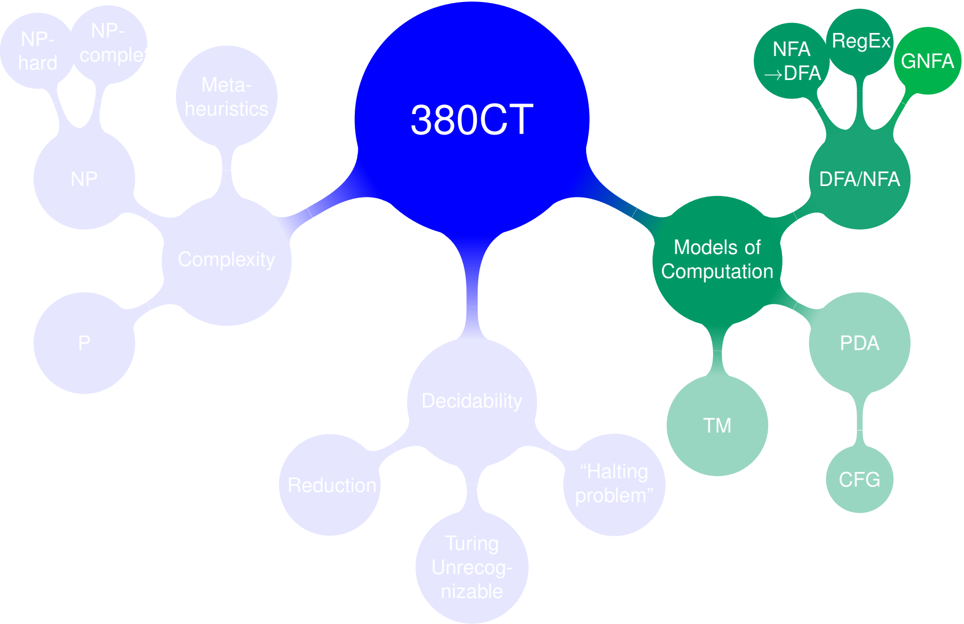
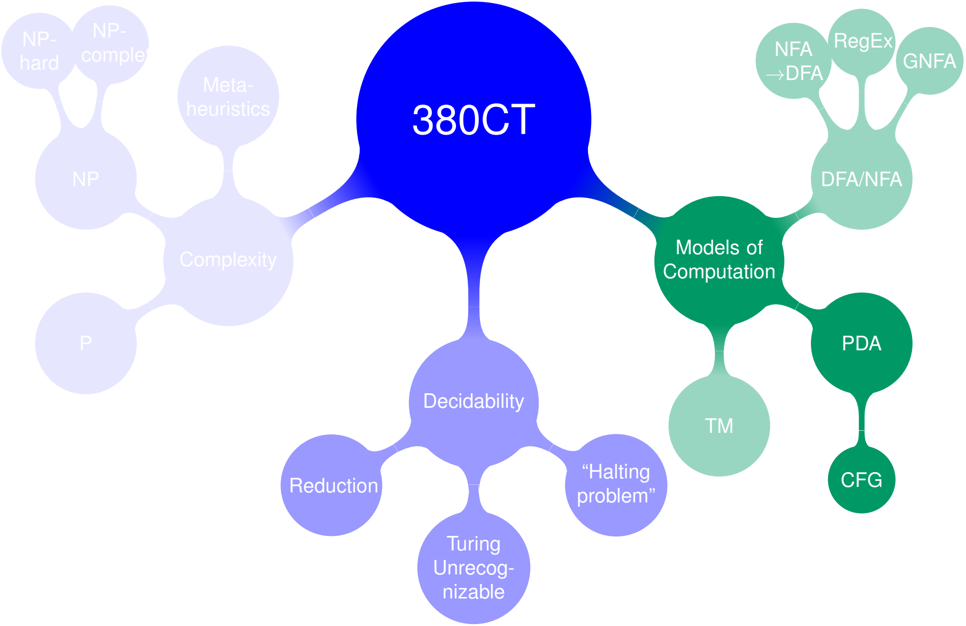
Decidability

