少儿图灵测试回顾

陆汝钤 1,2,3,4,5+, 张松懋 1,2
1(中国科学院 数学与系统科学研究院 数学研究所,北京 100080)
2(中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100080)
3(中国科学院 数学与系统科学研究院 管理、决策和信息系统重点实验室,北京 100080)
4(复旦大学 上海市智能信息处理重点实验室,上海 200433)
5(北京工业大学 北京市多媒体与智能软件重点实验室,北京 100022)

摘要: 本文对儿童图灵测验（CTT）进行了研究。我们的测试项目与其他测试项目的主要区别在于其基于知识的特点，这是由大量常识知识库支持的。本文介绍了 CTT 的动机、设计、技术、实验结果和平台（包括知识引擎和对话引擎）。最后，给出了一些关于 CTT 和 AI 的结论性思考。
关键词: 图灵测试; 会话系统; 常识知识库; 封闭世界假设

1 图灵测试和对话程序

TT（图灵测试）是一个公认的测试机器智能的标准，由图灵在 1950 年提出[1,2]。由于 TT 没有严格的定义，下面给出的定义符合作者的理解。两名受试者（也称为同盟者）A 和 B，其中一名是人（称为人同盟者），另一名是计算机（称为计算机同盟者），由法官 C 进行测试。C 看不到 A 和 B，也不知道其中谁是人或计算机。C 可以和他们中的任何人交谈，并且必须决定他们的身份。如果 C 不能做出正确的决定，那么就断言计算机联盟已经通过 TT（表明它有智能）。

在图灵时代，机器智能和人类智能还没有被广泛接受的定义。图灵没有试图给出这样的定义。相反，他试图设计一个测试，根据他的想法，它在功能上等同于给出一个定义（判断机器智能的标准），但更容易实现和检查。这个由图灵定义的测试，如上所述，被称为图灵测试。显然，他的定义本质上是实验性的和行为性的，这是后来争议的来源之一。

反对图灵论点的最著名论点是塞尔的所谓中国房间问题[3~5]。假设有一台计算机 C 通过了中文测试。每个不懂中文的人都可以用 C 与其他人进行对话。P 和 C 所在的房间被称为中文房。我们能说 P 懂中文是因为他可以用这种方式“用中文”与环境互动吗？

尽管如此，图灵的思想已经引发了许多关于人机对话的研究工作。因此，许多自由对话节目被制作出来。这些程序不同于结构化对话程序。在一个结构化的对话程序中，句型是预先定义好的，然后由对话伙伴跟随。有限

状态机用于解析和处理语音，并产生输出。这种对话模式在功能上非常有限，不符合图灵测试的要求。

然而，自由对话程序允许用户以自由的方式与计算机对话。没有预先定义句子的格式或模式。乍一看，这种程序似乎很难开发。但是，如果一个人把自己局限在一个非常谦逊的话语领域，那么有一些技巧可以用来让对话程序以一种聪明的方式运行。

最著名的早期例子之一是麻省理工学院约瑟夫·魏森鲍姆[6]写的《伊丽莎》。ELIZA 与人类伴侣进行简单而流畅的沟通。它使用一系列关键字、一系列语法模式和另一系列转换规则来对用户输入做出反应。

第二个例子是科尔比[7]开发的 PARRY 程序。与伊丽莎类似，帕里也没有使用任何语法分析器。它拥有大约 6000 条模式匹配规则。它让自己看起来像退伍军人医院的精神病人。帕里在 1973 年被放到了 Arpanet 上，当时已经成功地欺骗了很多人。

Ultra HAL 是此类对话程序的另一个例子[8]，它是 HAL 的新版本。它聚集在一起 1000 个所谓的常用短语（对话中最常用的句子形式）及其回应；其中一个常用短语可能有几个相应的回复。如果找不到合适的公共短语，则使用基于关键字的模式匹配机制。在这种情况下，它会从数据库中寻找合适的答案，或者改写问题，将其转换为答案。

自 1991 年以来，每年都会举办一项名为 Loebner 测试的国际竞赛，以鼓励实施图灵测试[9]。尽管铜奖一直颁发给参加每年测试的最佳项目，但一枚 10 万美元的金牌仍然悬而未决。

