How to Complete an Interactive Configuration Process?

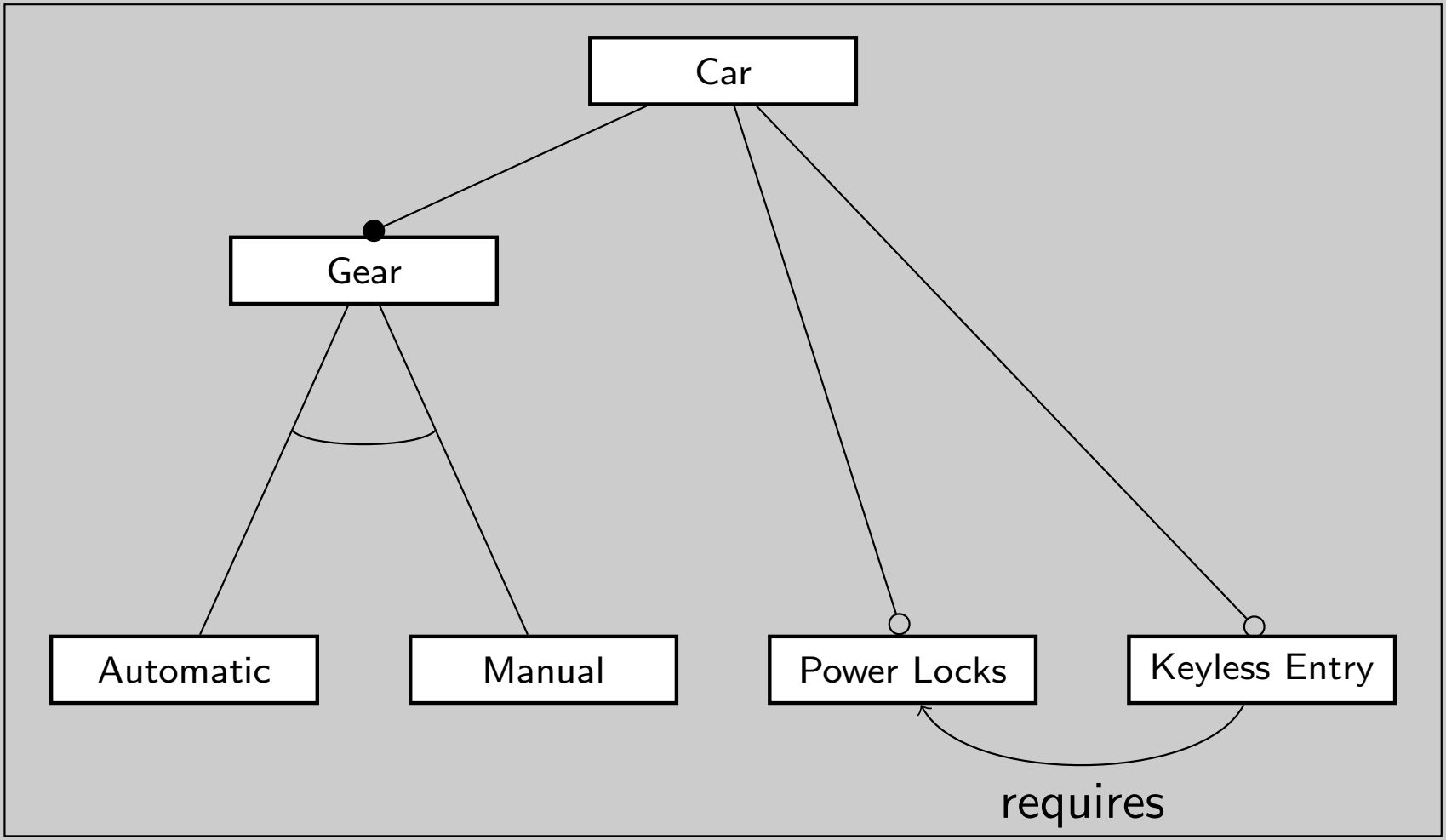
Mikoláš Janota Goetz Botterweck
Radu Grigore Joao Marques-Silva

