归约

维基百科，自由的百科全书

在可计算性理论与计算复杂性理论中，所谓的归约是将某个计算问题（computational problem）转换为另一个问题的过程。可用归约法定义某些问题的复杂度类（因转换过程而异）。

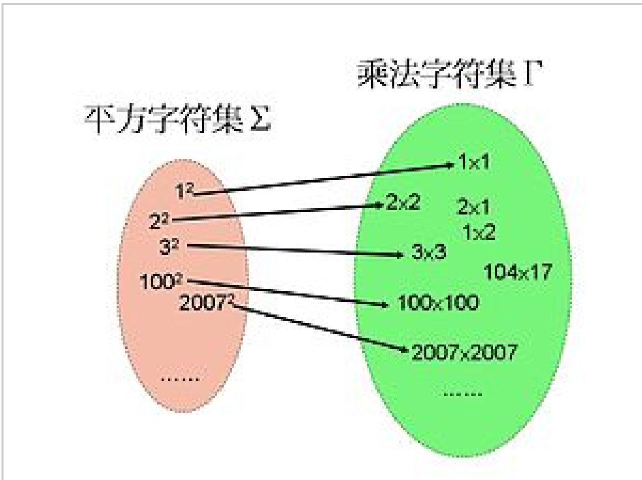
以直觉观之，“问题A可归约为问题B”，指问题B的答案可用于解决问题A。因此解决A不会难于解决B。吾辈写作A ≤ B，通常也在≤符号下标使用的归约手法。

目录

简易介绍

我们解题时常遇见似曾相识的题目。此时，我们若可将新题转换成已解旧题的一例，则新题亦解矣。

另一更微妙的用法是：若我们拥有一个已证明难以解决的问题，我们又获得另一个相似的新问题。我们可合理推想此新问题亦是难以解决的。我们可由下列谬证法得证：若此新问题本质上容易解答，且若我们可展示每个旧问题的实例可经由一系列转换步骤变成新问题的实例，则旧问题便容易解决，因此得到悖论。因此新问题可知亦难以解决。



以乘法与平方为例的多一归约示意图。

一个归约简例是从乘法化成平方。设想我们仅能以加、减、平方与除以二等操作，我们可运用此知识并结合下列方程式，以取得任两数的乘积：

$$a\times b = ((a + b)^2 - a^2 - b^2)/2.$$

我们亦可从另一方向归约此问题：显然地，若我们可以乘以任两数，则我们可以对任一数平方：

$$a^2 = a\times a.$$

因此可见两问题之难度似乎相等，此类归约称为图灵归约。上题的图灵归约关系为：

乘法$\leq_T$平方且 平方$\leq_T$乘法

然而，若我们增加条件：“此运算只能使用平方一次，且只能在结尾使用”，则更难寻找合适归约。在这条件下，即使我们使用所有基础运算，包括乘法，也找不到适当的归约步骤。因为我们不仅要运算有理数，也必须运算像是$\sqrt{2}$的无理数。而另一方向的归约，我们的确可用一次乘法简单地平方任何数，且此乘法的确是最后的运算。将此限制形式导入归约中，我们已展示其归约结论：普遍来说，乘法的确难于平方。此归约称为多一归约。上题的多一归约关系为：

平方$\leq_m$乘法（因为每个合法的整数平方式n^2都可归约成乘法$n\times n$，但反之不然）

定义

给予两个自然数N的子集A与B，以及一个函数集合F，型态为由N至N，并拥有复合封闭性。我们称在F下，A可归约成B若：

$$\exists f \in F \ . \ \forall x \in \mathbb{N} \ . \ x \in A \Leftrightarrow f(x) \in B$$

我们写做：

$$A \leq_F B$$

设S为P(N)的子集，另设≤的归约关系，则S称做封闭于≤之下若：

$$\forall s \in S \ . \ \forall A \in P(\mathbb{N}) \ . \ A \leq s \Leftrightarrow A \in S$$