这些作品的一个基本特点是缺乏知识的运用，尤其是常识性知识。这些系统都不是真正基于知识的。如果你问这样一个问题："一头牛有几条腿"，我们相信，这些系统都不能给出合理的答案。

2.知识型儿童的动机图灵测验

围绕 TT 意义的争论主要是哲学上的。我们研究的第一个动机不是证明或反驳 TT 作为测试机器智能标准的合理性。我们只是想探索进行此类测试和构建一个强大到足以让计算机赢得 TT 的平台的技术可行性。这是我们的主要动机。

由于很难使计算机具有足够的智能来通过其原始意义上的 TT，我们的第二个动机是探索削弱图灵提出的标准的可能性。更准确地说，我们正试图将 TT 的研究对象限制在某些特定的未成年人群体，即学龄前或初中年龄的儿童，比如 10 岁以下的儿童。这种限制有两层含义。首先，为了模

仿一个孩子，计算机联盟只需要拥有少量的知识和智力。其次，儿童年龄的法官智力较低，因此从 a 和 B 中判断谁是人，谁是计算机的可能性较小。

在我们的 TT 实验中，知识在人类智力中的作用是一个重要的研究课题。我们想研究如何以及在多大程度上利用知识使 TT 更成功。

提出这样一个问题可能很有意义：我们的电脑有多旧？如果一个人不能区分 n 岁的孩子和计算机，那么我们说这台计算机至少有 n 岁的智能。

最后，我们做这项工作的最后动机是研究与 TT 实施相关的主题，例如自然语言理解，自然语言生成，常识知识表示和处理、人类思维建模、对话和辩论策略等。

3 儿童图灵测试的第一个结果

对于测试来说可能有太多的变化。对于我们的第一个孩子 TT，我们将其规则细化如下：法官和儿童联盟成员均为 5 岁至 11 岁。使用该年龄段儿童的自然中文。对话仅以文本形式进行（没有语音识别或生成）。每节课分为几轮。每轮由双方各一句组成。有两种会话模式。在有限会话模式下，只允许进行 10 轮。在无限期开庭模式下，法官可以执行任意轮数，直到她认为自己收集了足够的信息做出决定。最后一条规则是不允许两个邦联之间进行信息交换。

第一次 CTT 于 2000 年初使用有限会话模式进行。已经进行了四次这样的测试。在前两个阶段中，三名 5 至 6 岁的儿童参加了测试。在每节课结束时，评委们被要求回答（A，B）中谁是孩子，谁是电脑。法官们犹豫了很长时间，然后正确地回答了问题，但信心不足。当调查人员问他们为什么认为 B 是个孩子时，他们无法解释原因。我们从这些测试中得到的主要印象是孩子们的年龄限制不当。法官太年轻了，无法理解 CTT 的含义。这些孩子只是对玩电脑感到好奇，他们没有思考如何打败电脑，以及如何检测 a 或 B 的实际容量。

在接下来的两个环节中，我们邀请了两个 7 到 11 岁的孩子。他们对 CTT 含义的理解显然要好得多。在测试结束时，他们还对 A 和 B 的角色做出了正确的猜测。见本文表 1。但这里我们看到了另一个问题。当我们问他们为什么得出这个结论时，他们说他们认为 A 是一台计算机，因为 A 给出的答案太完美了，不能被认为是真正的孩子给出的答案。相应地，在谈话过程中，他们试图用棘手的问题一次又一次地攻击他们认为是孩子的角色，以揭示孩子的真实身份。

Table 1 Result of CTT in H-M / teacher-child / unlimited session mode

Session	Role A	Role B	Number of rounds	Talk with A	Talk with B	Result
1	Child	Computer	4	2	2	Correct
2	Child	Computer	10	6	4	Correct
3	Computer	Child	8	4	4	Wrong
4	Child	Computer	11	3	8	Correct

