CS202 ASSIGNMENT

SUDOKU IN SAT

SUDOKU GENERATION CONSTRAINTS

- For each number (1-9) each cell has atmost 1 number.
- For each row all number occur at least one.
- For each column all number occur at least one.
- For each box number occur at least one.
- For each diagonal number occur at least one.
- Point 1 and 2 will suffice that every cell has exactly one number.
- Fixed values constraint.

ENCODING

- Expressed each number corresponding to a cell in the base-9 notation.
- 9^2 (row) + 9 (column) + (number in cell)
- Hence having total of 729 variables.

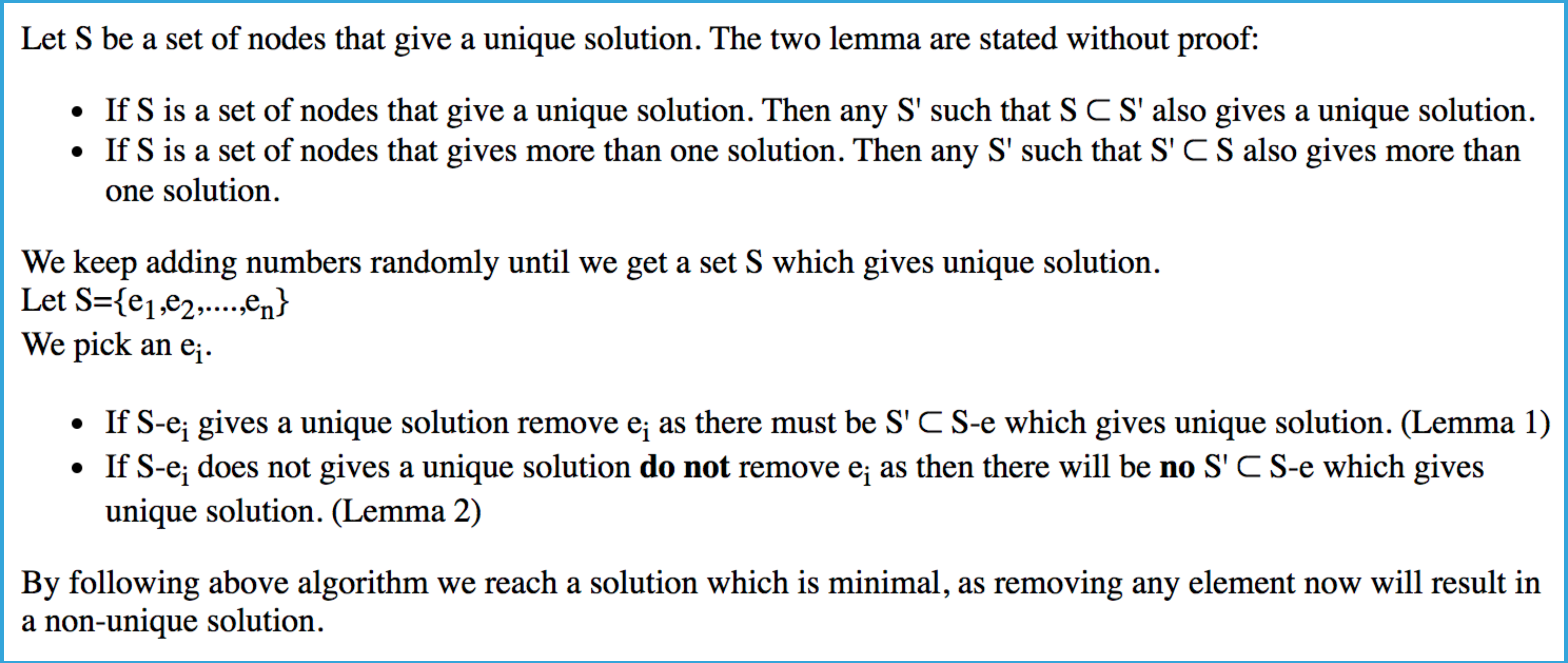
HOW IS MY SUDOKU ALWAYS DIFFERENT?

- Uses initial -rnd-seed function of minisat and seed itself is rand()
- Seed value set to time(NULL) to ensure it changes every time.

MINIMAL SUDOKU GENERATION

- Generate a sudoku using sudoku solver.
- Follow the algorithm in next slide.
- How to check if a sudoku has more than 1 solution?
- Negate the fixed values, if there exist another solution-SAT else SAT.

AS A COMPUTER SCIENCE STUDENT OUR DUTY IS TO GIVE ROCK SOLID PROOF FOR CORRECTNESS.



Let S be a set of nodes that give a unique solution. The two lemma are stated without proof:

- If S is a set of nodes that give a unique solution. Then any S' such that $S \subset S'$ also gives a unique solution.
- If S is a set of nodes that gives more than one solution. Then any S' such that $S' \subset S$ also gives more than one solution.

We keep adding numbers randomly until we get a set S which gives unique solution.
Let $S=\{e_1,e_2,....,e_n\}$
We pick an e_i.

- If $S\text{-}e_i$ gives a unique solution remove e_i as there must be $S' \subset S\text{-}e$ which gives unique solution. (Lemma 1)
- If $S\text{-}e_i$ does not gives a unique solution **do not** remove e_i as then there will be **no** $S' \subset S\text{-}e$ which gives unique solution. (Lemma 2)

By following above algorithm we reach a solution which is minimal, as removing any element now will result in a non-unique solution.

Link: http://home.iitk.ac.in/~abhyuday/minimalsudoku.html

IMPROVISATIONS

- Ref Slide 2, halved the time by removing additional constraints.

- Instead of removing numbers in generator.cpp, we added starting from an empty sudoku, almost reducing the time quadruple times*.

- *Elements in minimal sudoku lie in range 17-22 (It can’t be less than that). So, removing 60 values take almost quadruple time.

THANK YOU!