

The Non-Deterministic Constraint Logic and Its Applications in Computational Complexity

Ziwen Liu¹, Chao Yang^{2*}

¹School of Software Engineering, South China University of Technology, Guangzhou Guangdong

²School of Mathematics, Sun Yat-sen University, Guangzhou Guangdong

Email: *yangch8@mail.sysu.edu.cn

Received: May 2nd, 2017; accepted: May 19th, 2017; published: May 22nd, 2017

Abstract

The computational complexity of several combinatorial puzzles has been determined in the literature in the past few decades. We introduce the non-deterministic constraint logic (NCL) in this paper. As an application, we prove that Beanstalk, a Sokoban-like puzzle, is PSPACE-complete by reduction from NCL.

Keywords

Non-Deterministic Constraint Logic, Computational Complexity, Combinatorial Puzzles

非确定性限制逻辑及其在计算复杂度中的应用

刘孜文¹, 杨 超^{2*}

¹华南理工大学软件学院, 广东 广州

²中山大学数学学院, 广东 广州

Email: *yangch8@mail.sysu.edu.cn

收稿日期: 2017年5月2日; 录用日期: 2017年5月19日; 发布日期: 2017年5月22日

摘 要

过去几十年, 有许多组合谜题的计算复杂度被确定了。本文介绍了非确定性限制逻辑, 并用归约为非确定性限制逻辑的方法, 证明了一种类似于推箱子的种豆游戏的计算复杂度为多项式空间完全的。

*通讯作者。

文章引用: 刘孜文, 杨超. 非确定性限制逻辑及其在计算复杂度中的应用[J]. 计算机科学与应用, 2017, 7(5): 407-413.
<https://doi.org/10.12677/csa.2017.75049>

关键词

非确定性限制逻辑, 计算复杂度, 组合谜题

Copyright © 2017 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近二三十年, 许多组合谜题类的电脑游戏的计算复杂度得到广泛的研究。其中最具代表性的组合谜题之一是推箱子游戏。推箱子又称仓库番(Sokoban), 是日本人今林宏行于 1981 年最早开发的。在推箱子游戏中, 玩家需要控制一个搬运工在二维的仓库中把多个货物箱子推到指定的存放位置。1997 年, Culberson 通过归约为线性空间自动机问题的方法, 证明了推箱子问题, 即判断推箱子关卡是否有解是多项式空间完全(一般简称 PSPACE 完全)的[1]。2005 年, Hearn 和 Demaine 给出了推箱子是 PSPACE 完全问题的另外一个证明[2]。他们的方法是先引入了非确定性限制逻辑(Non-deterministic Constraint Logic, 缩写为 NCL), 并证明了 NCL 问题是 PSPACE 完全的。然后再把 NCL 问题归约到推箱子问题。NCL 方法非常有效, 此后又被其他学者用于证明推箱子的几个变种谜题的计算复杂度, 如[3] [4]。

除了推箱子游戏外, 许多滑块类游戏(Sliding Block Puzzles)的计算复杂度也是 PSPACE 完全的。与推箱子游戏不同的是, 在滑块类游戏中, 所有滑块都可以自主移动, 因此, 这可以作为机器人学中多机器人路径规划的一个理论模型, 在工程自动化中有重要的应用前景。文献[5]的引言部分对多机器人路径规划问题的研究历史和相关问题的计算复杂度有一个扼要的介绍。其中有一类特殊的滑块游戏 Rush Hour, 对滑块的移动方向作了限制, 有的滑块只能在二维平面中左右移动, 有的滑块只能上下移动, 问题是特定的滑块能否移动到指定位置。对于 Rush Hour, 如果滑块的大小都是 1×2 或 1×3 , 已经被证明是 PSPACE 完全的[2] [6]。但对滑块大小都是 1×1 的 Rush Hour 问题, 其计算复杂度还是个公开问题。文献[5]试图解决 1×1 的 Rush Hour 问题, 也是采取了归约为 NCL 问题的方法, 但是他们归约的实例中, 允许滑块移动半格, 所以本质上只是证明了 2×2 的 Rush Hour 问题是 PSPACE 完全的。

与滑块类游戏相关的组合谜题还有翻滚游戏(Rolling Block Puzzles), 游戏中平面的滑块被立体的 $1 \times 1 \times 2$ 的可翻滚物体替代了。这类翻滚游戏的两种变化, 均被证明是 PSPACE 完全的[7] [8], 运用的也都是 NCL 方法。

