Complexity Theory Spring 2023

Charles E. Hughes
Computer Science
College of Engineering and Computer Science
University of Central Florida

email: charles.hughes@ucf.edu

Structure: TR 0900-1015 (9:00AM-10:15AM); 28 class periods, each 75 minutes long.
Room: HEC-103 when we meet face-to-face; **https:tinyurl.com/4drwcssj**

Instructor: Charles Hughes; Contact: charles.hughes@ucf.edu; Use Subject COT6410
Office Hours: Tuesday/Thursday 10:45-12:00PM
OH Zoom Link: https://tinyurl.com/49sx2mde
GTA: Wei Zhang; Contact: we991015@knights.ucf.edu; Use Subject COT6410
Office Hours: Monday/Wednesday 2:30-4:00PM
OH Zoom Link: https://ucf.zoom.us/j/2408121701

Required Reading: All class notes linked from this site and available in Webcourses.
Recommended Reading:

- **Sipser, *Introduction to the Theory of Computation 3rd Ed.*, Cengage Learning, 2013.**
- **Hopcroft, Motwani&Ullman, Intro to Automata Theory, *Languages and Computation 3rd Ed.*, Prentice-Hall, 2006.**
- **Garey&Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1979.**
- Cooper, *Computability Theory 2nd Ed.*, Chapman-Hall/CRC Mathematics Series, 2003.
- Davis, Sigal&Weyuker, *Computability, Complexity and Languages 2nd Ed.*, Acad. Press (Morgan Kaufmann), 1994.
- Papadimitriou & Lewis, *Elements of the Theory of Computation*, Prentice-Hall, 1997.
- Bernard Moret, *The Theory of Computation*, Addison-Wesley, 1998.
- Oded Goldreich, *Computational Complexity: A Conceptual Approach*, Cambridge University Press, 2008.
- Oded Goldreich, *P, NP, and NP-Completeness: The Basics of Complexity Theory*, Cambridge University Press, 2010.
- Arora&Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.

Web Pages:
Base URL: https://www.cs.ucf.edu/courses/cot6410/Spring2023
Notes URL: Introductory Notes; Formal Language and Automata Theory; Computability Theory; Complexity Theory
2021 Videos URL: OlderVideos/

Assignments: 6 for sure and maybe a seventh; Paper + Video Presentation with Slides

Exams: Midterm and Final.

Exam Dates (Tentative): Mid Term: Tuesday, March 7 in HEC-101; Withdraw Deadline: Friday, March 24; Spring Break: March 13-19; Final: Thurs., April 27, 7:00AM-9:50AM in **HEC-101**.

Evaluation (Tentative):

1. Mid Term: 125 points; Final Exam: 125 points (balance between weights will be adjusted in your favor)
2. Assignments: 75 points;
3. Individual Paper Reviews and Presentations: 125 points
4. Extra -- 50 points used to increase weight of best exam, always to your benefit
5. Total Available: 500
6. Grading will be A >= 90%, B+ >= 85%, B >= 80%, C+ >= 75%, C >= 70%, D >= 50%, F < 50%; minus grades might be used.

.

Weeks#1: (1/10, 1/12) -- Syllabus; About Me; Preliminaries; Introduction; Formal Languages and Automata Theory,
Videos from Last Year (2022) are here
The old videos for this week are: Preliminary.mp4, Intro.mp4, Day1.mp4, Day2.mp4
New Videos: Day1.mp4, Day2.mp4

