]-strand   r5n7{c1 c9} - b9n7{89 98}
 27) r1c3 <> 3 pointing pair b4/c3
 28) r1c5 <> 4 nrc[9x11]-braid c6n4{r3 r4} - c6n6{r4 r8} - c7n6{r8 r7} - c7n2{r7
r5} - r6n2{c8 c4} - r6n3{c4 c3} - c3n6{r6 r1} - c2n6{r1 r4} - {5r4c2 .}
                  [7]-strand   c6n4{r3 r4} - c6n6{r4 r8} - c7n6{r8 r7} - c7n2{r7
r5} - r6n2{c8 c4} - r6n3{c4 c3} - r6n8{c3 c2}
                  [3]-strand   c6n4{r3 r4} - c6n6{r4 r8} - c9n6{r8 r2}
 29) r2c2 <> 6 nrc[9x16]-braid c3n6{r1 r6} - r6n3{c3 c4} - r6n2{c4 c8} - r9n2{c8
c2} - c1n2{r7 r3} - c1n3{r3 r1} - r1n4{c1 c3} - r1n8{c3 c2} - {8r6c2 .}
                  [5]-strand   c3n6{r1 r6} - r6n3{c3 c4} - r6n2{c4 c8} - r9n2{c8
c2} - c2n1{r9 r3}
                  [3]-strand   c9n6{r2 r8} - r8n7{c9 c4} - r1n7{c4 c5}
                  [3]-strand   c9n6{r2 r8} - r8n7{c9 c4} - r2n7{c4 c1}
                  [2]-strand   c3n6{r1 r6} - c3n3{r6 r5}
                  [1]-strand   r1n6{c2 c7}
 30) r3c1 <> 4 nrc[1x1]-whip  r1n4{c3 c5}
 31) r2c2 <> 2 nrc[9x17]-braid r9n2{c2 c8} - c8n7{r9 r4} - b4n7{42 51} - r2n7{c1
c4} - r2n9{c4 c3} - r8n9{c3 c4} - r1c4{d9 d5} - r2c6{d5 d8} - {8r2c1 .}
                  [5]-strand   r9n2{c2 c8} - c8n7{r9 r4} - b4n7{42 51} - r2n7{c1
c4} - r8n7{c4 c9}
                  [4]-strand   c6n2{r2 r3} - c6n4{r3 r4} - c6n6{r4 r8} - c9n6{r8
r2}
                  [3]-strand   r9n2{c2 c8} - c8n7{r9 r4} - r4n9{c8 c7}
                  [2]-strand   c6n2{r2 r3} - r3n4{c6 c5}
                  [2]-strand   c6n2{r2 r3} - r5c6{d2 d9}
 32) r3c6 <> 8 nrc[20x25]-braid c6n4{r3 r4} - c6n6{r4 r8} - b9n6{89 77} - c7n2{r7
r5} - r6n2{c8 c4} - r6n3{c4 c3} - c3n6{r6 r1} - r1n4{c3 c1} - c1n3{r1 r3} -
r3n2{c1 c2} - r9n2{c2 c8} - c8n7{r9 r4} - r5n7{c9 c1} - c1n8{r5 r2} - r1n8{c2 c7}
- c7n5{r1 r2} - r8c7{d5 d1} - c1n1{r8 r4} - c5n1{r4 r6} - r6c8{d1 d5}
                  [13]-strand   c6n4{r3 r4} - c6n6{r4 r8} - b9n6{89 77} - c7n2{r7
r5} - r6n2{c8 c4} - r6n3{c4 c3} - c3n6{r6 r1} - r1n4{c3 c1} - c1n3{r1 r3} -
r3n2{c1 c2} - r9n2{c2 c8} - c8n7{r9 r4} - r4n9{c8 c7}
                  [10]-strand   c6n4{r3 r4} - c6n6{r4 r8} - b9n6{89 77} - c7n2{r7
r5} - r6n2{c8 c4} - r6n3{c4 c3} - c3n6{r6 r1} - r1n4{c3 c1} - c1n3{r1 r3} -
c1n2{r3 r7}
                  [9]-strand   c6n4{r3 r4} - c6n6{r4 r8} - b9n6{89 77} - c7n2{r7
r5} - r6n2{c8 c4} - r6n3{c4 c3} - c3n6{r6 r1} - r1n4{c3 c1} - r1n3{c1 c8}
                  [3]-strand   c6n4{r3 r4} - c6n6{r4 r8} - c9n6{r8 r2}
                  [1]-strand   r9c6{d8 d5}
 33) r6c3 <> 8 nrc[20x25]-braid r6n3{c3 c4} - r6n2{c4 c8} - c7n2{r5 r7} - r7n6{c7
c5} - c6n6{r8 r4} - c6n4{r4 r3} - r3n2{c6 c1} - c1n3{r3 r1} - r1n4{c1 c3} -
