Context-Aware Clause Selection Using Symbol Name Meanings in Theorem Proving

Claudia Schon
Hochschule Trier
c.schon@hochschule-trier.de

Trier University
of Applied Sciences

HOCHSCHULE TRIER

Automated Reasoning

$subclass(bed, furniture)$
$subclass(doublebed, bed)$
$subclass(pillow, artifact)$
$subclass(blanket, fabric)$
$\ldots$

$\forall x(instance(x, hammock) \rightarrow$
$(material(x, fabric) \wedge \ldots$

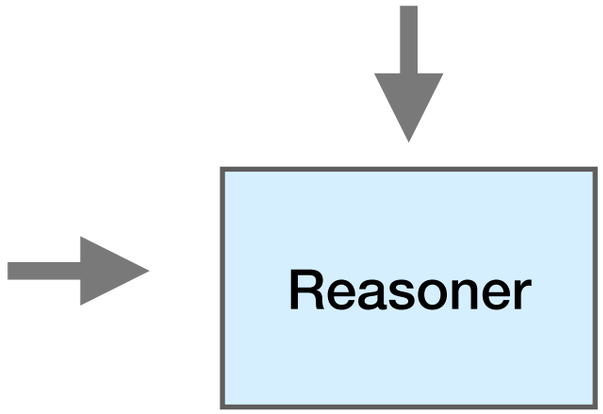
$\ldots$

$\forall x, y, z$
$((subclass(x, y) \wedge subclass(y, z)) \rightarrow$
$subclass(x, z))$

$subclass(weapon, device)$
$subclass(device, artifact)$

Knowledge Base

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$



Automated Reasoning

$subclass(bed, furniture)$
$subclass(doublebed, bed)$
$subclass(pillow, artifact)$
$subclass(blanket, fabric)$
$\ldots$

$\forall x(instance(x, hammock) \rightarrow$
$(material(x, fabric) \wedge \ldots$

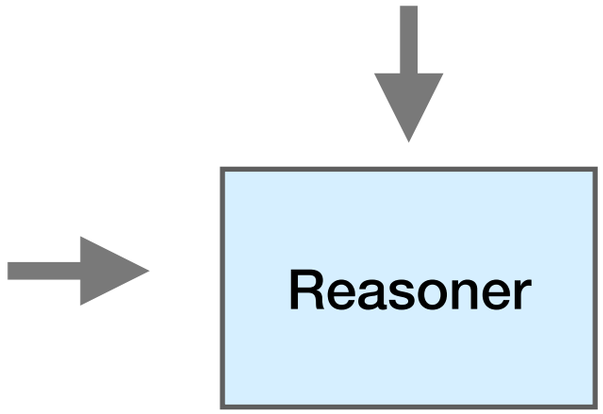
$\ldots$

$\forall x, y, z$
$((subclass(x, y) \wedge subclass(y, z)) \rightarrow$
$subclass(x, z))$

$subclass(weapon, device)$
$subclass(device, artifact)$

Knowledge Base

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$



Computes inferences!

Automated Reasoning

$subclass(bed, furniture)$
$subclass(doublebed, bed)$
$subclass(pillow, artifact)$
$subclass(blanket, fabric)$
$\ldots$

$\forall x(instance(x, hammock) \rightarrow$
$(material(x, fabric) \wedge \ldots$

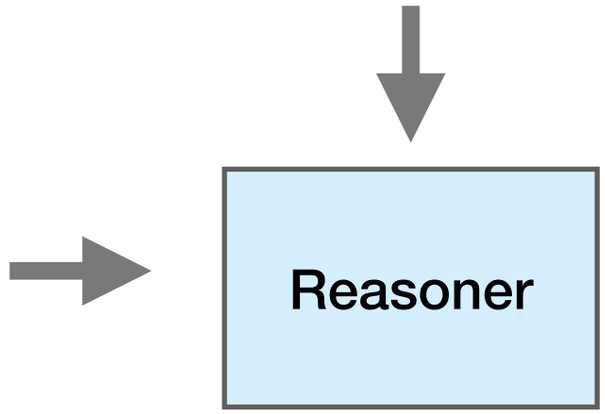
$\ldots$

$\forall x, y, z$
$((subclass(x, y) \wedge subclass(y, z)) \rightarrow$
$subclass(x, z))$

$subclass(weapon, device)$
$subclass(device, artifact)$

Knowledge Base

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$



Computes inferences!

Does not take the meaning of symbols into account!

Automated Reasoning

$s(b, f)$
$s(db, b)$
$s(p, a)$
$s(bl, fa)$
$\ldots$

$\forall x(in(x, h) \rightarrow$
$(m(x, fa) \wedge \ldots$

$\ldots$

$\forall x, y, z$
$((s(x, y) \wedge s(y, z)) \rightarrow$
$s(x, z))$

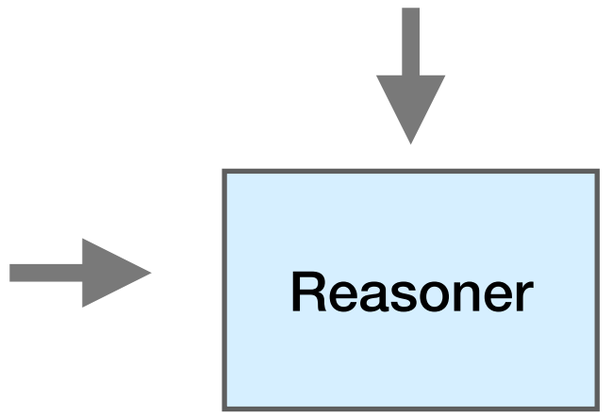
$s(w, de)$
$s(de, a)$

Knowledge Base

For the prover, the proof task looks like this:

Goal:

$\forall x(in(x, h) \rightarrow in(x, b))$



Computes inferences!

Does not take the meaning of symbols into account!

Automated Reasoning: Are Hammocks Beds?

Reasoner: PyRes

Wortwolken.com. (2024). Word cloud generator. Available at: https://www.wortwolken.com [Accessed: 2024-08-29].

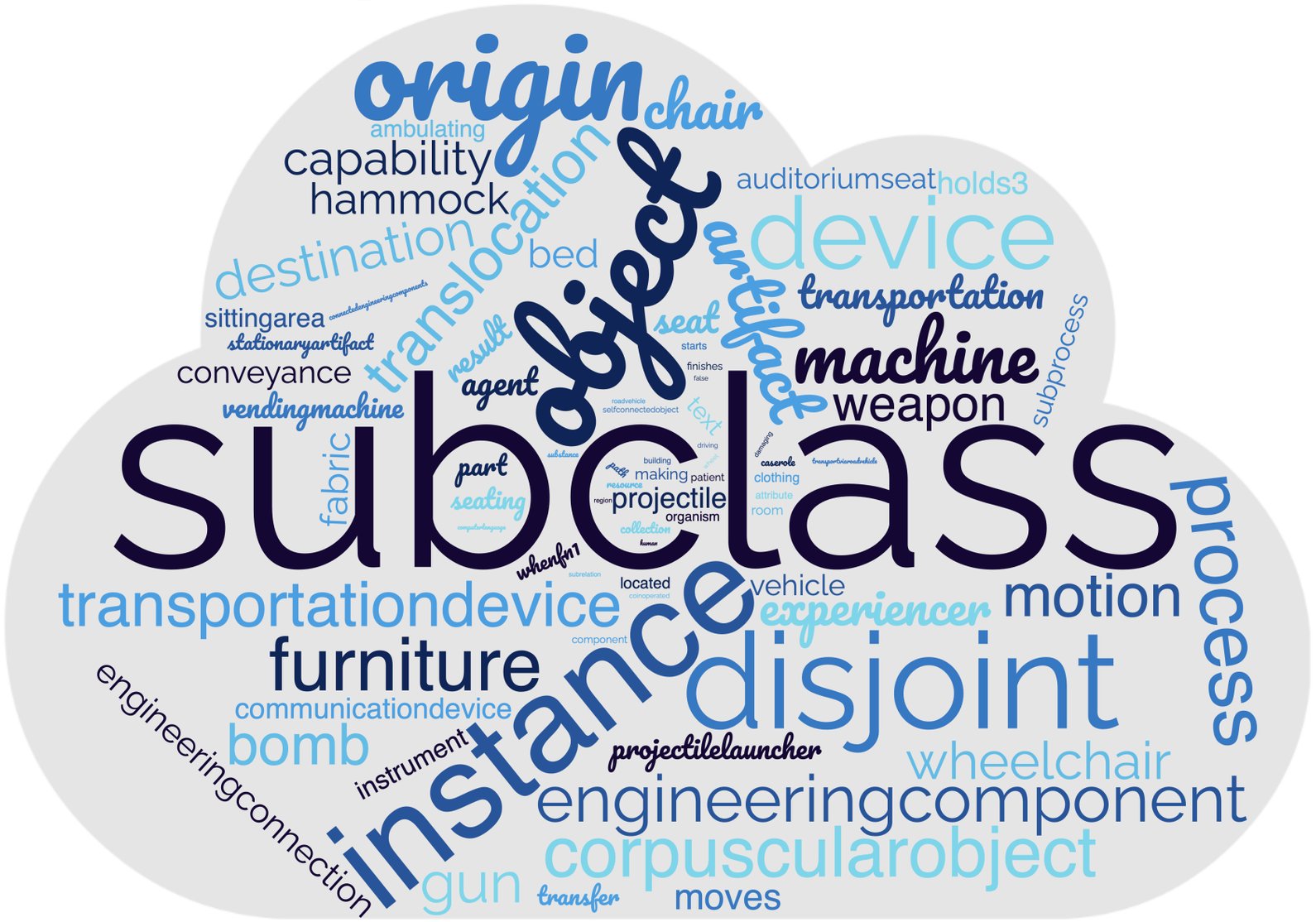
Automated Reasoning: Are Hammocks Beds?

Reasoner: PyRes

PyRes computes 538 resolvents!

Wortwolken.com. (2024). Word cloud generator. Available at: https://www.wortwolken.com [Accessed: 2024-08-29].