后来，我们改变了儿童 CTT 的形式→从儿童到成人→从有限会话模式到无限会话模式。也就是说，法官不再是孩子，而是老师。这有一些好处。首先，（成人）法官比儿童更了解 CTT 的含义，并会提供更好的合作。其次，老师对孩子行为的理解更可靠。第三，在这种模式下，可以检查和测试更多基于知识的原则。结果见表 2。

Table 2 Result of CTT in {H-M, H-H, M-M} / teacher-child / unlimited session mode

Session	Role A	Role B	Number of rounds	Talk with A	Talk with B	Judge's decision
1	M	H	15	7	8	A=M, B=H
2	M	M	19	12	7	A=M, B=H*
3	H	M	35	17	18	A=H, B=M
4	H	M	21	0	21	A=H, B=M
5	H	M	30	0	30	A=H, B=M

*当前位置法官对第二次开庭的判决是错误的

在最后的 CTT 实验中，我们增加了 CTT 法官的难度，允许两个联盟的所有可能组合：儿童+计算机、两个儿童或两台计算机。因此，教师评委需要更多（大约三倍）的对话才能做出决定。见表 3。

Table 3 Comparison of round numbers in Tables 1 and 2

	H-M mode	{H-M, H-H, M-M} mode
Maximal number of rounds	11	35
Average number of rounds	8.25	24

4 CTT 平台：知识引擎

支持 CTT 的主要设施包括两部分：知识引擎和对话引擎。这两部分构成了 CTT 平台。知识引擎是第一个显著特征，它不同于我们基于知识的 CTT 程序和传统的“基于聊天”的对话程序。传统的对话程序无法回答“一头牛有几条腿？”这样的常识性问题聪明人程序可能会试图通过建议另一个对话主题来避免回答它。但如果你坚持要求一个具体的答案，这些程序将无法给出合理的答复。因此，无论程序是否有足够的知识，以及如何利用其知识，这是我们的 CTT 对话引擎与许多其他对话程序的本质区别。对于 CTT，我们可以假设谈话对象既没有专业知识，也没有高水平的科学知识。所以我们需要的只是常识。因此，我们对话引擎的知识引擎是在一个名为 Pangu 的常识知识库上运行的。

4.1 盘古知识库

常识知识库应尽可能模块化。盘古中两种基本的常识知识模块是agent和ontology,它们以关系的形式存储。因此,我们拥有的是一个agent本体关系知识库。Agent和Agent类以两种方式组织。在纵向上，它们形成继承层次结构。在水平方向上，它们形成了称为本体的语义网络的不同层,用于以连接主义的方式组织代理和代理类。代理之间的通信通过BQML实现，BQML是标准KQML语言的变体[13]。

4.2 CBS代理的结构和功能

基于文献综述，代理可分为六类：被动代理（OO范式中的一个对象）、反应代理（检测环境中的任何变化并以适当的方式作出反应）[14]、BDI代理（带有信念、欲望和意图的蓄意代理）[15]，社会代理（具有合作和竞争的代理社会）[16]、进化代理（可能通过学习提高其智能）[17]、个性化代理（具有情感和感觉等人类特征的代理）[18]。

分析结果表明，各种智能体对构建盘古常识知识库及其应用是有用的。另一方面，上述代理商都不能满足我们的所有需求。为了对常识知识进行适当的表示，我们设计了一种新的代理，CBS代理。

为了模拟人类专家的行为，我们的CBS代理有两种类型的知识，组织在代理结构的两个部分：信念部分和策略部分。它们代表了agent的静态知识（信念）和动态知识（策略）。信念部分模仿专家的知识记忆。它就像一个信息库。策略部分由一组规则组成。它模仿专家根据信念部分进行推理以解决问题的能力。为了提高搜索效率，每个CBS代理都有一个功能部分，可以将其视为该代理可能提供的服务的列表。每次要求代理人提供服务时，首先会检查此列表，以确定提供所需服务的可能性。代理的链接部分指定了该代理所涉及的本体的列表。主体和本体之间的关系是多对多的。例如，本体论“学校”包含“教师”和“学生”等主体。代理“总线”涉及本体“旅行”和“交通”。关于agent语言类型的信息主要用于支持自然语言的理解和生成，也用于会话中的推理。这个代理所代表的每个概念的词汇类型、同义词和反义词都记录在这个槽中。