Lero
University College Dublin
Ireland

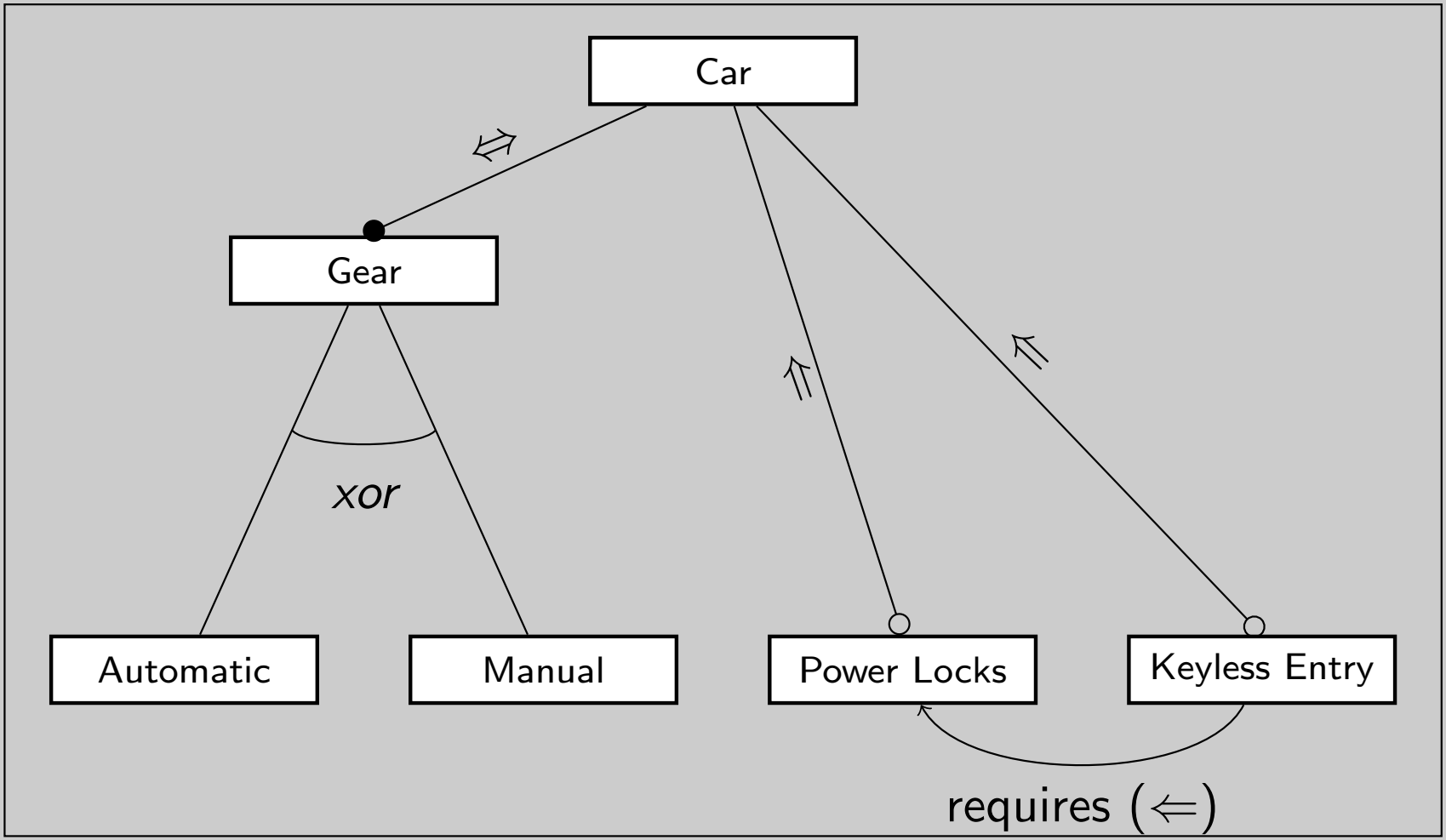
lero

SFI grant no. 03/CE2/I303_1

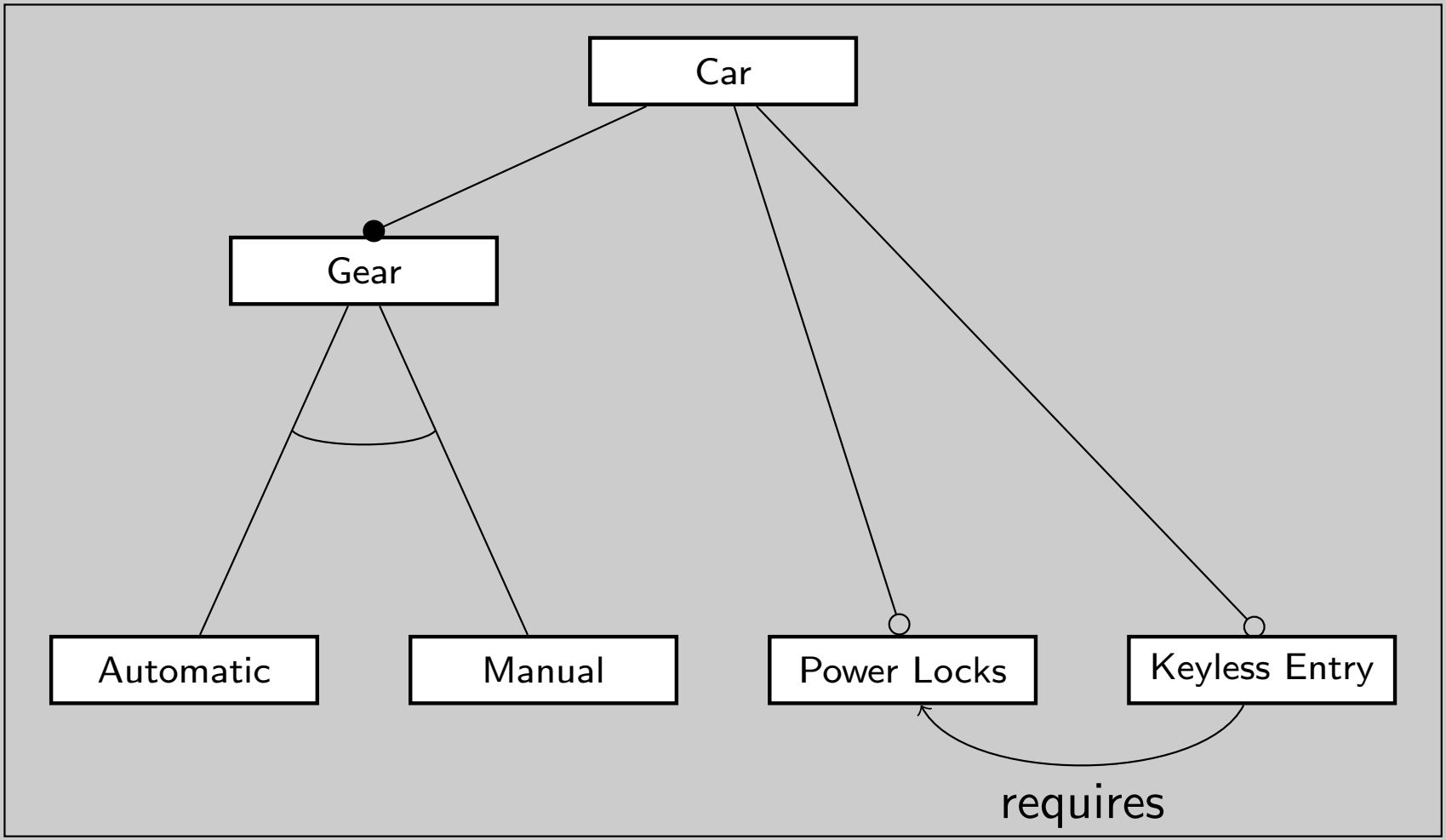
Feature Diagrams



Feature Diagrams



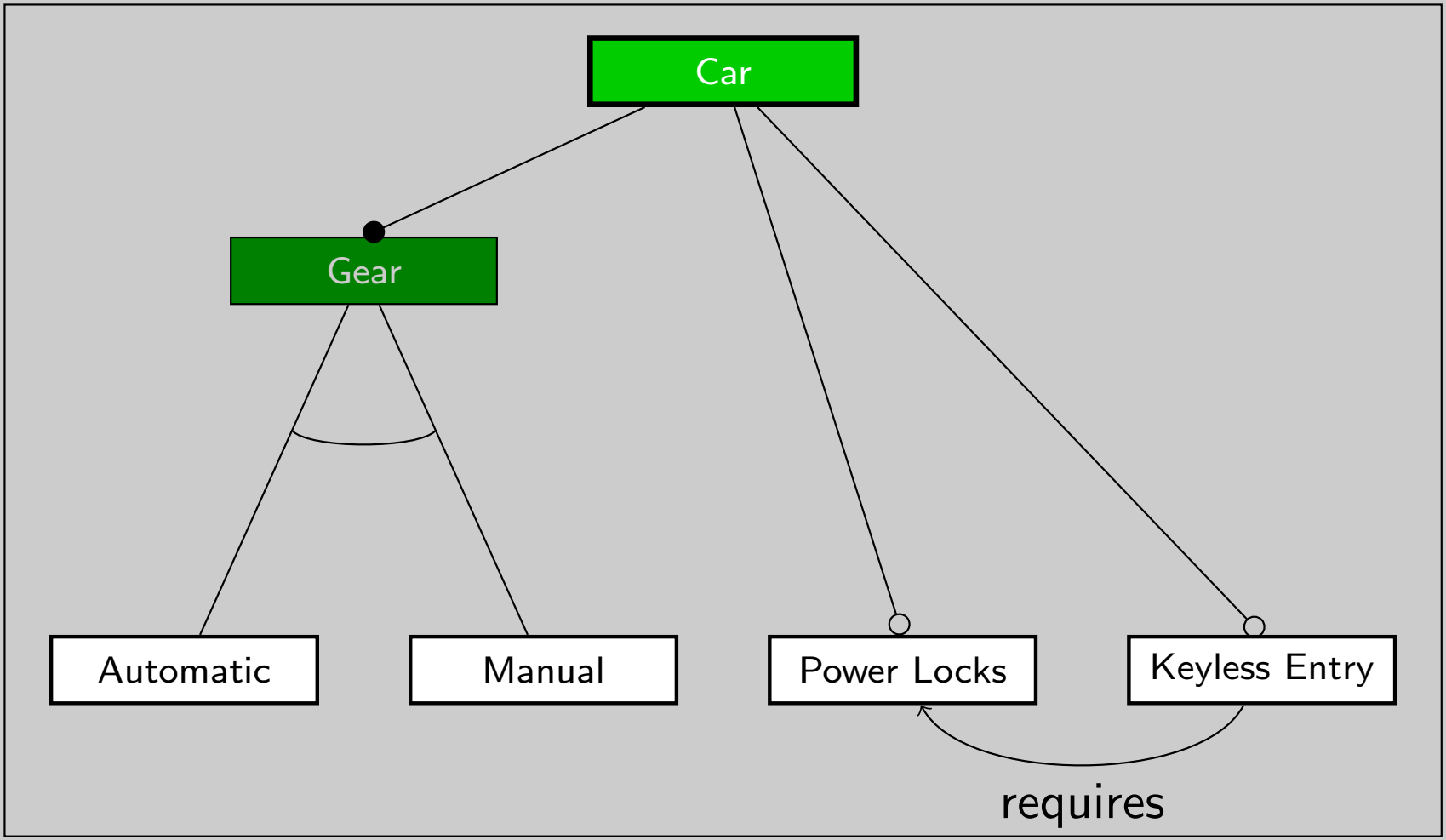
Configuration



Legend