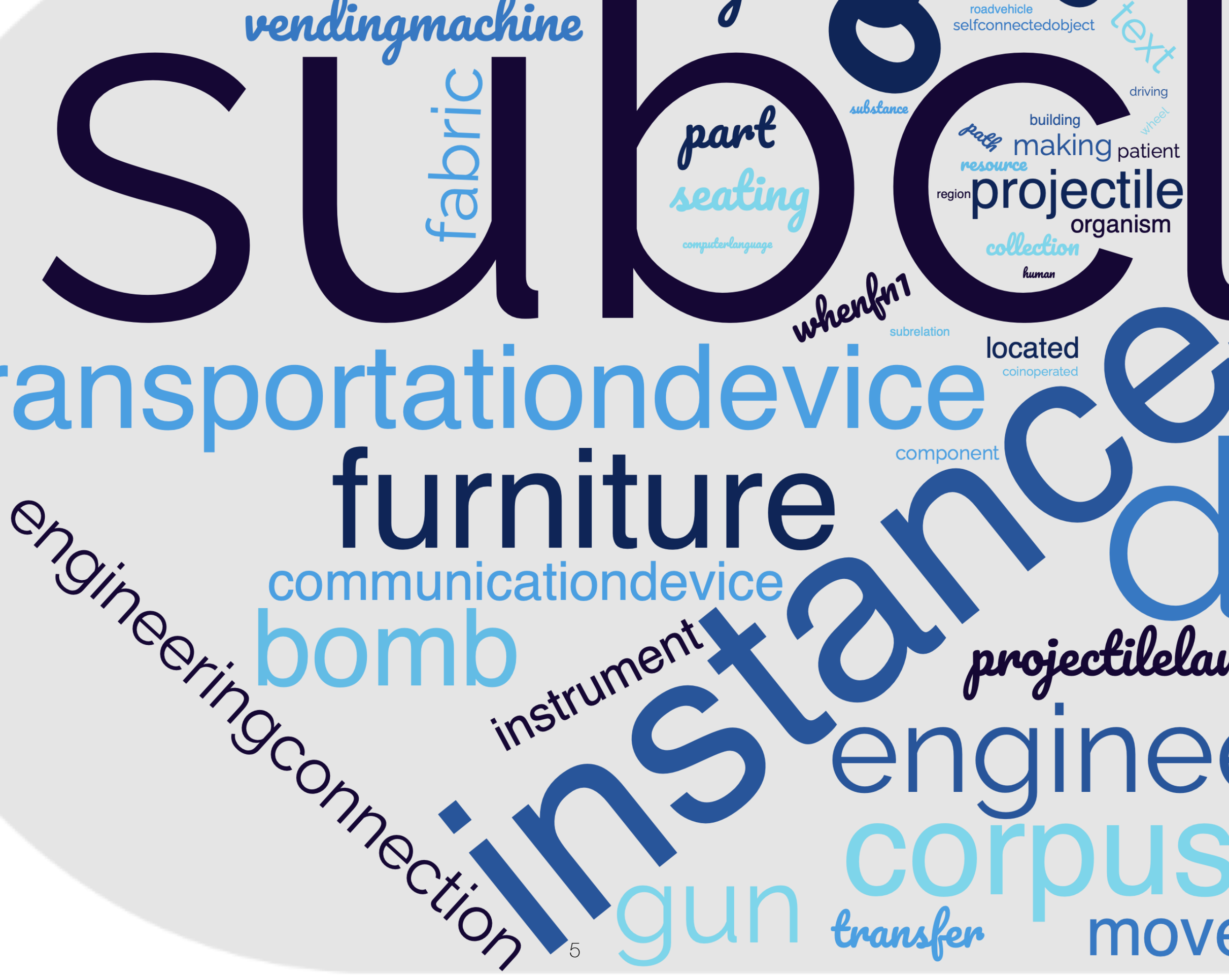
Automated Reasoning: Are Hammocks Beds?

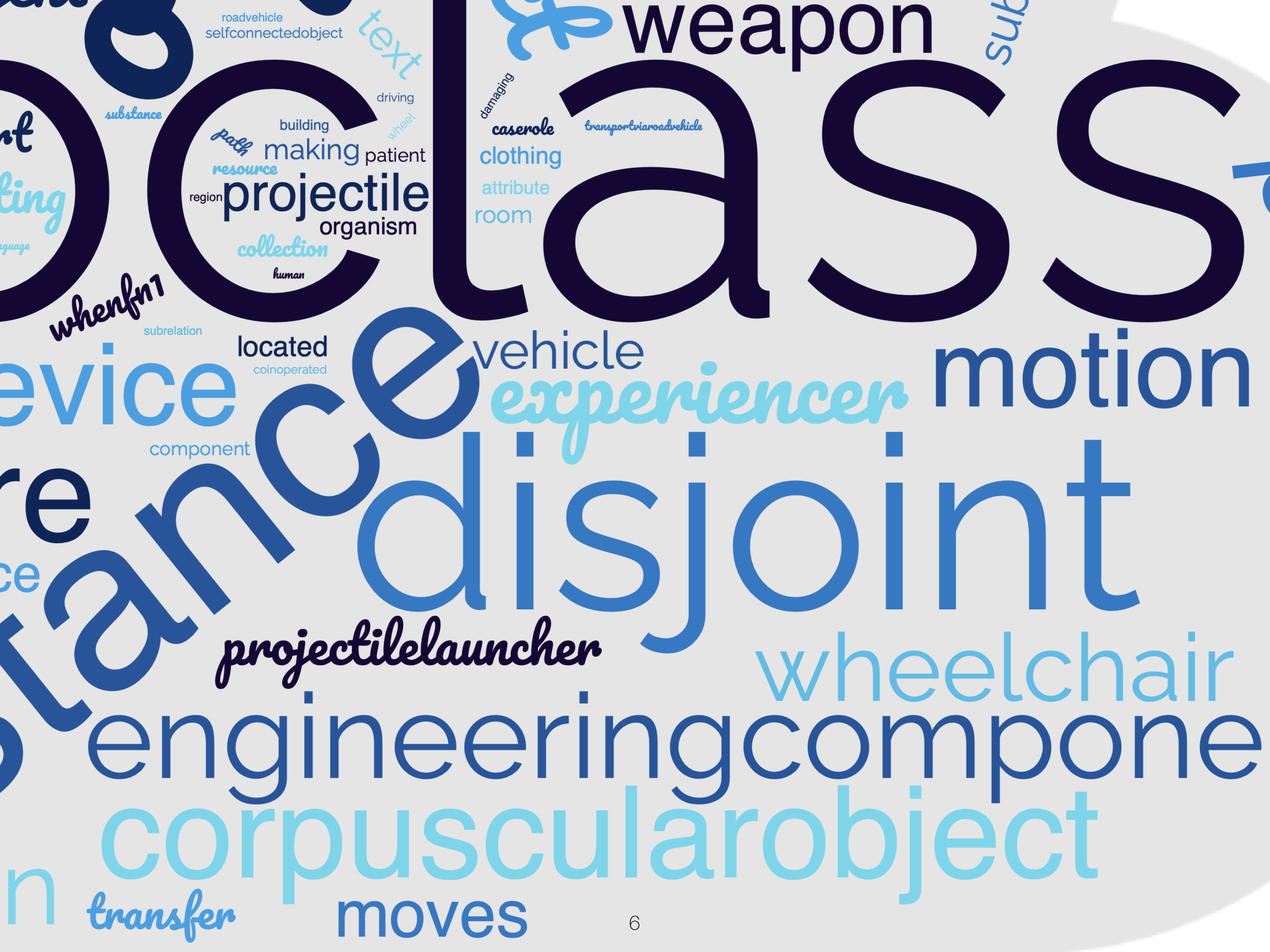


Reasoner: PyRes

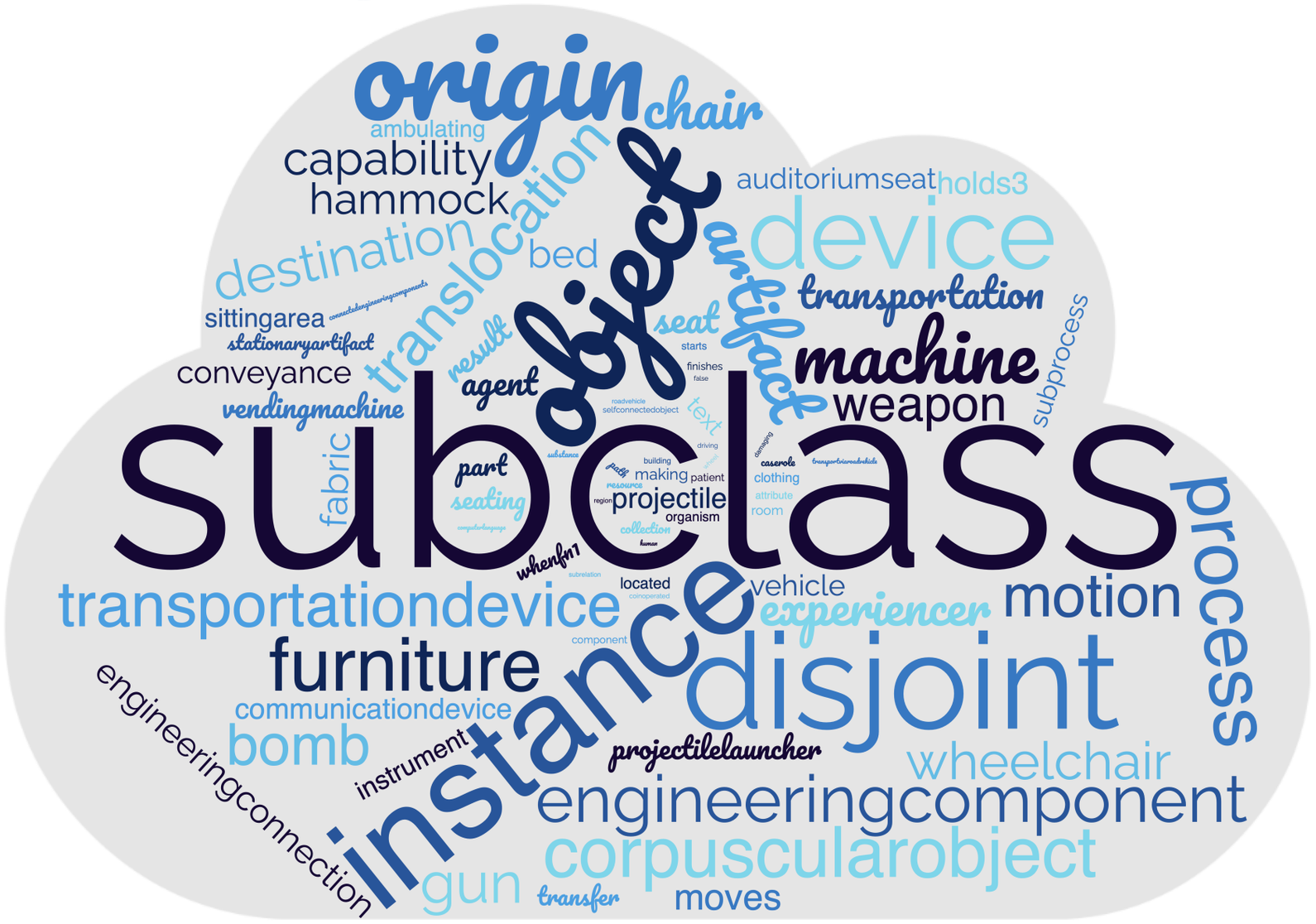
PyRes computes 538 resolvents!

Wortwolken.com. (2024). Word cloud generator. Available at: https://www.wortwolken.com [Accessed: 2024-08-29].


vendingmachine
fabric
roadvehicle
selfconnectedobject
text
substance
driving
wheel
part
building
path
making
patient
seating
resource
region
projectile
organism
computerlanguage
collection
human
whenfn1
subrelation
located
coinoperated
transportationdevice
component
furniture
engineeringconnection
communicationdevice
bomb
instrument
gun
transfer


weapon
text
roadvehicle
selfconnectedobject
driving
substance
building
path
making
patient
wheel
resource
region
projectile
organism
collection
human
damaging
caserole
transportviaroadvehicle
clothing
attribute
room
whenfn1
subrelation
located
coinoperated
vehicle
motion
experiencer
component
disjoint
projectilelauncher
wheelchair
engineeringcompone
corpuscularobject
transfer
moves

Automated Reasoning: Are Hammocks Beds?



Reasoner: PyRes

PyRes computes 538 resolvents!

Wortwolken.com. (2025). Word cloud generator. Available at: https://www.wortwolken.com [Accessed: 2024-08-29].

Why should we draw unrelated inferences in automated reasoning?

