max planck institut
informatik

Extending an Isabelle Formalisation of CDCL to Optimising CDCL

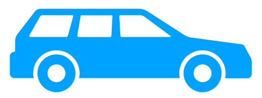
Mathias Fleury

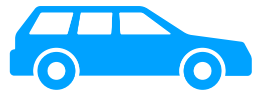
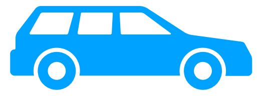
joint work with Christoph Weidenbach

SAARLAND UNIVERSITY
SAARBRÜCKEN GRADUATE SCHOOL OF COMPUTER SCIENCE

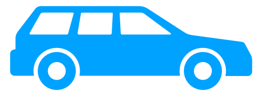
SIC Saarland Informatics Campus

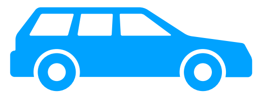
Let’s find a model with minimal weight

 $\vee$ 

 10 $\neg$  4

 20 $\neg$  13

Optimal partial model: 

Optimal total model: $\neg$  

max planck institut informatik

How reliable is the theory?

Conference version

Branch and Bound for Boolean Optimization and
the Generation of Optimality Certificates
Javier Larrosa, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell (SAT 2009)

A literal l is *true* in I if $l \in I$, *false* in I if $\neg l \in I$, and *undefined* in I otherwise.

A clause set S is true in I if all its clauses are true in I. Then I is called a *model* of S, and we write $I \models S$ (and similarly if a literal or clause is true in I).

How reliable is the theory?

Conference version

Branch and Bound for Boolean Optimization and
the Generation of Optimality Certificates
Javier Larrosa, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell (SAT 2009)

A literal l is *true* in I if $l \in I$, *false* in I if $\neg l \in I$, and *undefined* in I otherwise.

A clause set S is true in I if all its clauses are true in I. Then I is called a *model* of S, and we write $I \models S$ (and similarly if a literal or clause is true in I).

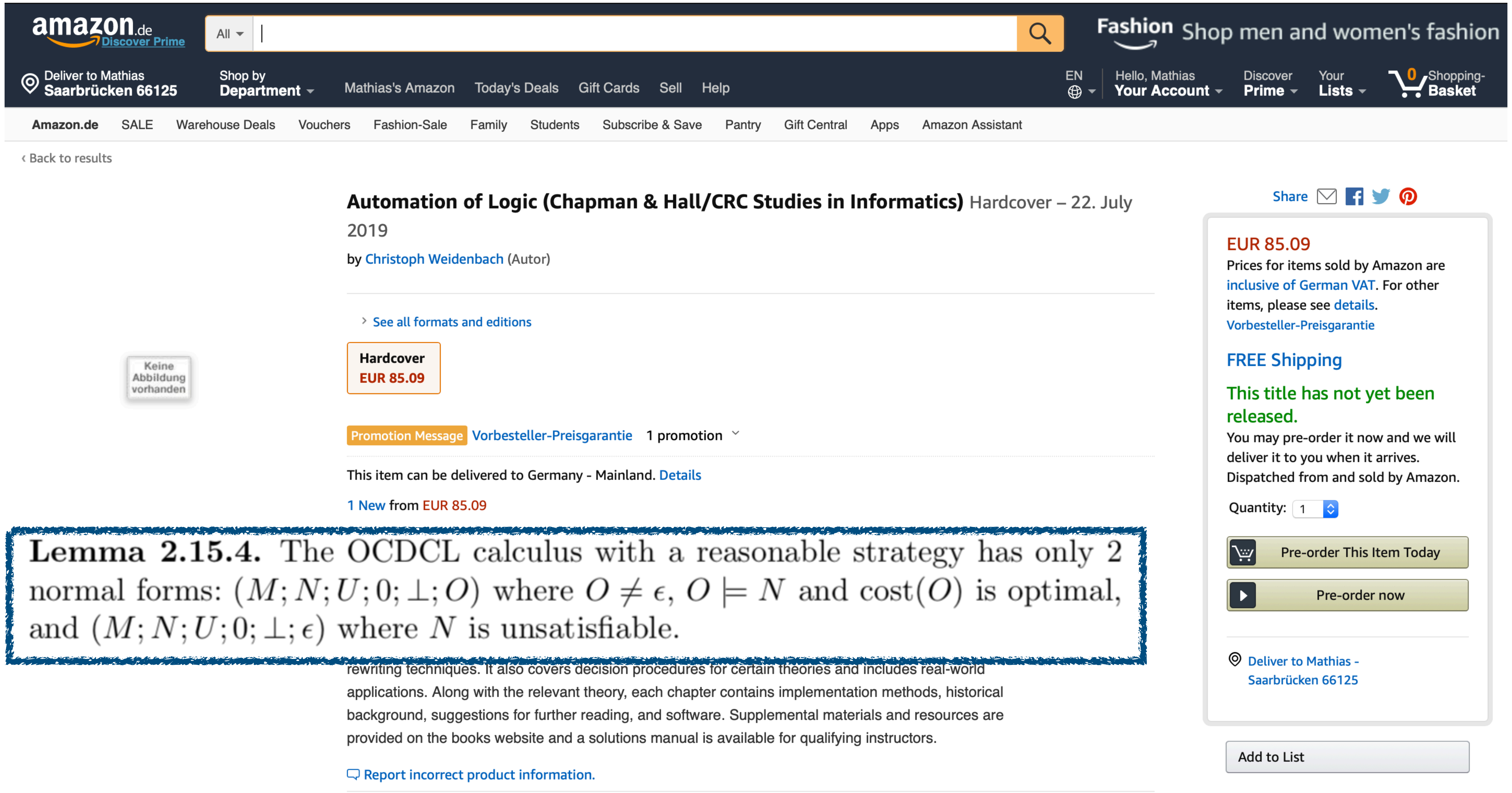
Journal version

A Framework for Certified Boolean Branch-and-Bound Optimization
Javier Larrosa, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell (JAR 2011)

literals of a clause C are false in I. A clause set S is true in I if all its clauses are true in I; if I is also total, then I is called a *total model* of S, and we write $I \models S$.

max planck institut
informatik

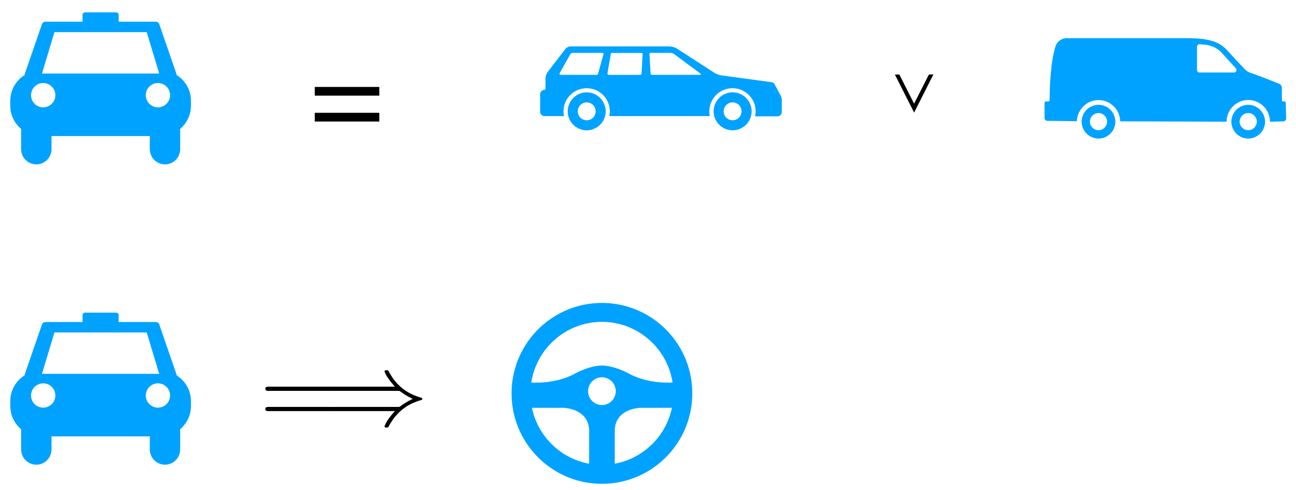
How reliable is the theory?

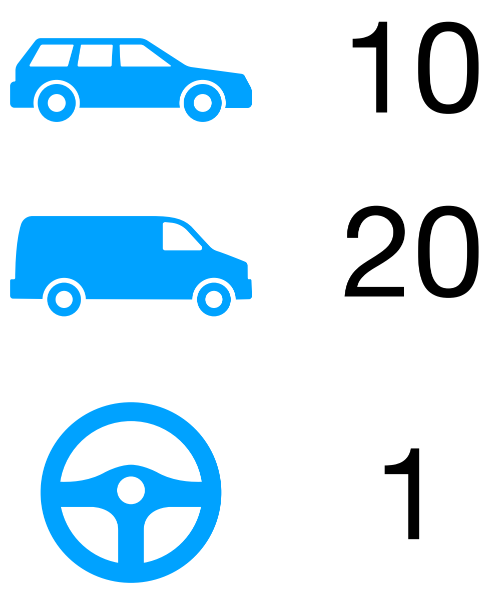


Lemma 2.15.4. The OCDCL calculus with a reasonable strategy has only 2 normal forms: $(M; N; U; 0; \bot; O)$ where $O \neq \epsilon$, $O \models N$ and $\text{cost}(O)$ is optimal, and $(M; N; U; 0; \bot; \epsilon)$ where N is unsatisfiable.