1. **Ground rules**
2. **Decision problems (HMU 1.5)**
3. **Solving vs checking**
4. **Procedures vs algorithms**
5. **Introduction to the theory of computation**
6. **Terminology, goals**
7. **Overview of automata (finite, pushdown, linear bounded, Turing machines)**
8. **Overview of formal languages (regular, context free, context sensitive, phrase structured)**
9. **Regular Languages (Sipser 1.1; HMU 2.1, 2.2)**
    - **Recognized by Deterministic finite state automata (DFA)**
10. **Closure Properties of Regular Languages (Sipser 1.2; HMU 4.2)**
    - **Complement**
    - **Union, Intersection, Difference, Exclusive Union, etc.**
        - **Based on parallel DFA construction with only Final States changing**
    - **Reversal**
        - **Motivates Non-Determinism when we reverse orientation of directed edges**
11. **Other Formal Models for Regular Languages**
    - **Recognized by Deterministic finite state automata (DFA) (Sipser 1.1; HMU 2.1, 2.2)**
    - **Recognized by Non-Deterministic finite state automata (NFA) (Sipser 1.2; HMU 2.3-2.5)**
        - **Lambda-Closure use in this construction**
    - **Denoted by Regular Expressions (Sipser 1.3; HMU 3.1-3.3)**
        - **Every Regular Set is a Regular Languages**
12. **Review material created by Prof. Jim Rogers, Earlam College**
13. **Review material from Prof. David Workman, UCF Retired**

Financial Aid (Assignment#1)

Survey at Webcourses

Due: Friday, January 13 at 11:59 PM

▲

Week#2: (1/17, 1/19) -- Formal Languages and Automata Theory

The old videos for this week are: Day3.mp4, Day4.mp4
New Videos: Day3.mp4, Day4.mp4

1. **Continue Automata/Formal Languages Review**
2. **Every Regular Language is a Regular Set (Sipser 1.3; HMU 3.1-3.3)**
    - **Rijk construction**
    - **State ripping**
    - **Solutions to Sets of Simultaneous Regular Equations (Arden's Theorem)**
        - **R = Q+RP, P does not contain lambda, then R = QP* is unique solution**
3. **State minimization using O($| Q |^2$) table. Note alphabet is constant size. (Sipser p. 327; HMU 4.4)**
    - **Why is this just O($| Q |^2$) cost, rather than O($| Q |^3$)?**
4. **More closure properties (Sipser 1.1; HMU 4.2-4.3)**
    - **Reversal and regular expressions**
    - **Substitution**
    - **Quotient using NFA**
    - **Quotient redone use meta technique based on substitution and intersection with Regular Sets**
    - **Meta technique for Init (Prefix), Last, (Suffix) Mid (Substring), Exterior, etc.**
5. **Closure under min and max and discussions of various Reaching Algorithms (Depth-First Search)**
6. **Pumping Lemma for Regular Languages (Pigeon Hole Principle) (Sipser 1.4; HMU 4.1)**
    - **Adversarial Nature of use of PL to show a language is not Regular**
7. **Myhill-Nerode Theorem (implicit in HMU 4.4)**
    - **Right invariant equivalence relations of finite index**
    - **Specific right invariant equiv. relation, R_L for language L**
8. **Myhill-Nerode as proof of min DFA uniqueness**
9. **Myhill-Nerode as a tool to show languages are not regular.**

▲

Week#3: (1/24, 1/26) -- Formal Languages and Automata Theory

The old videos for this week are: Day5.mp4, Day6.mp4
New Videos: Day5.mp4, Day6.mp4

1. **More Discussion of Pumping Lemma for Regular Languages (Sipser 1.4; HMU 4.1)**
    - **Adversarial Nature of use of PL to show a language is not Regular**
2. **The chosen language is**
   L = { a^n b^m | n is not equal to m}.
   This can be shown easily in an indirect manner by showing its complement is not regular. A direct approach is using right invariant equivalence classes as it is not amenable to the Pumping Lemma.
3. **Finite State Machines (Transducers): Mealy versus Moore Model (Sipser p. 87)**
4. **Decision Problems for Regular Languages (HMU 3.4; HMU 4.3**
5. **Introduction to Grammars (Regular and CFG mainly) (Sipser 2.1;**
6. **Regular Grammars and Regular Languages (HMU p. 180)**
    - **Rules of Form A →a; A → aB; A → Define Right Linear Grammars**
    - **Conversion of DFA to Equivalent Right Linear Grammar**
    - **Conversion of Right Linear Grammar to Equivalent NFA**
    - **Equivalence of Right and Left Linear Grammars as Generators of Languages**
    - **Why We Cannot Mix Right and Left Linear Grammars and stay in Regular Languages**