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

$\neg w \vee d$

$\neg d \vee a$

$\neg h \vee f$

$\neg h \vee s$

$\ldots$

Clause set S

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

$\neg w \vee d$

$\neg d \vee a$

$\neg h \vee f$

$\neg h \vee s$

$\cdots$

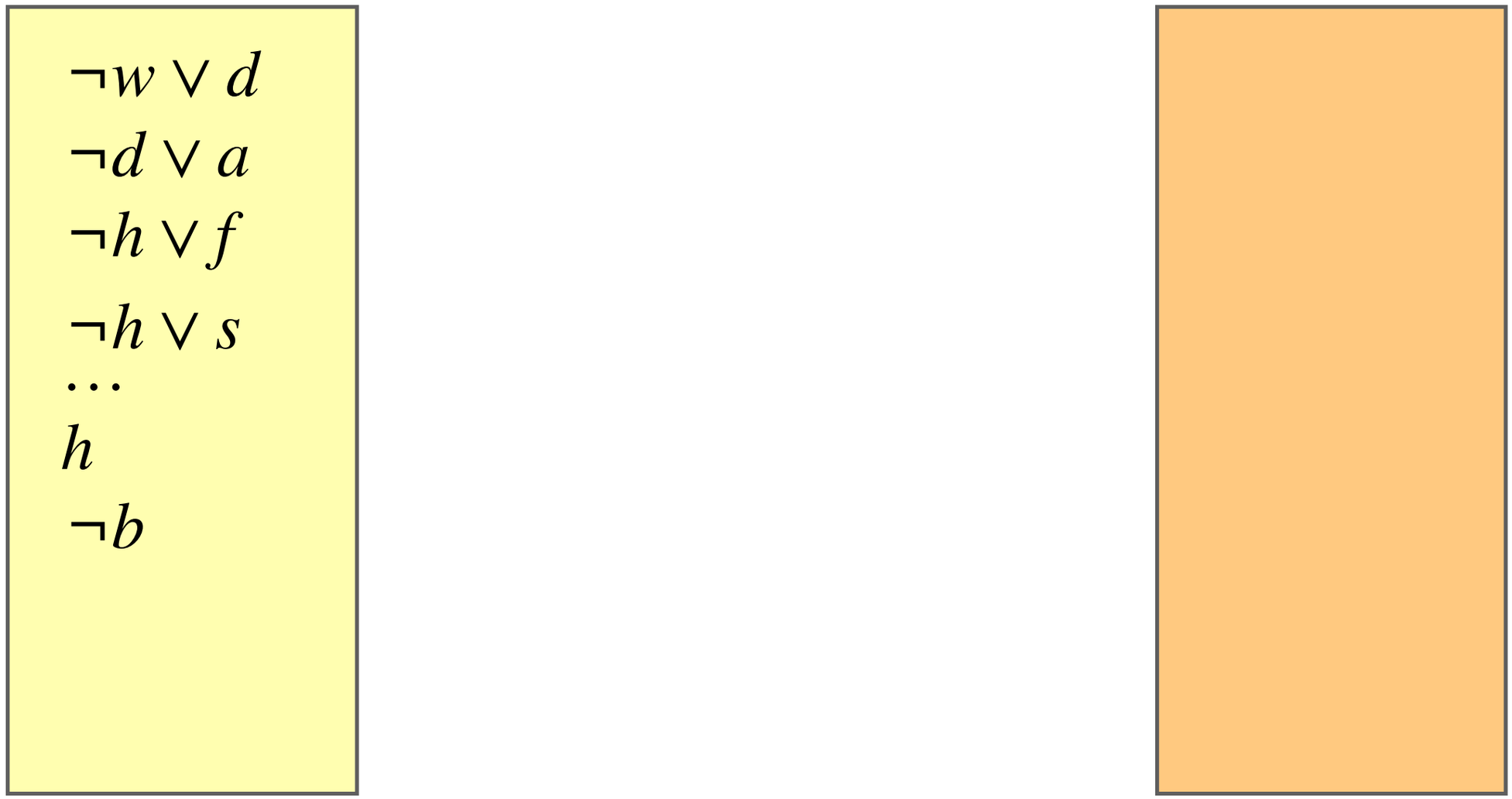
h

$\neg b$

Clause set S

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

