Non-Monotonic Reasoning

Hey! I thought about the problem of non-monotonic reasoning you mentioned yesterday. As I interpreted it, this is a problem because when we find a satisfiable subset, we cannot assume that all its subsets are also satisfiable, which would impact the skipping for finding all minimal cores.

I have an example here:

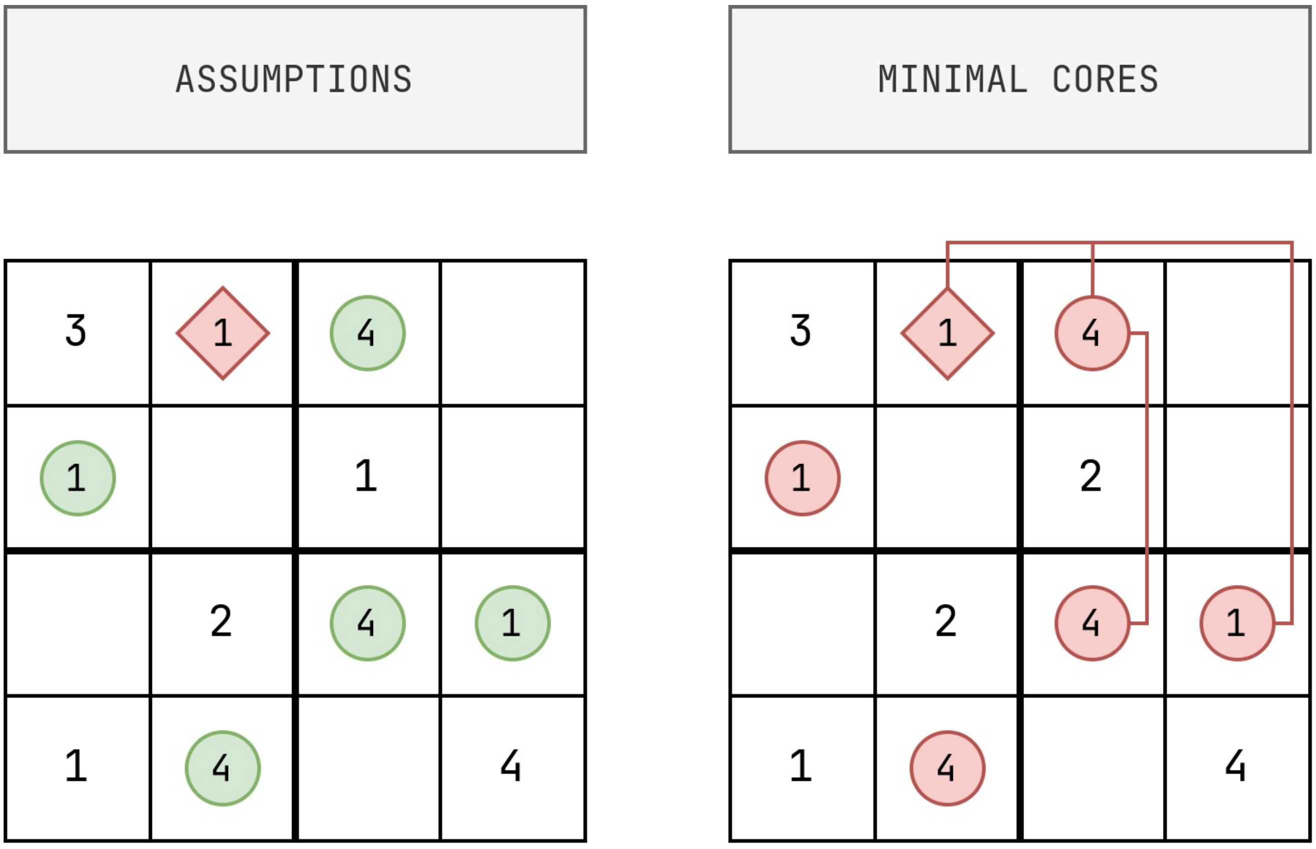
```
{a}. {b}. {c}. {d}.


:- a, b, c, d.
:- a, b, not c.
```

In this example $\{a, b, c, d\}$ would be `UNSAT` so my algorithm would start. Then its subset $\{a, b, c\}$ would be `SAT` again but its subset $\{a, b\}$ could be interpreted as `UNSAT`. That's because if we would run it with the assumptions $\{a, b, \neg c, \neg d\}$ the last integrity constraint would make it `UNSAT`. So we would have an unsatisfiable set as a subset of a satisfiable set.

I think this is a matter of interpretation and definition of our minimal core algorithm. Because to get this unsatisfiable subset we actively have to assume c to not be `True`. I always thought that the minimal unsat core algorithms should only look for minimal cores that are contained in the original assumption set. Because in our assumption set we don't assume $\neg c$ it also shouldn't be part of any minimal core found! And if we only run our encoding with the assumptions $\{a, b\}$ we get `SAT` because a valid assignment like $\{a, b, c, \neg d\}$ is possible. Thus $\{a, b\}$ wouldn't be a minimal core for our assumption set!

To maybe further illustrate why we would only want minimal cores that are subsets of our assumption set I created a small `4x4` Sudoku example:



Here our assumption set is looking the following way:

```
assume(sudoku(1,2,1), True).
assume(sudoku(2,4,4), True).
assume(sudoku(3,1,4), True).
assume(sudoku(3,3,4), True).
assume(sudoku(4,3,1), True).
assume(sudoku(2,1,1), False).
```

This would lead to the following minimal cores :

$$\begin{aligned}&\{\text{sudoku}(1,2,1)\},\\&\{\text{sudoku}(2,4,4)\},\\&\{\text{sudoku}(3,1,4), \text{sudoku}(3,3,4)\},\\&\{\neg\text{sudoku}(2,1,1), \text{sudoku}(3,1,4), \text{sudoku}(4,3,1)\},\end{aligned}$$

Even when a core contains a negation, to find all of them we also only have to look at our original assumption set. There all the assumptions that make up the MUCs are contained. You can imagine that if we would also look for cores containing other assumptions or negations of our assumptions, we would get extremely many other unrelated MUCs which have nothing to do with our original query.

This in contrast to the abstract example above which would only add the $\{a, b, \neg c\}$ minimal core, which is unrelated to the assumption set. But for all examples that are more complex (which is the most common case) this would lead to many more unrelated and unnecessary minimal cores.

So in my opinion when we are only looking for minimal cores that are contained in the assumption set we can skip over all subsets of an already found satisfiable set.

I'm not really sure if I got the concept of non-monotonic reasoning in this context right. Please let me know if there's a misunderstanding on my side or if there are any holes in my reasoning (Which is very well possible) :)