Selection | Deselection | Autoselect | Autodeselect

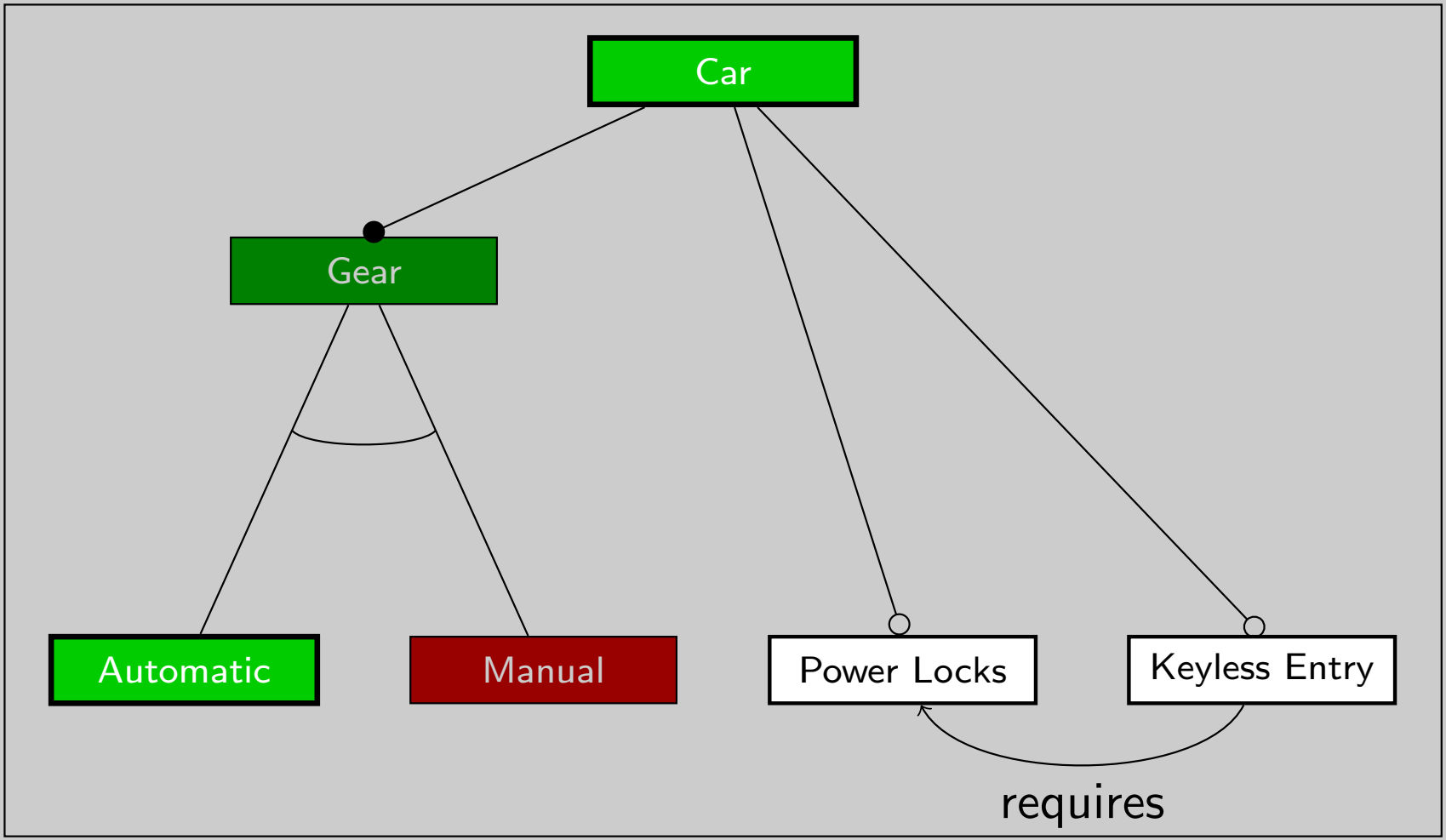
Configuration



Legend

Selection | Deselection | Autoselect | Autodeselect

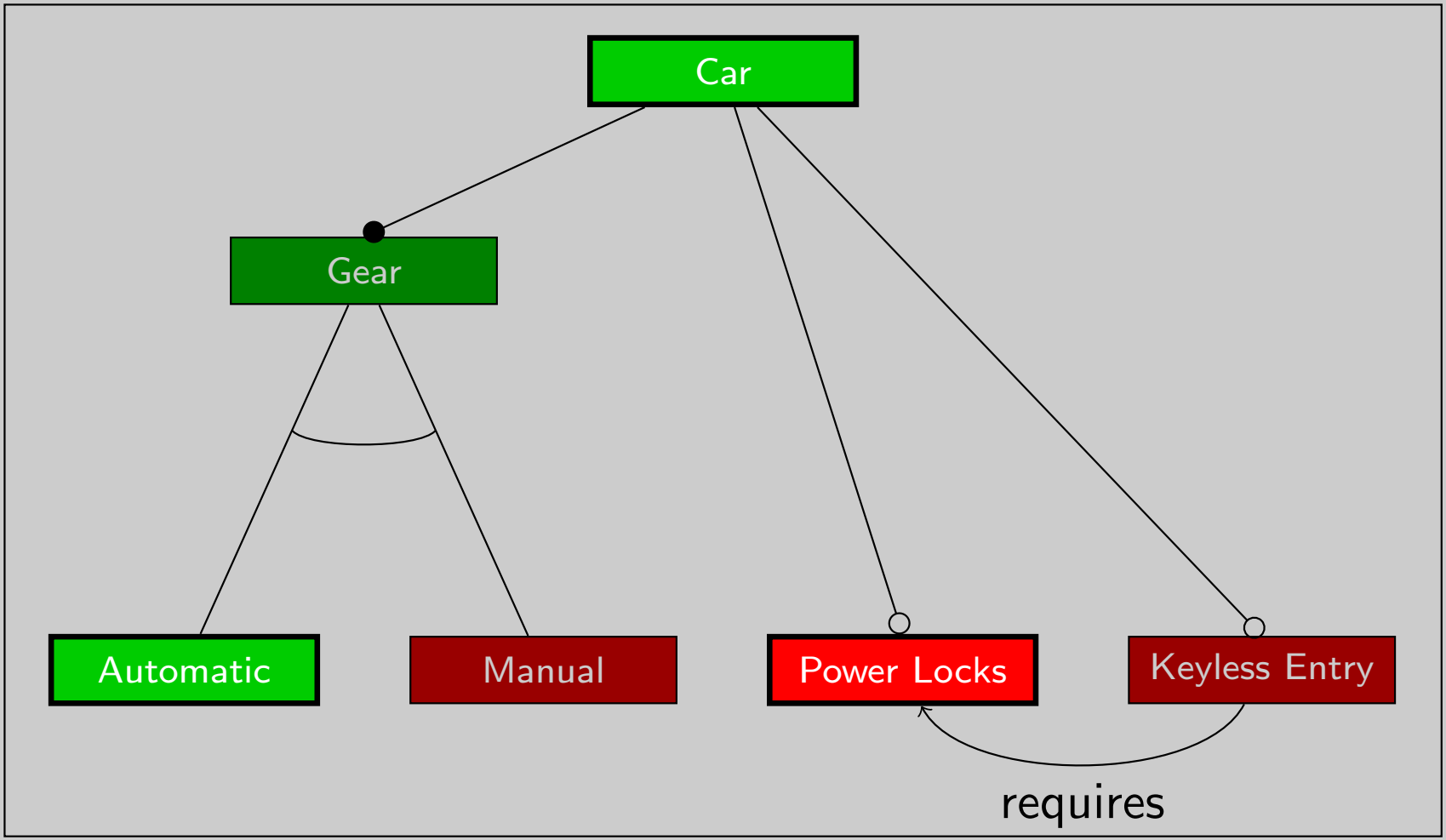
Configuration



Legend

Selection | Deselection | Autoselect | Autodeselect

