# AN AUTOMATON FOR ROAD TRANSPORT LEGISLATION

Moritz Müller, Raül Espejo Boix and Joost J. Joosten

## EUROPEAN ROAD TRANSPORT REGULATION No. 561/2006

Regulation 561 concerns activity sequences of truck drivers recorded by tachographs. Formally: finite words with letters $d, r, w$ representing activities **driving**, **resting** and **other work** per minute. E.g. $dddwwrrr$ means 3 minutes driving, followed by 2 minutes other work and 3 minutes resting. In practice, these words are very long. One needs an **algorithm** that checks legality.

Regulation 561 is a complex set of laws. Depending on accumulated durations of e.g. driving, it requires various kinds of resting periods, namely **breaks**, **daily rest periods** and **weekly rest periods**, each of which can be **regular** or **reduced**. E.g. Article 8.6 requires:

> In any two consecutive weeks a driver shall take at least two regular weekly rest periods, or one regular weekly rest period and one reduced weekly rest period of at least 24 hours. However, the reduction shall be compensated by an equivalent period of rest taken en bloc before the end of the third week following the week in question.

## STOPWATCH AUTOMATA

A **stopwatch automaton** is given by finitely many **states**, **transitions** and **stopwatches**. At each time-point of a computation the automaton resides in a state. It can either stay in this state or take a transition and switch to another state instantaneously. At each time-point the stopwatches show values, up to some **bound**. In a given state each stopwatch is **active** or not. When staying in a state the active stopwatches increment their value, the unactive ones keep their value.

Each transition has a guard and an action. The **guard** specifies a property of stopwatch values that must be met for the transition to be taken, e.g. certain upper or lower bounds of stopwatch values. The **action** updates stopwatch values, e.g. resets some to value 0. States are labeled by $d, w$ or $r$. The automaton **reads** the label for each time unit spent in the state.

The automaton **accepts** a word $w$ if there is a computation reading $w$ and leads from the state *start* to the state *accept*.

## CONSTRUCTION

We construct a stopwatch automaton that accepts a word if and only if the represented activity sequence is legal according to Regulation 561.

### ARCHITECTURE

The automaton uses the 11 states shown in the figure plus the state *start*.

Letters labeling states are shown in yellow. The label of *week* and *accept* do not matter.

Below each state there is the list of stopwatches active in it. Their bounds are

$x_{start}$: 1   $x_{cd}$: 271   $x_{week}$: 10081
$x_{break}$: 540   $x_{day}$: 1440   $x_{ww}$: 3601
$x_{dr}$: 660   $x_{dd}$: 601   $x_{dw}$: 3361
$x_{wr}$: 2700   $x_{cr}$: 1260

Additionally, the automaton operates 16 bits, 4 counters and 3 registers – these are stopwatches that are nowhere active.

The automaton has 102 transitions. An arrow between states indicates possibly many transitions. An arrow between boxes indicates transitions between all states within the boxes. The figure omits transitions from state *start* to all other states.

The table below lists the transitions excluding those leaving *start*, *compensate1* and *compensate2*.
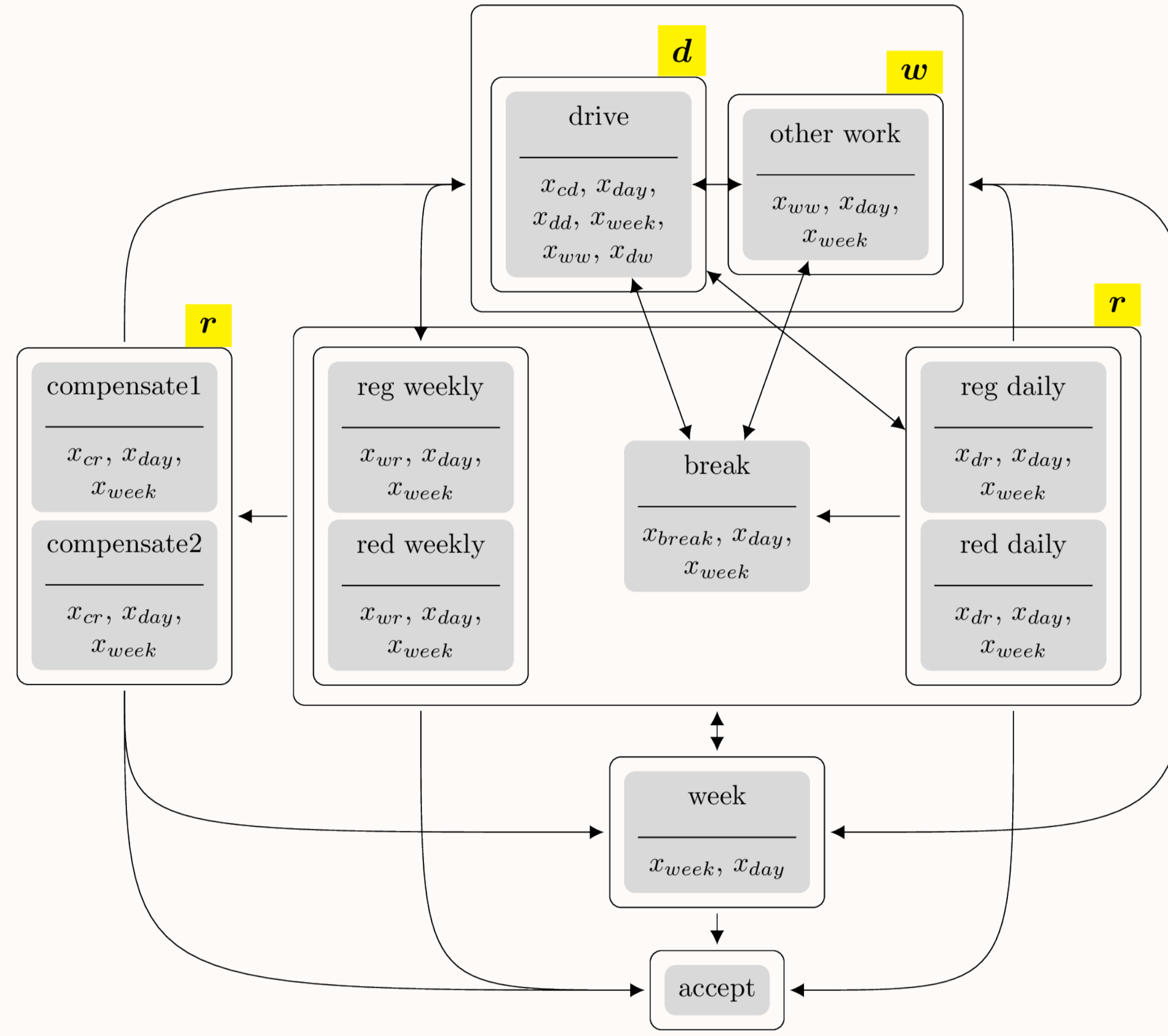


## TRANSITIONS

There is a transition from *break* to *red. daily*. Its guard checks that the stopwatch $x_{day}$ has value at most $15h = 900$, that the counter $c_{rd}$ has value at most 3 and that the bit $b_{rb}$ has value 0. Its action resets the stopwatch $x_{cd}$ to 0 and increments the counter $c_{cd}$ by 1.