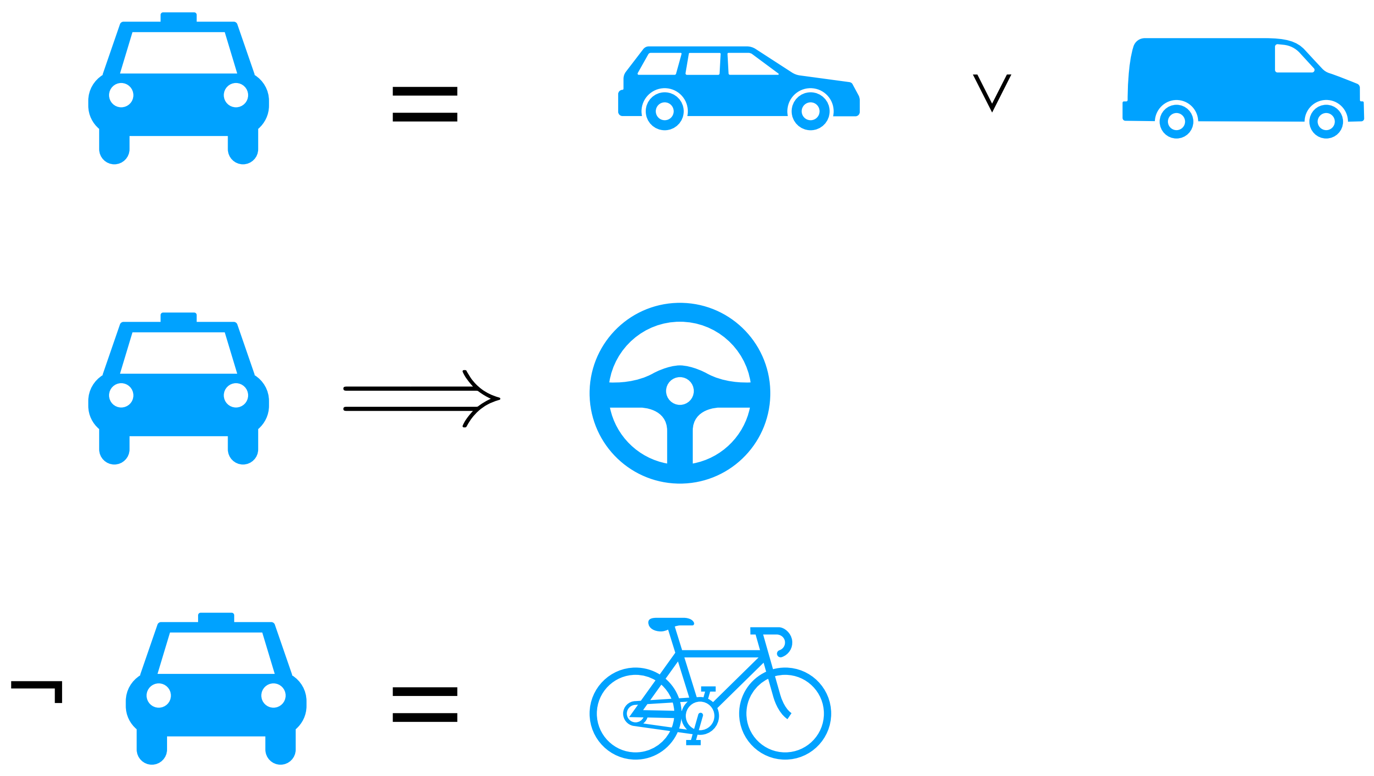
max planck institut
informatik

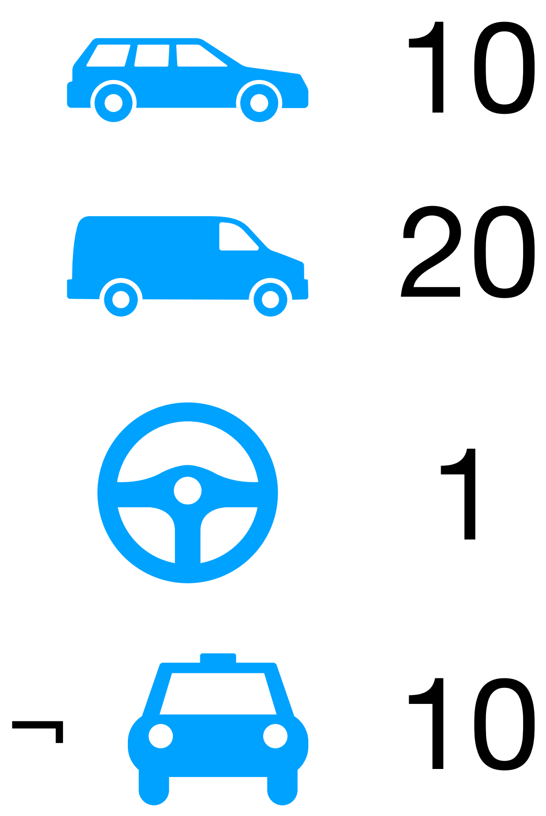
Let's optimise our problem





Let’s optimise our problem

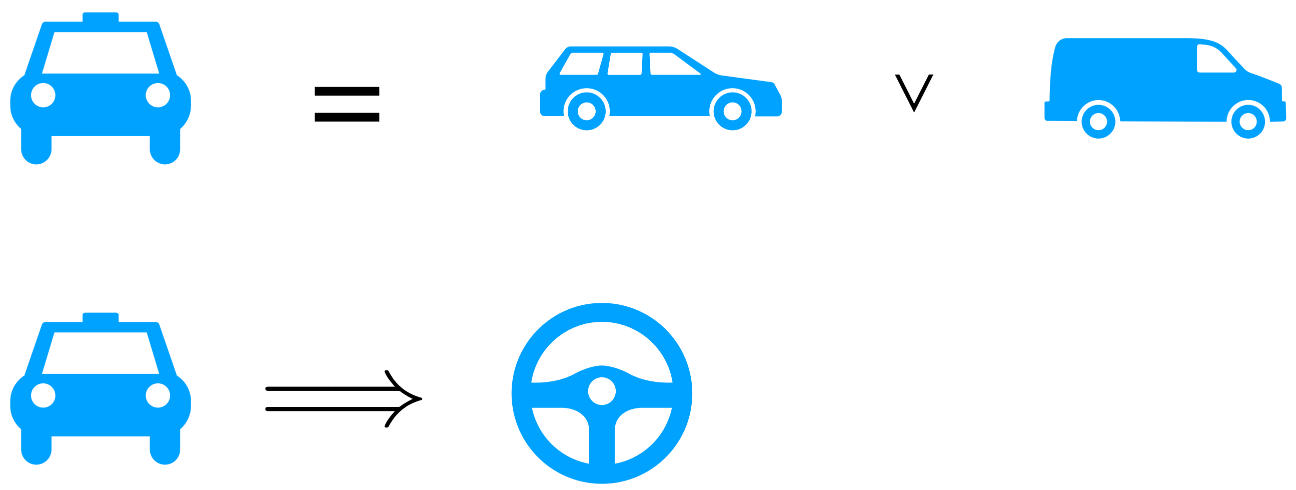


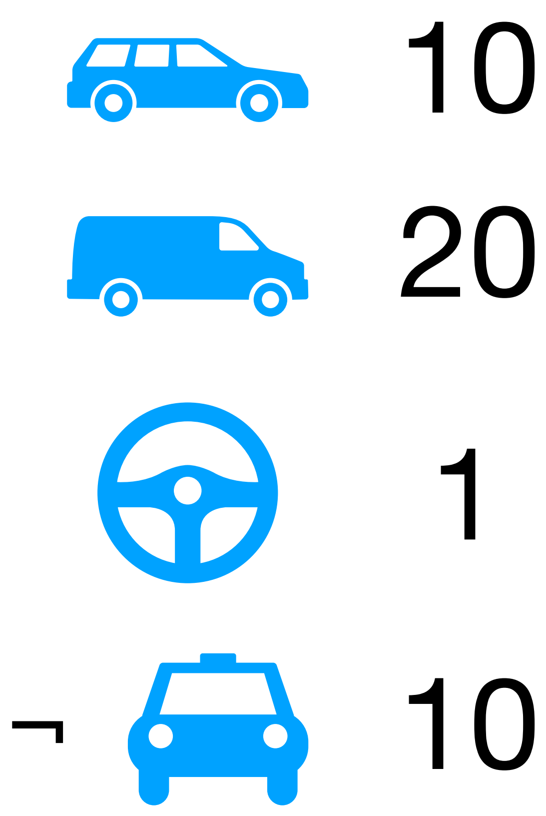


Let's optimise our problem

OCDCL = CDCL + identify better models
+ conflicts based on weights

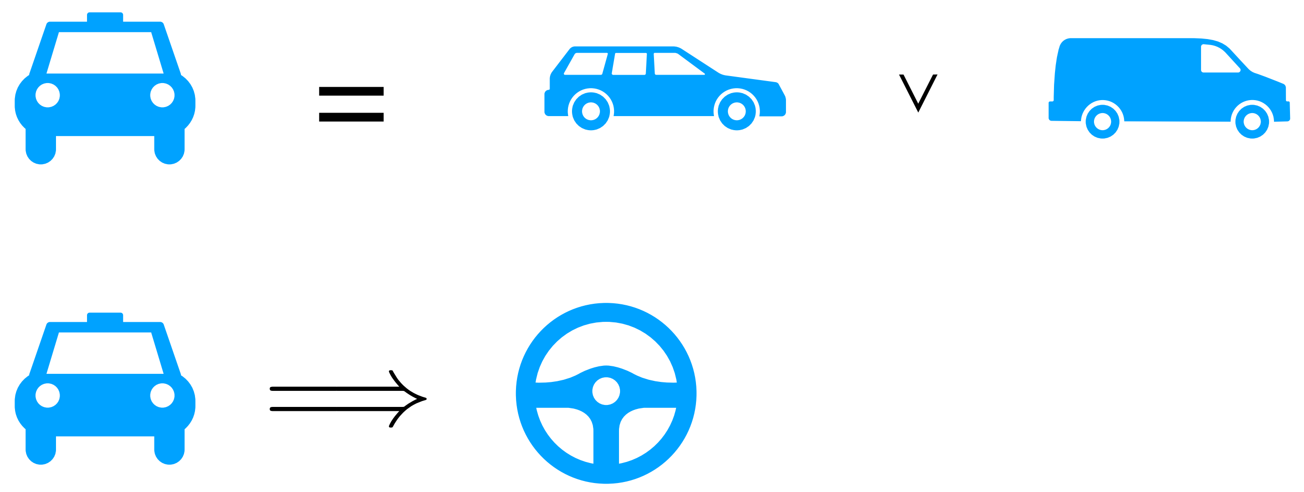
Let’s optimise our problem

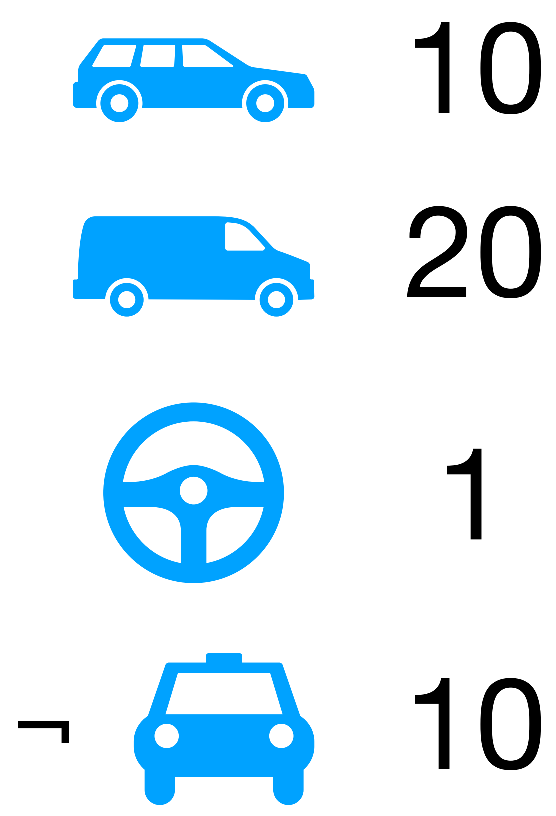




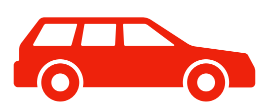
Optimal model

Let’s optimise our problem

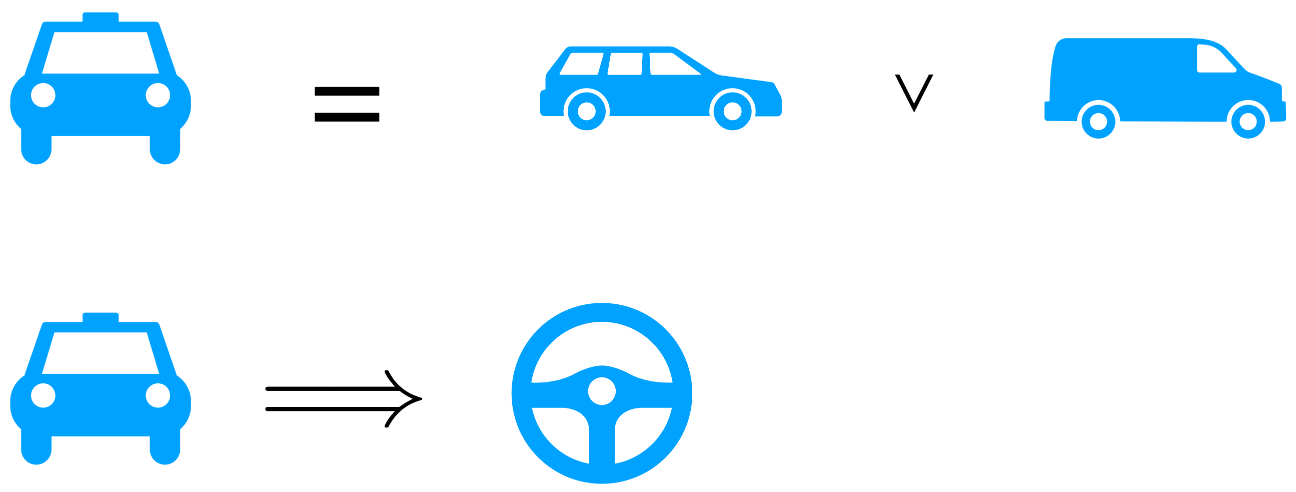


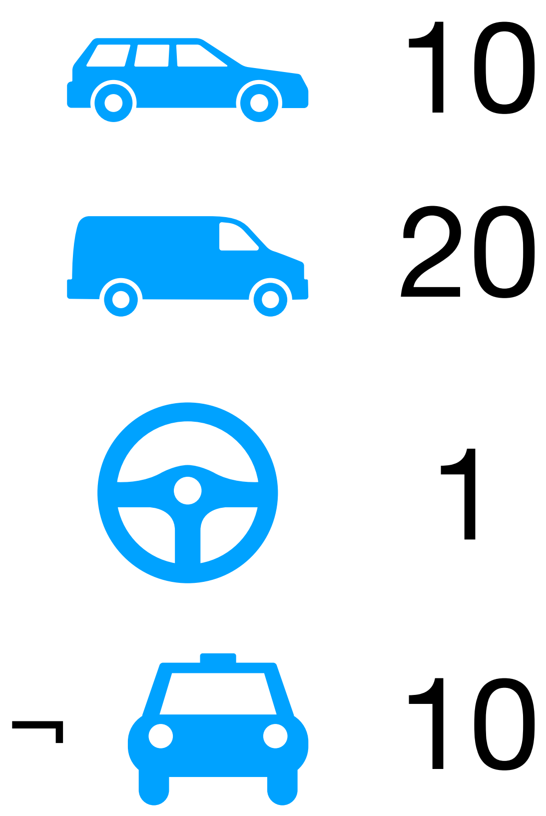