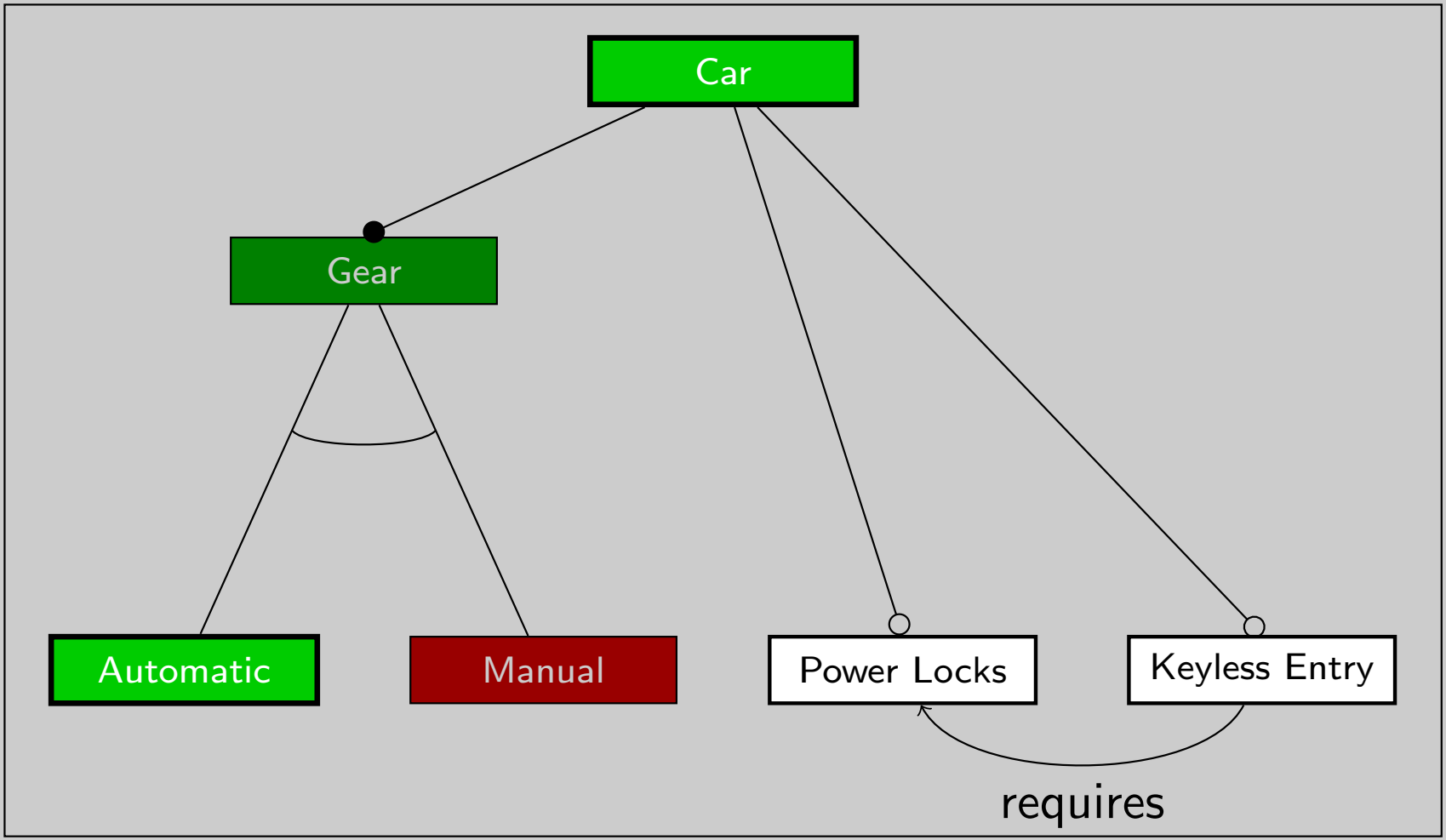
Configuration



Legend

Selection | Deselection | Autoselect | Autodeselect

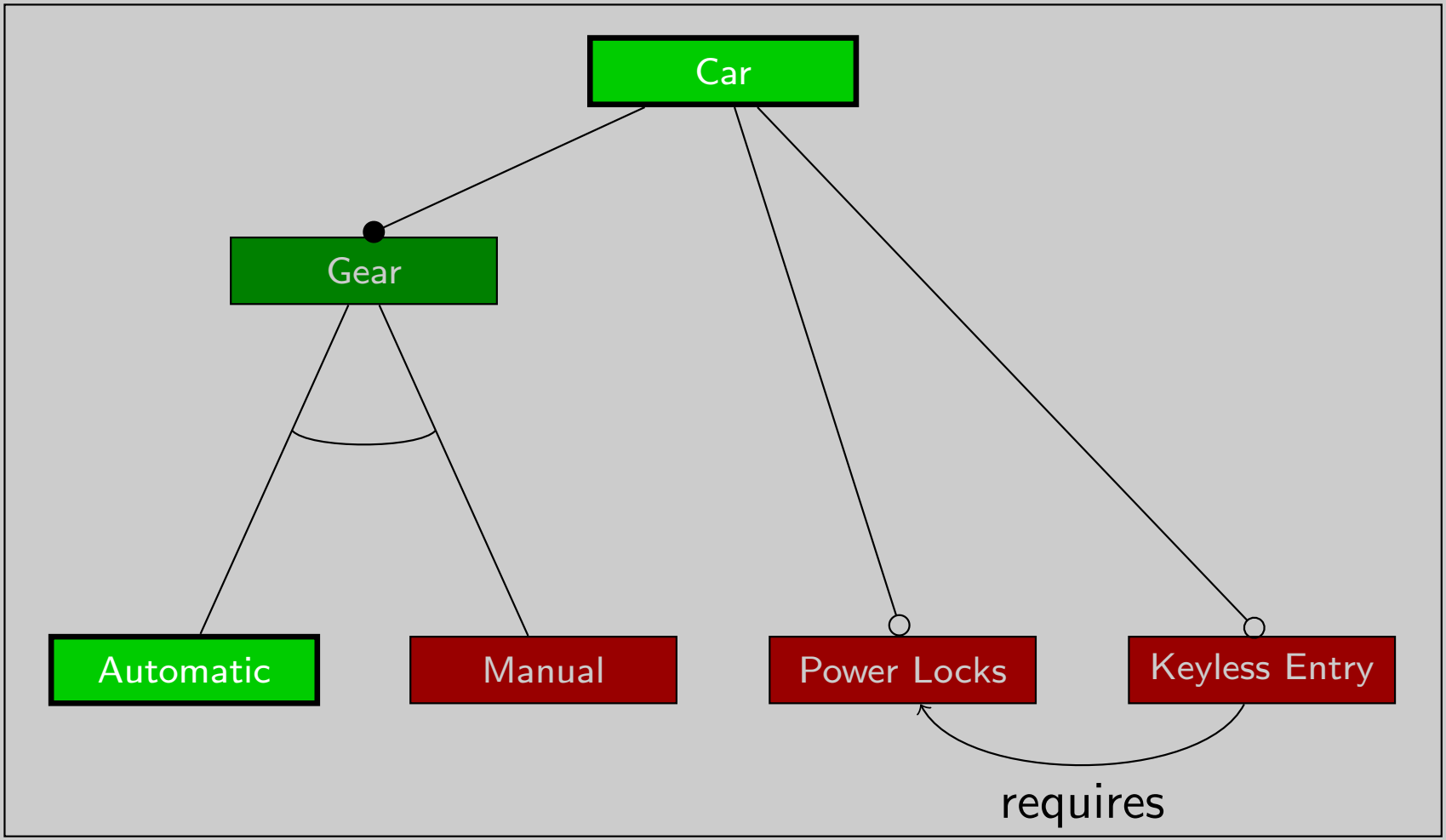
“I’m Done”



Legend

Selection | Deselection | Autoselect | Autodeselect

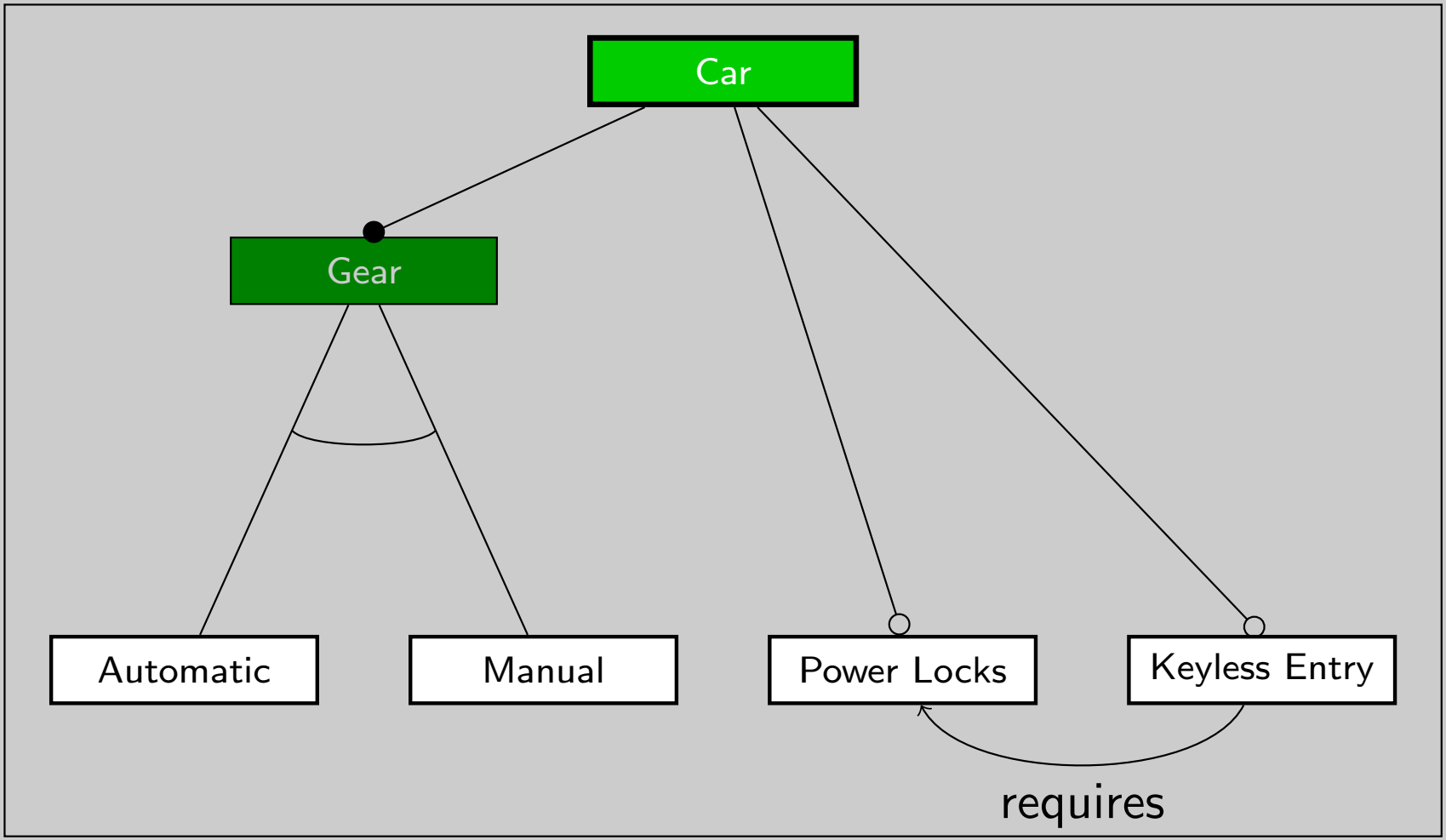
“I’m Done”