c3n6{r1 r2} - c9n6{r2 r8} - r8n7{c9 c4} - r1n7{c4 c5} - r1n8{c5 c2} - c1n8{r2 r7}
- r7n5{c1 c2} - r7c8{d5 d9} - r1n9{c8 c4} - c5n9{r3 r9} - r9n8{c5 c6}
                  [13]-strand   r6n3{c3 c4} - r6n2{c4 c8} - c7n2{r5 r7} - r7n6{c7
c5} - c6n6{r8 r4} - c6n4{r4 r3} - r3n2{c6 c1} - c1n3{r3 r1} - r1n4{c1 c3} -
c3n6{r1 r2} - c9n6{r2 r8} - r8n7{c9 c4} - c4n1{r8 r7}
                  [10]-strand   r6n3{c3 c4} - r6n2{c4 c8} - c7n2{r5 r7} - r7n6{c7
c5} - c6n6{r8 r4} - c6n4{r4 r3} - r3n2{c6 c1} - c1n3{r3 r1} - r1n4{c1 c3} -
r1n6{c3 c7}
                  [9]-strand   r6n3{c3 c4} - r6n2{c4 c8} - c7n2{r5 r7} - r7n6{c7
c5} - c6n6{r8 r4} - c6n4{r4 r3} - r3n2{c6 c1} - c1n3{r3 r1} - c1n4{r1 r8}
                  [3]-strand   r6n3{c3 c4} - r6n2{c4 c8} - r9n2{c8 c2}
                  [1]-strand   r6c9{d8 d5}
 34) r7c5 <> 8 nrc[8x12]-braid b8n6{75 86} - r4c6{d6 d4} - r3c6{d4 d2} - c1n2{r3
r2} - r7c1{d2 d5} - r4c1{d5 d7} - c8n7{r4 r9} - {7r9c5 .}
                  [4]-strand   b8n6{75 86} - c6n9{r8 r5} - r5c4{d9 d3} - r5c3{d3
d1}
                  [1]-strand   r7n6{c5 c7}
 35) r8c9 <> 5 nrc[11x19]-braid r8n7{c9 c4} - c4n1{r8 r7} - b8n5{74 96} - c6n8{r9
r2} - c7n8{r2 r1} - c3n8{r1 r5} - c3n3{r5 r6} - c3n6{r6 r1} - r1n4{c3 c1} -
r8c1{d4 d1} - c7n1{r8 r2}
                  [10]-strand   r8n7{c9 c4} - c4n1{r8 r7} - b8n5{74 96} - c6n8{r9
r2} - c7n8{r2 r1} - c3n8{r1 r5} - c3n3{r5 r6} - c3n6{r6 r1} - r1c2{d6 d9} -
r2c3{d9 d1}
                  [9]-strand   r8n7{c9 c4} - c4n1{r8 r7} - b8n5{74 96} - c6n8{r9
r2} - c7n8{r2 r1} - c3n8{r1 r5} - r5n3{c3 c4} - r6c4{d3 d2} - b6n2{68 57}
                  [4]-strand   r8n7{c9 c4} - c4n1{r8 r7} - b8n5{74 96} - r9n8{c6
c5}
                  [1]-strand   c9n6{r8 r2}
                  [1]-strand   r6c9{d5 d8}
 36) r9c3 <> 8 nrc[1x1]-whip  r7n8{c1 c5}
 37) r1c2 <> 7 nrc[11x16]-braid b2n7{15 24} - r8n7{c4 c9} - c9n6{r8 r2} - r1n6{c7
c3} - r6c3{d6 d3} - r6c4{d3 d2} - r5c6{d2 d9} - r8n9{c6 c4} - r7c5{d9 d6} -
r7c7{d6 d5} - {5r2c7 .}
                  [7]-strand   b2n7{15 24} - r8n7{c4 c9} - c9n6{r8 r2} - r1n6{c7
c3} - r6c3{d6 d3} - r6c4{d3 d2} - b6n2{68 57}
                  [5]-strand   b2n7{15 24} - r8n7{c4 c9} - c9n6{r8 r2} - r1n6{c7
c3} - c3n4{r1 r8}
                  [3]-strand   c5n7{r1 r9} - c8n7{r9 r4} - r4n9{c8 c7}
 38) r1c2 <> 8 nrc[8x23]-braid c3n8{r1 r5} - r5n3{c3 c4} - r6c4{d3 d2} - r2n2{c4
c6} - c1n2{r2 r3} - c1n3{r3 r1} - r1c8{d3 d9} - r7n9{c8 c5}
                  [7]-strand   c3n8{r1 r5} - r5n3{c3 c4} - r6c4{d3 d2} - r2n2{c4
c6} - c6n5{r2 r8} - r8c1{d5 d4} - r8c3{d4 d9}