因此，CBS代理的结构如下所示：

代理（⟨名称⟩）

父亲⟨对父类的有条件引用⟩

语法⟨这个代理的语言类型⟩

链接：⟨本体名称列表⟩

能力⟨CS网络列表⟩

相信⟨CS 网络列表⟩

策略⟨由推理规则组成的 CS-prolog 程序⟩

结束⟨名称⟩

CS-Net 是 commonsense-Net 的缩写，它是我们设计和实现的一种语义网络，用于表示 agent 的信念和能力。CS Prolog 是 commonsense Prolog 的缩写，是 Prolog 的进一步发展，带有一组内置的专用谓词和函数。CS Prolog 程序不是在谓词数据库上运行，而是在一组带有通信原语的分层语义网络上运行。

4.3 本体论：构建主体社会

粗略地说，本体论是一个由主体和子本体构成的结构化宇宙，与某个话题相关。

本体论的总体结构如下所示：

⟨本体类型⟩ 本体论[⟨名称⟩]

父亲⟨对父亲本体论的条件引用⟩

静态扩展⟨此本体中涉及的代理列表⟩

动态扩展⟨此本体所需的子本体列表⟩

属性⟨属性列表⟩

本体网⟨连接静态扩展和动态扩展的语义网络列表⟩

结束⟨名称⟩

在 Pangu 中，我们总结了 Agent 之间的各种关系，这些关系用于设计 10 多种不同类型的本体。它们有不同的语法和语义。本体类型包括：过程本体、描述本体、概念本体、因果本体和模拟本体。有关代理和本体设计的详细信息，请参阅参考文献。

4.4 效率

常识知识处理通常需要很多时间。为了处理一些查询，计算机经常需要扫描整个知识库以获得答案。其效率低下的部分原因是以 agent 和本体的形式组织知识。因此，我们增加了一个额外的知识表示层：关系。所有代理和本体都在关系数据库的关系中进行转换。我们利用了关系演算，效率提高了十倍。

提高效率的第二种技术是使用语义缓存。在每次对话中，都会检测到对话伙伴的兴趣，并提前将相关本体提取到主存中，以节省内存访问时间和知识库搜索时间。这再次带来了两倍的效率提升。我们的效率提高了 20 倍。

4.5 回顾

常识知识库盘古的发展一直是我们 CTT 平台建设的主要瓶颈。与任何教科书或百科全书中都能轻易找到的专业知识不同，常识知识虽然无处不在，但却没有明确的表述。

agent 的静态知识用 CS-Net 表示。语义网络的形式适合于以自然语言的形式表示常识知识。它的弱点是这种表述的模糊性。由于我们没有专业的程序员，不同的学生做了知识的工作在不同的时间编码。这使得模糊性问题更加严重。为了克服这个困难，我们从数千个句子样本中总结了 88 种正常的语义网络表示形式。然后将用户输入的理解过程分为三个阶段。在第一阶段，对自然语言句子进行分析，并将其转换为语义网络。在第二阶段，该语义网络被转换为规范形式，然后在第三阶段对其进行语义处理。这大大提高了盘古的知识处理能力。

本体概念在盘古中有多种用途。它不仅用于将代理组织为知识单元[10,11]，还用于组织知识丰富的对话[19,20]。这将在下一节中详细描述。本体论的第三个用途是保持知识存储的合理组织。它的第四个用途是为实现语义缓存提供下一个会话流的预测，以提高会话的运行效率。详见参考文献。

5 CTT 平台：对话引擎

对话引擎主要处理 CTT 平台的语言部分[27]。我们省略了大部分细节，这里只讨论两点。

5.1 期望的对话

我们认为对话是一个不断产生和消除期望的过程。“一头牛有几条腿？”产生一个知识预期，答案“四”涵盖了这一点。类似地，“我母亲病了”的演讲产生了一种安慰的期望，这在回答“我很抱歉听到这个”中有所体现。