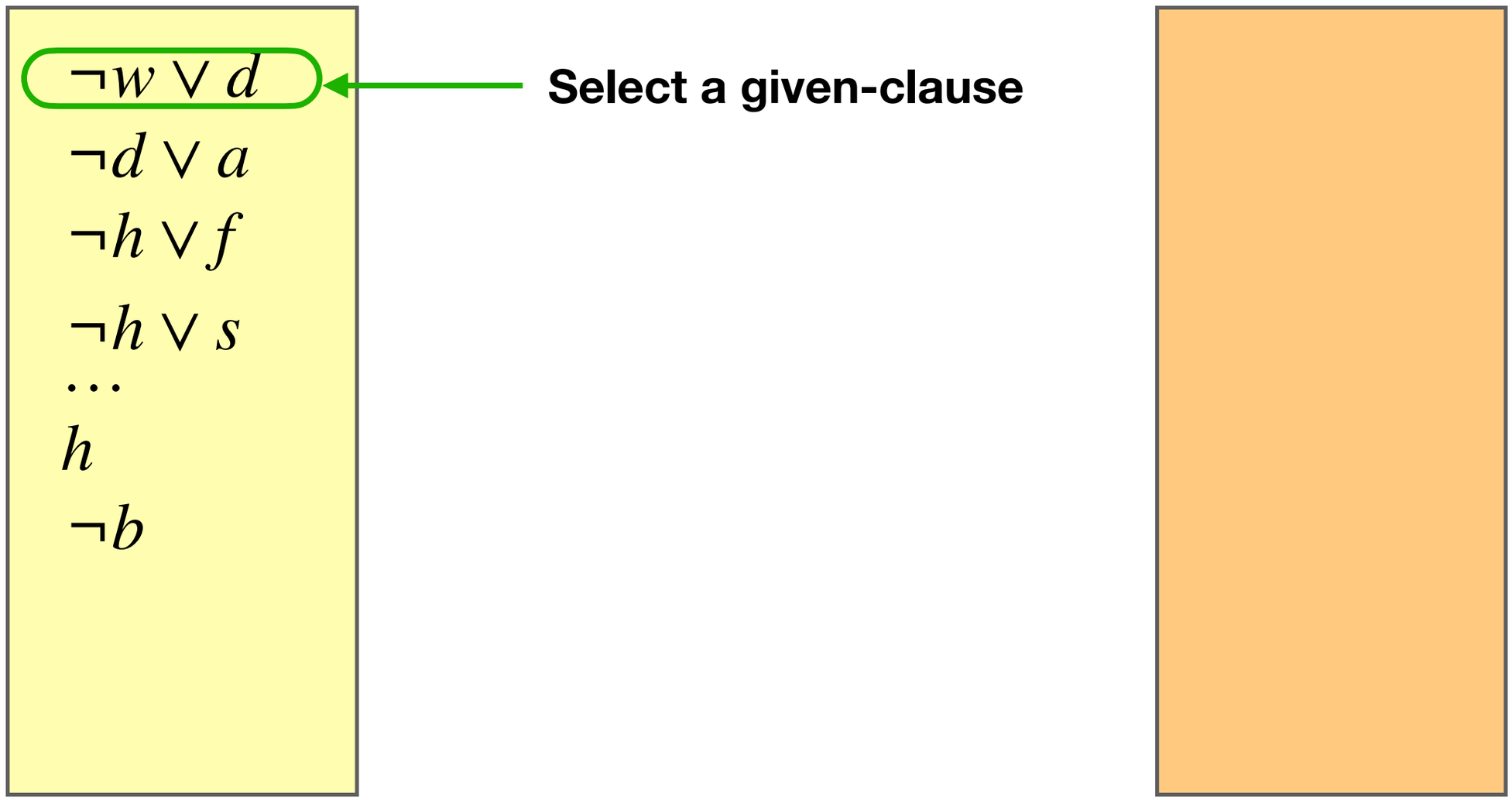


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

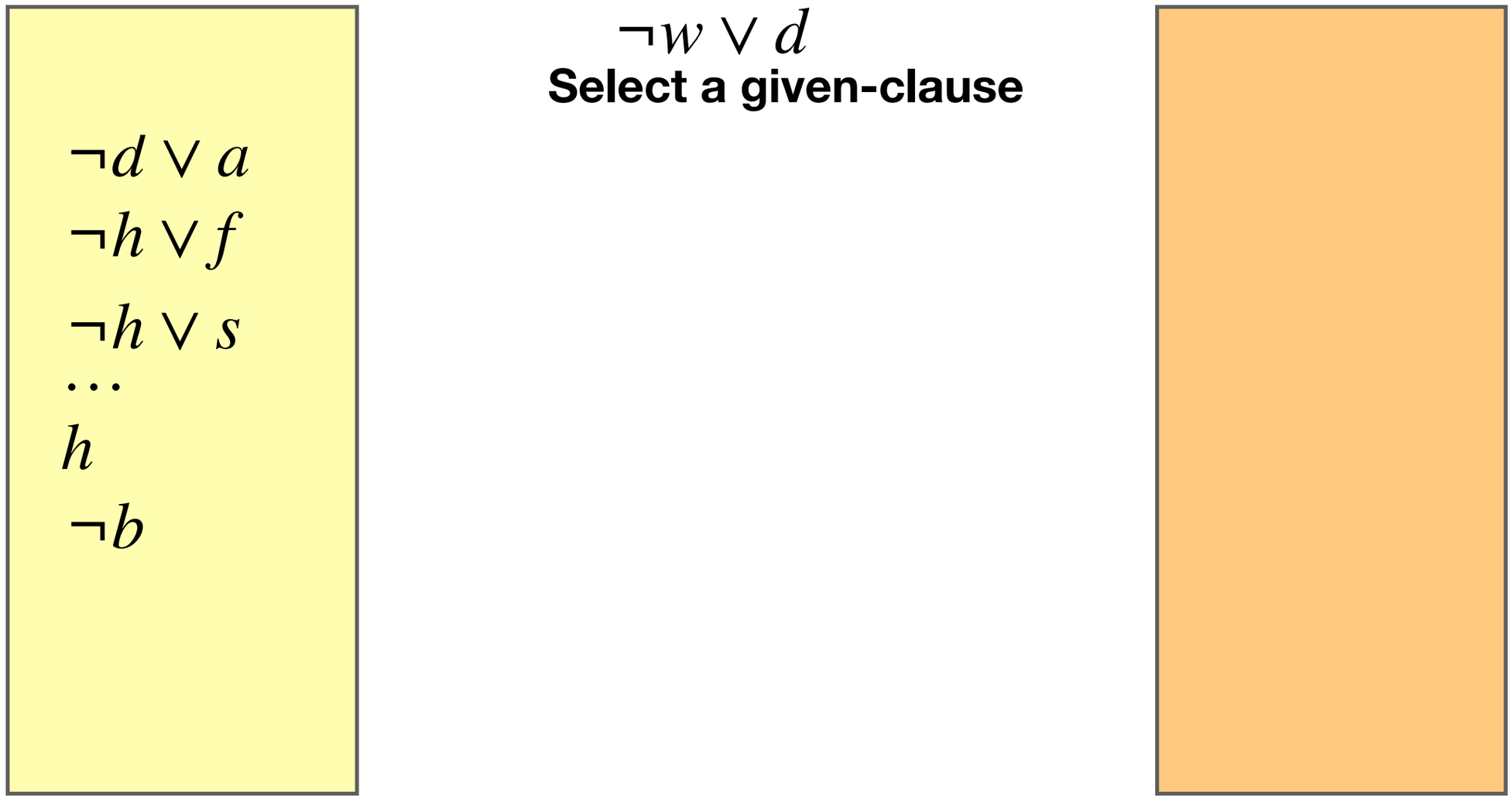


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

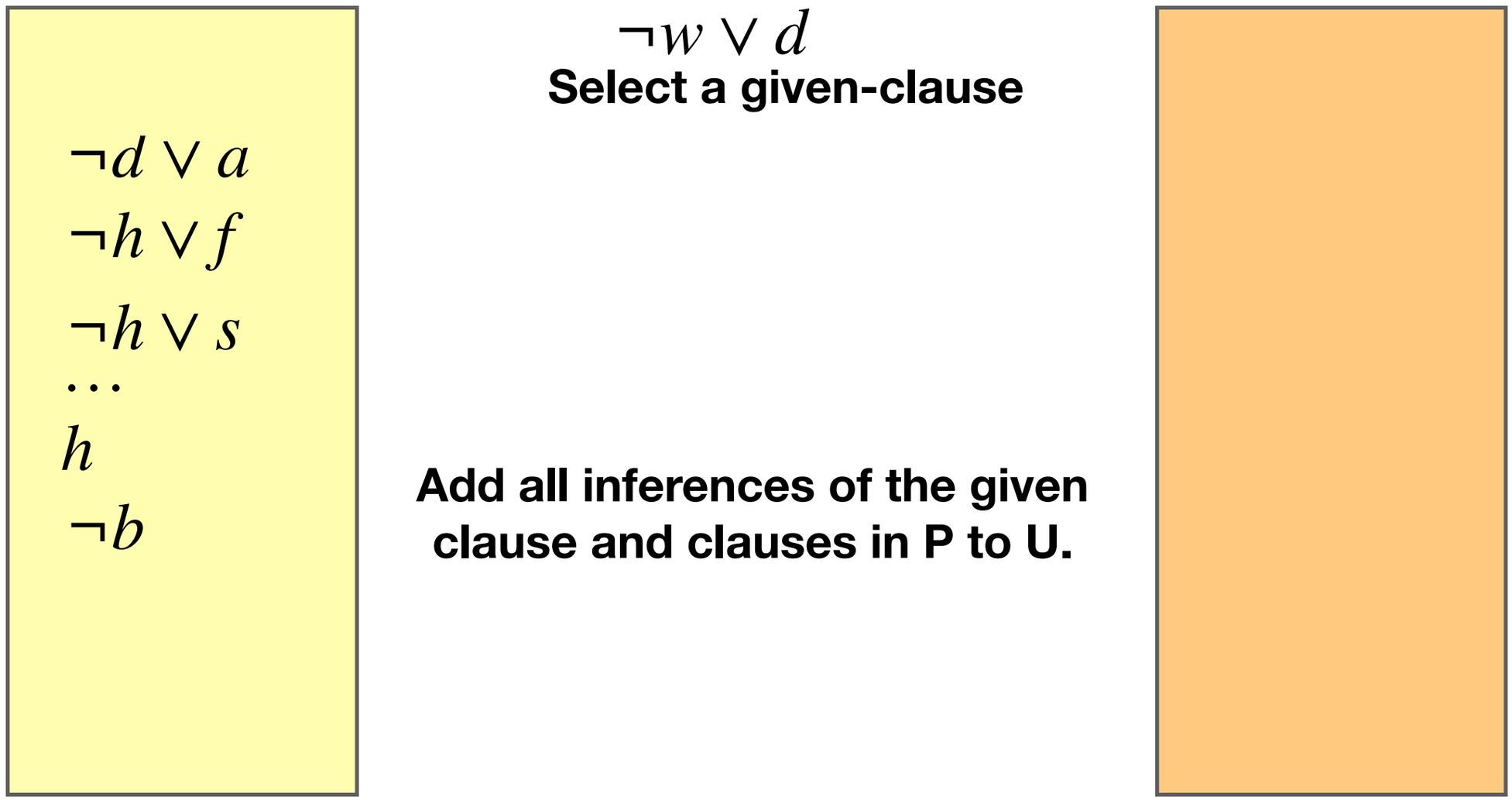


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

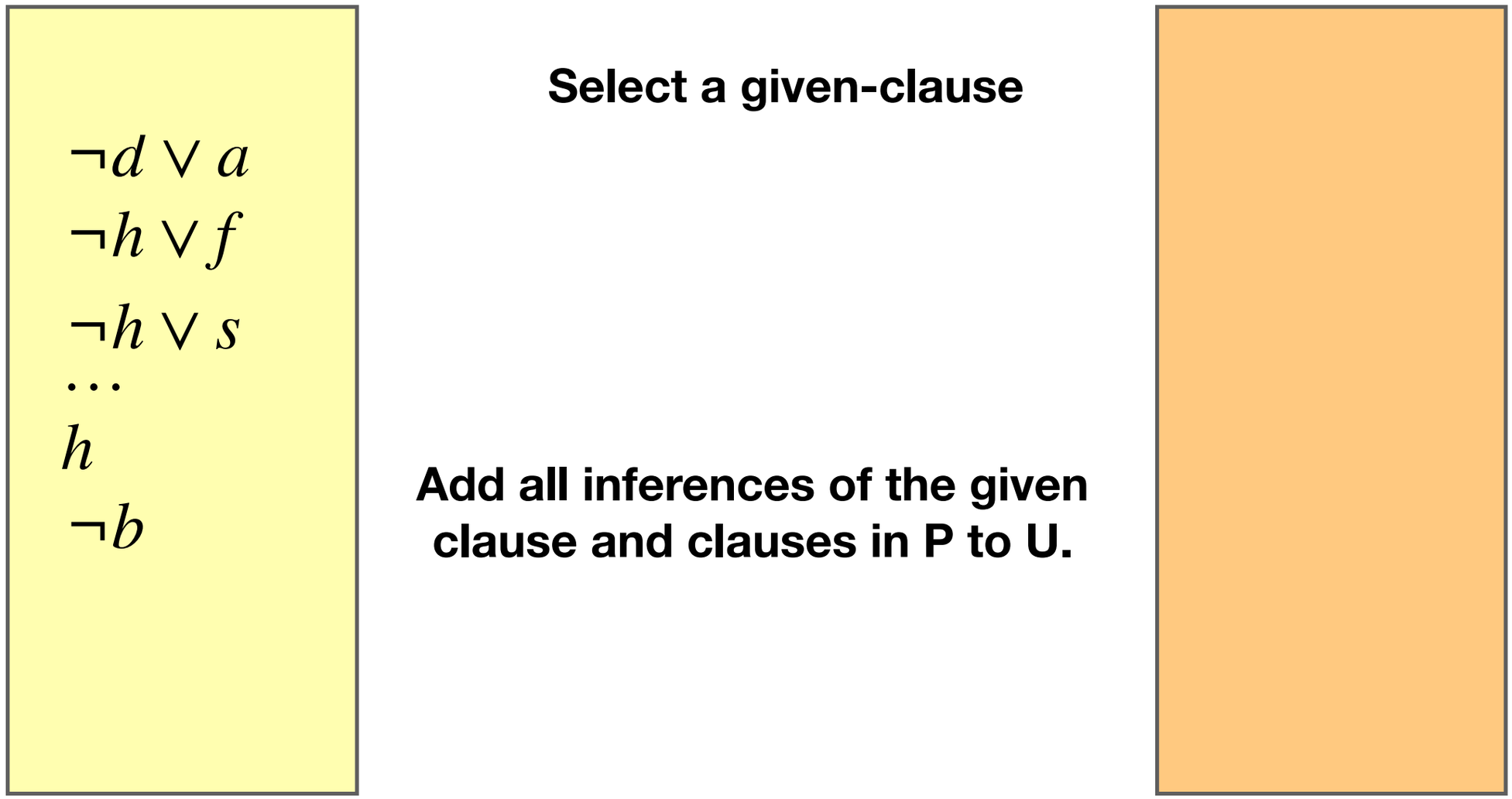


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

