Introduction to Logic

Jake Chandler

The Language of Sentential Logic

Last week

- The notion of deductive validity (and soundness)
- The idea that arguments can be valid just because of the form of the sentences
- Two types of form: sentential and subsentential
- Sentential form: the way in which complex sentences are built up from atomic ones using connectives

A language of sentential forms

Building blocks

- The sentential form of an argument is represented using special notation (the "language of sentential logic" aka $\mathscr{L}_S$).
- This notation is widely used in philosophy (also comp. sci.)
- Its sentences (aka "formulae") are built from:
  - a set of lower-case letters, called 'atomic sentences': $p, q, r, \ldots$
  - a set of connective symbols: $\&, \vee, \supset, \equiv, \sim$
  - a pair of bracket symbols '(', and ')'.

Metalanguage

- We will use upper case letters $P, Q, R, \ldots$ to represent arbitrary sentences in the language
- Also commonly used: lower case greek letters $\phi, \psi, \chi, \ldots$
- These upper case letters are *not themselves* part of the language!!
- They are part of a metalanguage: a language to talk about the language

Symbol translations

English	$\mathscr{L}_S$	common variants
…and…	&	$\wedge$
…or…	$\vee$	
if…, then…	$\supset$	$\rightarrow$
…if and only if…	$\equiv$	$\leftrightarrow$
it is not the case that…	$\sim$	$\neg$

Syntax

- Every natural language has syntax or set of grammatical rules.

Some ungrammaticalities
"He went to the bar never came back and"
" If, then I guess that's the end"
"Not a bad place to hang out it is."
"Bernard told."

- Our sentential formal language is no different

Syntax

- A well-formed sentence (wfs) is a sentence whose construction obeys the rules of syntax.
- Rulebook:
  (1) Any atomic sentence is a wfs.
  (2) If P is a wfs, $\sim P$ is a wfs.
  (3) If P and Q are wfs's, so are $(P \& Q)$, $(P \vee Q)$, $(P \supset Q)$ and $(P \equiv Q)$.
  (4) Nothing else is a wfs.
- Note: it is common to omit the outmost brackets of a wfs, so that we write, for e.g., '$(p \& q) \vee \sim r$' rather than '$((p \& q) \vee \sim r)$'.

Wfs: some examples

A wfs

$((p\&q) \supset r)$

Since p, q and r are atomic sentences, they are wfs's (by (1))

Since p and q are wfs's so too is $(p\&q)$ (by (3))

Since $(p\&q)$ and r are wfs's, so too is $((p\&q) \supset r)$ (by (3))

Not wfs's

$(p\&q \supset r)$

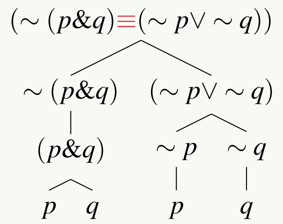
pq

$(\supset r)$

Parse trees again

- Again, parse trees can provide a nice representation of the stages involved in constructing a wfs from a set of atomic sentences.

Parse tree



- Parse trees mirror the information contained by the brackets (Can you see why?)

Introduction to Logic

Jake Chandler

The Language of Sentential Logic

La Trobe University

Formalisation

From English to Sentential Logic

- How do we ‘translate’ English sentences into sentential logic (formalise them)?
  (1) If required, rewrite the sentences in a format that “brings out” the sentential form
  (2) Figure out what the parse tree would look like
  (3) Set up a dictionary, establishing a one-to-one correspondence between the English atomic sentences and various letters
  (4) Replace each atomic sentence of the argument by its corresponding letter
  (5) For each sentence, gradually replace each English connective by its symbolic counterpart, working up from the bottom of the tree (adding brackets around each new representation of a complex subsentence)

Single sentence example

Single sentence translation

“If you go left, then if I go right, then we’ll collide. ”

(1) No need to rewrite here

(2) Figure out the parse tree

If you go left, then if I go right, then we’ll collide.