Dr Kamal Bentahar

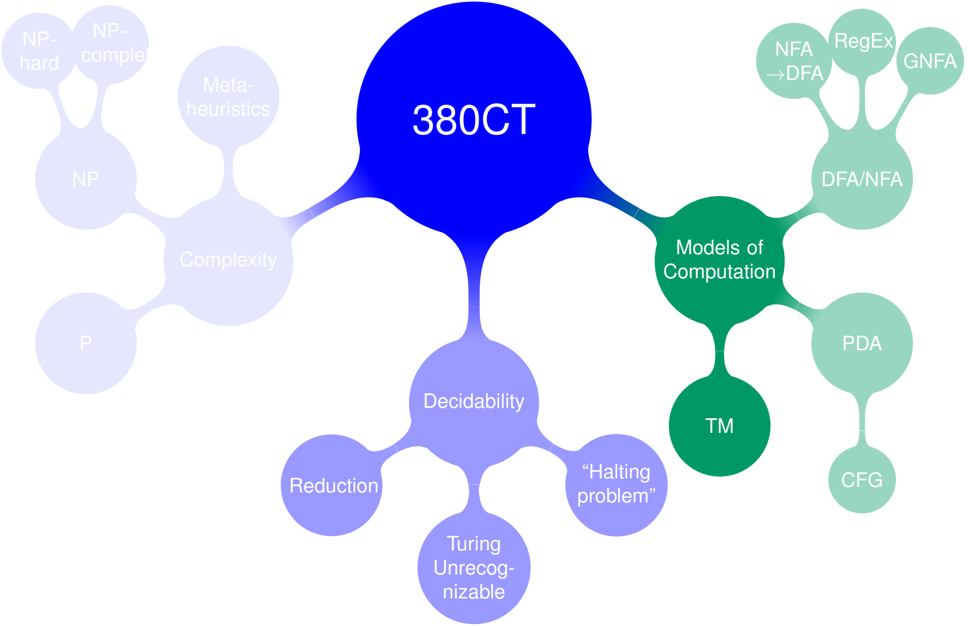
School of Computing, Electronics and Mathematics
Coventry University

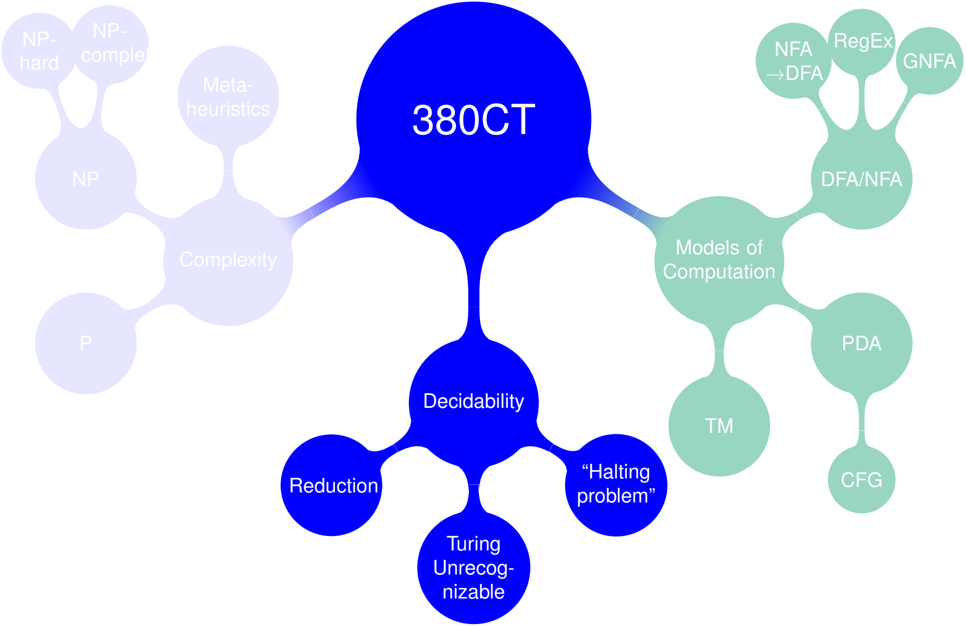
Week 8 – 12/03/2019


380CT
Models of Computation
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM
Decidability
Reduction
Turing Unrecog- nizable
“Halting problem”
Complexity
NP
NP- hard
NP- complet
Meta- heuristics
P



380CT
Complexity
NP
NP-hard
NP-complete
P
Meta-heuristics
Models of Computation
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM
Decidability
Reduction
Turing Unrecog-nizable
“Halting problem”


380CT
Complexity
NP
NP-hard
NP-complet
Meta-heuristics
P
Models of Computation
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM
Decidability
Reduction
Turing Unrecog-nizable
"Halting problem"


380CT
Complexity
NP
NP-
hard
NP-
complet
Meta-
heuristics
P
Models of
Computation
DFA/NFA
NFA
→DFA
RegEx
GNFA
PDA
CFG
TM
Decidability
Reduction
Turing
Unrecog-
nizable
“Halting
problem”

Last time...