Legend

Selection | Deselection | Autoselect | Autodeselect

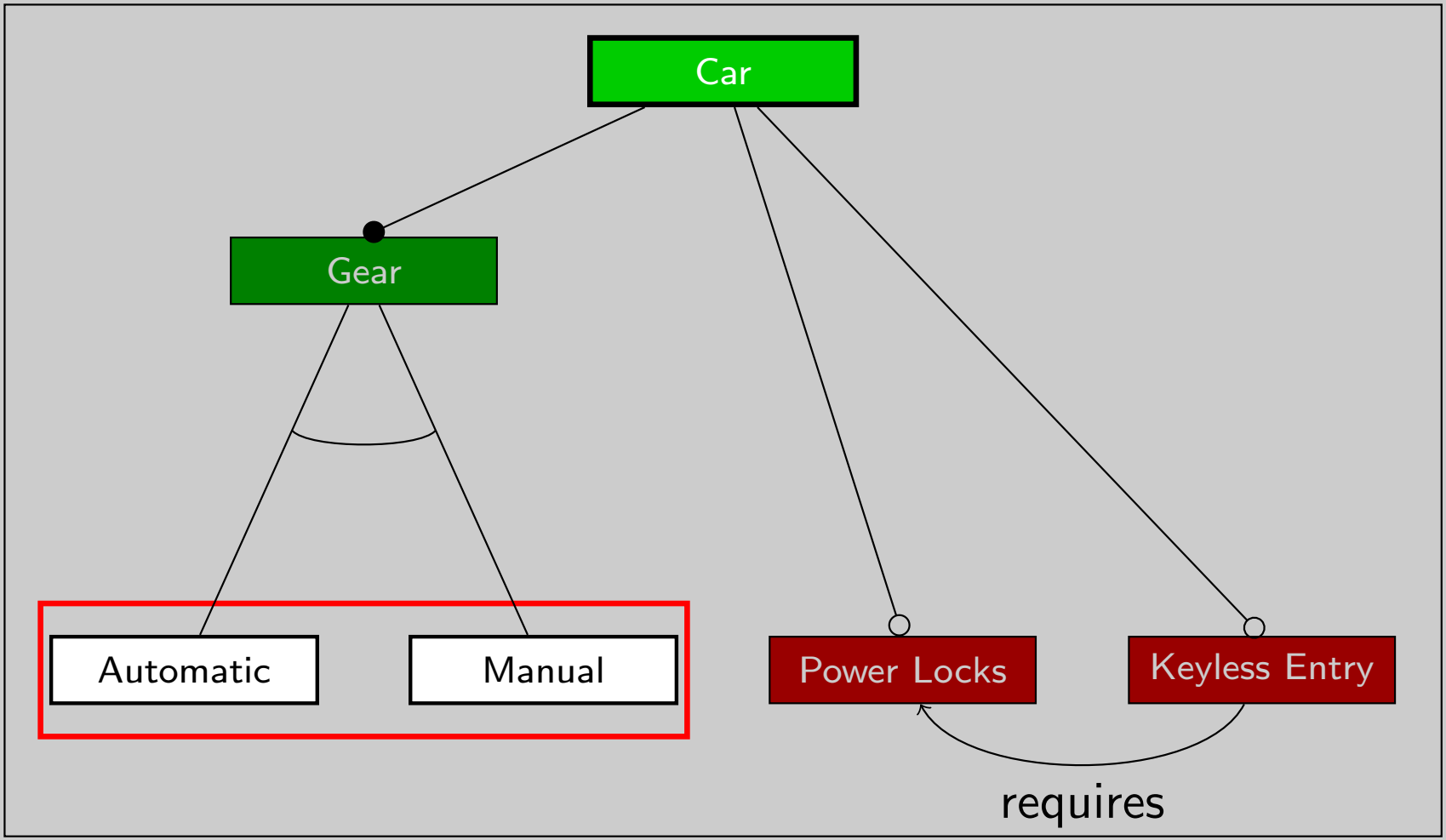
“I’m Done” sometimes doesn’t work



Legend

Selection | Deselection | Autoselect | Autodeselect

“I’m Done” sometimes doesn’t work



Legend

Selection | Deselection | Autoselect | Autodeselect

How is a configuration process completed?

Conditions

- All features must have a value.
- Values must conform to the constraints.

How is a configuration process completed?

Conditions

- All features must have a value.
- Values must conform to the constraints.

Scenarios

- **manual** — user fills in everything

How is a configuration process completed?

Conditions

- All features must have a value.
- Values must conform to the constraints.

Scenarios

- **manual** — user fills in everything
- **blind completion** — automated tool fills in everything

How is a configuration process completed?

