| | |
|---|---|
| **CS270 Combinatorial Algorithms & Data Structures** | **Spring 2004** |

Lecture 21: 4.6.04

*Lecturer: Prof. Satish Rao* — *Scribe: Qi Zhu*

**Disclaimer**: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

## 21.1 3SAT $\leq_p$ 3DM(3D-Matching)

**Definition 21.1 (3DM)** *Given a boys-girls-pets compatibility relation (i.e., a set of boys-girls-pets triangles), is there a complete matching which makes everyone happy (i.e., all boys, girls and pets are covered by a set of disjoint triangles)?*

*More general way: Given finite disjoint sets $X, Y, Z$ of size $n$, and a set of triples $\{t_i\} \subseteq X \times Y \times Z$, are there $n$ pairwise disjoint triples?*

**Theorem 21.2** *3SAT can be reduced to 3DM.*

**Reduction from 3SAT**

In 3SAT, there are literals, clauses, and consistency constraints. The general technique of 3SAT reduction has three aspects: construct a "flip-flop" for each literals; construct a "constraint" for each clause; and maintain consistency between a literal and its negative form (cannot be satisfied at the same time). A flip-flop is used to represent true-false assignment; a constraint is used to ensure that the clause is satisfied; and the consistency need to be maintained because of the nature of boolean function. To get the reduction from 3SAT, our task is to find the "mapping" relation between 3SAT and the reduced problem.

**From 3SAT to 3DM**

A 3SAT CNF $\varphi$ can be reduced to a 3DM $R(\varphi)$, by the following steps:

**Flip-Flops:** For each literal $x$ in $\varphi$, make a gadget with $2k$ triangles, where $k$ is the larger number of occurrence of positive($x$) or negative($\overline{x}$) literal form. This gadget is shown in Figure 21.1, on which the tips of the triangles are labeled $x$ or $\overline{x}$, alternatively. Each triangle represents a value assignment (true or false) which $x$ can take.

**Constraints:** For each clause in $\varphi$, add a pair of boy vertex and girl vertex to the graph. If $x$ appears in this clause, connect this boy and this girl to one of the $x$ vertices in $x$'s flip-flop gadget to form a triangle. If $\overline{x}$ is in the clause, then connect them to the $\overline{x}$ vertex from the flip-flop gadget. For other two literals in this clause($y, \overline{z}$ in the figure), we do the same thing. Figure 21.1 shows how this is done. Thus for the new boy and girl to be matched, one of the literals in the clause must be satisfied. And in this process, there are more pets than boys and girls, so some pets will be left homeless. This can be fixed by adding a boy and a girl for each homeless pet and make a triangle to connect them.

**Consistency:** The topology of the gardet ensures consistency. The construction guarantees that if any $x$ vertex is matched with some vertices outside of the this gadget then all $\overline{x}$ vertex can only be matched by the triangles inside this gadget, and vice versa. Thus the "availability" of a vertex to be matched by an outside vertex corresponds to the truth assignment.

$\varphi$ is satisfiable if and only if $R(\varphi)$ has a solution. This is evident from the way that $G$ is constructed: $R(\varphi)$ preserves all literals, clauses, and consistency of $\varphi$. Thus for a truth assignment that satisfies $\varphi$, we can find a complete match in $G$, which represents a solution to $R(\varphi)$; and vice versa.

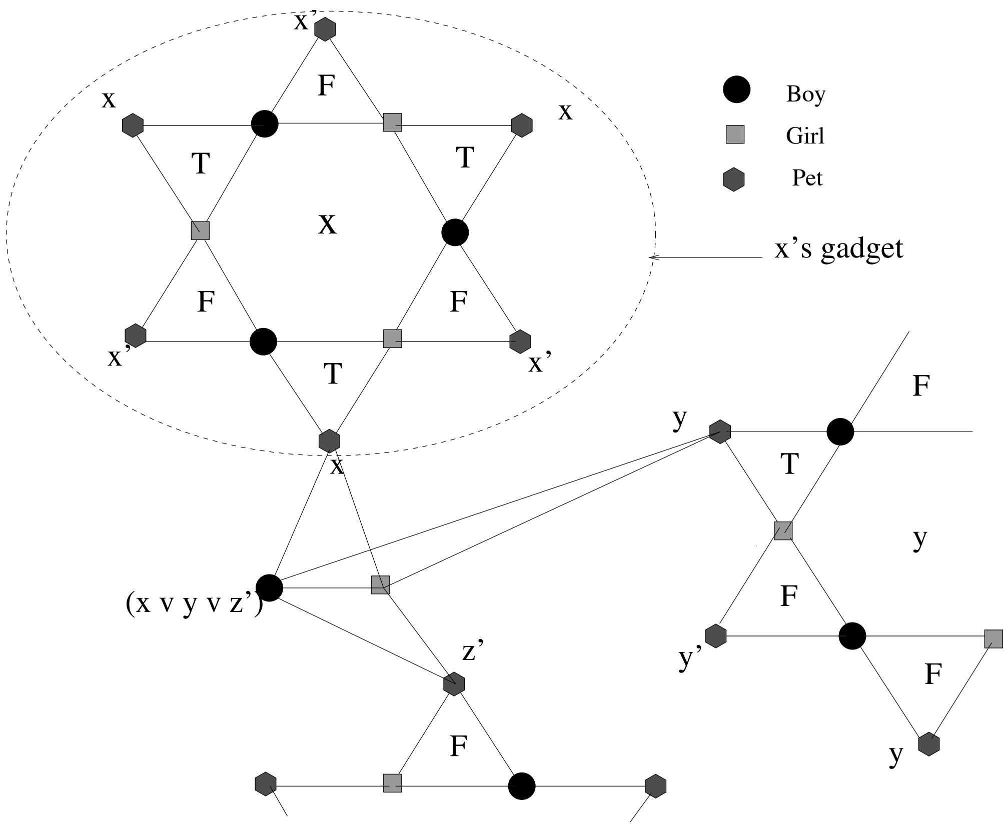


Figure 21.1: 3SAT to 3DM

Obviously 3DM problem is in NP, and it is also NP-hard because 3SAT can be reduced to 3DM. So 3DM is NP-complete.

## 21.2 Knapsack

**Definition 21.3 (Knapsack)** *This is also known as the "subset-sum" problem. Given a set of $n$ numbers $S = \{a_1, a_2, \ldots, a_n\}$ and a number $K$, can we find a subset $S\prime \subseteq S$, such that $\sum_{x \in S\prime} x = K$.*

**Theorem 21.4** *3DM can be reduced to Knapsack.*

### From 3DM to Knapsack

3DM can be reduced to Knapsack as follows: Suppose we have $n$ boys, $n$ girls, and $n$ pets. For each triangle connecting a boy, a girl and a pet, we create three $n$-digit numbers. Each number encodes the "index" of corresponding boy, girl or pet. All digits in that number are set to 0 except the one corresponding to the index of that boy (or girl or pet). For instance, suppose $n = 4$, then boy $No.3$ is represented by the number