Optimal model



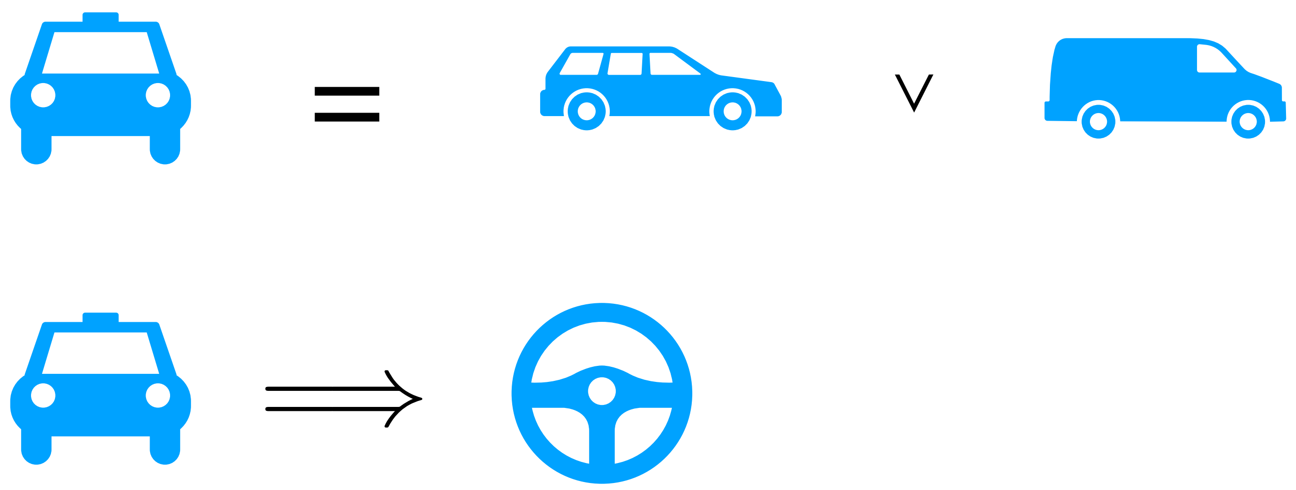
Let’s optimise our problem

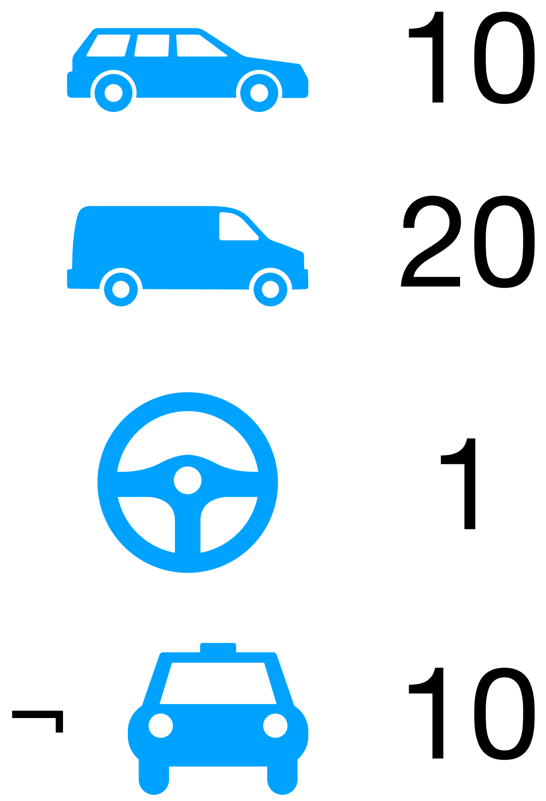




Optimal model

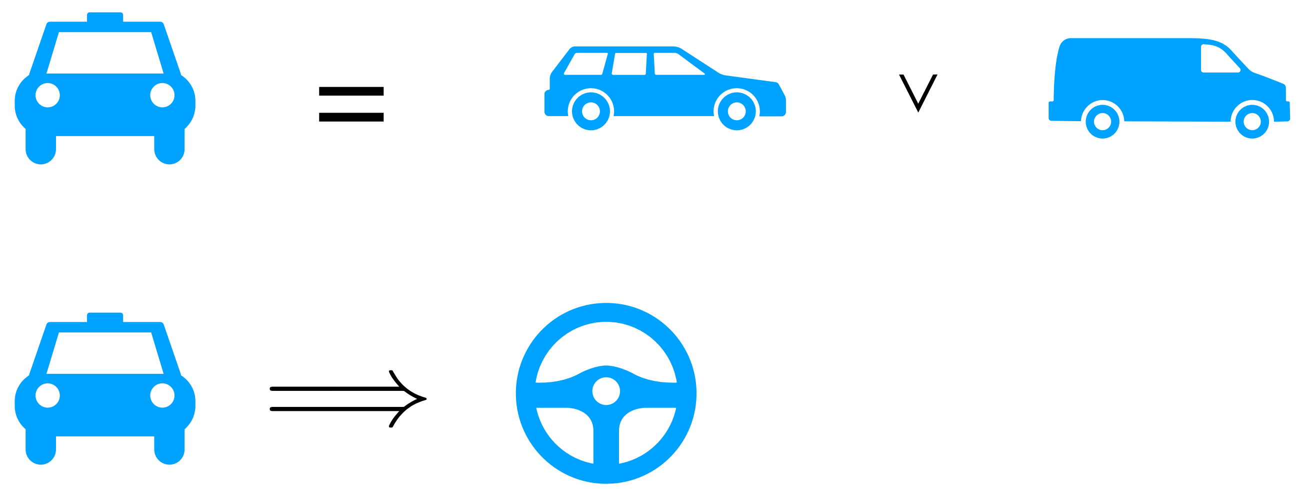
Let’s optimise our problem

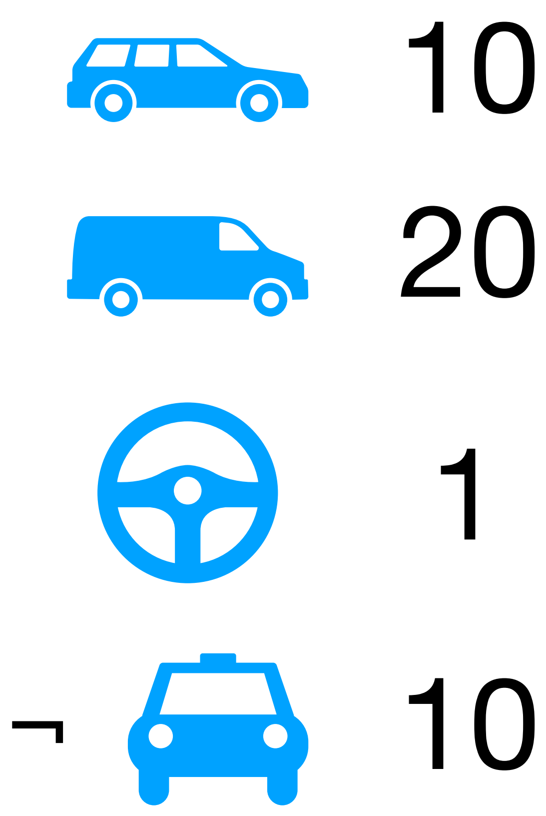




Optimal model

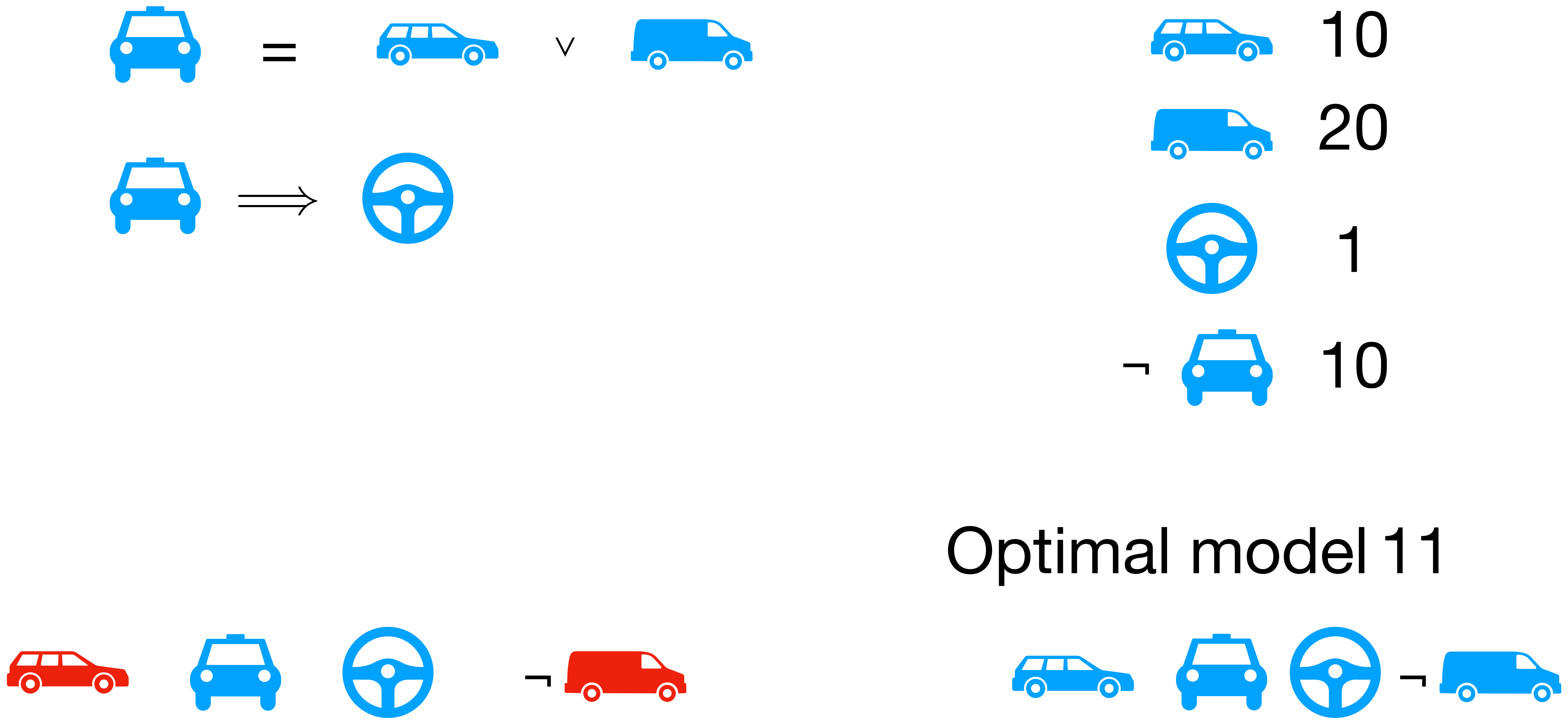
Let's optimise our problem



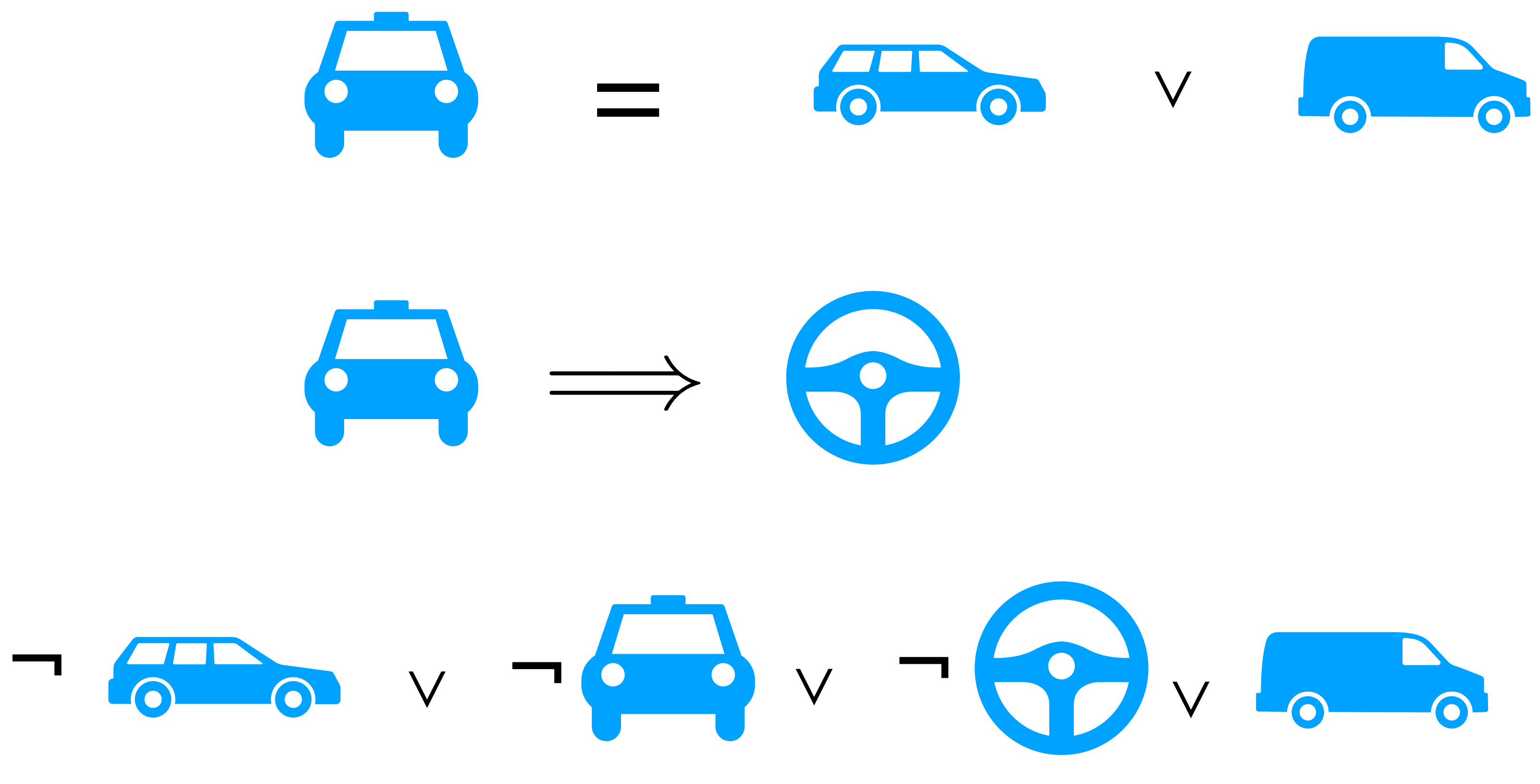


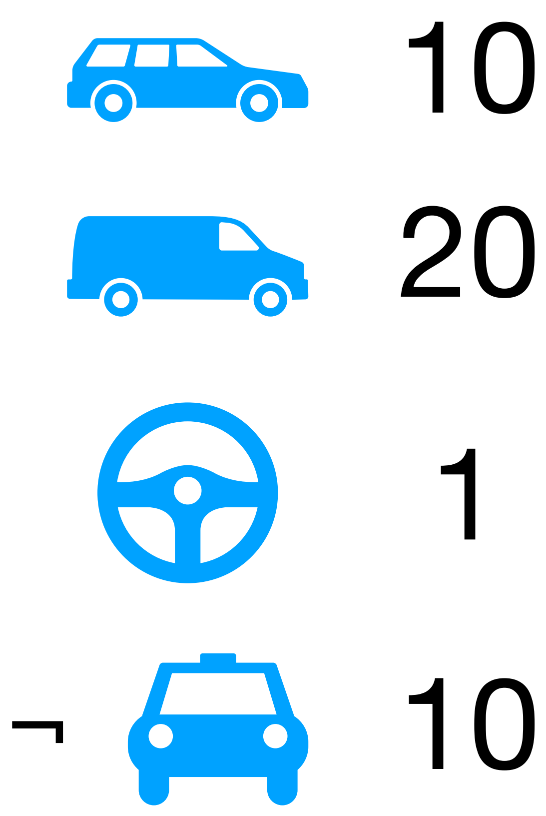
Optimal model

Let’s optimise our problem

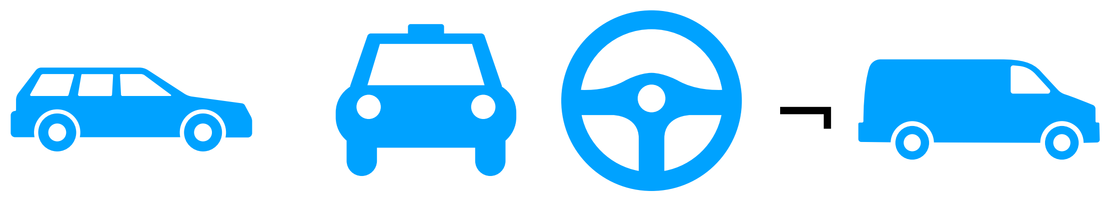


Let’s optimise our problem

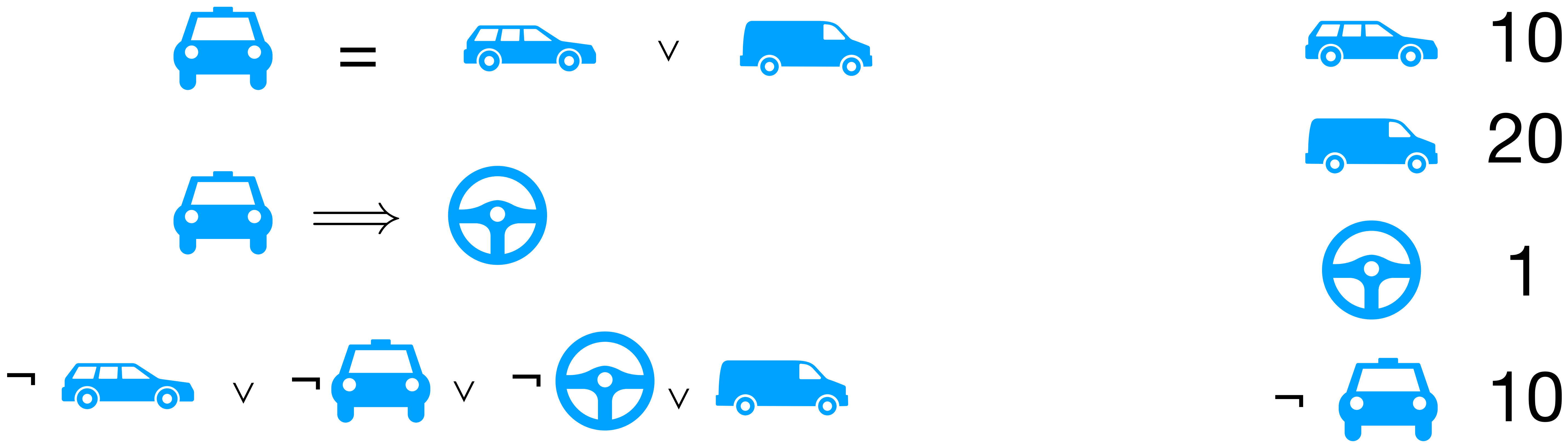




Optimal model 11

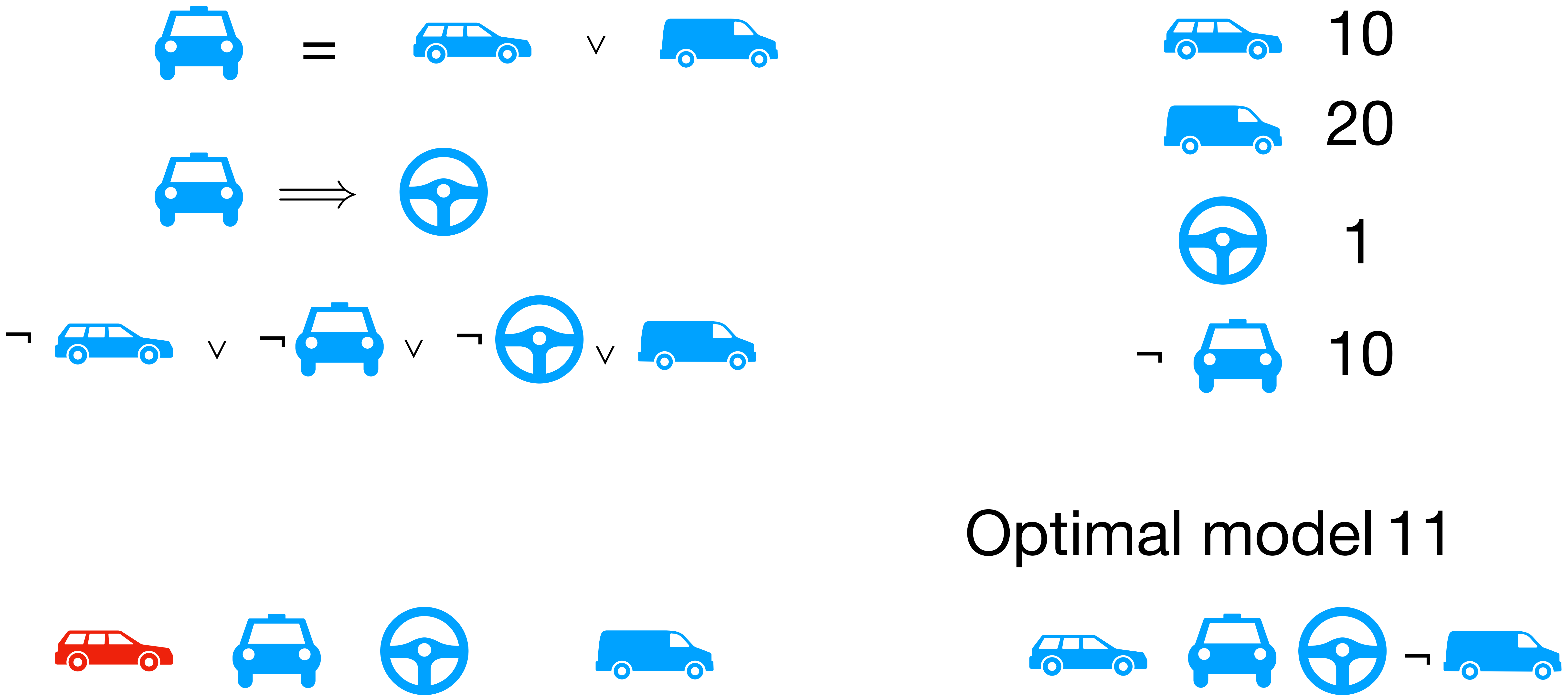