一N的子集A，称对S困难（hard），若：

$$\forall s \in S \ . \ s \leq A$$

一N的子集A，若A对S困难且A包含于S集合之内，则称A对S完备（complete）。

复杂性类的判别

- 若要证明一问题是不可在决定的，我们可以一可计算函数将它转换成另一已知不可决定的问题，例如，欲证P是不可决定的，可试将停机问题化约成问题P。
- 复杂度类P、NP与PSPACE拥有多项式时间归约的封闭性。
- 复杂度类L、NL、P、NP与PSPACE拥有对数空间归约（log-space reduction）的封闭性

详例

下例利用从停机问题至某个语言的转换，以证明该语言是不可决定的。设H(M,w)是问题：“判定给定的图灵机M会否在输入字串w后停机（接受或拒绝此字串）”。此语言已知是不可决定的[1]。又设E(M)是问题：“给定图灵机M，判定它所接受的语言是否空（意即M是否接受任何字串）”。我们可以借由从H归约成E以显示E也是不可决定的。

为了获得悖论，假设R是E的一个仲裁机器（decider，即一定会停的图灵机），我们将用此机器R产生问题H的仲裁机器S。给予输入资料——一个图灵机M与某些输入字串w，定义图灵机S(M,w)：S创造一个图灵机N，N仅接受输入图灵机M时会停止的字串w，输入其他字串则N进入死循环。仲裁机器S现在可评估R(N)，以验证被N接受的语言是否为空集合。如果R接受N，则被N接受的语言是空集合，所以M不会在输入为w时停止，所以S可以拒绝。如果R拒绝N，则被N接受

的语言是非空集合，则M不会在输入为w时停止，故S可接受。因此若我们有了E的一仲裁机器R，则我们将能产生停机问题H(M,w)及任何机器M与任何输入字串w的仲裁机器S。但我们已知此S绝对不存在，故得矛盾。因此可知语言E同样也是不可决定的。

注

归约亦是一种预序关系，意指在P(N)×P(N)，此P(N)上拥有自反关系与传递关系，此处的P(N)是自然数的幂集（power set）。

若在某个复杂度类别上的所有问题都可归约成某问题P，则可称P是完备（complete）的，且P自己也会处于此类别中。故问题P代表此类别，因其任一解都可经由归约解决此类别中的所有问题。[2]

归约种类与应用

依上例所述，在计算复杂度中，主要有两大类的归约：多一归约与图灵归约。多一归约将一问题的所有实例对应到另一问题的实例上；图灵归约计算一问题的解，并假设其他问题容易解决。多一归约弱于图灵归约。较弱的归约在分割问题的种类上效率较高，但它们的威力较弱，使本类归约较难设计。

然而，为了使归约有用，它们必须易于使用。例如实际研究中常常要将难以得解的NP完备问题，例如SAT问题，归约成显而易懂的问题，像借由效率为指数时间并在有解时输出整数零的机器，决定一数是否为零。但这并没有多少用处，因为我们可以执行如同解决旧问题一样难的归约以解决新问题。

因此，依照复杂度类别使用适当归约符号的学问兴起。在钻研复杂度类NP与更难的类别时，我们使用多项式时间多一归约。在多项式阶层中定义类别时，我们使用多项式时间图灵归约。当我们在类别P之内学习NC与NL类别时，我们使用对数空间归约。归约也用在可计算性理论中，以显示问题是否可不可被任何机器解决；在此情境下，归约仅局限于可计算函数上。

参考文献

1. ^ 例如：http://cs.wwc.edu/KU/Logic/turingMachine.html
2. ^ Thomas H. Cormen, Introduction to Algorithm, second edition, page. 970, figure 34.1.

参阅

- 多一归约
- 真值表归约
- 图灵归约
- Comparison of numberings
- 优化（计算机）

文献

- （英文） Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, 2001, ISBN 978-0-262-03293-3
- （英文） Hartley Rogers, Jr.: Theory of Recursive Functions and Effective Computability, McGraw-Hill, 1967, ISBN 978-0-262-68052-3.
- （英文） Peter Bürgisser: Completeness and Reduction in Algebraic Complexity

Theory, Springer, 2000, ISBN 978-3-540-66752-0.
- （英文） E.R. Griffor: Handbook of Computability Theory, North Holland, 1999, ISBN 978-0-444-89882-1.

取自“http://zh.wikipedia.org/w/index.php?title=歸約&oldid=33527065”
