Other Complexity Classes

Computational Complexity

Luis José Quintana Bolaño

March 31, 2014

Introduction

In computational complexity theory, a complexity class is a set of problems of related resource-based complexity.

A typical complexity class has a definition of the form:

The set of problems that can be solved by an abstract machine **M** using **O(f(n))** of resource **R**, where **n** is the size of the input.

The simpler complexity classes are defined by the following factors:

- **The type of computational problem:** The most commonly used problems are decision problems. However, complexity classes can be defined based on function problems (an example is FP), counting problems (e.g. #P), optimization problems, promise problems, etc.
- **The model of computation:** The most common model of computation is the deterministic Turing machine, but many complexity classes are based on nondeterministic Turing machines, boolean circuits, quantum Turing machines, monotone circuits, etc.
- **The resource (or resources) that are being bounded and the bounds:** These two properties are usually stated together, such as "polynomial time", "logarithmic space", "constant depth", etc.

Time-Bounded Complexity Classes

These are some of the most important complexity classes defined by bounding the time of the algorithm:

Class	**Computational Model**	**Resource Constraint**
DTIME	DTM	Time $f(n)$
P	DTM	Time $poly(n)$
EXPTIME	DTM	Time $2^{poly(n)}$
NTIME	NDTM	Time $f(n)$
NP	NDTM	Time $poly(n)$
NEXPTIME	NDTM	Time $2poly(n)$

Space-Bounded Complexity Classes

Some examples of classes defined by bounding the space of the algorithm:

Class	**Computational Model**	**Resource Constraint**
DSPACE	DTM	Space $f(n)$
L	DTM	Space $O(\log n)$
L^2	DTM	Space $O(\log^2 n)$
PSPACE	DTM	Space $poly(n)$
EXPSPACE	DTM	Space $2^{poly(n)}$
NSPACE	NDTM	Space $f(n)$
NL	NDTM	Space $O(\log n)$
NPSPACE	NDTM	Space $poly(n)$
NEXPSPACE	NDTM	Space $2^{poly(n)}$

Relations Between Time- and Space-Bounded Classes

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXPTIME} \subseteq \mathsf{NEXPTIME}$$

Where

$$\mathsf{PSPACE} = \mathsf{NPSPACE} \text{ and } \mathsf{P} \subset \mathsf{EXPTIME}$$

Other classes

Other important complexity classes that use different computational models include:

- BPP, ZPP and RP, which are defined using probabilistic Turing machines.
- AC and NC, which are defined using boolean circuits.
- BQP and QMA, which are defined using quantum Turing machines.

Complementary Classes

Complexity classes have a variety of closure properties; for example, decision classes may be closed under negation, disjunction, conjunction, or even under all Boolean operations.

Definition

Each **class X** that is **not closed under negation** has a **complement class co-Y**, which consists of the complements of the languages contained in X.

Example

For the **NP** class there's a **coNP** class which, for every decision problem P in **NP**, contains the complementary decision problem, denoted *coP* (i.e., the decision problem in which the "Yes" instances are "No" instances of **P** and vice versa).

If **C** is a deterministic time or space complexity class, then **coC** $=$ **C**.

References

[] John E. Savage. *Models of Computing: Exploring the Power of Computing*. Brown University, 2008.