A*-GAC, A General Constraint-Satisfaction Problem Solver

Kristian Våge and Bjørn Bråthen

October 4, 2015

1 Central aspects of your A*-GAC program

1.1 The agenda and how it is managed

The agenda is awesome. list and Heapq. . .

1.2 Generality of the program

Constraint Network, VIs and CIs

1.3 Evaluation procedure of the GAC

```python
""" Creates an anonymous lambda function from expression string """

def make_function(variable_names, expression, environment=globals()):
    """ Returns lambda given by expression and variables """
    arguments = ','.join([variable for variable in variable_names])
    statement = '(lambda {}: {})'.format(arguments, expression)

    return eval(statement, environment) # pylint: disable=eval-used
```