Let’s optimise our problem

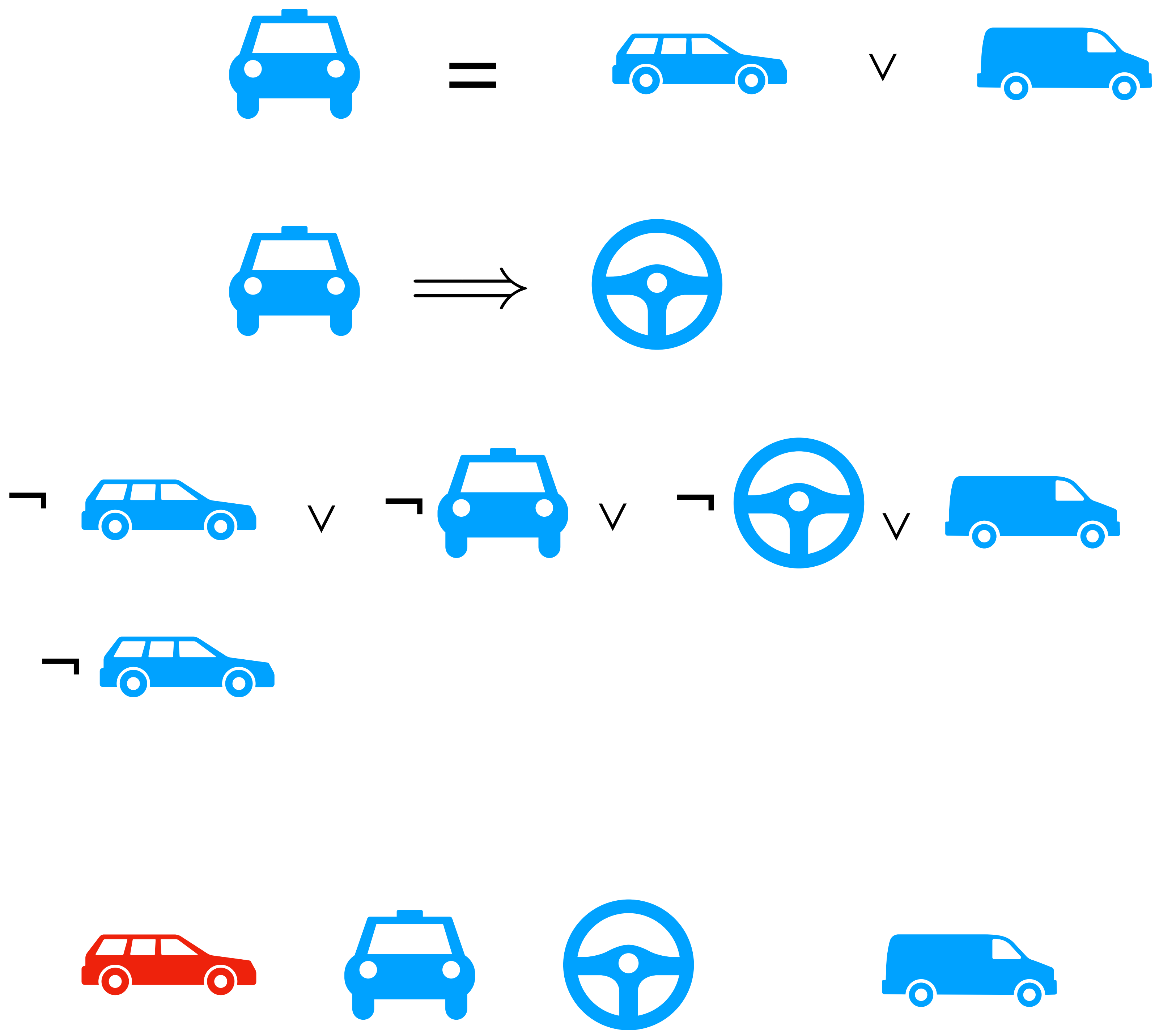


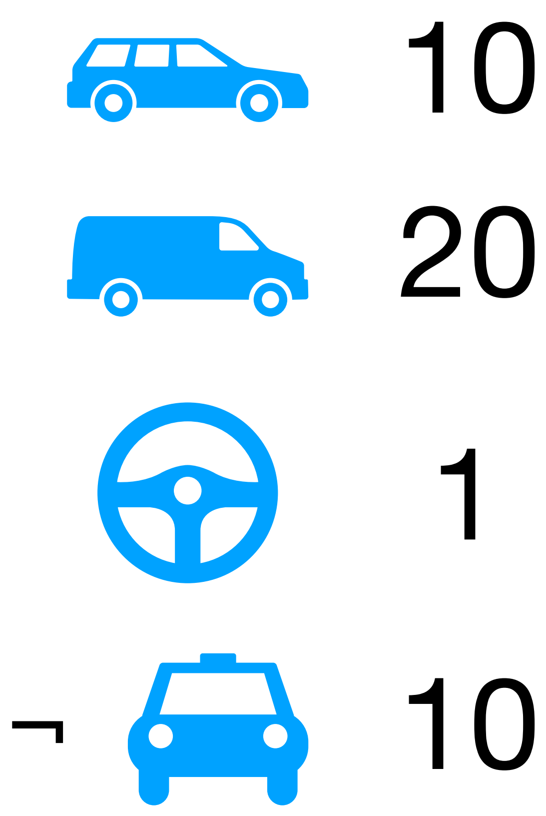
Optimal model 11

Let’s optimise our problem

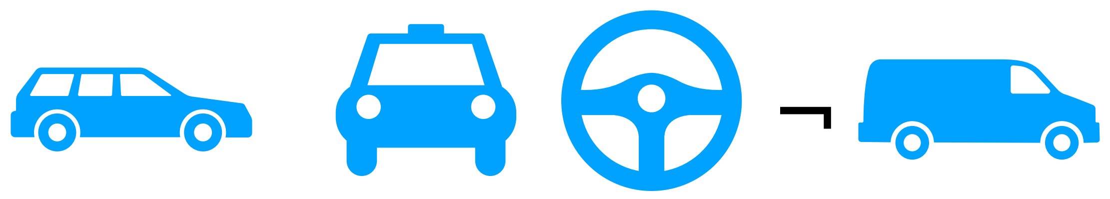


Let’s optimise our problem

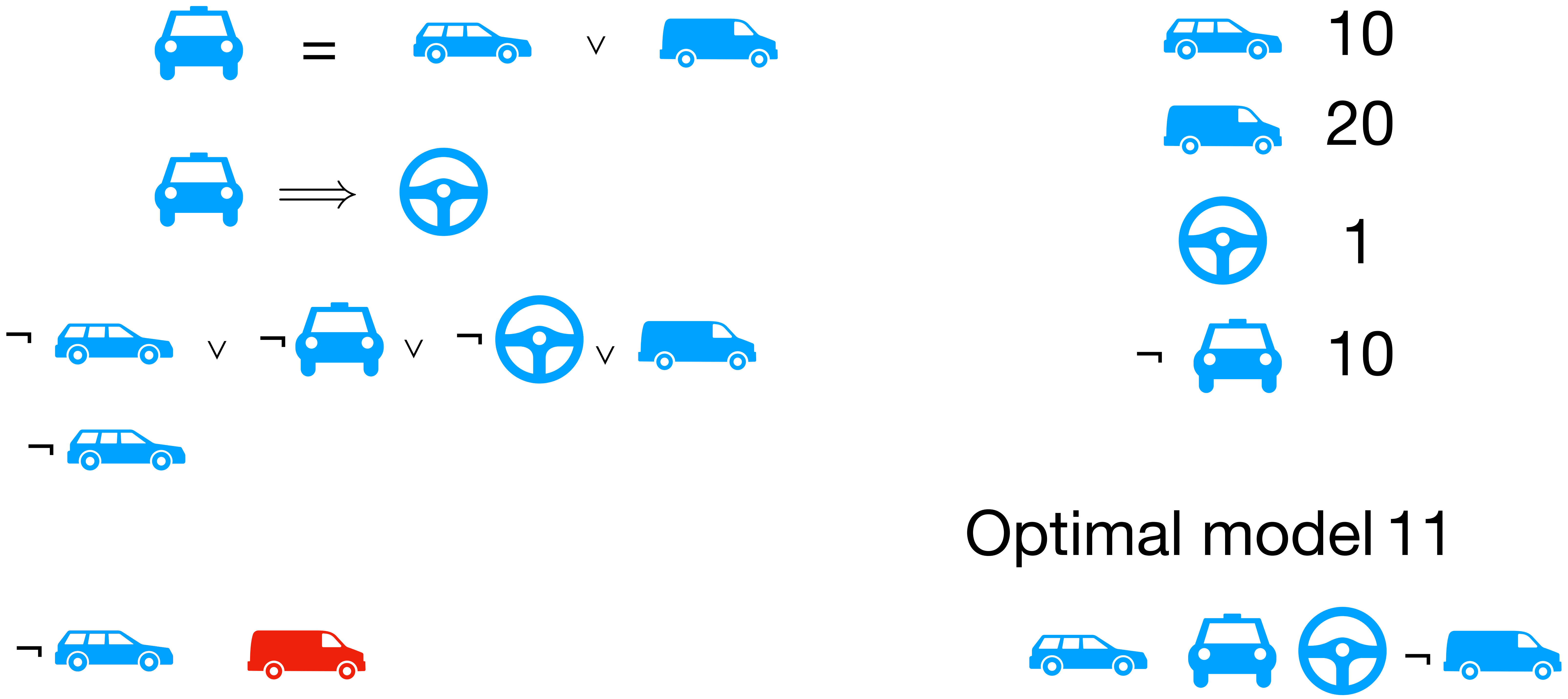




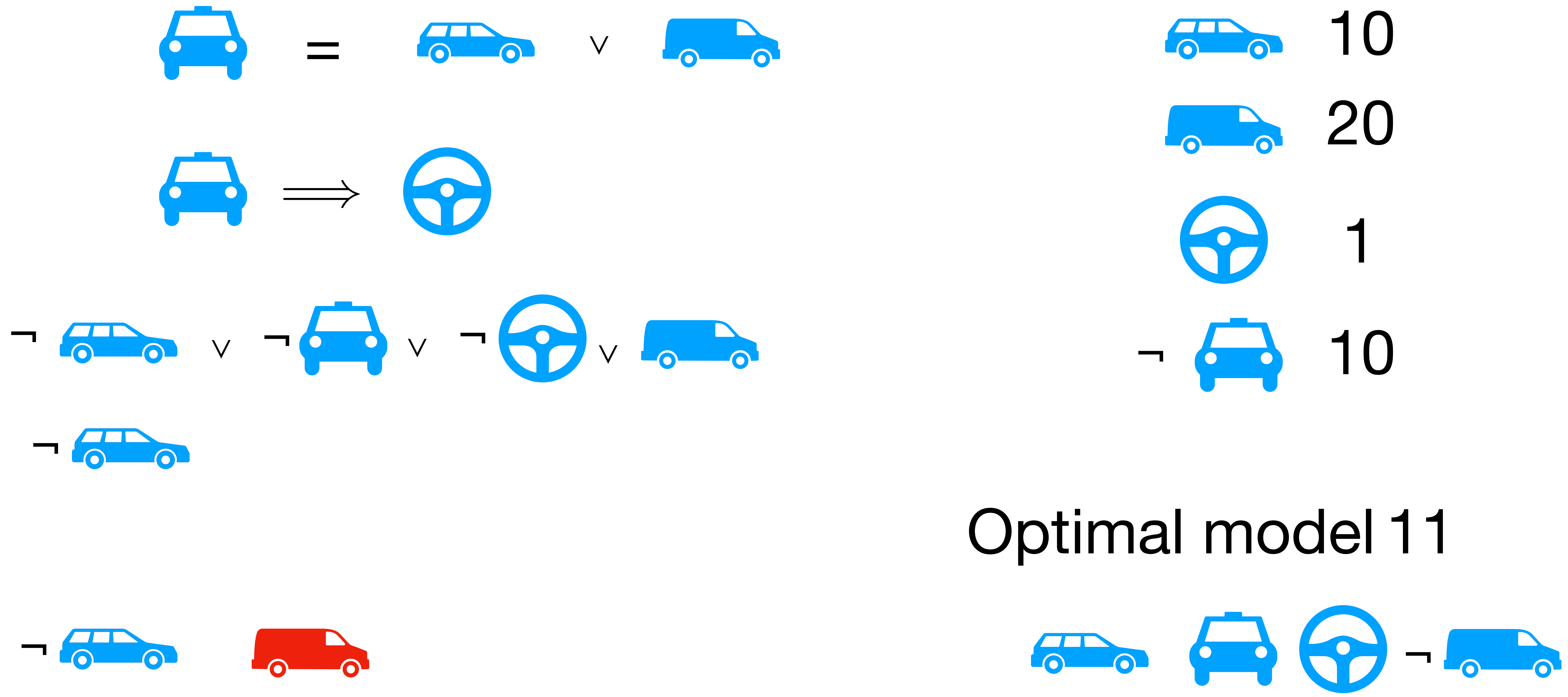
Optimal model 11



Let’s optimise our problem



Let’s optimise our problem



How lazy do you like your formalisation?

Christoph's view:

$OCDCL_W$ = CDCL + improve + conflict rules

copy-paste of proofs

How lazy do you like your formalisation?

Christoph's view:

$OCDCL_W$ = CDCL + improve + conflict rules