- Extended Right and Left Linear grammars Using String of Terminals

7. Context Free Grammars (A $\rightarrow$ x where x $\in$ $(V \sqcup \Sigma)^*$) (Sipser 2.1; HMU 5.1)
8. Use of context free grammars in parsing and notion of Ambiguity (Sipser 2.1; HMU 5.2, 5.4)
   - Bottom-Up (Shift-Reduce and conflicts)
   - Top-Down (Predictive and guessing -- recursive descent)
9. Review of reduced CFGs (HMU 7.1)
   - Remove Rules: Null, Unit, Non-Productive, Unreachable (order of removal matters)
10. Chomsky Normal Form (CNF) (Sipser 2.1; HMU 7.1)
    - A -> a and A -> BC are only allowed forms
11. The use of CNF in the Cocke-Kasami-Younger O(N^3) parsing of CFLs generated by CNFs (HMU 7.4)
    - Bottom-Up technique
    - Great Example of the value of Dynamic Programming
12. Pumping Lemma for CFLs (Sipser 2.3; HMU 7.2)
13. Show L1 = { a^n b^n c^n | n > 0 } and L2 = { ww | w is in $\{a,b\}^+$ } are not CFLs
14. Non-closure of CFLs under intersection and complement (Sipser 2.3)
15. Consider L3 = { a^n b^n c^m | n,m > 0 }, L4 = { a^m b^n c^n | n,m > 0 }, L1 = L3 ∩ L4
16. L3 and L4 are CFLs, [For L3 S->Sc | Tc; T->aTb | ab], but intersection is not
17. Consider the complement of L2
    { xy | |x| = |y| but x is not the same as y }
    can be first viewed as
    {x1 a x2 y1 b y2 | |x1|=|x2|, |y1|=|y2|} Union
    {x1 b x2 y1 a y2 | |x1|=|x2|, |y1|=|y2|}.
    But this can also be seen as
    {x1 a y1 x2 b y2 | |x1|=|x2|, |y1|=|y2|} Union
    {x1 b y1 x2 a y2 | |x1|=|x2|, |y1|=|y2|}.
    The above is easy to show as a CFL. We then union this with odd length strings and we have L2 complement.

    Assignment #2

    See Webcourses (Assignment # 2) for description
    Sample of Similar Problems with Solutions
    More Discussion of HALF
    Worked out Other Samples

    Due: 2/7 (Key)

Week#4: 1/31, 2/2) -- Formal Languages and Automata Theory; Computability Theory

The old videos for this week are: Day7.mp4, Day8.mp4
New Videos: Day7.mp4, Day8.mp4

1. Pushdown Automata (Notion of Instantaneous Descriptor (ID))
2. Top Down versus Bottom Up parsing and relation of PDAs to CFLs
3. Decision problems for CFLs: is L(G) empty or finite/infinite are fine (Sipser 4.1; HMU 7.4)
4. Closure of CFLs under substitution and intersection with Regular (HMU 7.3)
5. Checking ambiguity, equality to Sigma*, equivalence and non-empty intersection with another CFL are nasty
6. Show L1 = { a^n b^n c^n | n i> 0 } and L2 = { ww | w is in $\{a,b\}^+$ } are both CSLs
7. Very Basic Material on Context Sensitive Grammars (CSG) and Linear Bounded Automata (LBA)