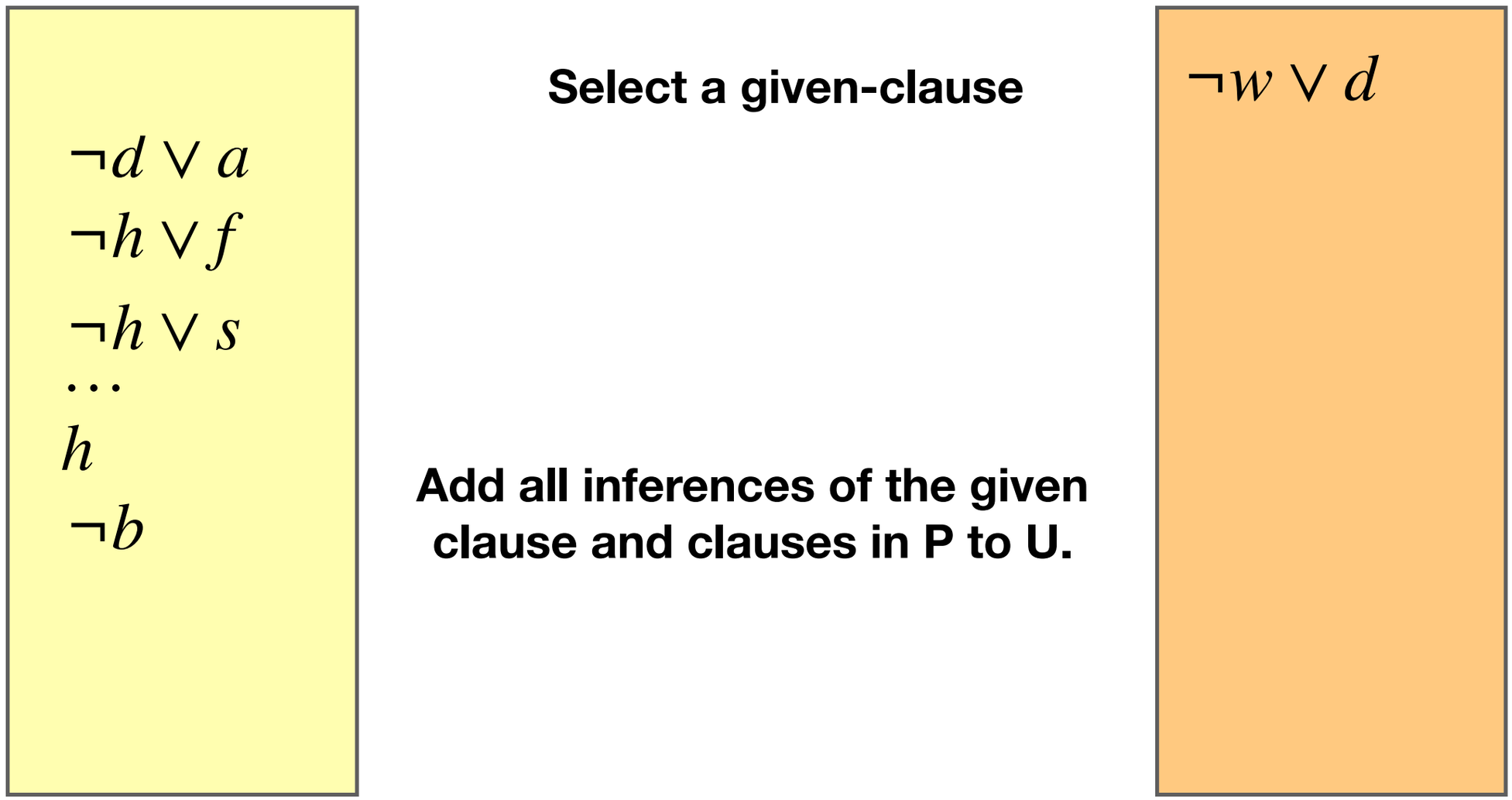


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

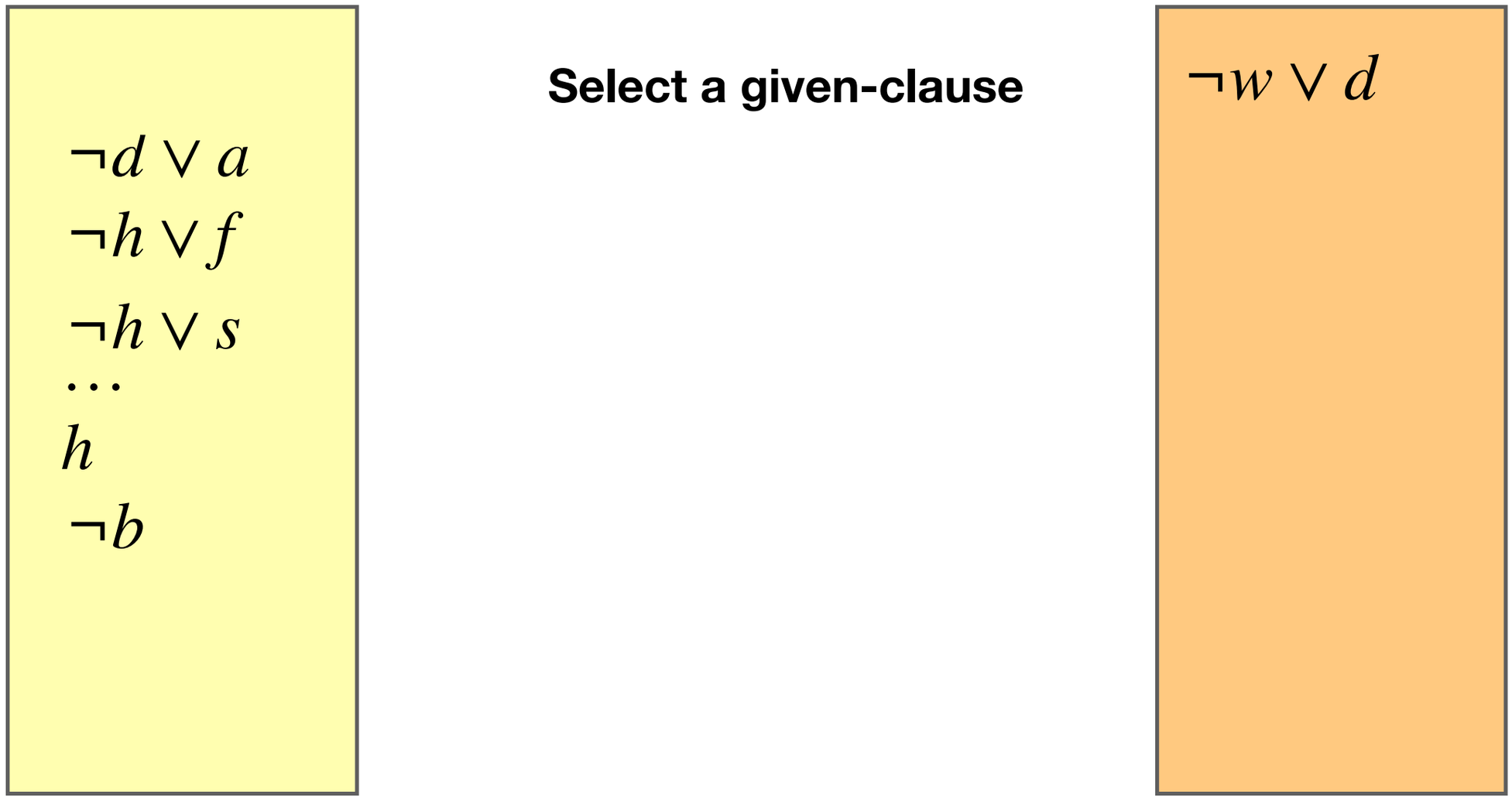


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

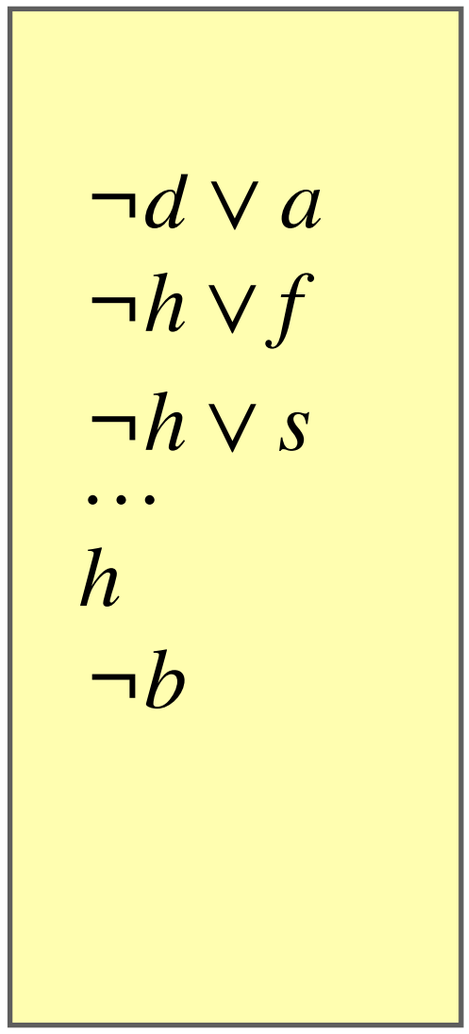


U: set of unprocessed clauses

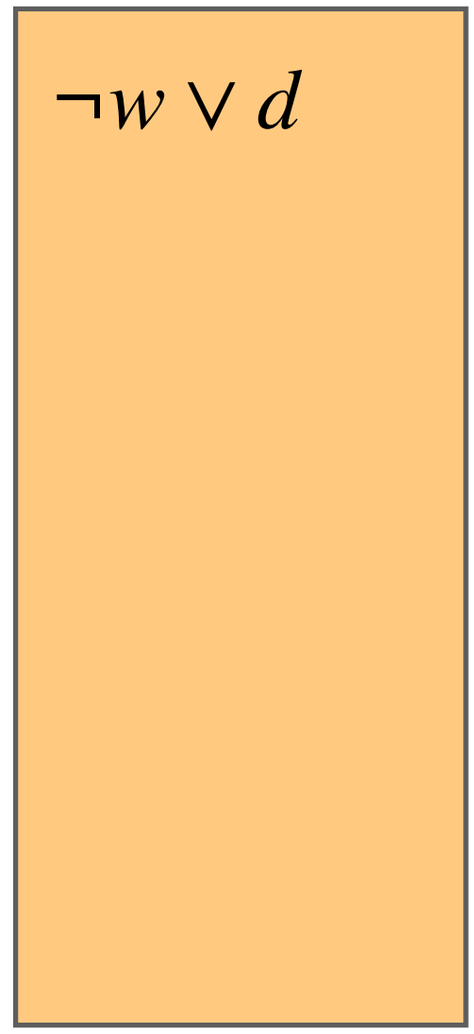
P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S