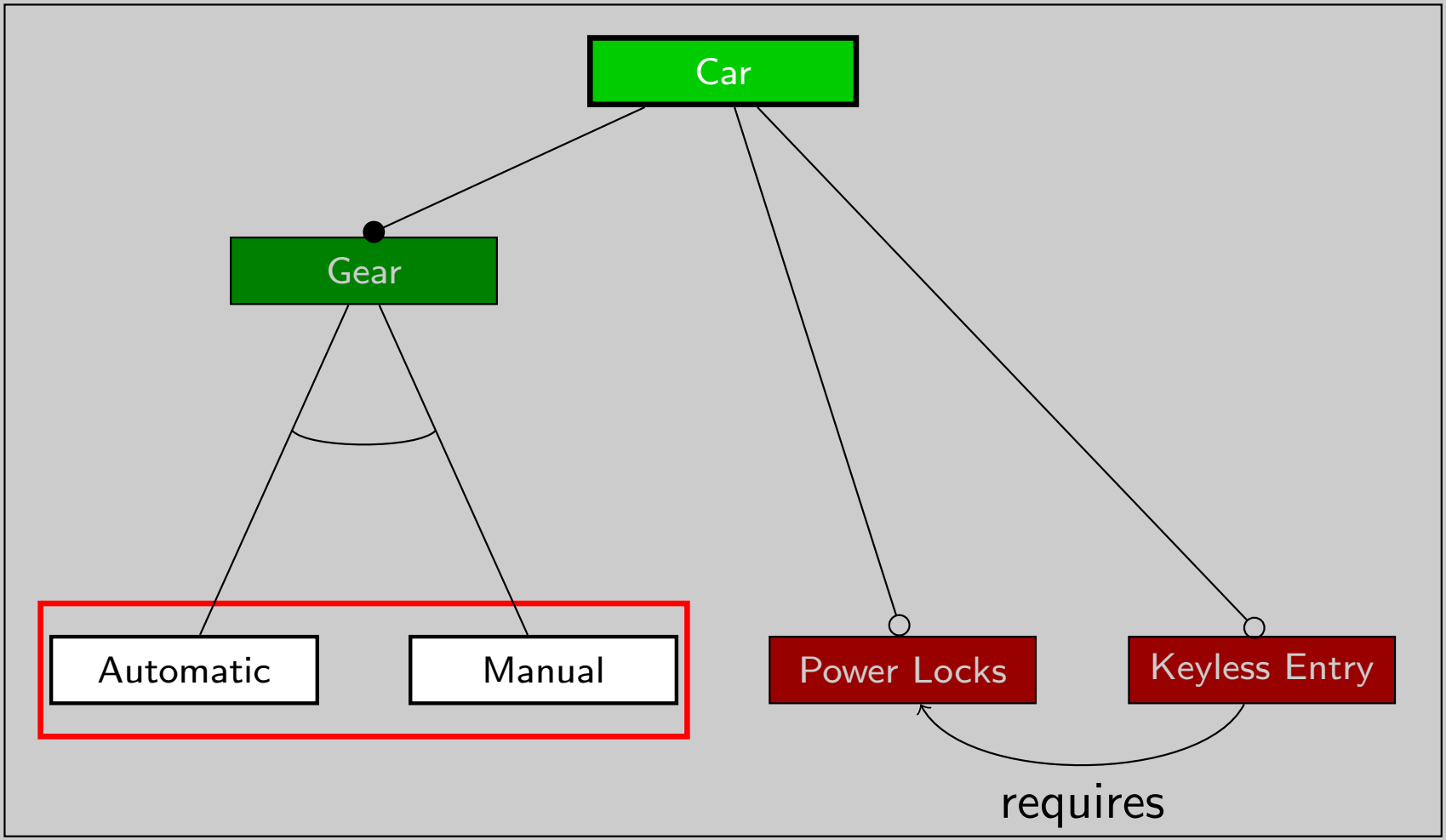
Conditions

- All features must have a value.
- Values must conform to the constraints.

Scenarios

- **manual** — user fills in everything
- **blind completion** — automated tool fills in everything
- **smart completion** — automated tool fills as much as possible *without making decisions* for the user

Smart Completion



Legend

Selection | Deselection | Autoselect | Autodeselect

Decisions and Dispensable Variables

- A set of variables is deselectable *iff* they can be all deselected all at once.

$$\mathcal{D}(\phi, X) \stackrel{\text{def}}{=} SAT\left(\phi \wedge \bigwedge_{v \in X} \neg v\right)$$

Decisions and Dispensable Variables

- A set of variables is deselectable *iff* they can be all deselected all at once.

$$\mathcal{D}(\phi, X) \stackrel{\text{def}}{=} SAT\left(\phi \wedge \bigwedge_{v \in X} \neg v\right)$$

- A set of variables X must be decided *iff*
  1. X it is not deselectable.
  2. All of the proper subsets of X are deselectable.

$$\neg\mathcal{D}(\phi, X) \wedge (\forall Y \subsetneq X)(\mathcal{D}(\phi, Y))$$

Decisions and Dispensable Variables

- A set of variables is deselectable *iff* they can be all deselected all at once.

$$\mathcal{D}(\phi, X) \stackrel{\text{def}}{=} SAT\left(\phi \wedge \bigwedge_{v \in X} \neg v\right)$$

- A set of variables X must be decided *iff*
  1. X it is not deselectable.
  2. All of the proper subsets of X are deselectable.

$$\neg\mathcal{D}(\phi, X) \wedge (\forall Y \subsetneq X)(\mathcal{D}(\phi, Y))$$

- A variable is dispensable *iff* it does not belong to any set that must be decided.

Examples

$x \vee y \vee z$

- Deselectable: $\{x, y\}$, $\{x, z\}$, $\{y, z\}$, $\{x\}$, $\{y\}$, $\{z\}$, and $\emptyset$
- Not deselectable: $\{x, y, z\}$
- $\{x, y, z\}$ must be decided and none of the variables are dispensable.

Examples

$x \vee y \vee z$

- Deselectable: $\{x, y\}$, $\{x, z\}$, $\{y, z\}$, $\{x\}$, $\{y\}$, $\{z\}$, and $\emptyset$
- Not deselectable: $\{x, y, z\}$
- $\{x, y, z\}$ must be decided and none of the variables are dispensable.

$x \Rightarrow (y \vee z)$

- $\{x, y, z\}$ is deselectable therefore does not have to be decided and all variables are dispensable.

Examples

$\mathbf{x} \vee \mathbf{y} \vee \mathbf{z}$

- Deselectable: $\{x, y\}$, $\{x, z\}$, $\{y, z\}$, $\{x\}$, $\{y\}$, $\{z\}$, and $\emptyset$
- Not deselectable: $\{x, y, z\}$
- $\{x, y, z\}$ must be decided and none of the variables are dispensable.

$\mathbf{x} \Rightarrow (\mathbf{y} \vee \mathbf{z})$

- $\{x, y, z\}$ is deselectable therefore does not have to be decided and all variables are dispensable.

$(\mathbf{x} \Rightarrow (\mathbf{y} \vee \mathbf{z})) \wedge \mathbf{x}$

- $\{y, z\}$ must be decided.

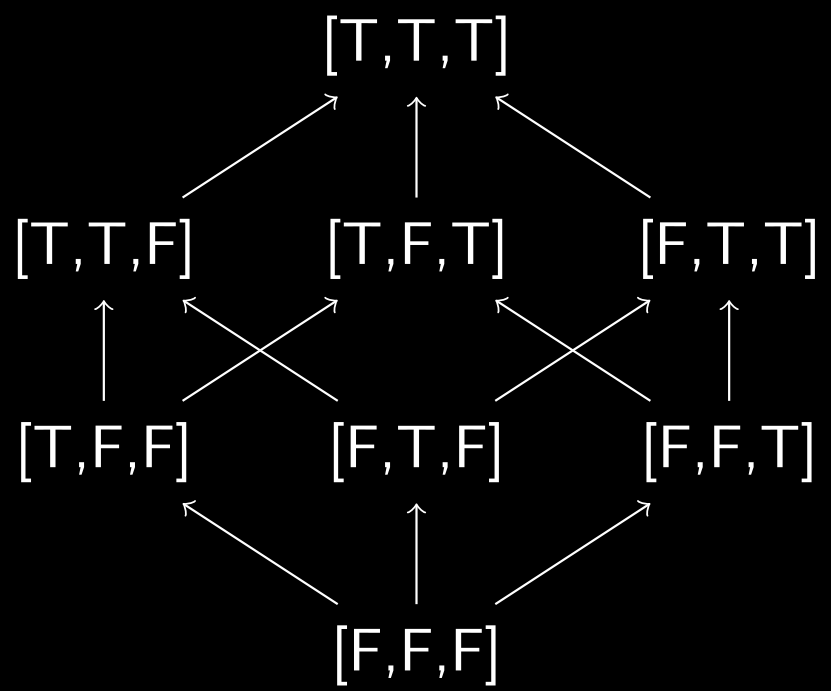
Relation to Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some **T**rue values to **F**alse, yields a non-model.

Relation to Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some **T**rue values to **F**alse, yields a non-model.

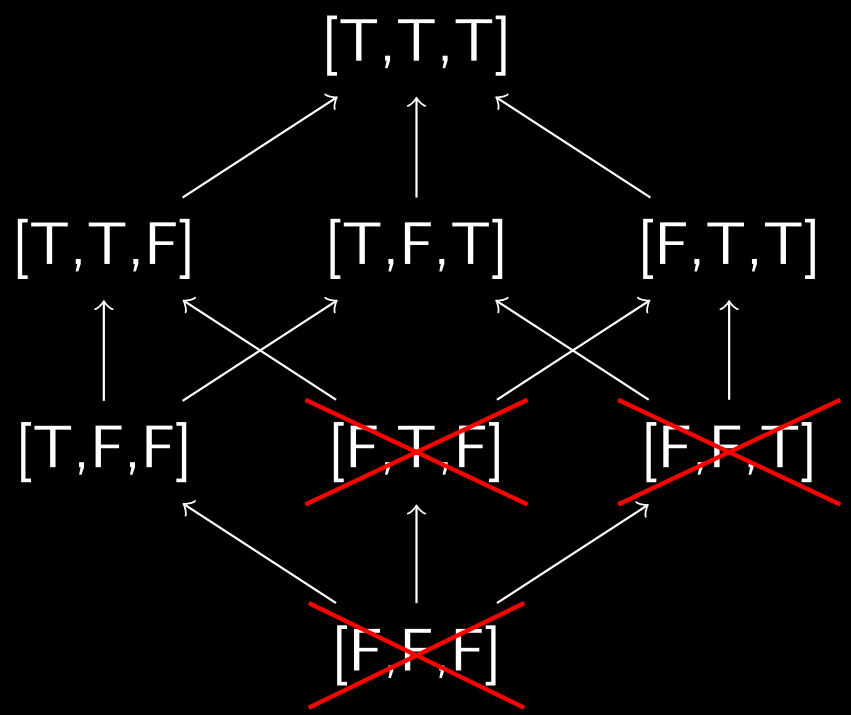
$\mathbf{x} \vee (\mathbf{y} \wedge \mathbf{z})$



Relation to Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some **T**rue values to **F**alse, yields a non-model.