8. **Insights from intro to computability material**
9. **Basic notions of computability and complexity**
10. **Existence of unsolvable problems (counting and diagonalization)**
11. **Solved, solvable (decidable, recursive), unsolved, unsolvable, re, non-re**
12. **Hilbert's Tenth**
13. **Undecidable problems made a bit more concrete**
14. **Lots about problems and their complexity**
15. **Halting Problem (HALT) is re, not decidable**
16. **Set of algorithms (TOT) is non-re**
17. **Halting Problem seen as fun**
18. **Some consequences of non-re nature of algorithms**

▲

Week#5: (2/7, 2/9) -- Computability Theory

The old videos for this week are: Day9.mp4, Day10.mp4
New Videos: Day9.mp4, Day10.mp4

1. **Models of computation**
    - Turing machines
    - Register machines
    - Factor replacement systems (FRS)
        - Originally introduced by John Conway, "Unpredictable Iterations," *Proceedings of the 1972 Number Theory Conference* (1972), pp. 49-52
        - Google Fractran for more information on Conway's fractional computation systems Weisstein, Eric W. "FRACTRAN." From MathWorld--A Wolfram Web Resource. https://mathworld.wolfram.com/FRACTRAN.html
        - The 3x+1 problem https://arxiv.org/pdf/math/0309224.pdf
2. **Systems related to FRS (Petri Nets, Vector Addition, Abelian Semi-Groups)**
3. **RM simulated by Ordered FRS**
4. **Primitive Recursive Functions (prf)**
5. **Initial functions**
6. **Closure under composition and recursion**
7. **Primitive recursive functions SNAP, TERM, STP and VALUE**
8. **Primitive Recursive Function in detail**
9. **Addition and multiplication examples**
10. **Sample functions and predicates**
11. **Closure under cases**
12. **Bounded minimization**
13. **Arithmetic functions that use bounded search**
14. **Pairing functions**
15. **Limitations of primitive recursive**
16. **mu-recursion and the partial recursive functions**
17. **Notions of instantaneous descriptions**
18. **Encodings**
19. **Equivalence of models**
20. **TMs to Register Machines**
21. **RM to Factor Replacement Systems**
22. **Factor Replacement to Recursive Functions**
23. **Gory details on FRS to REC**
24. **Universal machines**
25. **Recursive Functions to TMs**
26. **Consequences of equivalence**

Assignment #3

See Webcourses (Assignment # 3) for description
Sample of Similar Problems with Solutions

Due: 2/14 (Key)

▲

Week#6: (2/14, 2/16) -- Computability Theory

The old videos for this week are: Day11.mp4, Day12.mp4
New Videos: Day11.mp4, Day12.mp4

1. Recursive Functions to TMs (completes equivalence)
2. Consequences of equivalence
3. The primitive recursive predicate STP and function VALUE
4. Review Undecidability (Halting Problem, shown by diagonalization)
5. RE sets and semidecidability
6. Enumeration Theorem: The set of all re sets W_0, W_1, W_2, ...
7. The set K = { n | n is in the n-th re set } = { n | n is in W_n } is re, non-recursive
8. The set K_0 = HALT = { <n,x> | x is in the n-th re set }
9. Alternative characterizations of re sets
10. Parameter Theorem (aka Sm,n Theorem)
11. Quantification and re sets
12. Quantification and co-re sets
13. Quantification and non-re / non-co-re sets
14. Reduction
15. Classic sets Halt = Ko = Lu, NON-EMPTY (Lne), EMPTY (Le)
16. Reduction from Ko that shows undecidability of K, HasZero, IsNonEmpty
17. Complete re set (K and K_0 as examples)
18. Note: To be re-complete a set must be re
19. Reduction from Ko to HasIndentity, TOTAL, IsZero, IsEmpty, IsIdentity
20. Equivalence of certain re sets (K, HasZero, IsNonEmpty, HasIndentity) to Ko
21. Equivalence of certain non-re sets (IsZero, IsEmpty, IsIdentity) to TOTAL
22. Reducibility and degrees (many-one, one-one, Turing)
23. Hierarchy or RE equivalence class (m-1 and 1-1 degrees)
24. Rice's Theorem