You go left   if I go right, then we’ll collide

I go right   We’ll collide

(3) Make dictionary:

p: You go left; q: I go right; r: We’ll collide

Single sentence example (ctd)

Single sentence translation (ctd)

(4) Replace atomic sentences:

If p, then if q, then r.

p If q, then r

q r

(4) Replace connectives:

$(p \supset (q \supset r))$ ($\Leftarrow$ translation)

p $(q \supset r)$

q r

Most descriptive sentential forms

- Replacing different occurrences of the same sentence by different letters also yields a form
- But the above method yields the most descriptive sentential form (because it also describes the possible fact that two atomic sentences are identical)

Form vs most descriptive sentential form

“Either you went to the cinema or you didn’t.”

A form: $p \vee \sim q$

The most descriptive form: $p \vee \sim p$

Argument example 1: *Modus Ponens*

Intergalactic Renegades

(1) If Jake teaches logic, then he's an Intergalactic Renegade
(2) Jake teaches logic

---

(3) Jake is an Intergalactic Renegade

Most descriptive sentential form: Modus Ponens

(1) $p \supset q$
(2) p

---

(3) q

Argument example 2: *Modus Tollens*

Still out and about

(1) If he was back, then the lights would be on.
(2) It is not the case that the lights are on.

---

(3) It isn't the case that he is back.

Most descriptive sentential form: Modus Tollens

(1) $p \supset q$
(2) $\sim q$

---

(3) $\sim p$

Argument example 3: *Constructive Dilemma*

Nightclub

(1) Either it'll be crowded or it'll be empty.
(2) If it's crowded, then it'll be too hot.
(3) If it's empty, it'll be boring.

---

(4) It'll either be boring or too hot.

Most descriptive sentential form: Constructive Dilemma

(1) $p \vee q$
(2) $p \supset r$
(3) $q \supset s$

---

(4) $r \vee s$

Argument example 4: *Disjunctive Syllogism*

Cluedo

(1) Either Professor Plum did it, or Colonel Mustard did.
(2) It isn't the case that Professor Plum did it.

---

(3) Colonel Mustard did it.

Most descriptive sentential form: Disjunctive Syllogism

(1) $p \vee q$
(2) $\sim p$

---

(3) q

Argument example 5: *Hypothetical Syllogism*

Up Late

(1) If you stay up late, then you'll be tired tomorrow.
(2) If you are tired tomorrow, then you'll be late for class.

---

(3) If you stay up late, then you'll be late for class.

Most descriptive sentential form: Hypothetical Syllogism

(1) $p \supset q$
(2) $q \supset r$

---

(3) $p \supset r$

Formal validity

- *Any* argument that instantiates one of these five sentential forms is valid.
- This is also true of a number of other common sentential forms.
- We say of such forms that they are valid sentential forms (antonym: *in*valid sentential forms).
- In other words: a form is valid if and only if every argument that instantiates it is valid.
- An argument that instantiates a valid form is formally valid.
- Note: a formally valid argument can additionally instantiate invalid forms (e.g. a disjunctive syllogism: $p \vee q, \sim r \therefore q$)

Introduction to Logic

Jake Chandler

The Language of Sentential Logic

Decidability and Complexity

Spotting a valid sentential form

- Recognising the instantiation of a valid sentential form is obviously a very useful shortcut when evaluating an argument.
- But how can we know when a sentential argument form is valid in the first place?
- This is equivalent to an instance of the sentential (aka “Boolean”) satisfiability (SAT) problem:

    Determining whether a sentence in $\mathscr{L}_S$ is satisfiable (i.e. whether there could exist a true sentence in English that has that form)

- The sentence of interest here:

    The conjunction of the premises and the negation of the conclusion

    (If this isn’t satisfiable, the sentential form is valid)

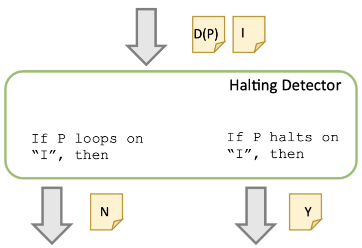
- SAT is important: many key problems can be reduced to it

Decidability

- A procedure takes a specification of a problem as an input and either provides an answer as an output (‘halts’) or carries on indefinitely (‘loops’)
- A question is decidable if and only if there is a procedure that will output the correct answer to it in a finite number of steps (aka “time”)
- Validity for a language is decidable if and only if the validity of every sentence is decidable
- Not every problem is decidable!