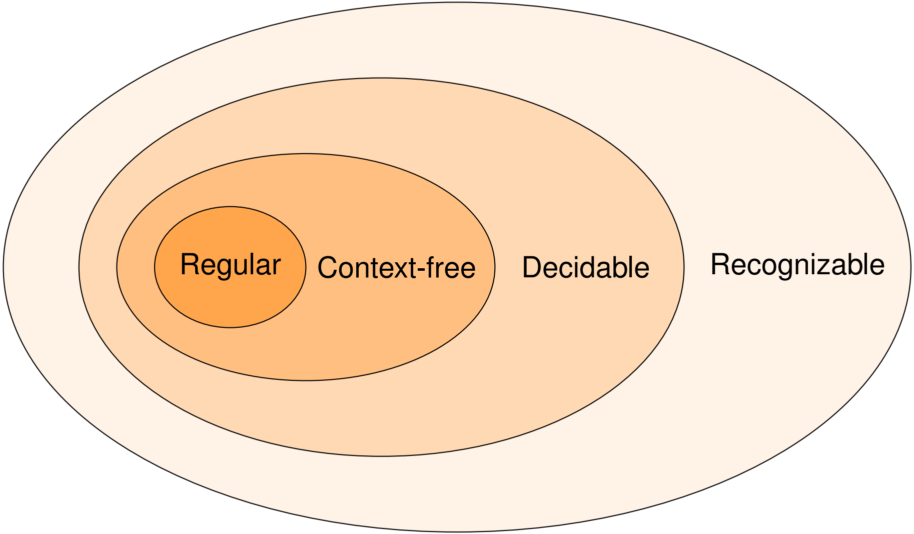
Turing Machines (TM) languages

- A language is **recognizable** if some TM **recognizes** it.
- A language is **decidable** if some TM **decides** it.
  (All branches of a NTM need to reject for it to reject a string.)

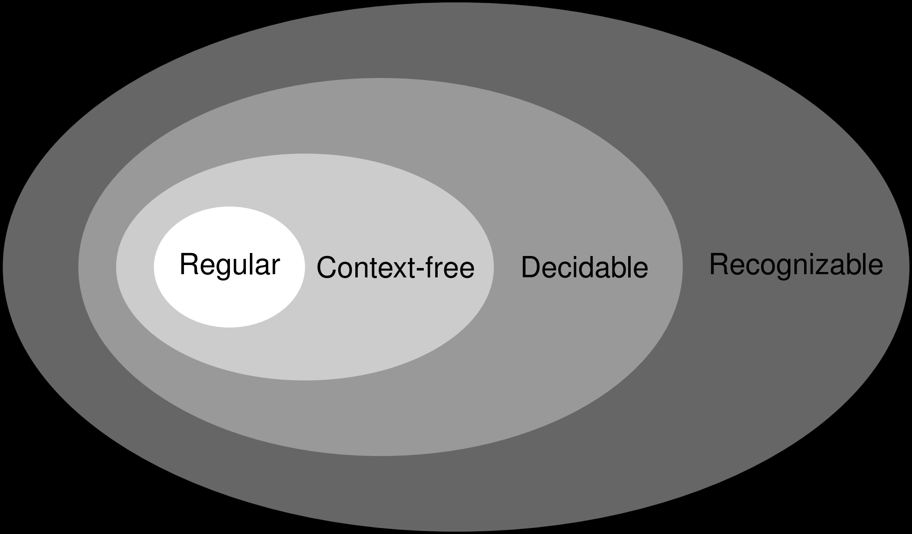
The Church-Turing Thesis – Algorithms

Intuitive concept of algorithms = Turing machine algorithms

Venn diagram



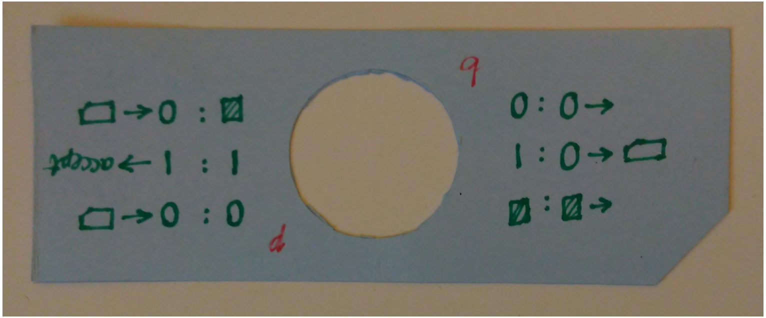
The "computation universe" discovered so far. . .



