1 Car-park Specification

1.1 The Specification

MODULE *carpark*

EXTENDS *Naturals*, *FiniteSets*
CONSTANT *Capacity*, *Cars*
VARIABLE *carpark*

$TypeInvariant \triangleq carpark \in \text{SUBSET } Cars$ Car park contains some set of cars

$Safety \triangleq Cardinality(carpark) \leq Capacity$ Car park is not over capacity

$Init \triangleq carpark = \{\}$ an empty carpark

$Enters(car) \triangleq$
$\land\ Cardinality(carpark) < Capacity$ is there space ?
$\land\ car \notin carpark$ a car in the carpark cant enter
$\land\ carpark' = carpark \cup \{car\}$ A car enters the carpark

$Leaves(car) \triangleq$
$\land\ car \in carpark$ only cars in the carpark can leave
$\land\ carpark' = carpark \setminus \{car\}$

$Next \triangleq$
$\lor\ \exists\, car \in Cars : Enters(car)$
$\lor\ \exists\, car \in carpark : Leaves(car)$

2 The Model

The car-park system has been run with two models, of different scales.

2.1 Model 1 Overview

Model 1 is a small model to quickly prove correctness of the specification.

The Behaviour specification is an *initial predicate and next-state relation*. These are filled in automatically if there is an *Init* and *Next* predicates.

The Model values assigned to declared constants

The set of cars to use is $Cars \leftarrow 1..10$

The capacity of the car-park is set $Capacity \leftarrow 5$

2.2 Checks and verifications

Invariants Two invariants are checked

The type invariant *TypeInvariant*

The safety case the *Safety* predicate.

2.3 Results

A summary of the results

Statistics a summaries of the actions and number of states found.

States found	5121
Distinct states	638

Action	Location	States Found	**Distinct states**
Init	Line 11	1	1
Enters	Line 13	2560	637
Next	Line 24	2560	0

2.4 Model 2 Overview

Model 2 is a large model to show how quickly states can build up. Apart from the model parameters, all the tests are the same.

The Model values assigned to declared constants

The set of cars to use is $Cars \leftarrow 1..100$

The capacity of the car-park is set $Capacity \leftarrow 10$

2.5 Results

A summary of the results

Statistics a summaries of the actions and number of states found. TLC (the model checker) took 27 minutes to find the number of states shown. After a further 20-hours of runtime the model-checking had not completed and was terminated. The remaining queue-size shows that there are still large numbers of states to add to the statistics.

States found	1 375 477 467
Distinct states	285 379 380
Queue size	271 624 604

Action	Location	States Found	**Distinct states**
Init	Line 11	1	1
Enters	Line 13	1 310 963 481	284 444 823
Next	Line 24	64 513 985	934 556

2.6 Discussion

2.6.1 Model description

The state of the car-park is the set of cars it contains. For the purposes of the specification we are not concerned with how the cars are arranged in the car-park or the order in which they enter and leave.

So a *set* is appropriate to model the state. Using a finite-set allows us to count the size of the set; how many cars are in the car-park.

The initial conditions are that the car-park is empty.
(The empty set {})

The type invariant is that the set modelling the car-park is some subset of the set of Cars in the model.

The safety invariant Is that the size of the set of cars in the car-park should not exceed the capacity. As the state-variable is a finite-set, it could be argued that this is part of the type-invariant.

The enter operation has a guard condition that there are spaces available, and that a car is not already in the car-park. The state is updated by adding the car to the set of cars in the car-park.

The exit operation is guarded by the condition that a car must be in the car-park to exit. The state is updated removing the car from the car-park

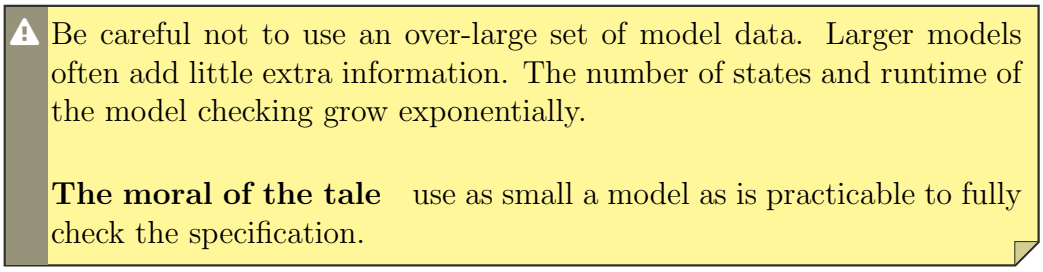
The Next relation is that either there is a car that can enter the car-park, or that there is a car in the car-park that can exit.

2.6.2 Interpretation of results

The specification makes no assumptions about the distribution of cars in the car-park, nor the order they enter and leave.

Model 1 is small enough to run in a reasonable amount of time, and is sufficient to verify the specification.

Model 2 generates a huge number of states and consequently requires a long time to run. The results from it add nothing to the verification of the model.



Be careful not to use an over-large set of model data. Larger models often add little extra information. The number of states and runtime of the model checking grow exponentially.

The moral of the tale use as small a model as is practicable to fully check the specification.