Assignment #4

See Webcourses (Assignment # 4) for description
Sample with key
A Tour of Undecidable Problems

Due: 2/23 (Key)

▲

Week#7: (2/21, 2/23) -- Computability Theory

The old videos for this week are: Day13.mp4, Day14.mp4
New Videos: Day13.mp4, Day14.mp4

1. **"Picture" proofs for Rice's Theorem**
2. **Constant Time and Mortal Machines**
3. **Introduction to rewriting systems (Thue, Post)**
4. **Union, intersection, complement for recursive, re, and non-re sets (can be, cannot be)**
5. **Rewriting systems**
6. **Post Canonical Forms**
7. **Thue and Semi-Thue systems (relation to group theory)**
8. **Word problems (Semi-Thue and Thue) and equivalence problems (Thue)**
9. **Simulating Turing Machine by Semi-Thue System**
10. **Simulating Turing Machines by Thue Systems**
11. **Grammars and re sets**
12. **Brief introduction to Post Correspondence Systems and Relation to Semi-Thue Systems**
13. **Post Correspondence Problem (in detail)**
14. **Unsolvable problems related to context-free grammars/languages**
15. **Ambiguity of CFGs**
16. **Non-Emptiness of CFL Intersections**
17. **Context-Sensitive Grammars and Unsolvability Results**
18. **Valid (CSL) and Invalid Traces (CFL)**
19. **Details on Valid (CSL) and Invalid Traces (CFL)**
20. **Intersection of CFLs revisited**
21. **Quotients of CFLs**
22. **Type 0 grammars and Traces**
23. **L = Sigma* for L a Regular or CFL**
24. **L = L^2 for L a CFL**
25. **Summary of Grammar Results**

▲

Week#8: (2/28, 3/2) -- Review and Complexity Theory

1. **Review session and sample exams**
   - **Exam Topics**
   - **Sample exam1; Sample exam1 key**
   - **Sample exam2; Sample exam2 key**
   - **Key to Samples from Notes**
   - **Yet Other Examples (Focused on Formal Languages and Automata Theory)**
   - **Yet Other Examples key (Focused on Formal Languages and Automata Theory)**
   - **Midterm Legal Cheat Sheet**
   - **A Tour of Undecidable Problems**
2. **Basics of Complexity Theory**
3. **Decision vs Optimization Problems (achieving a goal vs achieving min cost)**
4. **Polynomial == Easy; Exponential == Hard**
5. **Polynomial reducibility**
6. **Verifiers versus solvers**
7. **P as solvable in deterministic polynomial time**
8. **NP as solvable in non-deterministic polynomial time**
9. **NP as verifiable in deterministic polynomial time**
10. **Concepts of NP-Complete and NP-Hard**
11. **Canonical NP-Complete problem: SAT (Satisfiability)**
12. **Some NP problems that do not appear to be in P: Graph Coloring, Vertex Cover, SubsetSum**
13. **Million dollar question: P = NP ?**
14. **Boolean expressions: Tautologies, Satisfiability, Truth Tables**
15. **Axiomatic Systems for propositional logic: Substitution and modus ponens versus refutation/resolution**

16. **Unsolvability of deducibility in fragments of propositional calculus**

▲

Week#9 (3/7, 3/9: Midterm Exam on 3/7 and then Complexity Theory

1. **Midterm (Tuesday, March 7 in HEC-101)**
2. **Midterm Key**
3. **More Comments on unsolvability of deducibility in fragments of propositional calculus**
4. **Construction that maps every problem solvable in non-deterministic polynomial time on TM to SAT**
5. **SAT is polynomial reducible to (<=$_P$) 3SAT**
6. **3SAT as a second NP-Complete problem**
7. **Integer Linear Programming**
8. **3SAT <=$_P$ SubsetSum**
9. **SubsetSum <=$_P$ Partition**
10. **Partition equivalence to SubsetSum**

