

我是个发烧友，但不是疯子，因为我对于如何建构飞机有些心得。我希望分享我所知的一切，然后，如可能的话，出一分力去帮助那将来会达到最终成功的人。

— Wilbur Wright

# Genifer 3.0 白皮书

YKY (甄景贤)

May 18, 2015

## Abstract

介绍通用人工智能 Genifer 3 的理论。  
写得很仓促，如看不懂请提问。

## 1 背景

Genifer 3 是「新古典主义」的、基于逻辑的人工智能，和 OpenCog、NARS 等相似。在 Genifer 4 我企图脱离经典逻辑，转移到向量空间中使用连续逼近法，但其具体细节现在仍未落实。

**命题逻辑**关心的是有真假值的命题，原子命题没有内部结构。例如，可以有这个逻辑式子： $P \wedge Q$ 。命题逻辑是 isomorphic to Boolean algebra。在命题逻辑中做逻辑推导，是著名的 NP-complete 问题，叫「可判定性」(satisfiability, SAT)。**Resolution**（消解原理）是命题逻辑的推导算法。

**谓词逻辑**赋予命题的内部结构，例如：爱(小明, 小娟)。谓词逻辑容许有一些 patterns 例如 爱(X, Y)，其中 X, Y 是变量。这些模式要和实际句子做配对(pattern-matching)，这配对的算法叫归一化(unification)。例如，以下这式子定义「爷爷」的概念：

爷爷(X, Z)  $\leftarrow$  爸爸(X, Y)  $\wedge$  爸爸(Y, Z)。

谓词逻辑的推导算法，需要结合 **unification** 和 **resolution**。结合后，算法的复杂性在最差情况下是可以没有终止的 (non-terminating)：因为一阶逻辑是 **Turing universal**，而「停机定理」(halting problem) 证明了这样的逻辑推导不存在必然会回覆的算法。

Genifer 3 与经典逻辑有以下区别：

- 用 **fuzzy-probabilistic** (机率/模糊) 的真假值
- 不使用变量 (variables)

**Fuzzy-probabilistic** 推导法的原理很简单，属例行公事，但因为要服从机率的法则，需要用到 **Bayesian belief propagation** (机率传播算法)。

不使用变量的原因是因为变量的思维方式很不自然。例如人类认识「爷爷」这概念，是「爸爸的爸爸」(在关系代数中记作 **爸爸  $\circ$  爸爸**)，而不是像 **X 是 Y 的爸爸** 这样使用变量。而且，在推导过程中，变量令我们需要做颇复杂的 **substitution management**，这在经典的 *“Structure and Interpretation of Computer Programs”* 中有详细描述，但我们可以不理。例如，图中所示，谓词逻辑的式子中，变量之间的连结 (binding) 关系可以很复杂：

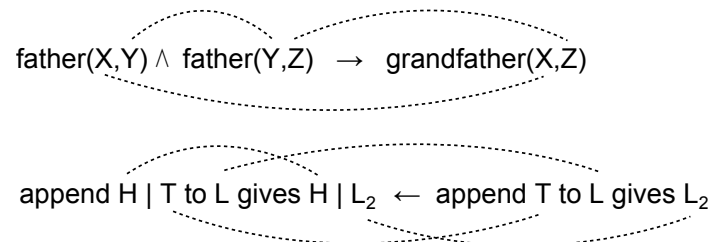


Figure 1: 谓词逻辑中变量之间的连结：「爷爷」的定义，和 PROLOG 中将一个序列 **append** 到另一个序列的定义。

**组合逻辑** (combinatory logic) 是一种没有变量的逻辑；而 **关系代数** (relation algebra) 也是一种组合逻辑，但局限在关系的运算上。Genifer 3 使用的是一种简化的关系代数。

AI 的瓶颈在於其 **学习算法**，在下文会讨论。

## 2 逻辑

Genifer 3 的逻辑式子很简单，就是一串原子概念 (atomic concepts) 的乘积：

$$c_1 \cdot c_2 \cdot \dots \cdot c_n.$$

例如：小明 · 爱 · 小娟。

由於不使用变量，Genifer 3 需要利用包含关系  $\subseteq$ ，例如：

猫  $\subseteq$  动物。

这关系容许逻辑上的一般化 (generalization)，例如，从「人是会死的」推导出「苏格拉底会死」。

一个逻辑式子，可以是事实 (fact) 或规则 (rule)；规则的特点是具有前件、后件的结构：

pre-condition  $\rightarrow$  post-condition。

作为例子，考虑如何从这些已知事实：

小明 · 爸爸 = 小强      (小明的爸爸是小强)

