Language
to logic mapper
to logic model checker

Colin James III
info@cec-services.com

ESMC
Colorado Springs

What is automated reasoning.

- This goal is to obtain logical proof of sentences.

- The motivation avoids costly mistakes sooner for efforts based on requirement documents.

- The approach is two-fold:
  - 1. Input from the language to logic mapper (LLM)
  - 2. Output from the logic model checker (LMC)

What is logic model checker (LMC).

- Jan Łucasiewicz invented logic Ł4 but did *not*:
  - Map modal logic for necessity and possibility; or
  - Find acceptance academically.
- Garry Goodwin (UK) and I (USA) fixed Ł4:
  - Variant Ł4 (V Ł4) uses 5 logical models; and
  - A 4-valued logic system of {F,N,C,T} and {U,I,P,E}.
- The Meth8 logic model checker implements it.
  - LMC is a *recent advance* in mathematical logic.

Variant Ł4 (VŁ4)

- Uses two sets of 4-valued logic for 5 models
- Output of proof result is a truth table such as:

CTCT	UEUE0	**EEEE**	PEPE	IEIE
Model 1	Model 2.1	**Model 2.2**	Model 2.3.1	Model 2.3.2

  - Proves all models in agreement as **T**rue, **E**valuated
- □(□ p →p) → □p: The Gödel-Löb axiom (GL)
  - It is suspicious with only one valid model of five.
  - If GL fails, then so also does the Zermelo-Fraenkel set theory (ZF) with axiom of choice (ZFC) as a foundation.

What GL wished it was in words.

- $\square(\square p \rightarrow p) \rightarrow \square p$ (GL) [1]

  The necessity of choice, as always implying a choice, implies *the necessity of choice*. (Circular nonsense is no cigar.)

- $\square(\square p \rightarrow p) \rightarrow \square(p \vee \sim p)$ [2]
  - The necessity of choice, as always implying a choice, implies *the necessity of a choice or not a choice*.

- $\square(\square p \rightarrow p) \leftrightarrow \square(p \vee \sim p)$ [3]
  - The necessity of choice, as always implying a choice, is equivalent to *the necessity of a choice or not a choice*.

TTTT	EEEE	EEEE	EEEE	EEEE
Model 1	**Model 2.1**	**Model 2.2**	**Model 2.3.1**	**Model 2.3.2**

  - VŁ4 validates all models in agreement as **T**rue, **E**valuated

What is language to logic mapper.

- What is natural language?
  - Simple logic dictates a simplistic approach:
  - whatever fits in sentences is parts of speech (POS).
- POS are abstract groups of:
  - Noun, Verb, Modifier (NVM).
    - The modifier is an adjective and adverb.
- POS are not grammatically significant here.
  - A subject, object, or direct object is still a “Noun”.

How to find POS by look up table

- Public domain POS list for 180K English words
  - Sub-types of POS are further grouped for use here.
    - Nouns: singular, plural, pronoun, noun clause
    - Verbs: (in)transitive, participle, gerund, conjunction
    - Modifier adjective: preposition, (in)definite article
    - Ignored: nominative, interjection
- Sequential access of the (un)sorted word list:
  - Searches on average ½ of the list; and
  - Avoids overhead of sorting before a binary search.

Some POS are ambiguous.

- A word may be several POS.
  - These words are noun, verb, adjective, and adverb:
    - free, firm, like, prompt, smooth
- The correct POS is from the context (intent).
  - Intent derives from a pattern search for the usage.
  - Example 1: *Tango* is leaders-followers.
    - Pattern is *Tango* – verb - nouns, hence **N**vn.
  - Example 2: Leaders-followers *tango*.
    - Pattern is nouns - *tango*, hence n**V**.

How to map POS to logical symbols.

- Nouns are literals for
  - Propositions as lower case {p,q,r,s, …}
  - Theorems as upper case {A,B,C,D, …}
- Verbs are connectives assigned {&+-<=>@\} for
  - { And, Or, Nor, Imply, Xor, Nand}
- Modifiers are operators assigned {~#%} for
  - Adjectives as {not, necessary, possible}
  - Adverbs as {never, necessarily, possibly}

How to prove a sentence.

- A sentence is a proof table of logical values, eg:
  - “A floor of a factory contains robots and computers.”
  - “A floor [and necessarily] of a factory [is] robots ...”
  - (p & #q) = (r & s).

Model 1	Model 2.1	Model 2.2	Model 2.3.1	Model 2.3.2
TTTC TTCC	EEEU EEEU	EEEE EEEE	EEEP EEEP	EEEI EEEI
TTTC FFFN	EEEU UUUE	EEEE UUUU	EEEP UUII	EEEI UUUP

- Proof is `TTTT` in Model 1 and `EEEE` in others:
  - The example sentence is not proved as true.

How to fix a sentence to prove it.

- The Technique of Unfolding
  - Makes the sentence itself more descriptive
  - Divides a sentence into simpler descriptive parts
- The Technique of Contraction
  - Abstracts a sentence into general, generic content
  - Builds sentences with high informational content
- Seek assistance from:
  - Business proposal, grant, and technical writing

How to map sentences together.

- S1 is nouns A, B, verb “is” =, so: S1=(A=B).
- S2 is nouns A, B, verb “and” &, so: S2=(A&B).
  - P1 is S1 then S2, so: P1=S1>S2=(A=B)>(A&B). The implication connective “>” is arbitrarily inserted between S1 and S2 based solely on the reason that S2 follows S1: one follows another.
  - In other words, “If S1, then S2.”
  - But: if S1 is True, then S2 as False cannot follow.
- Subsequent Pn form a requirements document.

Summary

- Automated reasoning is achieved by mapping language to logic and by checking logic models.
  - Words in sentences are looked up for the POS.
  - POS map to symbols for logical expressions.
- Sentences are proved by five logical models.
- Serial sentences imply proof of paragraphs.
- Serial paragraphs validate requirements docs.

Colin James III info@cec-services.com