▲

Week#10 Spring Break

▲

Week#11: (3/21, 3/23, (3/24 is Withdrawal Deadline)) -- Complexity Theory

1. **Discussion of presentations**
2. **Reduction of 3SAT to k-Vertex Cover**
3. **3-SAT to 3-Coloring**
4. **Isomophism of k-Coloring with k-Register Allocation of live variables**
5. **Scheduling problems introduced**
6. **Scheduling on multiprocessor systems**
7. **Scheduling problems (fixed number of processors, minimize final finishing time)**
8. **N processors, M tasks, no constraints**
9. **Partition and scheduling problems**
10. **Greedy heuristics**
11. **2-processor scheduling -- greedy based on list, sorted long to short, sorted short to long, optimal. Tradeoffs.**
12. **Precedences (lists, delays, preemption)**
13. **Anomalies (reducing precedence, increasing processors, reducing times)**
14. **Unit Execution Time: Trees, forest, anti forests**
15. **UET: DAGs and m=2**
16. **Hamiltonian Path**
17. **Traveling Salesman**
18. **Knapsack (relation to SubsetSum), Dynamic Programming Pseudo-polynomial Solution**
19. **Bin packing (fixed capacity, minimize number of bins)**
20. **Pseudo polynomial-time solution for Knapsack using dynamic programming with changed parameters,**
    n*W versus 2^n but W = 2^log(W). This kind of problem is called Weak NP Complete. I'll chat about that later.
21. **You will each, individually, develop a paper and a presentation, based on an existing research paper, just as if you had to present to your peers and advisors. In addition to presenting the results and the**

way in which these were proven, you should comment on the paper's importance, readability, and replicability, and even its validity if you question that, just as if you were a journal or conference reviewer. Your report must be a tutorial on the topic of the paper so those with less time to delve into the paper can get a strong sense of the results and the context of those results. Specifying new open problems that you see as interesting is also a goal, but may not be attainable for some of these. The length of your paper is 6 to 12 pages double-spaced, 1" margins all around, using either Times Roman or Calibri 11 point, or Arial 10-point. The references are in addition to the narrative and appear on separate pages that do not count against the 6 to 12-page limit. Images may be used but if the total number of images exceeds a page, then all past that one-page aggregate will lead to a requirement for additional text. The presentation slides must be designed to support your 8 to 10-minute video. This means that there are likely 10 to 15 slides.

22. **Towards the end of the semester you give a presentation to a small group of other students and they to you. This will be done in a breakout Zoom session where your presentation and any questions/answers are captured. You may, if you wish, create the presentation in advance and share a video, but you need to be there to participate in the Q&A and to hear other students' presentations. To do this, I assume you all have access to a webcam and use Zoom or a similar tool that captures your slides and you in one extended screen. Let me know if you lack a webcam. Note that you will be asked to provide some brief feedback to me on the papers of your fellow students taking part in teh same breakout.**
23. **I have placed about 55 sample papers at Sample Topics. You may choose from any of these except for ones that have already been taken. You may also choose your own separate from these with permission from me. In general, the minimum page length in IEEE 2-column format is 8 pages or 10 pages in single column, single-spaced layout. The prefix CLAIMED_NAME means NAME has already chosen that. Please send me an email to assure you succeed in making your "claim."**
24. **Realize that many of these suggested papers will be from arXiv and, while arXiv is a fantastic source, it is not refereed so if you question the validity of some result, that is actually a reasonable outcome so long as you present their approach and your counterarguments just as if you were a reviewer.**
25. **I also put out a folder that contains some examples (papers and presentations) from past semesters.**

Final Paper and PowerPoint

Read above. Its weight is 125 points. The split is nominally 75 for the paper and 50 for your presentation
and feedback on other presentations you provide.