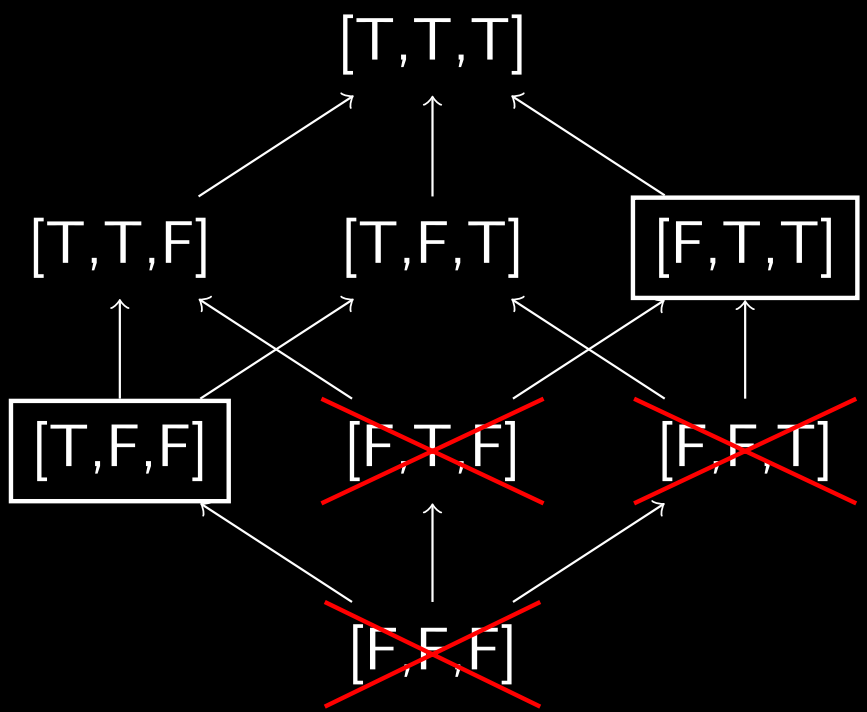
x $\vee$ (**y** $\wedge$ **z**)



Relation to Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some **T**rue values to **F**alse, yields a non-model.

$\mathbf{x} \vee (\mathbf{y} \wedge \mathbf{z})$



Dispensable Variables and Minimal Models

- A variable is dispensable *iff* it is False in all minimal models.

Dispensable Variables and Minimal Models

- A variable is dispensable *iff* it is False in all minimal models.

$\mathbf{x} \vee \mathbf{y} \vee \mathbf{z}$

x	**y**	**z**
T	F	F
F	T	F
F	F	T

Dispensable Variables and Minimal Models

- A variable is dispensable *iff* it is False in all minimal models.

$\mathbf{x} \vee \mathbf{y} \vee \mathbf{z}$

x	y	z
T	F	F
F	T	F
F	F	T

$\mathbf{x} \Rightarrow (\mathbf{y} \vee \mathbf{z})$

x	y	z
F	F	F

Other Relations

Non-monotonic Reasoning

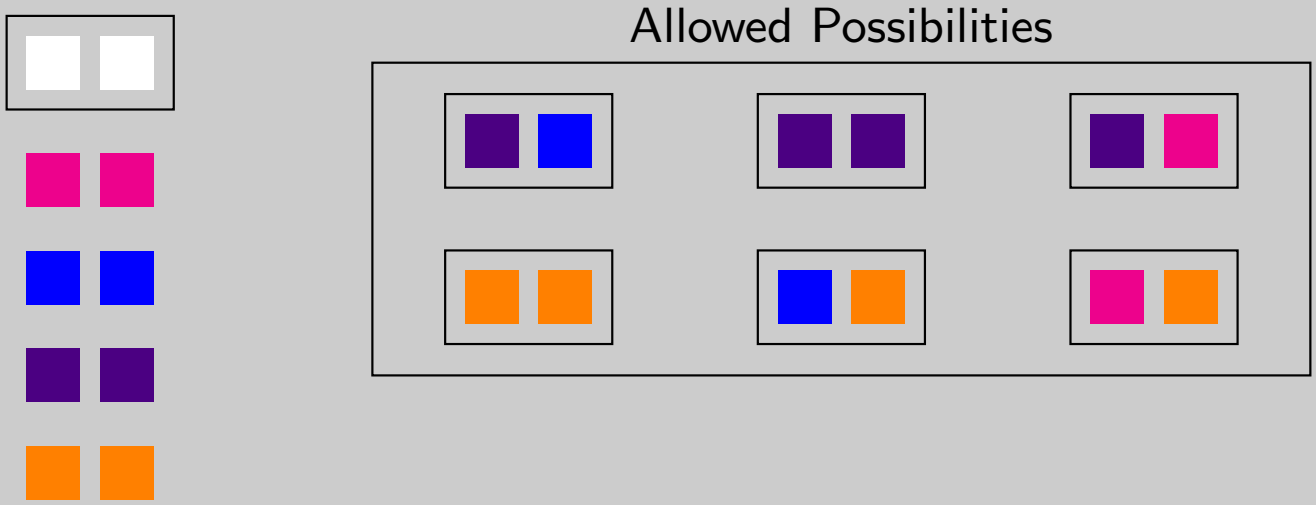
- Propositional Circumscription

$$\phi \models_{\min} \psi$$

- Generalized Closed World Assumption (GCWA)

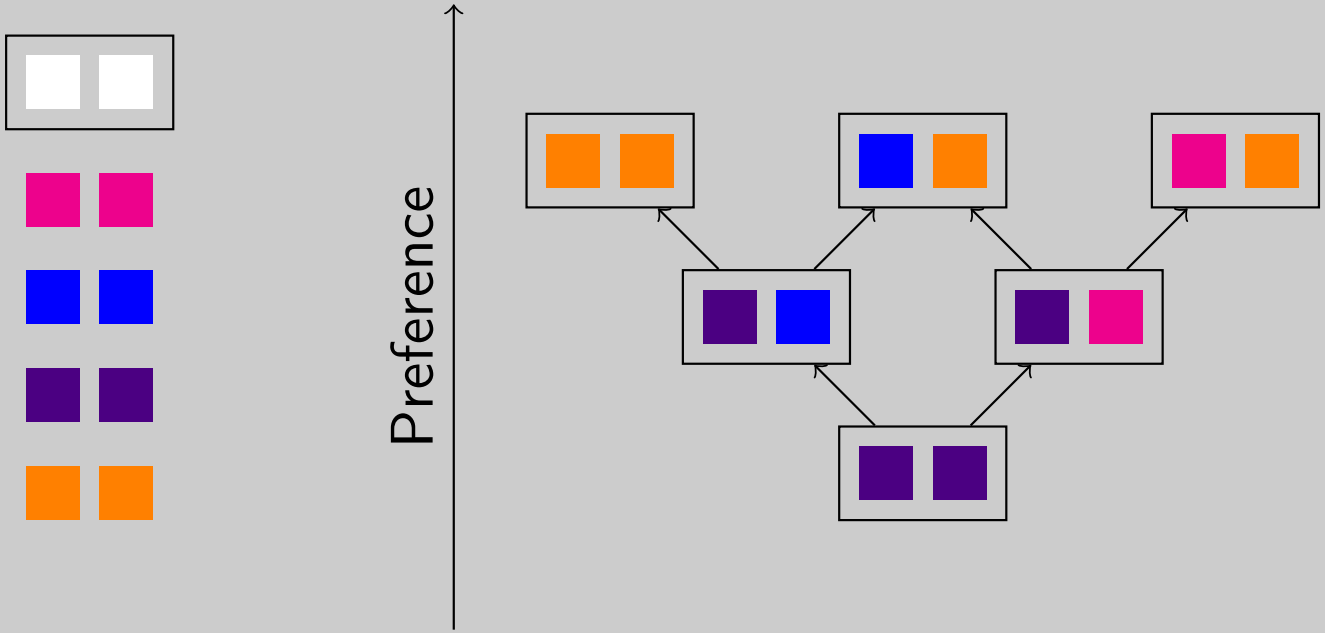
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.