本文将运用 NCL 方法证明一个类似于推箱子的组合谜题种豆游戏(Beanstalk)也是 PSPACE 完全的。通过这些研究, 进一步揭示这几类相关谜题的内在联系, 以及探索 NCL 方法能否用于最终解决 1×1 的 Rush Hour 问题。

2. 种豆问题和非确定性限制逻辑

种豆游戏(Beanstalk)是 Braingle.com 网站开发的一个在线小游戏[9], 图 1 是游戏截图。

种豆游戏是在一个由一些墙体围成的二维的格子迷宫中, 控制游戏中的一个人上下左右移动, 并且推动游戏中的四种物品: 铲子、种子、化肥、浇水壶, 在迷宫中标记 X 的目标格子中完成种豆。标记 X 的格子和空地一样, 人可以自由通过。

要完成种豆, 须做四步, 即按顺序把铲子、种子、化肥、浇水壶依次推到目标格子。和经典的推箱

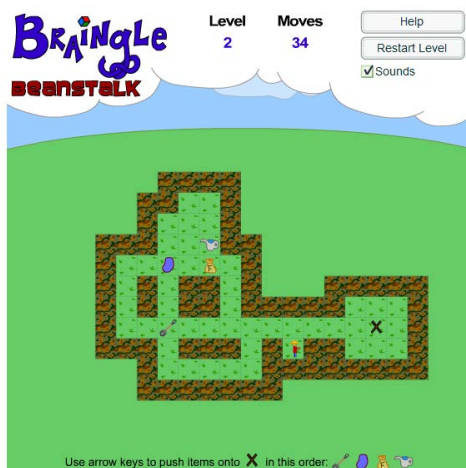


Figure 1. Screenshot of the Beanstalk Puzzle
图 1. 种豆游戏截图

子一样, 人一次只能推动一个物品。第一步, 把一个铲子推到标记 X 的目标格子, 这个铲子便会从迷宫中消失, 同时目标格子会变成一个坑。人若推着种子、化肥或浇水壶可无障碍的通过标记为 X 目标格子, 如同空地一样。第二步, 把一颗种子推到坑里, 这颗种子便会消失, 同时目标格子变成放了一颗种子的坑。对于铲子、化肥和浇水壶, 空着的坑如同空地, 人推着铲子、化肥或浇水壶可以自由通过。第三步, 把一包化肥推到放入种子的坑, 则这包化肥消失, 同时目标格子进一步变成一颗豆苗。人推着铲子、种子或浇水壶则可以自由通过已放入种子的坑。第四步, 把最后一物品浇水壶推到已经演化成豆苗的目标格子, 则浇水壶消失, 豆苗长成一棵豆树, 完成种豆。人可以推着除浇水壶之外的其它三样物品, 即铲子、种子或化肥自由通过豆苗。变成豆树后的格子如同墙体, 人和其他物品不能再通过。

原始的在线种豆游戏每个关卡只需要种豆一棵, 在这种情况下, 问题肯定是在多项式时间内可解的。我们把问题推广到一个关卡需要种豆多棵的情况, 定义其判定问题如下。

定义 1: 种豆问题(Beanstalk)

实例: 问题的一个实例是一个二维的关卡, 关卡中有铲子、种子、化肥、浇水壶和目标格子各 n 个。

问题: 判定这个关卡是否有解, 即按照上面规则是否可以完成种豆 n 棵。

本文主要结论是证明以下定理。

定理 1: 种豆问题是 PSPACE 完全的。

我们的方法是把文献[2]中引入的 NCL 问题归约到种豆问题。NCL 问题是定义在 3 正则的有向图上, 称之为 NCL 图。NCL 图的每条边都赋予权 1 或者 2。同时 NCL 图只有两类顶点, 称为 AND 顶点和 OR 顶点。AND 顶点与两条权 1 边和一条权 2 边关联, OR 顶点则与三条权 2 边关联。两种顶点如图 2 所示, 红边表示权为 1, 蓝边表示权为 2。边的方向不必和图示的例子完全一致, 只要满足下文对入边权的限制即可。

NCL 图必须满足一个限制: 每个顶点的入边的权和必须大于等于 2。在满足这个限制的情况下, 图的有向边可以逐地改变方向。即要求该边方向改变前和改变后, 每个顶点的入边权和都大于等于 2。现若给定 NCL 图的一个初始定向, 和一个目标定向, 问能否在满足限制的情况下, 通过逐一改变某些边的方向, 使 NCL 图从初始定向变为目标定向。

定义 2: 非确定性限制逻辑(NCL)

实例: 给定一个 NCL 图, 并给定一个初始定向和一个目标定向。

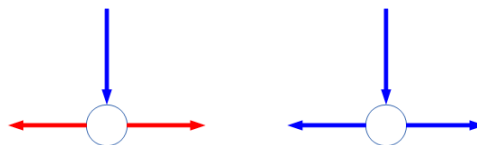


Figure 2. (Left) AND vertex (right) OR vertex

图 2. (左)AND 顶点(右)OR 顶点

问题: 能否在满足入边权和大于等于 2 的条件下, 逐一改变若干边的方向, 从初始定向变到目标定向?

