Homework Collection format:

- on paper, also for modeling/programming
- please copyedit questions, models, data, command line input and output into file (word, latex, handwritten but neat)

We described a way to find all basic solutions for an LP in standard form.

- These lie in the affine space $Ax=b$ but can be feasible $(x \geq 0)$ or infeasible $(x \ngeq 0)$.
- The basic feasible solutions correspond to vertices.
- Different basis/nonbasis splits can give the same vertex.

$\Rightarrow$ We can find an optimal vertex (if one exists)

Problem: This is not an efficient approach as there may be exponentially many vertices.

Reading: AMPL Chapter 2

Excursion: An Introduction to AMPL

AMPL is a modeling language (matrix generator), not a solver:

- enables the formulation of an LP in a natural language
- translates this language to input for a solver

AMPL links to many commercial and open-source solvers

Cplex, Gurobi, Highs

Example: A Steelmaking Operation

Problem: A steel company must decide how to allocate next week's time on a rolling mill. The mill takes unfinished slabs of steel as input and can produce two products: bands or coils. The mill's products come off the line at different rates and have different profitabilities. Weekly production amounts must be limited to not exceed currently booked orders. How many bands or coils should be produced to bring the greatest total profit?

variables constraints objective

Model Formulation

Input	P	set of products (bands, coils)
	b	hours available on the mill (24·7)
	c_j	profit per ton of product $j \in P$
	u_j	maximum tons of product $j \in P$
	a_j	tons per hour of product $j \in P$