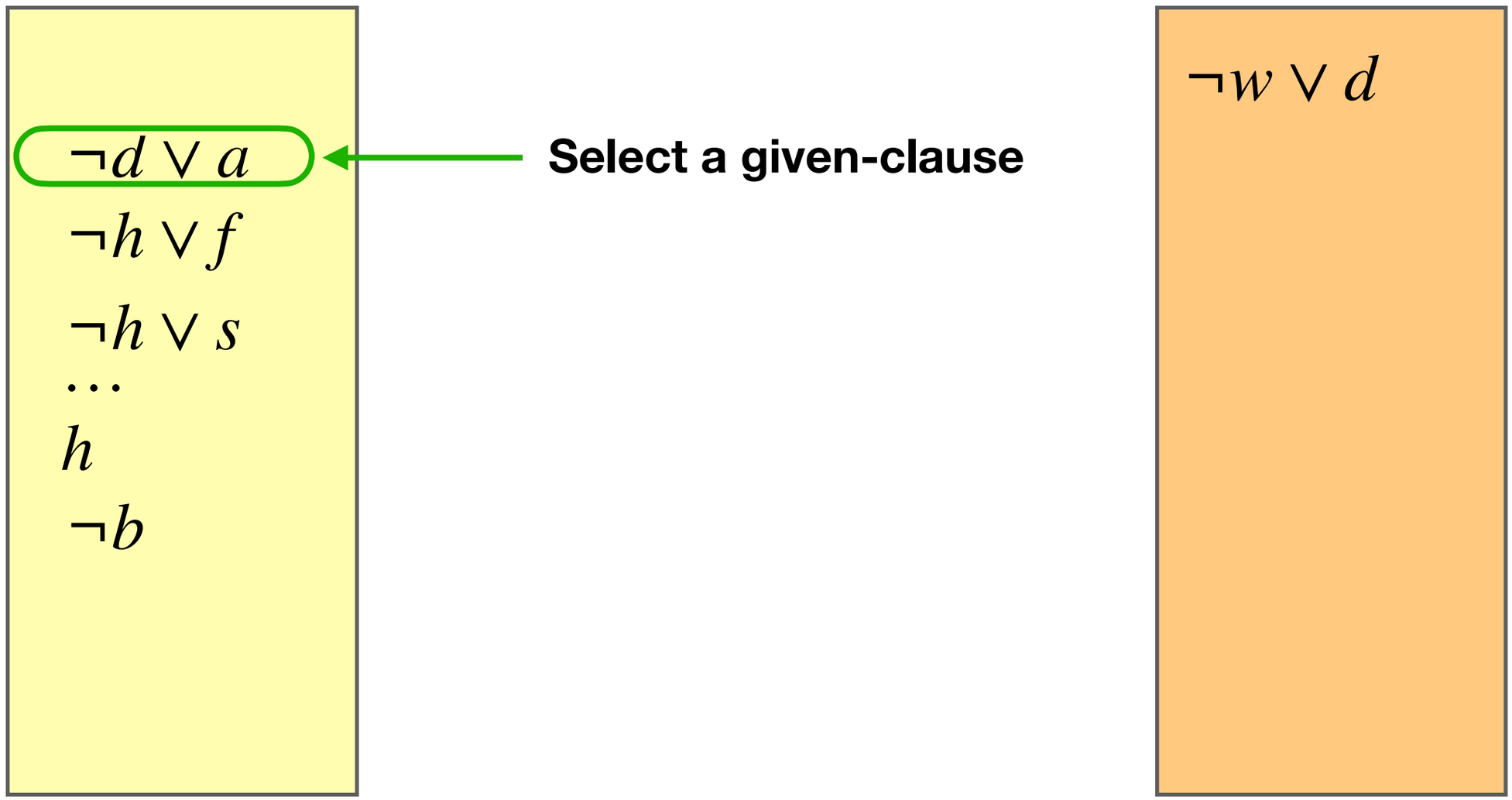
U: set of unprocessed clauses



P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

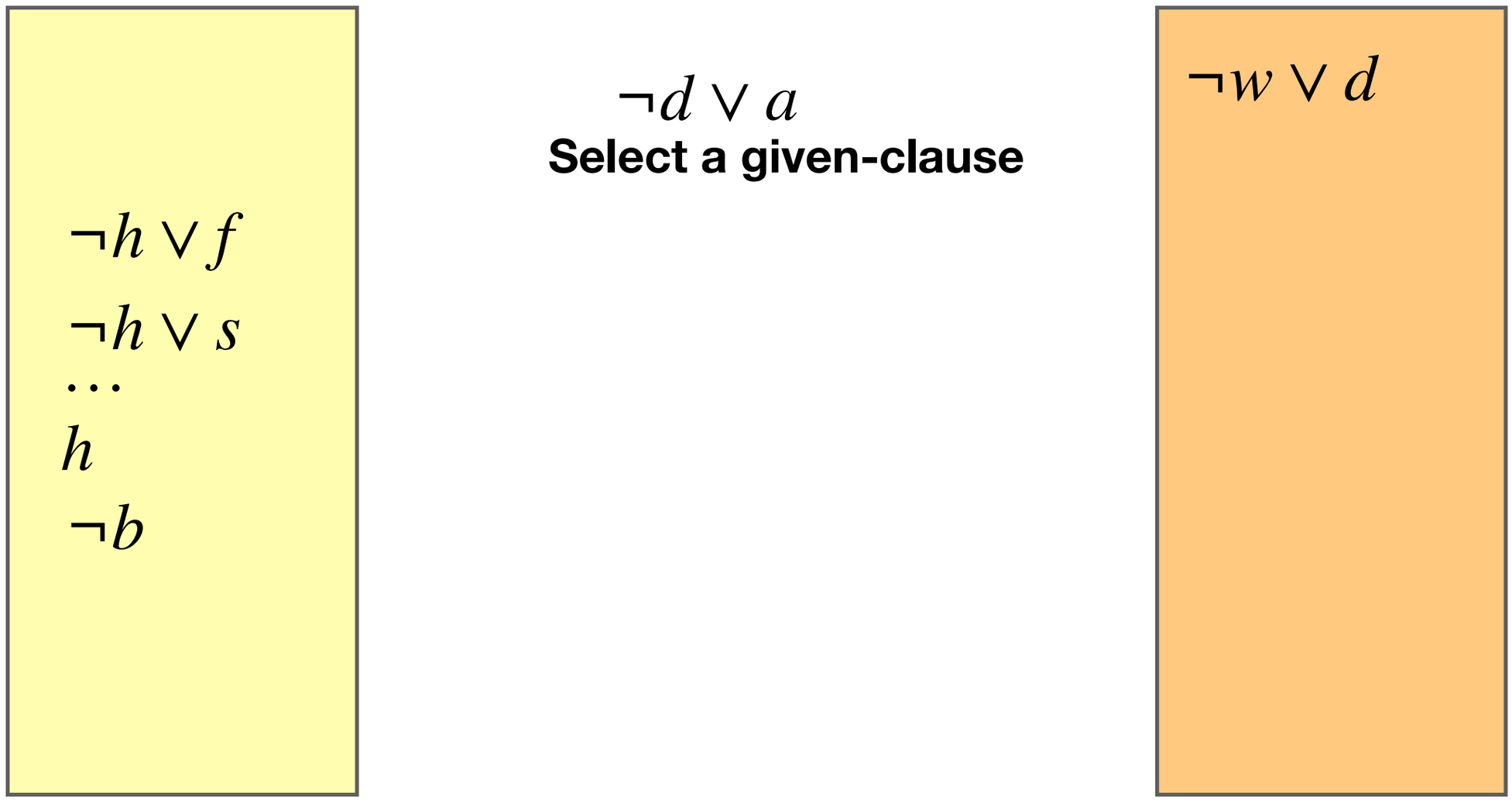


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

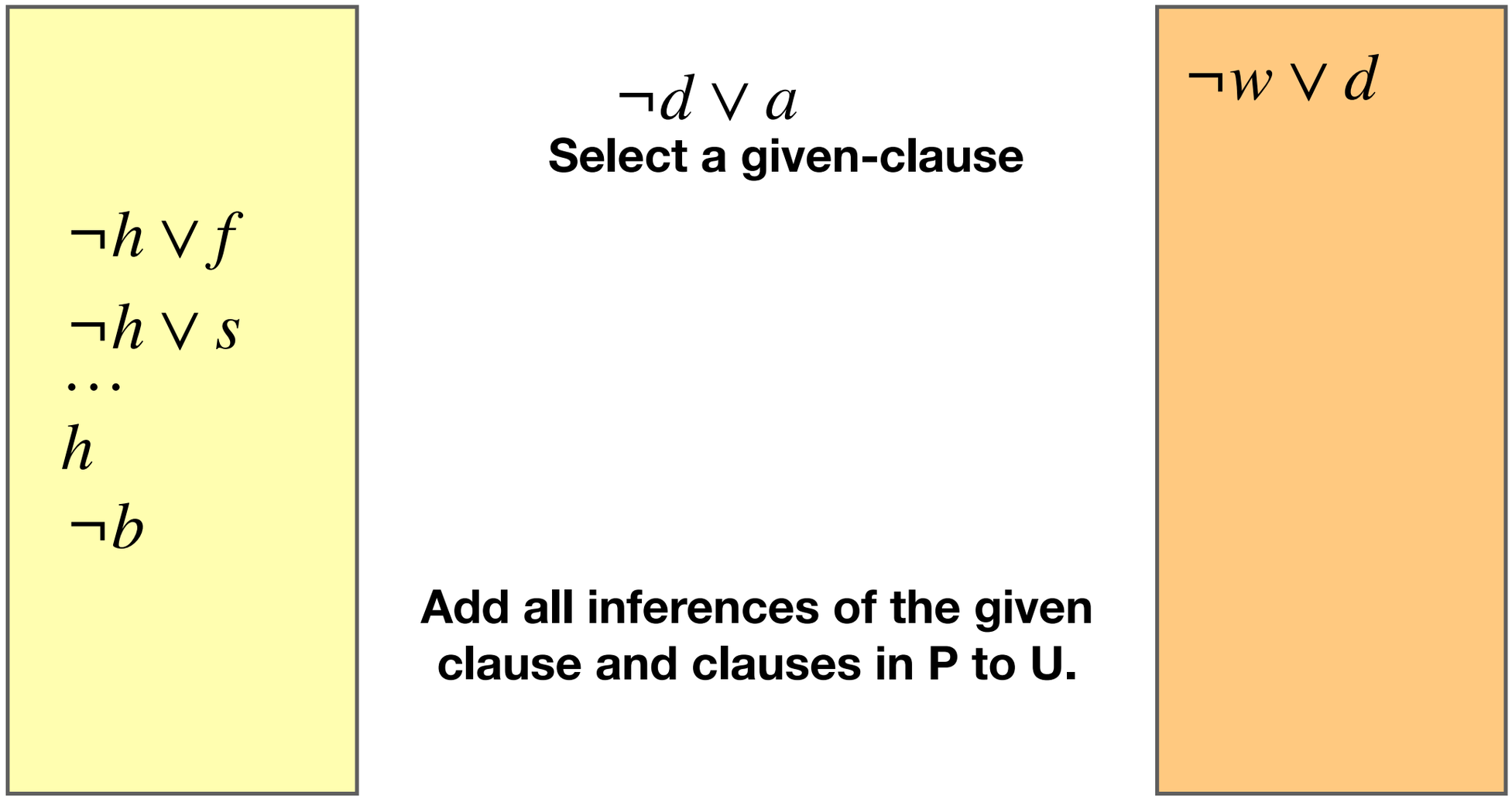


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

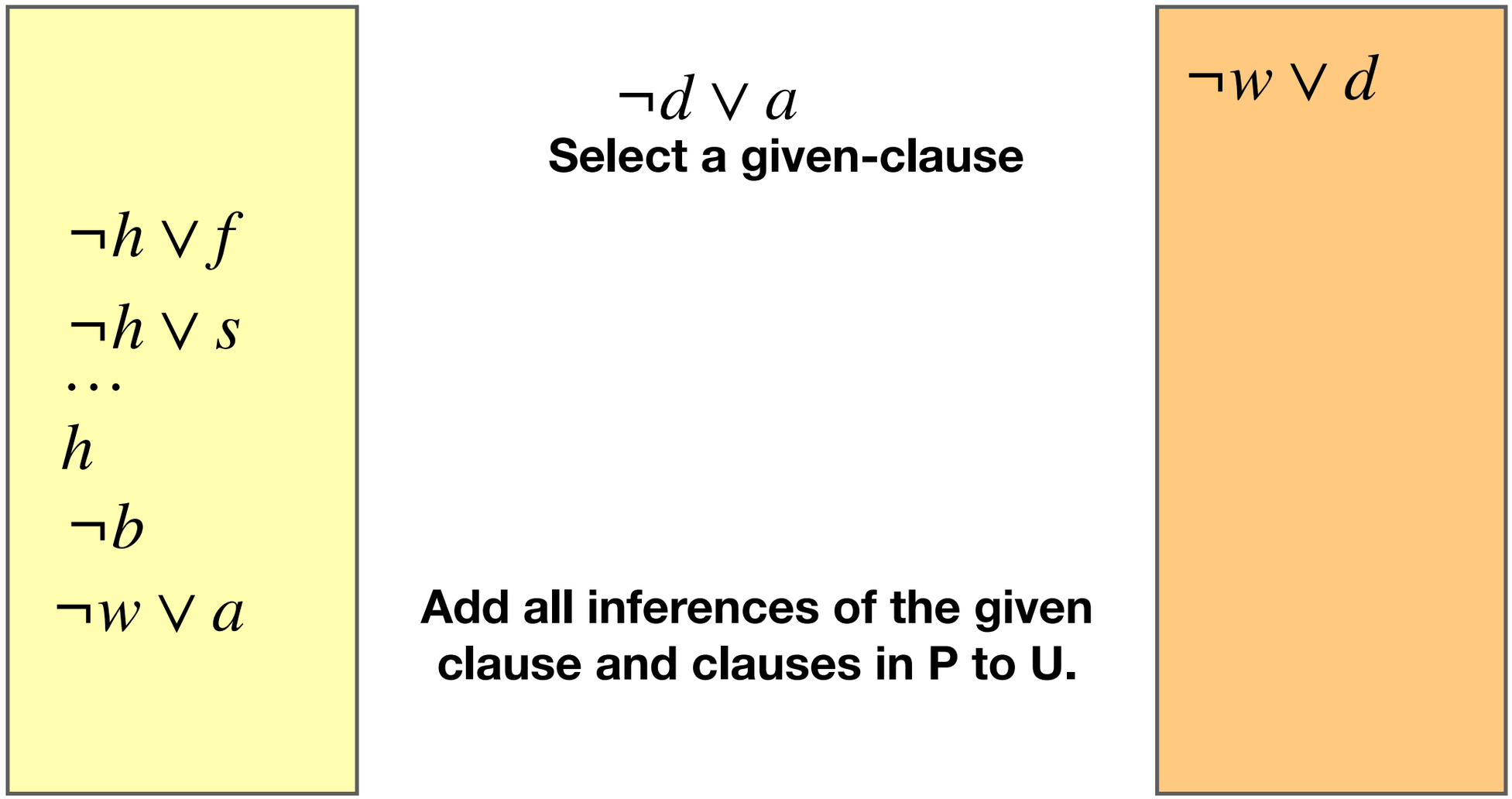


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

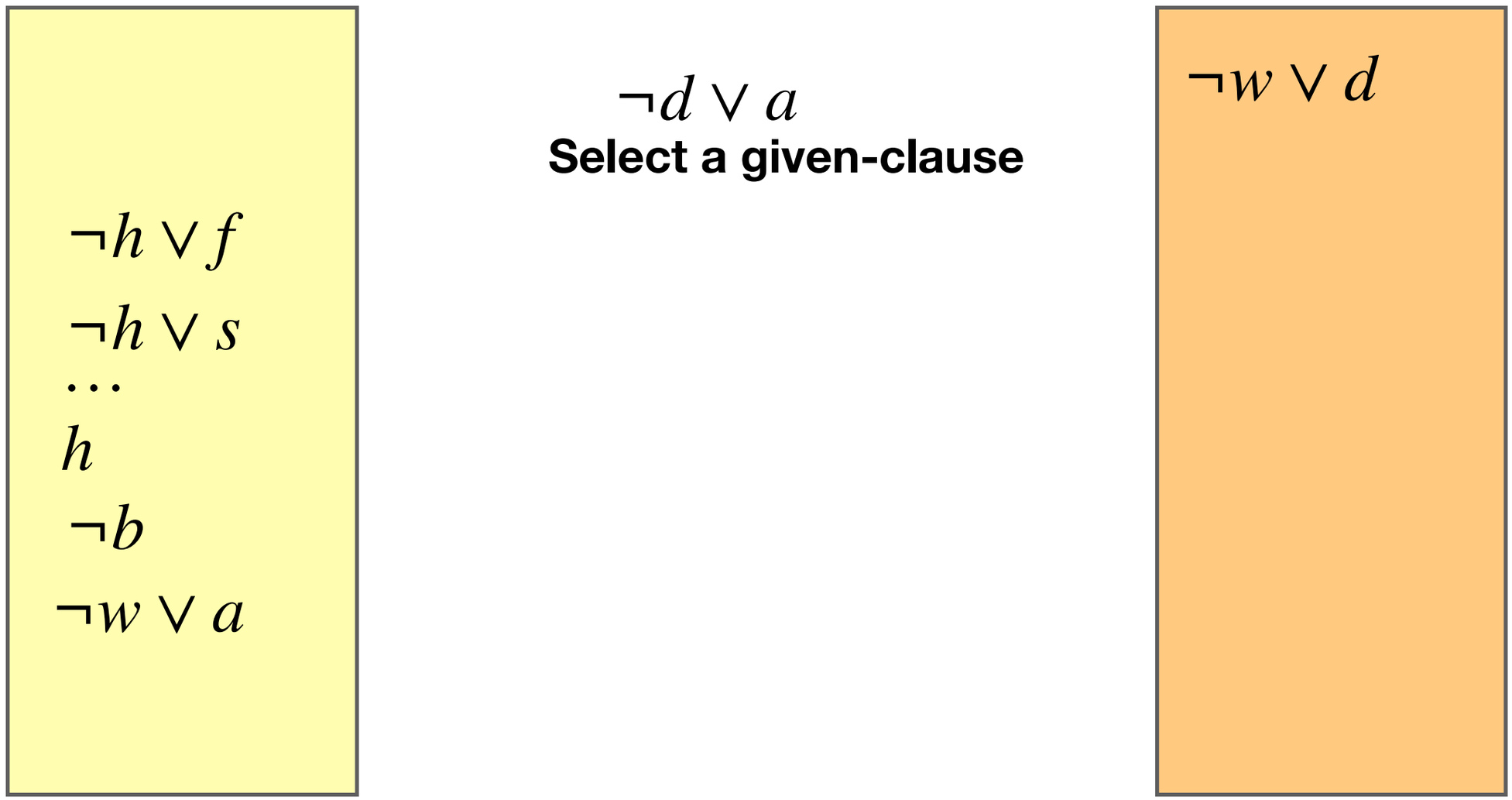


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

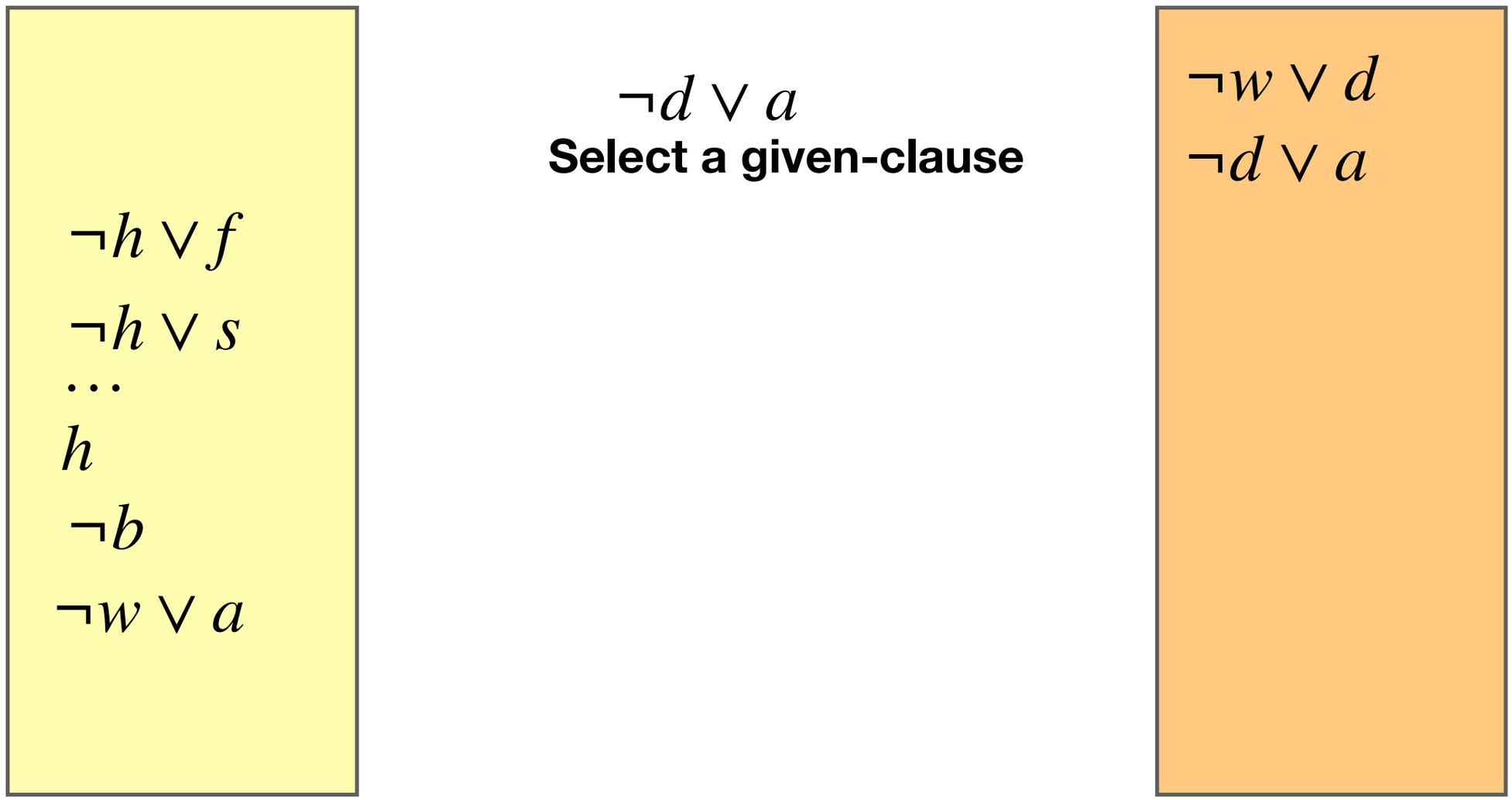


U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S



U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $h \rightarrow b$ follows from clause set S

$\neg h \vee f$

$\neg h \vee s$

$\cdots$

h

$\neg b$

$\neg w \vee a$

$\neg d \vee a$

Select a given-clause

$\neg w \vee d$

$\neg d \vee a$

U: set of unprocessed clauses

P: set of processed clauses

Selection of given clause using heuristics.
Does not take the meaning of symbol names into account.

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg weapon \vee device$
$\neg device \vee artifact$
$\neg hammock \vee fabric$
$\neg hammock \vee sleep$
…
$hammock$
$\neg bed$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg weapon \vee device$

$\neg device \vee artifact$

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

Select a given-clause

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg device \vee artifact$

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

…

$hammock$

$\neg bed$

$\neg weapon \vee device$

Select a given-clause

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg device \vee artifact$