Hearn 和 Demaine 在[2]中证明了即使只考虑平面图, NCL 问题也是很困难的。

定理 2: [2]平面图 NCL 问题是 PSPACE 完全的。

在下一节, 我们将利用定理 2 来证明定理 1。

3. 主要结论的证明

显然种豆问题属于 PSPACE。下面证明对平面图 NCL 问题的每一个实例 I , 我们都可以按某种规则转换为种豆问题的一个关卡 I^* , 使得 I 有解当且仅当 I^* 有解。我们先用种豆问题的局部小部件来模拟 NCL 图的边和顶点, 然后用这些小部件拼接成一个完整的种豆问题的关卡。

3.1. 模拟边的小部件

一个模拟 NCL 图的边的小部件如图 3 所示。这是一个条半封闭的通道, 两头(黑色箭头所示)将分别连接两个顶点的小部件。整条通道被两个出入口(红色双箭头所示)分成三段, 这使得以后拼接成的完整关卡是开放性的, 人可以在整个关卡中自由移动到任意地方。图 3 中标记上 1、2、3、4 的正方形分别表示铲子、种子、化肥和浇水壶。在完成种豆之前, 浇水壶可以依次在三段中转移位置。如图中, 中间一段的浇水壶可以先被推到右边一段, 然后左边一段的浇水壶可以推到中间的一段。因此, 在边小部件中, 进入两个关联顶点的通道, 总会恰有一个被堵上。这样, 就模拟了一条从被堵住的顶点指向没别堵住的顶点的一条有向边; 并且, 若被堵住的那个顶点小部件另有出入口, 则堵住和没被堵住的关系还可以互换, 模拟了有向边方向的变化。若边小部件内完成了种豆, 则边方向就固定下来, 不能再改变了; 因此, 应在最后阶段才完成种豆。

3.2. 模拟顶点的小部件

边的小部件以不同方式连接起来, 就可以形成模拟 OR 顶点或者 AND 顶点的小部件。

图 4 是模拟了 OR 顶点的小部件。OR 顶点本身可视为有三个出入口的区域(图中红色方框标识), 每个出入口与一条边的小部件连接。如图所示, 左侧和中间两条边把出入口堵住了, 因此若右侧的边要想改变状态变成堵住, 则左侧和中间两条边至少有一条要先改变状态变成没有堵住出入口, 否则 OR 顶点的三个出入口都堵住, 关卡就再也无法解出了。这就刚好模拟了 OR 顶点的逻辑。

图 5 是模拟了 AND 顶点的小部件。与 OR 顶点的小部件相比, 有两处不同。一是 AND 顶点本身的区域只有两个出入口, 分别直接和两条边的小部件连接。二是第三条边(左侧浅红色的边)接到中间的边的一个出入口, 从而间接地接到 AND 顶点上去。所以左侧和中间的边必须同时变化到没有堵住的状态, 右侧的边才能堵住 AND 顶点的另一个出入口。这样就模拟了 AND 顶点的逻辑。

总之, 不管是 OR 顶点小部件还是 AND 顶点小部件, 要保持关卡有解的可能性, 任何一个顶点区域至少有一个出入口没有被堵住, 这恰好模拟了 NCL 图的入边权和大于等于 2 的限制。