                  [6]-strand   c3n8{r1 r5} - r5n3{c3 c4} - r6c4{d3 d2} - b6n2{68
57} - c7n1{r5 r8} - r8n6{c7 c9}
                  [5]-strand   r6n8{c2 c9} - r3n8{c9 c5} - c5n4{r3 r4} - c5n1{r4
r6} - r6c8{d1 d5}
```

```
                  [5]-strand   r6n8{c2 c9} - r3n8{c9 c5} - c5n4{r3 r4} - r4c6{d4
d6} - c5n6{r6 r7}
                  [3]-strand   r6n8{c2 c9} - r3n8{c9 c5} - r3n4{c5 c6}
                  [1]-strand   r7n8{c2 c1}
 39) r1c3 <> 9 nrc[7x15]-braid r1c2{d9 d6} - b4n6{42 63} - r6n3{c3 c4} - r6n2{c4
c8} - r7n2{c8 c7} - r7n6{c7 c5} - c5n9{r7 r3}
                  [6]-strand   r1c2{d9 d6} - r2c3{d6 d8} - c6n8{r2 r9} - b8n5{96
74} - r7n1{c4 c8} - r3c8{d1 d9}
                  [6]-strand   r1c2{d9 d6} - r2c3{d6 d8} - c6n8{r2 r9} - b8n5{96
74} - r1c4{d5 d7} - c5n7{r1 r9}
                  [5]-strand   r1c2{d9 d6} - r2c3{d6 d8} - c6n8{r2 r9} - b8n5{96
74} - c4n1{r7 r8}
 40) r1c5 <> 9 nrc[7x18]-braid r1n7{c5 c4} - r8n7{c4 c9} - r5n7{c9 c1} - b4n8{51
62} - c9n8{r6 r5} - c9n1{r5 r3} - {1r3c2 .}
                  [5]-strand   r1n7{c5 c4} - r8n7{c4 c9} - c9n6{r8 r2} - r2n9{c9
c3} - r9c3{d9 d1}
                  [3]-strand   r1c2{d9 d6} - b4n6{42 63} - c3n3{r6 r5}
                  [2]-strand   c5n7{r1 r9} - c5n8{r9 r3}
                  [2]-strand   c5n7{r1 r9} - c8n7{r9 r4}
                  [2]-strand   r7c5{d9 d6} - r6c5{d6 d1}
 41) r1c8 <> 9 nrc[8x8]-whip  r1c2{d9 d6} - b4n6{42 63} - r6n3{c3 c4} - r6n2{c4
c8} - r7n2{c8 c7} - r7n6{c7 c5} - r7n9{c5 c2} - b1n9{32 23}
 42) r2c1 <> 7 nrc[9x17]-braid c2n7{r2 r4} - c8n7{r4 r9} - r9n2{c8 c2} - c1n2{r7
r3} - r3c6{d2 d4} - c5n4{r3 r4} - c5n1{r4 r6} - c8n1{r6 r4} - r5c7{d1 d8}
                  [8]-strand   c2n7{r2 r4} - c8n7{r4 r9} - r9n2{c8 c2} - c1n2{r7
r3} - r3c6{d2 d4} - c5n4{r3 r4} - c5n1{r4 r6} - r6c8{d1 d2}
                  [7]-strand   c2n7{r2 r4} - c8n7{r4 r9} - r9n2{c8 c2} - c1n2{r7
r3} - c1n3{r3 r1} - r1n4{c1 c3} - r1n8{c3 c7}
                  [5]-strand   c2n7{r2 r4} - c8n7{r4 r9} - r9n2{c8 c2} - c1n2{r7
r3} - r3n3{c1 c8}
                  [3]-strand   r5n7{c1 c9} - r8n7{c9 c4} - c4n1{r8 r7}
 43) r2c2 <= 7 hidden single in b1
 44) r1c2 <> 6 nrc[11x20]-braid c3n6{r1 r6} - r6n3{c3 c4} - r5c4{d3 d2} - c7n2{r5
r7} - r7n6{c7 c5} - r4n6{c5 c6} - c6n4{r4 r3} - r3n2{c6 c1} - r3n3{c1 c8} -
r1c8{d3 d5} - c7n5{r1 r2}
                  [10]-strand   c3n6{r1 r6} - r6n3{c3 c4} - r5c4{d3 d2} - c7n2{r5
r7} - r7n6{c7 c5} - c5n9{r7 r9} - r9c3{d9 d1} - r8c1{d1 d4} - r8c3{d4 d9} -
c7n9{r8 r4}
                  [6]-strand   c3n6{r1 r6} - r6n3{c3 c4} - r5c4{d3 d2} - c7n2{r5
r7} - r7n6{c7 c5} - r8c6{d6 d5}