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

…

$hammock$

$\neg bed$

$\neg weapon \vee device$

Select a given-clause

Add all inferences of the given clause and clauses in P to U.

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg device \vee artifact$
$\neg hammock \vee fabric$
$\neg hammock \vee sleep$
…
$hammock$
$\neg bed$

Select a given-clause

Add all inferences of the given clause and clauses in P to U.

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg device \vee artifact$

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

…

$hammock$

$\neg bed$

Select a given-clause

Add all inferences of the given clause and clauses in P to U.

$\neg weapon \vee device$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg device \vee artifact$

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

Select a given-clause

$\neg weapon \vee device$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg device \vee artifact$

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

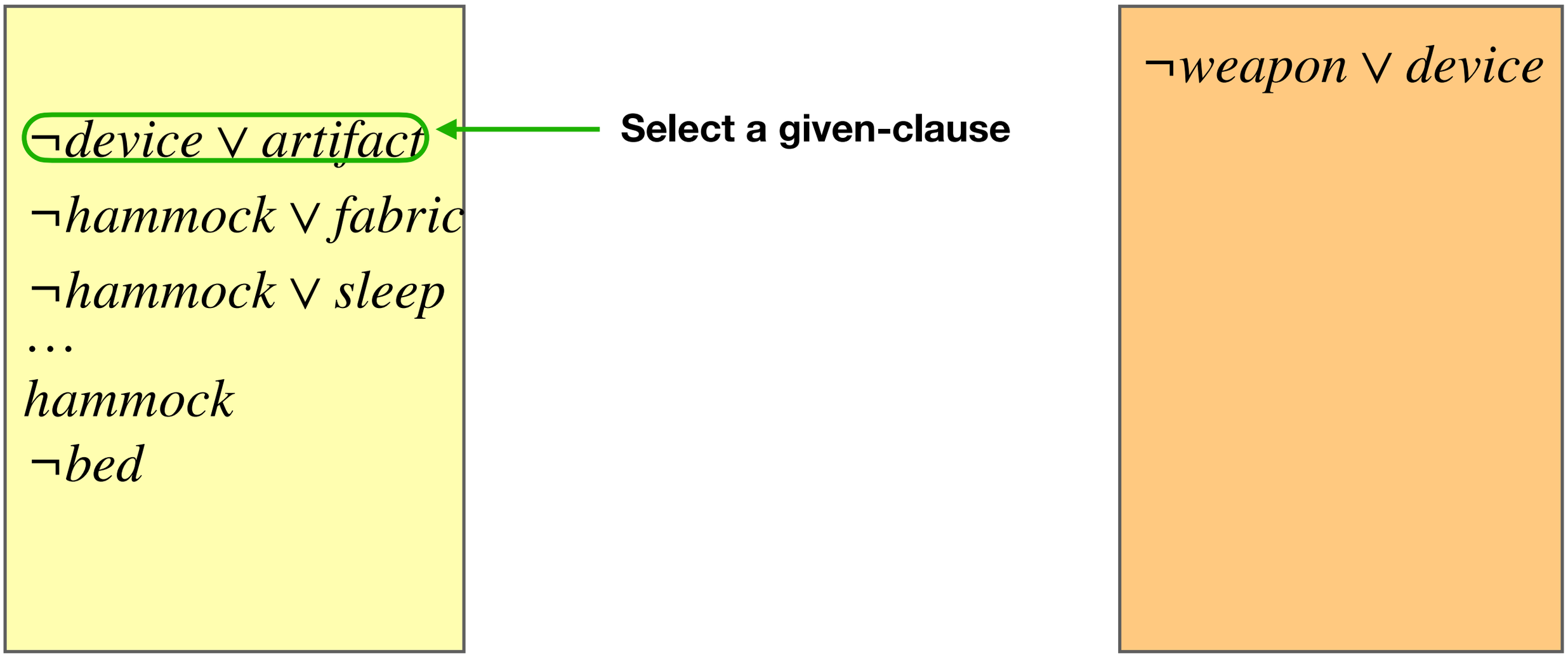
$\neg weapon \vee device$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses



U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

$\neg device \vee artifact$

Select a given-clause

$\neg weapon \vee device$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

$\neg device \vee artifact$

Select a given-clause

$\neg weapon \vee device$

$\neg device \vee artifact$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

$\neg device \vee artifact$

Select a given-clause

Add all inferences of the given clause and clauses in P to U.