3.3. 拼接

现在我们完整地描述如何根据给定的一个平面图 NCL 问题的实例, 构造出一个与之等价的种豆问题

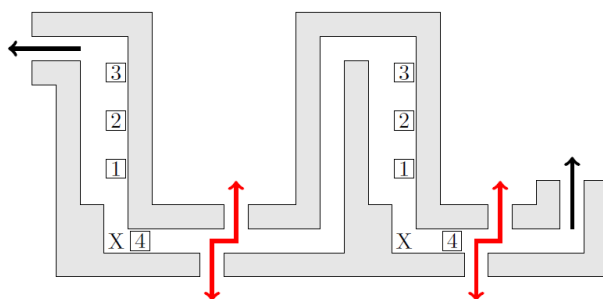


Figure 3. Edge gadget

图 3. 边小部件

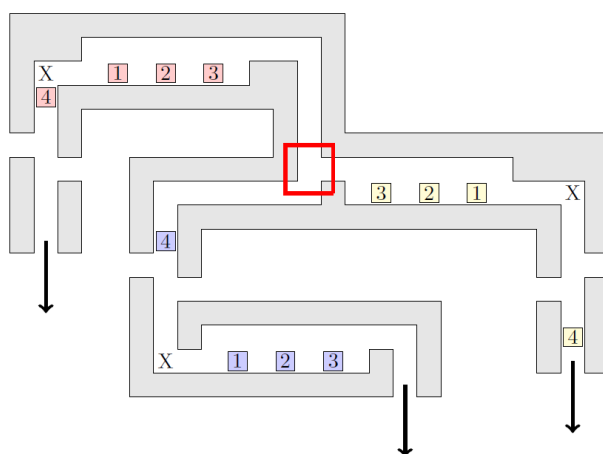


Figure 4. OR vertex gadget

图 4. OR 顶点小部件

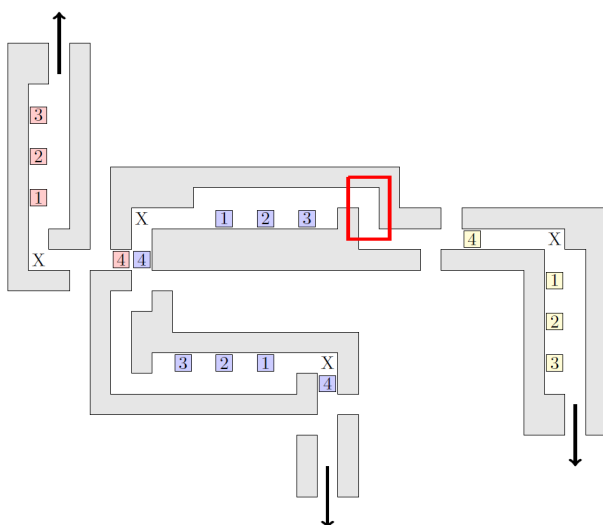


Figure 5. AND vertex gadget

图 5. AND 顶点小部件

的关卡。对 NCL 图的每一条边，若初始时从顶点 u 指向顶点 v ，则相应的边小部件就堵住顶点 u ，且不堵住顶点 v 。在 NCL 的实例中，若边的初始方向与目标方向一致，则在相应的边小部件中，浇水壶都与目标点 X 处于同一段；若边的初始方向与目标方向不一致，则浇水壶与目标点不处于同一段。考虑到边

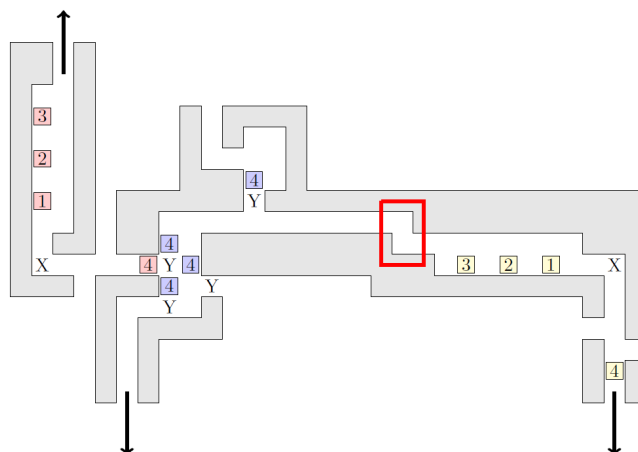


Figure 6. AND vertex gadget (II)

图 6. AND 顶点小部件之二

小部件可以随意转弯, 甚至可以增加段数, 我们的 NCL 图又是平面图, 所以把前文所描述的边小部件和顶点小部件拼接成一个完整的关卡基本上没有太大困难。

但是, 按上述方法拼接完整的关卡, 还是有一处难点出现在 AND 顶点小部件。即若有一条权 1 的边小部件直接连接到 AND 顶点区域, 并且该边小部件的目标状态对应着不堵住 AND 顶点区域。在这种情况下, 3.2 小节的图 5 所介绍的标准方法无法使用, 我们需要用图 6 的方法来实现。在图 6 中, 这条直接连接 AND 顶点的权 1 边用蓝色标识, 且处于堵住 AND 区域的状态。图中标记 Y 的格子表示变化成豆苗的目标格子。若要变成不堵住状态, 只能把图 6 中从下数起第二个蓝色浇水壶向下推动一格, 否则会导致关卡立刻无解。此外, 从上数起第一个蓝色浇水壶形成一个单向通道, 以便在下文提到的第二阶段完成种豆后, 人能够离开封闭的 AND 顶点区域。

