# Assignment 1

# 2016-2017

# Constraint Programming (2IMI25)

# The Problem

The problem you are asked to solve in this first assignment comes from the actual practice of a drama society. The complexity of a good model to solve the problem is rather modest, making it a good candidate for a first assignment. The drama society in question wants to stage a play. The play has a series of scenes, each scene having different characters in it, where the issue is that there are so many characters, that they do not have a separate actor available for each character. What you want to find out is how many actors you minimally need and how they should be assigned to characters so that all characters are covered, this while making sure that assignment satisfies a number of constraints.

Once an *actor* plays a certain *character* in a scene for example, he or she needs to play that character in the whole play. Another constraint is that you cannot have two actors together play a character as that will confuse the audience. An actor obviously also cannot play more than one character in the same scene. There are furthermore a couple of *leading characters* and the actors assigned to those characters cannot play any other character as that would again confuse the audience. There are also parts for males that can only be played by men, parts for females that can only be played by women, etc. Another constraint is that to allow people to change costume, an actor cannot play one character in one scene and another in the scene that is directly next, i.e., at least one scene needs to be in between any actor playing two different characters. A final constraint is that no actor can be assigned to more than a given *maximal number of characters*, this as assigning too many characters to an actor will again confuse the audience.

As said, we are looking for an assignment of actors to characters that satisfies the above given constraints and that minimizes the number of required actors.

# The Data

Please find included the file SmallPlay.dat that provides a small instance that you can use to start developing a solution:

CharacterTypes = {"Male", "Female"};

Characters = {

<"John","Male">,

<"Alex","Male">,

<"Father","Male">,

<"Young Man","Male">,

<"Stacy","Female">,

<"Mother","Female">,

<"Grandma","Female">,

<"Woman #1","Female">,

<"Lawyer","Female">

};

LeadingCharacters = {"Stacy"};

maxNrOfCharacters = 2;

Scenes = {

<"Act I Opening",{"Father","Mother","Grandma"}>,

<"Theatre Lobby",{"Stacy", "John"}>,

<"An Audition",{"Woman #1","Alex","John","Lawyer", "Stacy"}>,

<"Stacy's Apartment",{"Stacy", "Young Man"}>

};

CharacterTypes gives the set of character types there exist in the play. Here there are two character types: male and female.

Characters gives the set of characters in the play, consisting of their names together with the character type of the character.

LeadingCharacters gives the set of leading characters.

maxNrOfCharacters gives the maximal number of characters any one actor can be assigned to.

Scenes gives the set of scenes in the play, consisting of the name or location of the scene and the set of characters in that scene. The scenes are played in the sequence given in this set. A name/location can be used more than once.

Note that you are explicitly not allowed to rename any of the data in the .dat files. More generally, you are ***not allowed to change anything in the provided .dat files*** as that may prevent me from doing the tests I will be doing on your model. Your model should thus run on these files as they are. If that’s not the case, your work will not be graded.

# Reporting Solutions

To report the solution you find I want you to fill in the following data structures with the result of your model. *You need to put this code after the “subject to” block as you will be using the values for the variables that your model found*.

{Character} CharactersPlayedByActor[i in 0..NrOfActorsNeeded-1] =

fill in from your decision variables.

int nrOfActorsOfType[ct in CharacterTypes] =

fill in from your decision variables.

With those structures you ***have to use the following code to generate the output*** I want your model to generate:

execute {

writeln("Actors needed: ", NrOfActorsNeeded);

for(var ct in CharacterTypes) {

writeln(ct, " needed: ", nrOfActorsOfType[ct]);

}

for(var i=0; i<NrOfActorsNeeded; i++) {

writeln("Actor ", i, " plays ", CharactersPlayedByActor[i]);

}

}

For SmallPlay.dat this is the output my model provides using the code above:

Actors needed: 6

Male needed: 3

Female needed: 3

Actor 0 plays {<"Stacy" "Female">}

Actor 1 plays {<"John" "Male">}

Actor 2 plays {<"Mother" "Female"> <"Woman #1" "Female">}

Actor 3 plays {<"Alex" "Male"> <"Father" "Male">}

Actor 4 plays {<"Young Man" "Male">}

Actor 5 plays {<"Grandma" "Female"> <"Lawyer" "Female">}

# The Tests

Besides SmallPlay.dat, please also find included the data files

* Actual.dat (coming from an actual play)
* ActualMax4.dat
* 3Types.dat
* 3TypesTwice.dat

These files provide the 5 instances in LTI (see “Assignment CP 2016-2017 General.doc”). Your model should be able to solve these 5 instances fairly efficiently. For all 5 you should define a time limit of 5 seconds. Also, always use sequential search (number of workers = 1) in all tests. The following code, to be included before the “subject to” block, reflects these two points:

execute {

cp.param.Workers = 1;

cp.param.TimeLimit = 5;

…

Proving optimality within this time limit is possible, for all instances in LTI you can in fact be quite a bit faster. If you use more than 5 seconds or more than 100K fails, you should do some more thinking as the instances can be solved considerably faster/requiring considerably less fails. If you believe you have a machine that is so slow that this time limit prohibits any decent performance, you can contact me to discuss. Note that the time limit is about the “Time spent in solve” and you can find that time in the "Engine log" tab.

Finally I want to point you to the file “Assignment CP 2016-2017 General.doc” that provides the general organization of the assignments as well as some guidelines I want you to follow.

Have fun!

Wim Nuijten.