Due: 4/21

Assignment #5

See Webcourses (Assignment # 5) for description
Sample with Key

Due: 4/4 (Key)

▲

Week#12: (3/28, 3/30) -- Complexity Theory

1. **Tiling the plane and Bounded Tiling**
2. **Bounded PCP**
3. **Co-NP**
4. **Reduction techniques**

5. P = co-P ; P contained in intersection of NP and co-NP
6. NP-Hard
7. QSAT as an example of NP-Hard, possibly not NP
8. NP-Hard problems are in general functional not necessarily decision problems
9. NP-Complete are decision problems as they are in NP
10. NP-Easy -- these are problems that are polynomial when using an NP oracle
11. NP-Equivalent is the class of NP-Easy and NP-Hard problems
    - In essence this is functional equivalent of NP-Complete but also of co-NP Complete
    - NP-Equivalent uses a Turing machine polynomial time reduction so just uses rather than accepts answers from oracle
12. Optimization versions of SubsetSum and K-Color
13. Validity of SubsetSum optimization reduction to SubsetSum Decision Problem Oracle
14. 2SAT (linear time complexity)
15. Uniform Min-1 is NP Equivalent
16. Triangle Strip versus Triangle List
17. Weakly versus Strongly NP-Hard/NP-Complete

Assignment #6

See Webcourses (Assignment # 6) for description
Sample with Key

Due: 4/18 (Key)

▲

Week#13: (4/4, 4/6) -- Complexity Theory

1. PSPACE, NPSPACE, CO-PSPACE, PSPACE-COMPLETE
2. EXPSPACE, EXPTIME, NEXPTIME
3. ATM (Alternating NDTM)
4. QSAT, Petri Nets, Presburger
5. Why does PSPACE = NPSPACE? Key is Savitch's Theorem
6. PSPACE-Complete problems (CSL membership, QSAT)
7. FP is functional equivalent to P; R(x,y) in FP if can provide value y for input x via deterministic polynomial time algorithm
8. FNP is functional equivalent to NP; R(x,y) in FNP if can verify any pair (x,y) via deterministic polynomial time algorithm
9. TFNP is the subset of FNP where a solution always exists, i.e., there is a y for each x such that R(x,y).
    - Task of a TFNP algorithm is to find a y, given x, such that R(x,y)
    - Unlike FNP, the search for a y is always successful
    - FNP properly contains TFNP contains FP (we don't know if proper)
10. Factoring is in TFNP, but is it in FP?
    - If TFNP in FP then TFNP = FP
11. P = (NP intersect Co-NP) is interesting analogue to intersection of RE and co-RE but may not hold here
12. It appears that TFNP does not have any complete problems!!!
    - The deal is that there is no known recursive enumeration of TFNP but there is of FNP
    - This is similar to total versus partially recursive functions (analogies are everywhere)
13. Khot's Conjecture and its implications

Week#14: (4/11, 4/13) More Computability Notes

1. **Cleanups (topics that were left hanging earlier)**
   - **Mortality Problem**
   - **Constant Execution Time**
   - **Extensions of Finite Power Property**
2. **Final Exam Topics**
3. **More stuff for final**
4. **Sample Final Exam (Key)**
5. **Exam Review (emphasis on Formal Languages and Computability topics)**

Week#15: (4/18, 4/20)

1. **Exam Review (emphasis on complexity topics and sample exam questions)**
2. **On 4/20 we will have the breakout sessions in which you give your presentations and hear those of a few of your fellow students (typically five per breakout room). These will be set up prior to the 20th.**
3. **Rules of the game and group composition for Breakout Rooms on the 20th**
4. **Breakout Room Videos**

Assignment #7

Just turn in your discussion of someone's or several someone's presentation(s)

Due: 4/23

Week#16: (4/25 -- Review Session; 4/27 -- Final Exam in HEC-101)

1. **I will run a help session on Tuesday, 4/25 at the normal class time**
2. **All Project Videos, Papers, and Presentations are Due on Friday, 4/21 by midnight**
3. **Final Exam is Thursday April 27; 7:00AM to 9:50AM**