copy-paste of proofs

My first idea:

OCDCL = CDCL + improve +
{-M. cost M ≥ min_cost}

reuse CDCL proofs

How lazy do you like your formalisation?

My first idea:

OCDCL = CDCL + improve +
$\{\neg M.\ \text{cost}\ M \geq \text{min\_cost}\}$

reuse CDCL proofs

My second idea:

CDCL_{bnb} = CDCL + improve +
$\mathcal{T}$(min_cost)

reuse CDCL proofs

max planck institut
informatik

How lazy do you like your formalisation?

$CDCL_{bnb}$ = CDCL + improve + $\mathcal{T}$(min_cost)

OCDCL = $CDCL_{bnb}$ where $\mathcal{T}$(min_cost) = {-M. cost M ≥ min_cost}

$OCDCL_W$ = OCDCL + restrictions

Reuse!

$$
\begin{tikzcd}[column sep=huge]
\mathrm{CDCL} \arrow[r, bend right, "\text{properties}"'] & \mathrm{CDCL}_{bnb} \arrow[l, bend right, "\text{can be seen as a fragment of}"'] \arrow[r, bend left, "\substack{\text{by removing the additional component,} \\ \text{is a special case of}}"] & \mathrm{CDCL} \arrow[l, bend left, "\text{definitions and invariants}"]
\end{tikzcd}
$$

Reuse!

can be seen as a fragment of

by removing the additional component,
is a special case of

CDCL

CDCL_{bnb}

CDCL

properties

definitions and invariants

Reuse!

Propagate rule in Isabelle

$C \lor L \in N \Longrightarrow M \models as\ \neg C \Longrightarrow \text{undefined\_lit}\ M\ L \Longrightarrow$