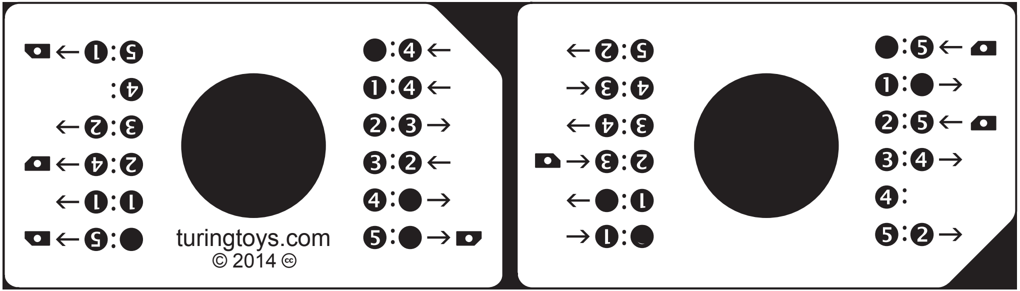
Business card TMs

TM with $q_{\text{start}} = q$ and δ:

State	Tape symbol	Transition
q	0	$(q, 0, R)$
q	1	$(p, 0, R)$
q	$\square$	$(q, \square, R)$
p	0	$(q, 0, L)$
p	1	$(q_{\text{accept}}, 1, R)$
p	$\square$	$(q, 0, L)$



Universal Turing Machines (UTM)



This TM is... **universal** with only 4 states and 6 symbols!
What does this mean?

There are TMs that can simulate any other TM!

$\rightarrow$ a **computer** is a simple idea – it is programming it that is hard. . .

Universal Turing Machines (UTM)

Example

U to simulate T:

- Place a description of T's transition function on the tape of U.
- Design U to read the description of T off the tape and do what T would have done to the tape.
- This is a systematic process, so it has to be possible.
- The part of the tape of U to the right of the description of T serves as T's tape.

In modern terminology, both the **program** and the **data** are stored in the memory of the UTM.
UTMs are what we now call **stored-program computers**.

Notation: encoding of an object

We need to **encode** objects so that (U)TMs can operate on them.

We denote the encoding of *object* by writing it between **angled brackets**:

$$\langle \mathit{object} \rangle$$

Decidable problems – examples

- Problems about regular languages
  - **Acceptance**
    - $A_{\text{DFA}} = \{\langle D, w\rangle \mid D \text{ is a DFA that accepts the input string } w\}$
    - $A_{\text{NFA}} = \{\langle N, w\rangle \mid N \text{ is an NFA that accepts the input string } w\}$
    - $A_{\text{REX}} = \{\langle R, w\rangle \mid R \text{ is a RegEx that generates the string } w\}$
  - **Emptiness**
    - $E_{\text{DFA}} = \{\langle D\rangle \mid D \text{ is a DFA and } L(D) = \emptyset\}$
    - $EQ_{\text{DFA}} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$
- Problems about context-free languages
  - **Acceptance**
    - $A_{\text{CFG}} = \{\langle G, w\rangle \mid G \text{ is a CFG that generates the string } w\}$
  - **Emptiness**
    - $E_{\text{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}$

Undecidability

Is $EQ_{\mathrm{CFG}} = \{\langle G, H\rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$ decidable?

No!

- Computers seem so powerful – can they solve all problems?
- Theorem: Computers are limited in a fundamental way.
- One type of unsolvable problem: given a computer program and a precise specification of what that program is supposed to do, verify that the program performs as specified.
- $\rightarrow$ **Software verification** is, in general, not solvable by computer!

Liar paradox

“I am lying.”

If the liar is lying, then the liar is telling the truth, which means the liar is lying.

“This sentence is a false.”

If it is true, then the sentence is false.
But if the sentence states that it is false, and it is false, then it must be true.
But . . .

Russel Paradox

Does “the list of all lists that do not contain themselves” contain itself?

If it does then it does not belong to itself and should be removed.
But, if it does not list itself, then it should be added to itself.
But, . . .

Undecidability – the **Acceptance Problem**

Consider

$$A_{\mathrm{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

Suppose that a decider D exists such that

$$D(\langle M\rangle) = \begin{cases} \textbf{reject} & \text{if } M \textbf{ accepts } \langle M\rangle \\ \textbf{accept} & \text{if } M \textbf{ does not accept } \langle M\rangle \end{cases}$$

Now run it on itself:

$$D(\langle D\rangle) = \begin{cases} \textbf{reject} & \text{if } D \textbf{ accepts } \langle D\rangle \\ \textbf{accept} & \text{if } D \textbf{ does not accept } \langle D\rangle \end{cases}$$

Does it accept or reject?