期望在这里意味着语用学。请注意，我们对期望的定义是第三方观察者的定义，这与 Austin[22]和 Searle[23]给出的定义不同。他们将期望定义为对话中一个参与者 A 对另一个参与者 B 的期望。我们将其定义为 B 对 A 对 A 对 B 的演讲的任何可能反应。例如，当 A 要求 B 帮他的忙时，A 期望 B“是的，我会”。但像“抱歉，我不能”这样的回答也属于观察者的期望——A 的反应越恰当，他们的对话就越自然，图灵测试就越成功。

我们定义了四组期望类型，其中 x 和 y 表示两个对话伙伴。它们是：信息期望，如果 x 输入的句子是一个查询，或者它包含信息缺陷，那么 y 可能会试图指出它，甚至纠正它。情绪预期，如果 x 输入的句子包含可能

影响 y 情绪的信息（x 对 y 的态度好或坏的信息，x 对 y 的态度好或坏）。态度预期，如果 x 输入的句子包含可能刺激 y 对 x 表达态度（欣赏、贬低、安慰、祝贺、感谢、抱怨等）的信息。回顾预期，如果 x 输入的句子包含可能刺激 y 表达观点的信息（新闻、事件）。计算机联盟成员不仅试图理解法官的期望并覆盖（实现）它，还试图检查法官是否理解并尊重他自己的期望。实现的基本思想是有一个状态转换图。但是没有内存的简单状态转换图不能支持对话，更不能支持上下文敏感的对话。我们使用多个堆栈来存储多个期望值。

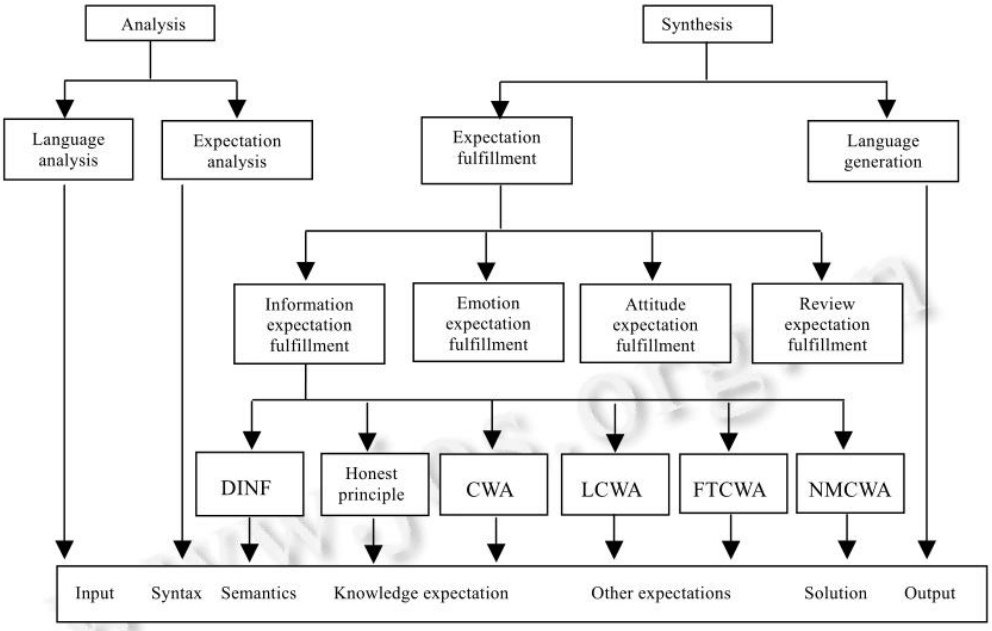

Fig.1 The expectation blackboard

5.2 期望处理黑板

对话引擎的黑板系统 BLAS[24]用于期望处理。BLAS 是一个等级化的 BS。它的架构是树状的，有几个层次。0 级的知识来源是分析和综合。在 1 级中，分析组件在语言分析（语法和语义）和期望分析（语用学）中被分离，而综合在期望生成和移除中被分离，期望生成和移除在四种类型的期望的处理组件中被进一步分离。每个期望处理组件依次有其子组件。例如，knowledge Expection 组件控制前面几节提到的不同的查询回答策略。