                  [4]-strand   c3n6{r1 r6} - r6n3{c3 c4} - r6n2{c4 c8} - r9n2{c8
c2}
                  [1]-strand   r1n9{c2 c4}
 45) r1c2 <= 9 naked single
 46) r6c3 <> 6 pointing pair b1/c3
 47) r6c3 <= 3 naked single
 48) r6c4 <= 2 naked single
 49) r5c4 <= 3 hidden single in c4
 50) r5c7 <= 2 hidden single in r5
 51) r5c6 <= 9 hidden single in r5
 52) r3c9 <> 8 pointing pair b6/c9
 53) r1c1 <> 3 nrc[4x4]-whip  r1n4{c1 c3} - r1n6{c3 c7} - r8n6{c7 c9} - {7r8c9 .}
 54) r1c8 <= 3 hidden single in r1
 55) r3c1 <= 3 hidden single in c1
 56) r2c7 <> 1 naked subset[2] r3c89.<19>
 57) r2c9 <> 1 naked subset[2] r3c89.<19>
 58) r3c2 <> 1 naked subset[2] r3c89.<19>
 59) r2c9 <> 9 naked subset[2] r3c89.<19>
 60) r3c5 <> 9 naked subset[2] r3c89.<19>
 61) r2c4 <= 9 hidden single in r2
 62) r8c7 <= 1 hidden single in c7
 63) r4c7 <= 9 hidden single in c7
 64) r7c4 <= 1 hidden single in b8
 65) r1c1 <> 4 nrc[5x10]-braid r8n4{c1 c3} - r8n9{c3 c9} - r3c9{d9 d1} - r5c9{d1
d8} - r5n8{c1 c3}
                  [4]-strand   r8n4{c1 c3} - c3n9{r8 r9} - r9c5{d9 d8} - r3n8{c5
c2}
                  [4]-strand   r8n4{c1 c3} - r8n9{c3 c9} - r9c9{d9 d7} - r5n7{c9
c1}
 66) singles from here to step 108

894571632176932845325684719512846973748359261963217458287163594439725186651498327
puzzle 1 givens 25 nrczt 25.0     1402.5830 ms
Alpha (x86_64-w64-mingw32-g++) nrczt engine ver 0.8
C:\msys\1.0\home\Paul\sudoku_v1\sudoku.exe -bv -Xprln -B1 -S1:1 -E1
1 out of 1 solved leaving 0 unsolved using 46 queues size 372 25.00 avg givens
25.00 avg scores
(25 1 25.00)
level counts 0) 11384 1) 58416 2) 10228 3) 39643 4) 9300 5) 35156 6) 6585 7)
23276 8) 3236 9) 10196 10) 2482 11) 6919 12) 1834 13) 5594 14) 1557 15) 4000 16)
1198 17) 2365 18) 781 19) 1049 20) 126 21) 318 22) 33 23) 87 24) 33 25) 151 26)
46 27) 115 28) 23 29) 115 30) 23 31) 23 32) 46 33) 115 34) 16 35) 48 36) 28 37)
84 38) 84 39) 27
total cpu time =        1402.7132 milliseconds
  solving rate =           0.7129 puzzles/second
                        1402.7132 msecs/puzzle
```

```
             singles updates/calls 56/55              101.82% eff       0.2651
msec tot       0.0048 msec/call       0.0047 msec/update
             boxline updates/calls 3/49                 6.12% eff       0.2724
msec tot       0.0056 msec/call       0.0908 msec/update
             subsets updates/calls 5/46                10.87% eff       4.6201
msec tot       0.1004 msec/call       0.9240 msec/update
          nrc_chain updates/calls 44/45                97.78% eff    1396.9857
msec tot      31.0441 msec/call      31.7497 msec/update
```