It rejects if it accepts, and it accepts if it doesn't accept!!

There is a problem with the assumption that such a D exists.

The acceptance problem A_{TM} therefore cannot be a decidable problem.

Undecidability – the **Halting Problem**

The Halting Problem

The Halting Problem is

$$HALT_{\mathrm{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \textit{ halts} \text{ on input } w\}$$

$HALT_{\mathrm{TM}}$ is also undecidable.
Proof uses "**reduction**" to A_{TM}.

The **reducibility** concept

How do we show that $HALT_{\mathrm{TM}}$ is undecidable?

We have: $HALT_{\mathrm{TM}}$ is decidable $\Longrightarrow$ A_{TM} is decidable.

Idea: use $HALT_{\mathrm{TM}}$'s decider to decide A_{TM} $\rightarrow$ **reduce** A_{TM} to $HALT_{\mathrm{TM}}$.

Proof

- Suppose there exists a TM H that decides $HALT_{\mathrm{TM}}$.
- Construct TM D to decide A_{TM} as follows:

  $D =$ "On input $\langle M, w \rangle$:
  1. Run TM H on input $\langle M, w \rangle$.
  2. If H rejects, *reject*.
  3. If H accepts, simulate M on w until it halts.
  4. If M has accepted, *accept*; if M has rejected, *reject*."
- If H decides $HALT_{\mathrm{TM}}$, then D decides A_{TM}.
- Since A_{TM} is undecidable, $HALT_{\mathrm{TM}}$ must also be undecidable.

More undecidable problems – there are lots of them!

Using reducibility we can show that the following problems are all undecidable

1. $E_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$ Reduce A_{TM} to it.
2. $REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is regular}\}$ Reduce A_{TM} to it.
3. $EQ_{TM} = \{\langle M, M' \rangle \mid M, M' \text{ are TMs and } L(M) = L(M')\}$ Reduce E_{TM} to it.
4. Post Correspondence Problem (PCP). Reduce A_{TM} to it – see lab.

Summary

(red: undecidable, blue: decidable)

1. Acceptance problems
   1. $A_{DFA} = \{\langle B, w\rangle \mid B$ is a DFA that accepts input string $w\}$
   2. $A_{NFA} = \{\langle B, w\rangle \mid B$ is an NFA that accepts input string $w\}$
   3. $A_{CFG} = \{\langle G, w\rangle \mid G$ is a CFG that generates string $w\}$
   4. $A_{TM} = \{\langle M, w\rangle \mid M$ is a TM and M accepts $w\}$
2. Language emptiness problems
   1. $E_{DFA} = \{\langle A\rangle \mid A$ is a DFA and $L(A) = \emptyset\}$
   2. $E_{CFG} = \{\langle G\rangle \mid G$ is a CFG and $L(G) = \emptyset\}$
   3. $E_{TM} = \{\langle M\rangle \mid M$ is a TM and $L(M) = \emptyset\}$
3. Language equality problems
   1. $EQ_{DFA} = \{\langle A, B\rangle \mid A$ and B are DFAs and $L(A) = L(B)\}$
   2. $EQ_{CFG} = \{\langle G, H\rangle \mid G$ and H are CFGs and $L(G) = L(H)\}$
   3. $EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1$ and M_2 are TMs and $L(M_1) = L(M_2)\}$
4. Miscellenious
   1. $HALT_{TM} = \{\langle M, w\rangle \mid M$ is a TM and M **halts** on input $w\}$
   2. $REGULAR_{TM} = \{\langle M\rangle \mid M$ is a TM and $L(M)$ is a regular language$\}$
   3. Post Correspondence Problem (PCP).

Unrecognizable languages

Theorem

L is **decidable** $\iff$ **both** L and $\overline{L}$ are **recognizable**

- Take $L = A_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$
- We know L is recognizable.
- If $\overline{L} = \overline{A}_{\text{TM}}$ were also recognizable then A_{TM} would be decidable.
- But we know A_{TM} is not decidable!
- So $\overline{A}_{\text{TM}}$ cannot be recognizable.

Corollary

$\overline{A}_{\text{TM}}$ is **not recognizable.**

Next week: Time Complexity

- Being decidable means that an algorithm exists to decide the problem.
- However, the algorithm may still be *practically* ineffective because of its **time** and/or **space** cost.