BLAS 的示意图如图 1 所示。布拉斯自然风格对话的诀窍在于巧妙地运用知识。在我们的知识库中，我们不仅有诸如动物园、老虎、狮子等代理，还有诸如参观动物园、喂养动物、乘坐公共汽车等本体。每个本体都是一个话题。一旦计算机联盟检测到覆盖法官语音（部分或全部）的本体，它就会利用该本体提供的知识，生成知识丰富的句子。例如，如果法官输

入的第一句话让人想起本体论“参观动物园”，那么本体论中包含的知识，比如：孩子们在参观动物园时经常有父母或老师陪同；参观动物园的目的是看动物园里的动物；参观动物园的一个有趣的项目是看动物饲养员喂动物，等等。可以用来产生知识丰富的回复。

在每次会议期间，计算机联盟维护所有对话主题的全局列表，并将检测法官演讲中的主题变化，然后相应地调整其策略以适应这种变化。

6、完善封闭世界假设

如果对话引擎没有足够的知识来构建对其对话伙伴的回复，最简单的方法就是说“不知道”。我们称之为诚实原则。但它也可能诉诸于这一原则逻辑互补。其中一个原则就是雷特所谓的封闭世界假设（CWA）。它说，每一个不能被知识库证明的命题，都应该被视为虚假的知识库。例如，如果知识库中没有说一头牛有两条尾巴，那么“一头牛有两条尾巴”的命题是错误的。从这个意义上讲，我们想用以下简单的形式重新表述 CWA：

定义 1（CWA1）。给定一个一致的知识库 K 和一组命题 P，关于 K 和 P 的封闭世界假设假定：

1.在域 p={p}上存在一个总函数 cwaK（p）

2。cwaK 的值域是{true，false}。

对于 p 的每一个 p，如果可以用 K 证明 p 为真，那么 cwaK（p）=真，否则 cwaK（p）=假。

这个 CWA 并不完美。它忽略了三个可能答案中“不知道”的可能性{yes，no，don know}，并将原始问题转换为布尔类型。传统的方法可能对数据库查询有用，但肯定不适合图灵对话。当且仅当我们的知识完整时，CWA 才能安全地应用于图灵对话。假设我们有一个大的知识库 K，对于谓词集 p 的每个 p，p 或~p 都可以用 K 证明。在这种情况下，我们可以从 K 中删除所有只用于证明~p 的知识项，以形成一个较小的知识库 K'。那么，如果 q 不能被证明，任何~q 都可以被认为是被 K'证明的。

但是，如果我们的知识不完整，会发生什么呢？在这种情况下，我们无法构建上述知识库 K'。这正是常识的情况，其范围是开放的。在这里，使用 CWA 是危险的。所有尚未在知识库中编码的内容都将被视为错误。

这一事实提醒我们区分“不”和“不知道”的必要性，即扩展 CWA 原则的必要性。为了简单起见，以下形式化以 CWA1 的形式给出。但它也适用于 CWA2。

定义 2（参数理论）。给定一组 n 个概念 C={Ci | i=1,2，…，n}，其中 C 形成一个树状继承层次结构。每个 Ci 都附有一个理论 Ti，即

1．如果 Cj 是 Ci 的父概念，如果命题 p（Cj）可以用 Tj 证明，那么 p（Ci）也可以用 Ti 证明。

2.如果对于 Cj 的所有子概念 Ci，p（Ci）都可以用 Ti 证明，那么 p（Cj）也可以用 Tj 证明。集合{（Ci，Ti）}被称为参数理论。

定义 3（LCWA）。给定一个知识库 K={（Ci，Ti）|i=1,2，…，n}，这是一个参数理论，以及一组命题 Q={qj（Ci）|i=1,2，…，n；j=1,2，…}，关于 K 和 Q 的格闭世界假设为：