$\quad (M, N) \Rightarrow_{CDCL} (L \# M, N)$

Propagate rule in Isabelle

$C \lor L \in N \Longrightarrow M \models as\ \neg C \Longrightarrow \text{undefined\_lit}\ M\ L \Longrightarrow$
$\quad (M, N, O) \Rightarrow_{CDCLbnb} (L \# M, N, O)$

Reuse!

Propagate rule in Isabelle

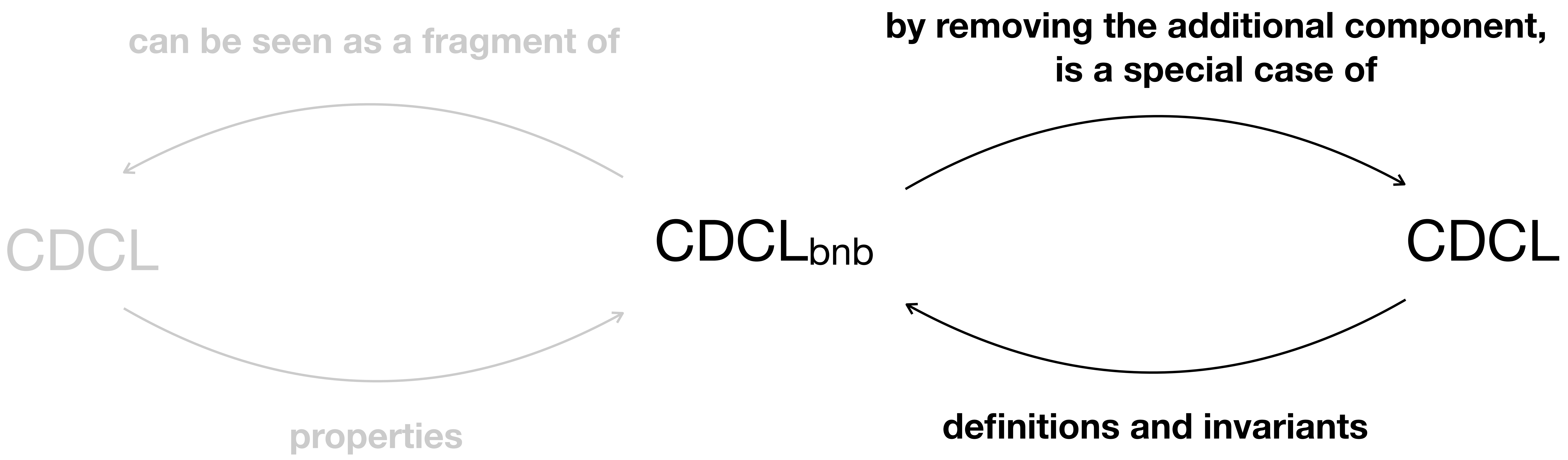
```
C v L ∈ N ⟹ M ⊨as ¬C ⟹ undefined_lit M L ⟹
    (M, N) ⇒CDCL (L # M, N)
```

obtained for free, thanks to abstraction over the state!
also invariants and theorems can be reused

Propagate rule in Isabelle

```
C v L ∈ N ⟹ M ⊨as ¬C ⟹ undefined_lit M L ⟹
    (M, N, O) ⇒CDCLbnb (L # M, N, O)
```