小强 · 爸爸 = 大强

推导出这个新的事实：

小明 · 爷爷 = 大强。

这个问题可以用规则 爸爸 · 爸爸 = 爷爷，再用代入 (substitution) 的方式解决，但我暂时还不肯定可不可将这动作化简为更原始的操作？

## 3 动作 (actions)

不使用变量的逻辑可能会变得太弱，意思是失去 Turing universal。补救的办法是使用「记忆储存器」。

大家记得 Turing machine 包含一条读写磁带和有限个状态的转移规则：

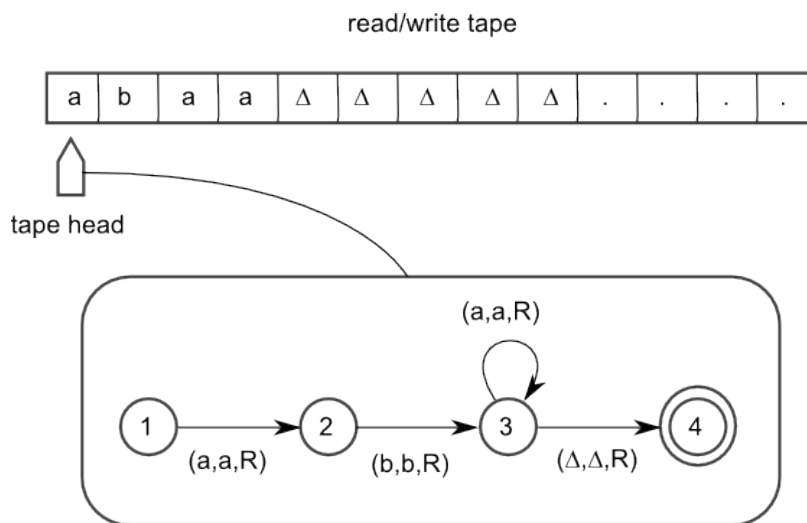


Figure 2: 接受  $aba^*$  的图灵机。

如果我们在逻辑上加上某种记忆体，就自然地可以重获 Turing universal。

具体地说：

下雨了  $\rightarrow$  草地是湿的

是一句经典逻辑句子，但我们可以引入一种带有**动作**的逻辑句子：

如果储存器 **A** 的值是 **0**  $\rightarrow$  将 **1** 写入储存器 **B**。

甚至我们不使用个别的储存器，而是用一些资料结构，例如 linked list 或 tree。

在 Genifer 3.0 我尝试用 list。

## 4 学习

学习的目的是寻找一组逻辑式子去**解释**这世界。所谓解释即推导。

学习的方法是 inductive learning，即由事实**诱导**出法则 (induce rules from facts)。

用逻辑表达就是：

$$KB \cup H \models E$$

其中 KB 是**知识库** (knowledge base) 或背景知识，H 是我们想学习的新的**假设** (hypothesis)，E 是一些实例 (examples)，或者可以说是新的经验

(experiences), 学习的目的是用  $H + KB$  去解释这些新经验,  $\models$  表示逻辑蕴涵(entailment)。

Inductive learning 就是在所有可能的逻辑句子空间中搜寻, 这个空间异常地大。这空间如果有格 (lattice) 的序结构, 搜寻会快很多。那就是 **general-to-specific order**。传统逻辑中这个序可以由两种方法达成:

1. 某个概念比另一个概念更一般, 例如:

动物  $\supseteq$  狗

2. conjunctions 的增加, 例如:

戴眼镜  $\wedge$  长头发 比 长头发 更特殊。

在 Genifer 4 我企图用连续空间的梯度下降法, 但这条路线离成功还很远。Genifer 3 不需要连续空间, 那就只需用 genetic algorithm 去学习, 这样很容易。容易未必表示不好, 例如在机器学习比赛的实践中, 人们发现最有效率的分类算法是理论上很简单的决策树 (decision tree), 而不是理论很复杂的支持向量机 (SVM)。可能 Genifer 3 用进化 / 遗传算法已经足够?

## 致谢

首先必须感谢王培 [2] [3] 与 Ben Goertzel [1] 对通用人工智能 (AGI) 的重大贡献。此外, 我和 Abram Demski、Russell Wallace 花了很多年讨论各种关于逻辑的想法。还有 Matt Mahoney 和其他在 AGI 电邮论坛上的参与者。William Taysom、Seh、Joseph Cheung 帮手编写 Genifer 程式, 现开源在 Github 上。

## References

- [1] Goertzel, Pennachin, and Geisweiller. Building better minds: engineering beneficial general intelligence, 2011.
- [2] Wang. *Rigid Flexibility - The Logic of Intelligence*. Springer applied logic series, 2006.
- [3] Wang. *Non-axiomatic logic: a model for intelligent reasoning*. World Scientific (in press), 2013.