1、在域 Q，上存在一个总函数 lcwaK（x）。

2、lcwaK（x）的值域是 L={true，false，不知道，依赖}，

3。如果 qj（Ci）可以用 Ti 证明，那么 lcwaK（qj（Ci））=true，

4。如果 qj，那么 qci 可以被证明是错误的。

5、如果 qj（Ci）不能用 Ti 来判定，但 qj（Ch）可以用 Th 来证明≠h、然后是

5.1。如果 Ch 不是 Ci 的子概念，那么 lcwaK（qj（Ci））=false，

5.2。如果 Ch 是 Ci 的一个子概念，那么 lcwaK（qj（Ci））=

6。否则 lcwaK（qj（Ci））=不知道。它被称为格化 CWA，因为它的值域不是对{true，false}，而是格 L={depends<{true，false}<不知道}

主题

1.在 LCWA 中，映射{qj（Ci）}→L 是独一无二的，

2。一个命题在 LCWA 中为真当且仅当它在 CWA 中为真时。

证据容易的

在代理继承层次结构中组织的知识库，例如我们的盘古知识库，可以被视为一个参数理论，其中每个代理都是一个概念，代理 Ai 的所有信仰以及从其祖先继承的那些信仰构成了理论。让我们考虑以下四个问题 QJ（鸟）：一只鸟需要食物。（对，因为它的父亲“动物”相信“需要食物”）

鸟会弹钢琴。（错，因为鸟不相信“会弹钢琴”，但人相信）

鸟会飞。（=视情况而定，因为鸟有“能飞”的信念，例外）

鸟会被艾滋病感染。（=不知道，因为在知识库的任何信仰中都没有关于艾滋病的说法）

7.智力年龄和智商

让我们回忆一下图灵在其创始论文中做出的著名预测：我相信在 50 年后，将有可能为计算机编程，存储容量约为 109，让他们在模仿游戏中玩得如此之好，以至于一个普通的审问者在五分钟的审问后做出正确身份的几率不会超过 70%。[2] 根据过去 10 年的洛布纳测试实践，我们可以得出这样的结论：图灵的预测已经失败了。但图灵没有透露审讯者（即法官）的任何细节，只使用了“平均”一词，以避免进一步讨论有关该审讯者的具体情况，如年龄、性别、国籍、职业等。然而，如果对测试设计的各种因素有一些限制，结果会完全不同。也就是说，如果我们考虑图灵检验的成功率（=做出正确的识别的机会）作为这些因素的函数，那么我们可以用下面的方法来重新表述图灵的预测。

一个建议的预测框架：我们相信，在 f（x，y，t）年的时间里，将有可能对存储容量约为 g（x，y，t）的计算机进行编程，使它们能够很好地玩模拟游戏，使 x 岁的普通审问者在 t 分钟的提问后做出正确识别的机会不超过 y%。这个预测框架，其中 f 和 g 都是实函数和连续函数，反映了我们孩子图灵测试的动机之一。我们希望通过限制法官（审讯人员）的能力来降低图灵测试的难度。从历史上看，这并不是削弱图灵测试要求的第一次尝试。科尔比让电脑模仿偏执狂的实验就是一个例子[7]。另外两个参数是提问时间和成功率（百分比）。

图灵认为五分钟的问话应该足以让法官有 30%的机会做出正确的身份证明。他没有说如果允许更多的提问时间，这个比例是否会更高。在这方面，我们有以下几点：

猜想 1。提问时间的临界长度（以分钟为单位）CRT>0。对于任何 t>CRT，我们有 f（x，y，t）=f（x，y，CRT），g（x，y，t）=g（x，y，CRT），这意味着，在一定限度内，更多的对话时间不会带来更多的信息，这将有助于正确识别。有实验证据表明这个猜想可能是真的。以下是 2000 年罗布纳测试报告的引文：

每位法官在每个终点站最多花 15 分钟。他们被要求在五分钟后判断被告是人类还是计算机，然后在 15 分钟后再次做出判断。在一些案件中，法官在 15 分钟后改变了判决，但大多数最初的判决保持不变

我们的第二个猜想如下：

猜想 2。审讯人员的临界年龄 CRA>0。对于任何 x>CRA，我们有 f（x，y，t）=f（CRA，y，t），g（x，y，t）=g（CRA，y，t）。这意味

着，一般来说，超过某个阈值，增加询问者的年龄不会降低通过计算机图灵测试的机会。事实上，Loebner 报告中没有提到法官的年龄。

接下来是我们的第三个猜想。

猜想 3。∞= >−),,（lim 0 CRTyCRAf y，∞= >−),,（lim 0 crtycragy）这意味着永远不会有一天 TT 会真正成功。换句话说，TT 的成功只能是一个无限的过程。我们的这个猜测可能会被批评为过于悲观。如果在可预见的未来被实践推翻，我们将非常高兴。