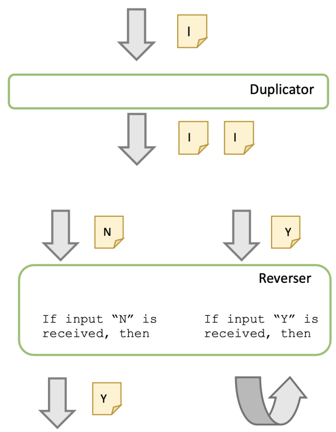
Turing’s Halting Problem

- Problem: Will an arbitrary procedure halt when given an arbitrary input?
- Assume for sake of argument that this general question *is* decidable
- A “Halting Detector” takes a description of a procedure and an input and says whether the procedure would halt with that input



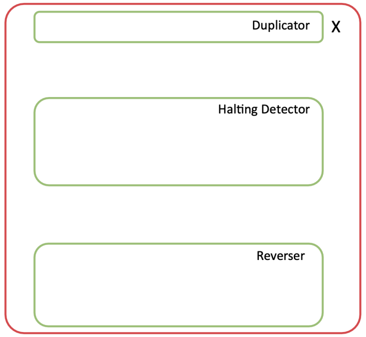
Turing’s Halting Problem (ctd.)

- Now consider two further procedures:



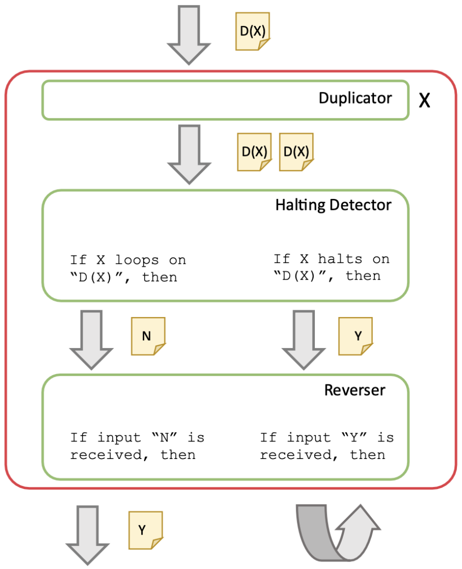
Turing's Halting Problem (ctd.)

– We chain all three together to make the procedure X



– Now we feed X with its own description as an input…

Turing's Halting Problem (ctd.)



Assume H outputs "N", then, because of R, X outputs "Y", contradicting H.

Assume H outputs "Y", then, because of R, X loops, contradicting H.

Complexity

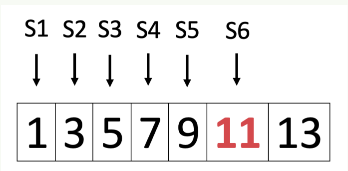
– Some decidable questions are more difficult to decide than others

– Time complexity of a question: worst case number of steps as a function of input size required by most efficient procedure

Efficiency

Task: Find a specific number in an ordered list of n numbers.

Linear search:

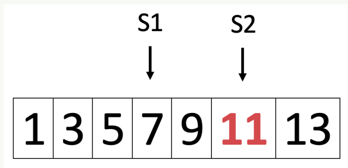


Worst case scenario: n (linear time: input $\times 2 \rightarrow$ time $\times 2$)

Complexity (ctd)

Efficiency (ctd)

Binary search:

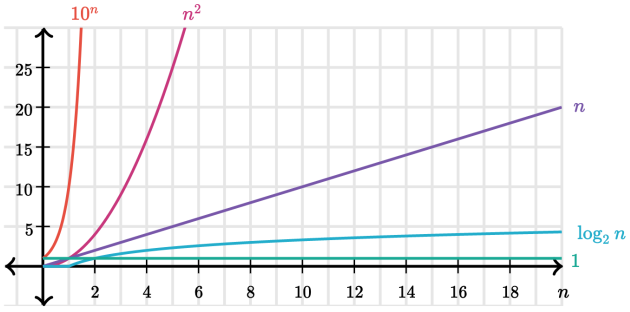


Worst case scenario: $\log_2 n$ (log. time: input $\times 2 \rightarrow$ time $+1$)

log time < linear time

Tractability

- Towards the top end of complexity: exponential time (e.g. 2^n)



- A problem is intractable iff it has exponential time complexity
- Question: Is SAT decidable and if so, what is its complexity?