Reuse!


can be seen as a fragment of
CDCL
properties
CDCL$_{bnb}$
by removing the additional component,
is a special case of
CDCL
definitions and invariants

Reuse!

$$
\begin{tikzcd}[column sep=huge]
\mathrm{CDCL} \arrow[r, bend right, "\textbf{properties}"'] & \mathrm{CDCL}_{bnb} \arrow[l, bend right, "\textbf{can be seen as a fragment of}"'] \arrow[r, bend left, "\substack{\textbf{by removing the additional component,}\\ \textbf{is a special case of}}"] & \mathrm{CDCL} \arrow[l, bend left, "\textbf{definitions and invariants}"]
\end{tikzcd}
$$

Translate to reuse

Propagate rule

in Isabelle

$C \vee L \in N \Longrightarrow M \models as \neg C \Longrightarrow$ undefined_lit M L $\Longrightarrow$

$(M, N, 0) \Rightarrow_{CDCLbnb} (L \# M, N, 0)$

Propagate rule

in Isabelle

$C \vee L \in N{+}\mathcal{T}(min\_cost) \Longrightarrow M \models as \neg C \Longrightarrow$

undefined_lit M L $\Longrightarrow$

$(M, N + \mathcal{T}(min\_cost), 0) \Rightarrow_{CDCL}$

$(L \# M, N + \mathcal{T}(min\_cost), 0)$

Reuse!

$$\begin{tikzcd}[column sep=huge]
\text{CDCL} \arrow[r, bend right, "\text{properties}"'] & \text{CDCL}_{\text{bnb}} \arrow[l, bend right, "\text{can be seen as a fragment of}"'] \arrow[r, bend left, "\substack{\text{by removing the additional component,} \\ \text{is a special case of}}"] & \text{CDCL} \arrow[l, bend left, "\text{definitions and invariants}"]
\end{tikzcd}$$

Reuse in practise!

ignore the additional component

CDCL_{bnb} = CDCL + improve + $\mathcal{T}$(min_cost)

Inherited:

Definitions (for free)

Reuse in practise!

no strategy
but terminating

well-founded
for most applications

$\mathrm{CDCL}_{\mathrm{bnb}}$ = CDCL + improve +
$\mathcal{T}$(min_cost)

Inherited:

Termination (for free) Definitions (for free)

max planck institut
informatik

Reuse in practise!

strategy $\Longrightarrow$ **strategy invariants holds**

improve can always be applied on total models

CDCL_{bnb} = CDCL + improve + $\mathcal{T}$(min_cost)

Inherited:

Termination (for free)   Definitions (for free)

Ends with an unsat set (nearly for free)

max planck institut informatik

Reuse in practise!

CDCL_{bnb} does not know anything about what is optimised!

Inherited:

Termination (for free)

Definitions (for free)

Ends with an unsat set (nearly for free)

Why does it work?

OCDCL = CDCL_{bnb} where

$\mathcal{T}(\text{min\_cost}) = \{-M.\ \text{cost } M \geq \text{min\_cost}\}$

Why does it work?

OCDCL = CDCL_{bnb} where
$\mathcal{T}$(min_cost) = {-M. cost M ≥ min_cost}

Lemma

If I is a total model of N with cost < min_cost, then I is a model of N+$\mathcal{T}$(min_cost)

max planck institut informatik

Why does it work?

Lemma

If I is a total model of N with cost < min_cost, then I is a model of N+$\mathcal{T}$(min_cost)

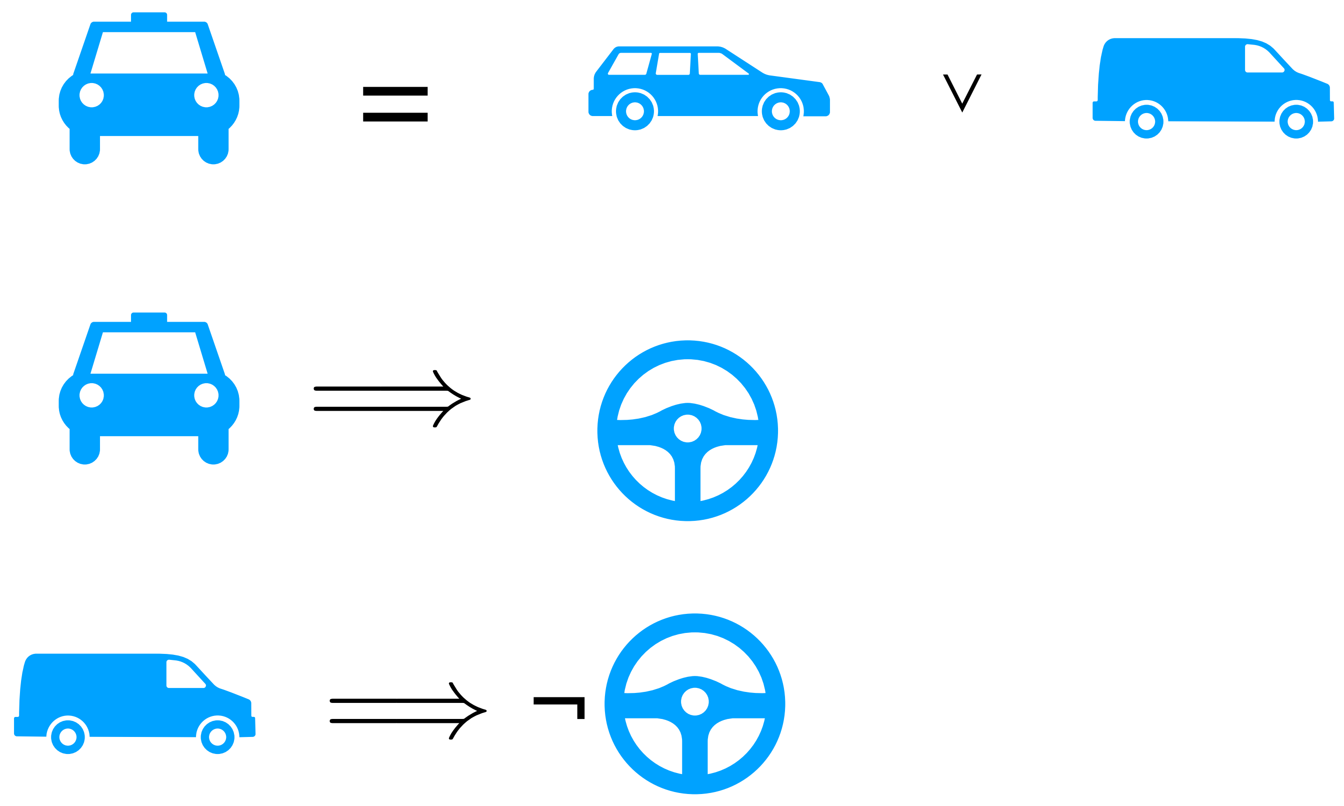
Fails for partial models!

How lazy do you like your formalisation?

make sure that the rules on paper and in Isabelle are the same

$\mathrm{OCDCL}_W = \mathrm{OCDCL} + \text{restrictions}$

Another application: Dead features



Can every option be true?

How lazy do you like your formalisation?

Christoph's view:

CDCLcm_W = CDCL + improve + conflict rules

copy-paste of proofs

My idea:

$\text{CDCLcm} = \text{CDCL}_{\text{bnb}}$ where

$\mathcal{T}$(models_founds) = {-M. there is a model with more trues in models_founds}

How lazy do you like your formalisation?

How lazy do you like your formalisation?

	Lines of codes (kloc)
CDCL_{bnb}	2,0
OCDCL	1,8
OCDCL_{W}	1,0
Partial Encoding	1,2

(a lot of boilerplate)

How lazy do you like your formalisation?

	Lines of codes (kloc)
CDCL_{bnb}	2,0
OCDCL	1,8
OCDCL_{W}	1,0
Partial Encoding	1,2
MaxSAT	0,4

(a lot of boilerplate)

can solve

Conclusion

Concrete outcome

- CDCL with branch and bound
- Via an encoding, also partial optimal models

Methodology

- Locales, locales, locales
- Be lazy!

Future work

- CDCL($\mathcal{T}$)

max planck institut informatik

Conclusion

Concrete outcome

- CDCL with branch and bound

OCDCL = CDCL_{bnb} where

$\mathcal{T}$(min_cost) = {-M. cost M ≥ min_cost}

OCDCL = CDCL_{bnb} where

$\mathcal{T}$(min_cost) = {-D. {M. cost M ≥ min_cost} ⊨ D}

Future work

- CDCL($\mathcal{T}$)

Conclusion

Concrete outcome

- CDCL with branch and bound
- Via an encoding, also partial optimal models

Methodology

- Locales, locales, locales
- Be lazy!

Future work

- CDCL($\mathcal{T}$)

Conclusion: How about CDCL($\mathcal{T}$)?

But isn't CDCL($\mathcal{T}$) exactly:

CDCL$_{bnb}$ where
$\mathcal{T}$ = {clauses entailed theory}

Not exactly, because the wrong conflict clause (negation of the trail) is used

max planck institut informatik

Translate to reuse

Theory propagation

Propagate rule

in Isabelle

C ∨ L ∈ N + $\boxed{\mathcal{T}\text{(min\_cost)}}$ ⟹ M ⊨as ¬C ⟹
undefined_lit M L ⟹
(M, N + $\mathcal{T}$(min_cost), 0) ⇒CDCLbnb
(L # M, N + $\mathcal{T}$(min_cost), 0)