It took my latest alpha 0.8 engine about 1.4 seconds with a pb-NRCZT score of 25.0. It's unoptimized code since I don't trust the new x86_64 gcc 4.5.0 compiler yet and there may be bugs, so caveat emptor.

Cheers,
Paul

[edit] I forgot to state that my pseudo-braids are composed of potentially multiple strands where each strand is a branch in the tree with the z-target as the topmost (0) level. The size of an [NxM] braid is the M part which is the total combined unique parts from each participating strand. The N indicates the length of the current or supporting strand(s). You can see that some strands contain common elements - these are only counted one time since they are "shared" - hence the uniqueness to qualify length. I think this is the correct way to determine the overall length/size of a multi-strand braid. Each strand is exactly linear, but there are cross-connections and shared runs that are difficult to express without making it impossible to read. I'm still working on presentation, so let me know if you have any ideas on how to improve legibility.

Back to top profile pm

denis_berthier

Joined: 19 Jun 2007
Posts: 1164
Location: Paris, France

Posted: Fri Jan 15, 2010 10:31 am Post subject: quote edit

Hi **Paul**,

Glad to see you back with fresh results.

> **PIsaacson wrote:**
>
> It took my latest alpha 0.8 engine about 1.4 seconds with a pb-NRCZT score of 25.0. It's unoptimized code ...

1.4 seconds for a SER=9.5 puzzle is far below anything I could have dreamt of (especially for "unoptimised" code).
With all the good theoretical properties of the pB-NRCZT rating, this makes it a very attractive alternative to SER.

As the nrczt-braid theories have the confluence property, the pB-NRCZT ratings should be totally implementation independent - contrary to the pNRCZT-rating based on whips, for which there may be a very small discrepancy (~ 1 in 10,000).

As you find a pB-NRCZT rating of 25 and Mauricio 26, there are only two possibilities:
- either Mauricio misses a few braids (which could happen only very rarely as he gets the good rating for the 10,000 puzzles in sudogen0)
- or you still have a (probably also rare) problem with length.

Did any of you two check manually each of the braids in your resolution paths?

If you want to test your algorithm on a large collection of puzzles, the 10,000 sudogen0 pB-NRCZT ratings (computed with SudoRules) are here:
http://www.carva.org/denis.berthier/HLS/Classification/sudogen0-pB-NRCZT-1-10000.txt

Back to top profile pm www

Display posts from previous: All Posts Oldest First Go

You **can** vote in polls in this forum