最后，我们提出了我们的最后一个猜想：

猜想 4。没有 ot 她（被广泛接受的）衡量人工智能（证明计算机智能）最终成功的标准，比 TT 更早实现。

在计算理论中，没有其他计算机制（递归函数、无限制乔姆斯基语法等）比图灵机更强大。同样，对于猜想 4，我们认为没有其他机制（符号推理、计算智能、神经网络、非单调推理等，或它们的组合）能帮助 AI 比 TT 更快地达到最终目标。

8.对成就和未来工作的评估

我们迄今取得的成果令人鼓舞，但仍有许多问题有待澄清。在我们高兴地看到我们的计算机已经通过了 CTT 的一些会议之后，我们想问自己两个问题：第一，我们有理由说我们的 CTT 没有违反图灵测试的本质，正如图灵在 1950 年定义的那样？第二，任何 TT 的（部分）成功能在多大程度上证明计算机联盟具有智能？

8.1 除了中文房间问题

要回答这两个问题，请注意，关于 TT 的重要性存在着众所周知的争议和辩论，其中包括著名的 Searle 的中文房间论证[3]。我们在本文中将不讨论这一点，只对其进行补充。不管 TT 是否真的能检查计算机的智能，有一点是明确的：它只能检查语音中显示的那部分智能。例如，它忽略了与模式识别有关的一切。一个完美的 TT 至少应该包括计算机视觉和语音识别。在我们的 TT 中，只使用了文本形式的消息。因此，即使在言语智能的意义上，它也不是完美的。

这还不是全部。人类智力应该包括劳动智力，一种涉及劳动的智力。例如，TT 的成功并不能说明计算机是否能用它在沙滩上用沙建造一座塔。

8.2 知识瓶颈

目前几乎所有的谈话节目都使用了很多聊天策略。它们使用模式匹配用户输入，并在回复语句中转换它们，而无需解析过程。这种方式与塞尔

描述的情况非常相似。坦率地说，这些程序是在“欺骗”用户。他们的节目中使用了很多作弊策略。作弊的需要是由知识的缺乏引起的，包括语言知识的缺乏。正如前面所说，我们的系统与当前对话程序的主要区别在于，我们的系统是基于知识的，尤其是基于常识的。但常识的数量是巨大的。这意味着我们的知识库永远不会完整。因此，会话引擎也面临同样的知识瓶颈问题。只是知识的缺乏程度不同。

8.3 法官的资格从上述讨论中可以看出，TT 的另一个问题是：TT 的结果取决于法官的智力。没有抽象的人。每个人都是具体的。每个人都有不同的智力水平。哪个人可以被认为有资格担任 TT 的法官？儿童成人？普通人？人工智能专家？似乎法官不应该是一个人，而应该是一大群人的平均数。测试不应该是一个单一的测试，而是一个无限系列的测试。这些测试一旦成功，将见证机器智能向真正的人类智能的提升。因此，我们的结论是：永远不会有一天，TT 会在真正意义上取得成功（图灵先生所指的意义）。换句话说，TT 的成功只能是一个无限的过程。

8.4 未来工作我们已经注意到，图灵·赛尔夫给出的 TT 定义并不是唯一的。可能会有不同的变化。我们将在下一个项目中更详细地研究它们。为了探索这些变化，并比较它们在确定机器智能方面的意义，必须测试不同的 TT 设计。

承认我们在图灵测试中的实验是 CNSF 项目“常识知识的实践方面”的一部分。该项目的其他参与者包括清华大学的石春义教授和中山大学的李世贤教授及其研究小组；美国医学科学院金志进教授、魏子初教授、刘冷宁、杨、陆平凡、赵冲、杨帆、金小龙、郑红、刘洪格、舒成、胡思康、易南、赵晓玲。我们感谢他们的贡献。