parameters
data

Variables x_j tons of product $j \in P$

→ Formulation

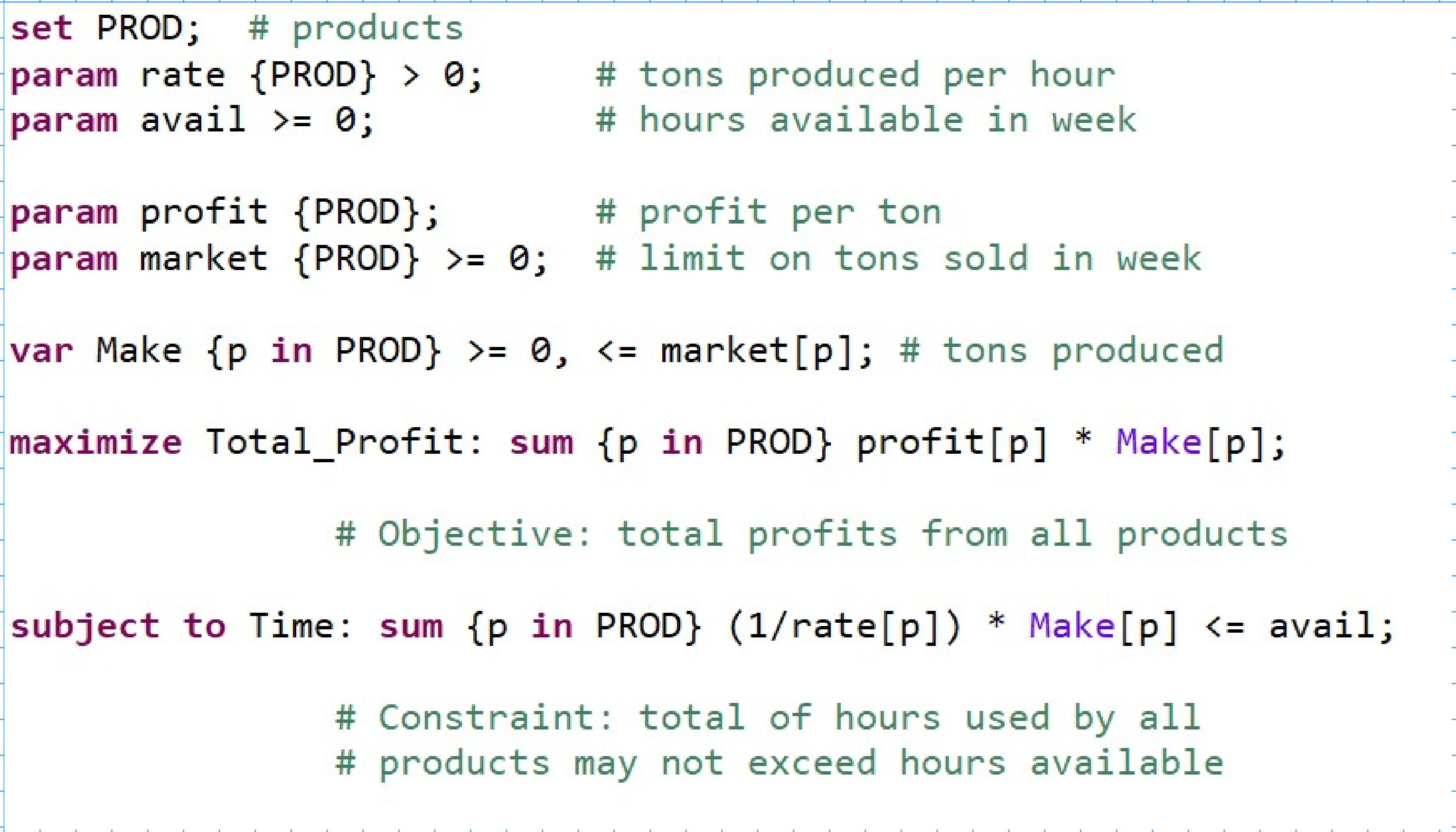
$$\max \sum_{j \in P} c_j \cdot x_j$$

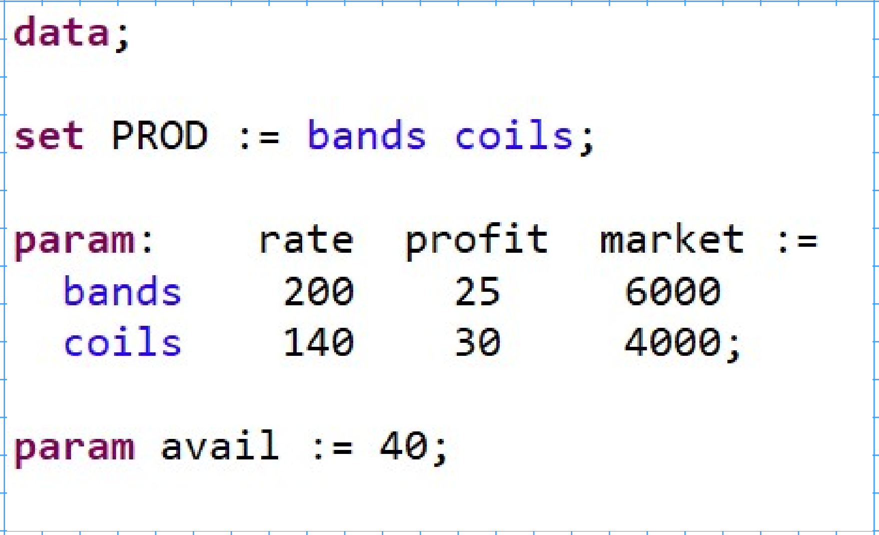
main constraints (translated to $A'x = b'$)

$$\left\{ \begin{array}{lll} \sum_{j \in P} \frac{1}{a_j} x_j \leq b & \\ x_j \leq u_j & \forall j \in P \end{array} \right.$$

domain constraints

$$x_j \geq 0 \quad \forall j \in P$$


```
set PROD;  # products
param rate {PROD} > 0;     # tons produced per hour
param avail >= 0;          # hours available in week

param profit {PROD};       # profit per ton
param market {PROD} >= 0;  # limit on tons sold in week

var Make {p in PROD} >= 0, <= market[p]; # tons produced

maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];

               # Objective: total profits from all products

subject to Time: sum {p in PROD} (1/rate[p]) * Make[p] <= avail;

               # Constraint: total of hours used by all
               # products may not exceed hours available
```


```
data;

set PROD := bands coils;

param:    rate  profit  market :=
  bands    200    25      6000
  coils    140    30      4000;

param avail := 40;
```

Typical Command Line Input

```
model ~ .mod;
data  ~ .dat;
option solver cplex;
solve;
display ~ ;
```

(AMPL Chapter 12)

HW 4 : Exercise 1-2 from AMPL book