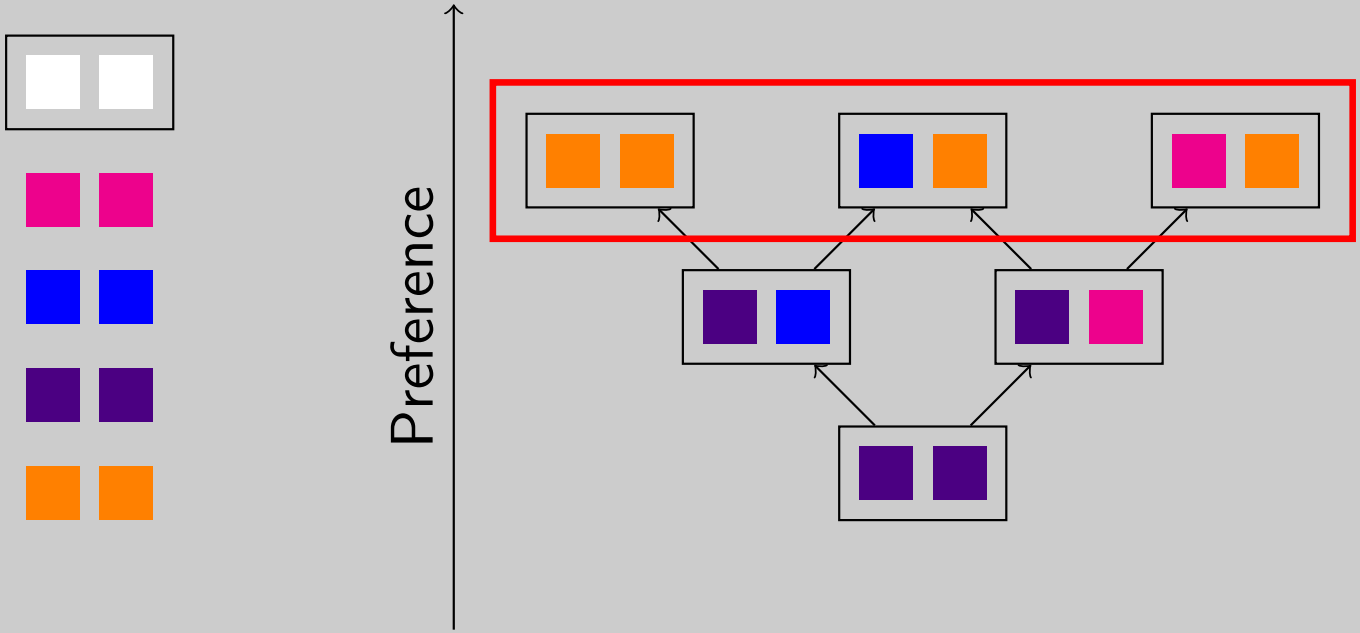
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
- But it is possible, if there is a preference on the possibilities.



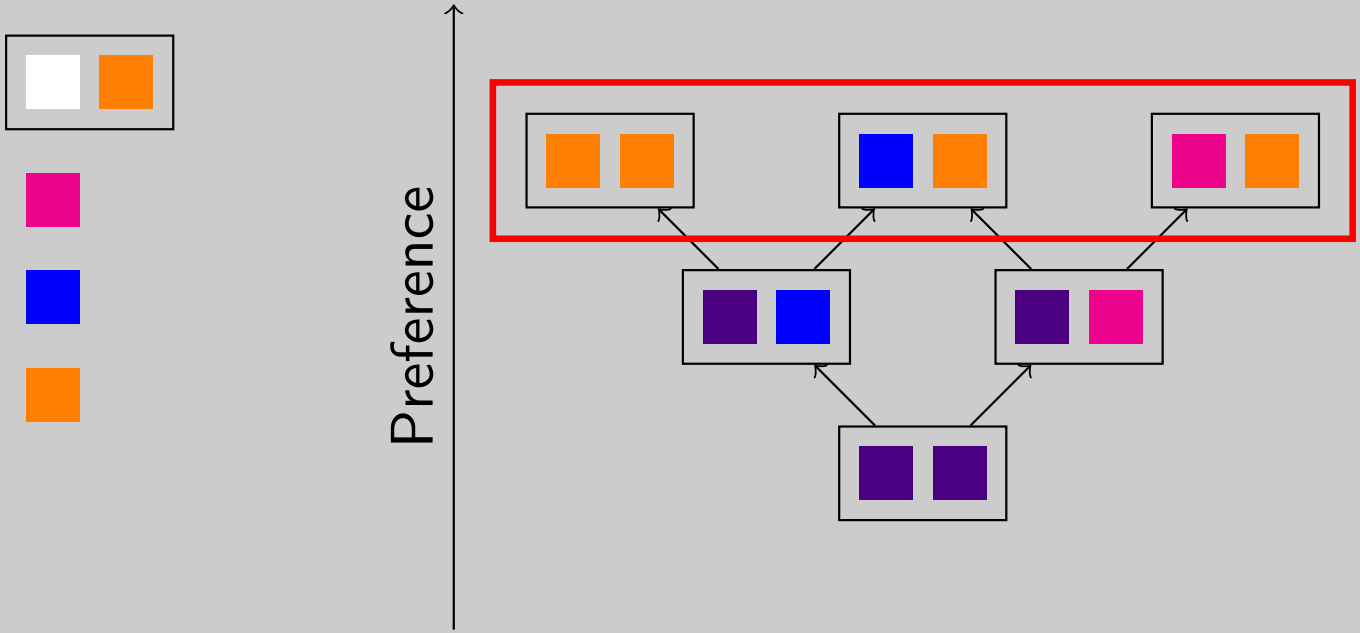
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
- But it is possible, if there is a preference on the possibilities.
- We can focus on the most preferred possibilities.



What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
- But it is possible, if there is a preference on the possibilities.
- We can focus on the most preferred possibilities.



General Case Meets the Boolean Case

- A value c is settled for a variable v *iff* v has the value c in all most preferred possibilities.

General Case Meets the Boolean Case

- A value c is settled for a variable v *iff* v has the value c in all most preferred possibilities.

Settled Values and Dispensable Variables

- A variable is dispensable iff False is settled for it under the point-wise preference for False.

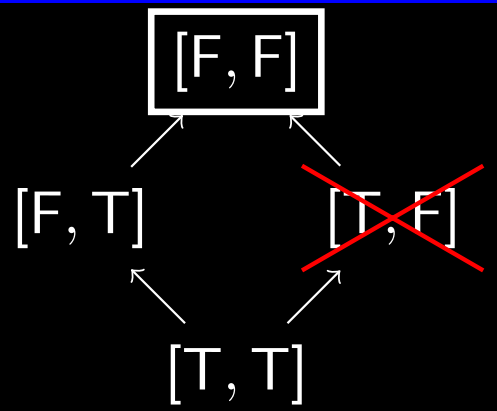
General Case Meets the Boolean Case

- A value c is settled for a variable v *iff* v has the value c in all most preferred possibilities.

Settled Values and Dispensable Variables

- A variable is dispensable iff False is settled for it under the point-wise preference for False.

$\mathbf{x} \Rightarrow \mathbf{y}$



Summary

- Motivation was to provide smart completion.

Summary

- Motivation was to provide smart completion.
- In configuration of Boolean constraints it led to defining dispensable variables.

Summary

- Motivation was to provide smart completion.
- In configuration of Boolean constraints it led to defining dispensable variables.
- Dispensable variables are closely related to CWA.

Summary

- Motivation was to provide smart completion.
- In configuration of Boolean constraints it led to defining dispensable variables.
- Dispensable variables are closely related to CWA.
- In non-Boolean case, smart completion can be provided in the presence of preference.

Summary

- Motivation was to provide smart completion.
- In configuration of Boolean constraints it led to defining dispensable variables.
- Dispensable variables are closely related to CWA.
- In non-Boolean case, smart completion can be provided in the presence of preference.
- Dispensable variables can be seen as a *preference for deselecting*.
- Analogously CWA as a preference for False.

Experimental Results

Name	Features	Clauses	Length	Done	Minimal models
tightvnc	21	22	5.5	5.5	1.0 ± 0.0
apl	27	41	12.2	11.9	1.0 ± 0.0
gg4	58	139	10.0	3.8	15.3 ± 22.6
berkeley	94	183	26.6	17.9	1.7 ± 1.1
violet	170	341	56.1	47.1	1.6 ± 0.9
E-shop	287	420	143	N/A	N/A