我们这样得到的一个种豆关卡, 解关过程可分为两个阶段。第一阶段, 把所有边的浇水壶都转移到与目标点 X 在同一段的状态, 这对应着把 NCL 图的边都定向到目标方向。若第一阶段无法实现, 这个种豆关卡是无解的(用逆向思维, 若关卡可解, 则一定可从过关状态退回到此状态)。若第一阶段实现了, 我们可以进入第二阶段, 把每一条边每一段里的一组铲子、种子、化肥和浇水壶依次推到目标点完成种豆。因为每个顶点小部件至少有一个出入口未被堵住, 此时第二阶段的种豆任务是一定可以完成的。因此原 NCL 问题的实例可解当且仅当相应的种豆关卡可解。

综上所述, 种豆问题是 PSPACE 完全的。

4. 结语

本文通过把 NCL 问题归约到种豆问题, 证明了种豆问题是 PSPACE 完全的。这体现了 NCL 问题在确定组合谜题的计算复杂度中的威力。与使用 NCL 问题证明滑块谜题[2] [5]或翻滚类谜题[7] [8]相比, 我们的证明中, AND 顶点小部件的设计要困难和复杂一些, 这表明了推箱子类谜题具有独特之处。我们的设计方法可以用来解决 1×1 的 Rush Hour 问题吗? 另外, 我们设计的 AND 顶点小部件中, 有一种情况(图 6)用到了中间物品豆苗。能否找到另外的设计只用到初始物品(即铲子、种子、化肥、浇水壶)呢? 这些都是值得进一步研究的问题。

致 谢

感谢匿名审稿人对本文初稿提出的宝贵意见。

基金项目

国家自然科学基金青年科学基金项目(11201496), 广东省自然科学基金(2015A030313222)。

参考文献 (References)

- [1] Culberson, J.C. (1997) Sokoban is PSPACE-Complete. Technical Report TR 97-02, Department of Computing Science, University of Alberta, Edmonton.
- [2] Hearn, R.A. and Demaine, E.D. (2005) PSPACE-Completeness of Sliding-Block Puzzles and Other Problems through the Nondeterministic Constraint Logic Model of Computation. *Theoretical Computer Science*, **343**, 72-96. <https://doi.org/10.1016/j.tcs.2005.05.008>
- [3] He, W., Liu, Z. and Yang, C. (2017) Snowman is Pspace-Complete. *Theoretical Computer Science*, **677**, 31-40. <https://doi.org/10.1016/j.tcs.2017.03.011>
- [4] Pereira, A.G., Ritt, M. and Buriol, L.S. (2016) Pull and Push Pull Are PSPACE-Complete. *Theoretical Computer Science*, **628**, 50-61. <https://doi.org/10.1016/j.tcs.2016.03.012>
- [5] Solovey, K. and Halperin, D. (2016) On the Hardness of Unlabeled Multi-Robot Motino Planning. *The International Journal of Robotics Research*, **35**, 1750-1759. <https://doi.org/10.1177/0278364916672311>
- [6] Flake, G.W. and Baum, E.B. (2002) Rush Hour is Pspace-complete, or Why You Should Generously Tip Parking Lot Attendants. *Theoretical Computer Science*, **270**, 895-911. [https://doi.org/10.1016/S0304-3975\(01\)00173-6](https://doi.org/10.1016/S0304-3975(01)00173-6)
- [7] Buchin, K. and Buchin, M. (2012) Rolling Block Mazes Are PSPACE-Complete. *Journal of Information Processing*, **20**, 719-722. <https://doi.org/10.2197/ipsjip.20.719>
- [8] Van der Zanden, T.C. and Bodlaender, H.L. (2015) Pspace-Completeness of Bloxorz and of Games with 2-Buttons. In: Paschos, V.T.H. and Widmayer, P., Eds., *Algorithm and Complexity*, Springer, Berlin, 403-415. https://doi.org/10.1007/978-3-319-18173-8_30
- [9] Braingle (2017) Beanstalk Puzzle Game. <http://www.braingle.com/games/beans/index.php?play=1>

7 和 8 涉及了
立方块翻滚
游戏

期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org