$\neg weapon \vee device$

$\neg device \vee artifact$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

$\neg weapon \vee artifact$

$\neg device \vee artifact$

Select a given-clause

Add all inferences of the given clause and clauses in P to U.

$\neg weapon \vee device$

$\neg device \vee artifact$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

…

$hammock$

$\neg bed$

$\neg weapon \vee artifact$

$\neg device \vee artifact$

Select a given-clause

$\neg weapon \vee device$

$\neg device \vee artifact$

U: set of unprocessed clauses

P: set of processed clauses

Given-Clause Algorithmus

Task: proof that $hammock \rightarrow bed$ follows from the set of clauses

$\neg hammock \vee fabric$

$\neg hammock \vee sleep$

$\ldots$

$hammock$

$\neg bed$

$\neg weapon \vee artifact$

$\neg device \vee artifact$

Select a given-clause

$\neg weapon \vee device$

$\neg device \vee artifact$

U: set of unprocessed clauses

P: set of processed clauses

Idea: Use clauses with symbols similar to the proof task as given clause!

Symbol Name Heuristics

Idea: Use clauses with symbols similar to the proof goal as given clause!

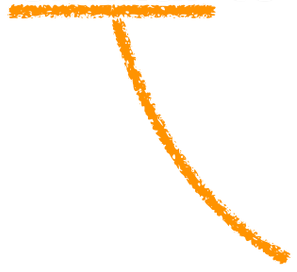
Symbol Name Heuristics

Idea: Use clauses with symbols similar to the proof goal as given clause!

How to determine similarity?

Symbol Name Heuristics

Idea: Use clauses with symbols similar to the proof goal as given clause!



How to determine similarity?

We use Word Embeddings!

Meaning of Symbol Names using Word Embeddings

“a word is characterized by the company it keeps” (Firth, 1957)

Word embedding: a function $f : Voc \rightarrow \mathbb{R}^n$

Measuring Similarities

Let $u, v \in \mathbb{R}^n$. The cosine similarity of u and v is defined as

$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u|| \quad ||v||}$$

assuming that both u and v are non-zero.

Measuring Similarities

Let $u, v \in \mathbb{R}^n$. The cosine similarity of u and v is defined as

$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u|| \ \ ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (v_i)^2}} = \cos(\phi)$$

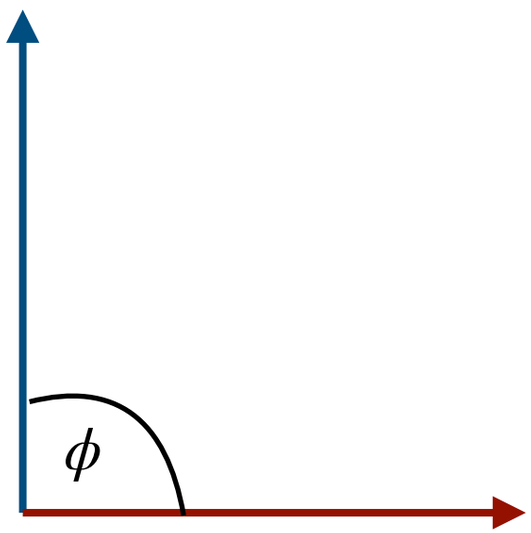
assuming that both u and v are non-zero.

Measuring Similarities

Let $u, v \in \mathbb{R}^n$. The cosine similarity of u and v is defined as

$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u|| \ \ ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (v_i)^2}} = \cos(\phi)$$

assuming that both u and v are non-zero.



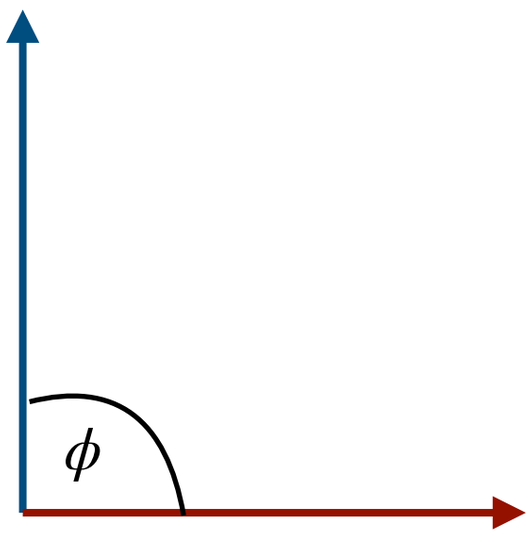
$\cos(\phi) = 0$

Measuring Similarities

Let $u, v \in \mathbb{R}^n$. The cosine similarity of u and v is defined as

$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u|| \ \ ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (v_i)^2}} = \cos(\phi)$$

assuming that both u and v are non-zero.



$\cos(\phi) = 0 = cosine\_similarity(u, v)$

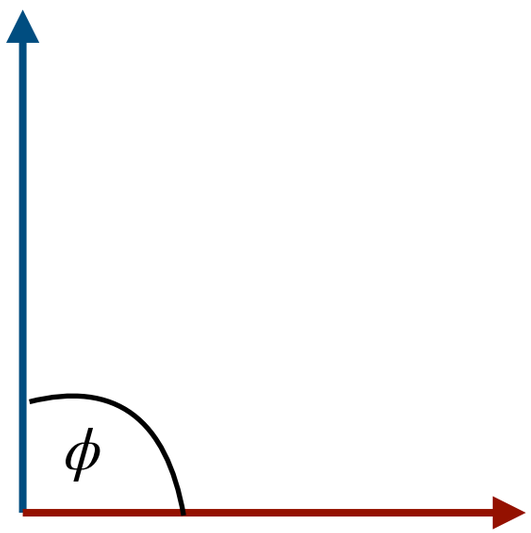
u and v are very different

Measuring Similarities

Let $u, v \in \mathbb{R}^n$. The cosine similarity of u and v is defined as

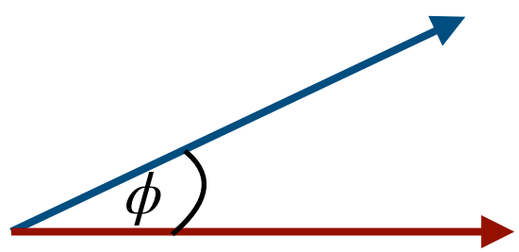
$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u||\ \ ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (v_i)^2}} = \cos(\phi)$$

assuming that both u and v are non-zero.



$\cos(\phi) = 0 = cosine\_similarity(u, v)$

u and v are very different



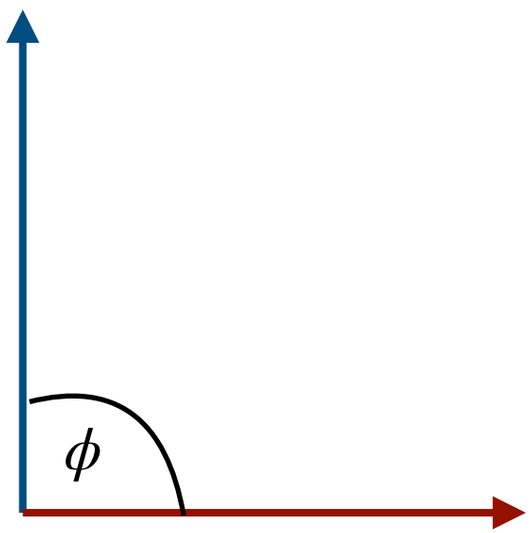
$\phi = 25$ $\cos(\phi) = 0.9$

Measuring Similarities

Let $u, v \in \mathbb{R}^n$. The cosine similarity of u and v is defined as

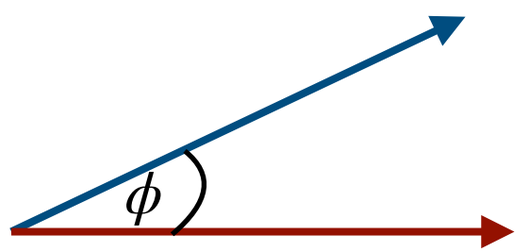
$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u||\ \ ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (v_i)^2}} = \cos(\phi)$$

assuming that both u and v are non-zero.



$\cos(\phi) = 0 = cosine\_similarity(u, v)$

u and v are very different



$\phi = 25$ $\cos(\phi) = 0.9 = cosine\_similarity(u, v)$

u and v are similar

Word Embeddings: Similar Words

Comparing words to the word *bed* in the Numberbatch word embedding:

Word	Cosine Similarity to *bed*
bed	1.0000
pillow	0.6285
sleep	0.6225
blanket	0.4787
vehicle	0.1025
weapon	0.0201

Word Embeddings: Similar Words

Comparing words to the word *bed* in the Numberbatch word embedding:

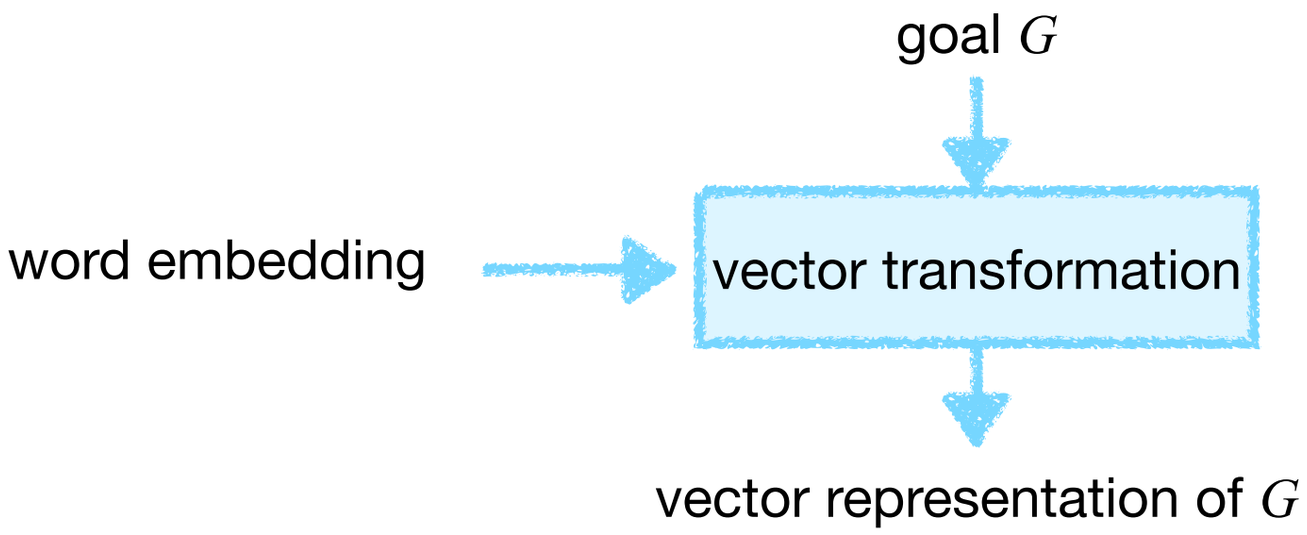
Word	Cosine Similarity to *bed*
bed	1.0000
pillow	0.6285
sleep	0.6225
blanket	0.4787
vehicle	0.1025
weapon	0.0201

We use these similarities to select the given clause!

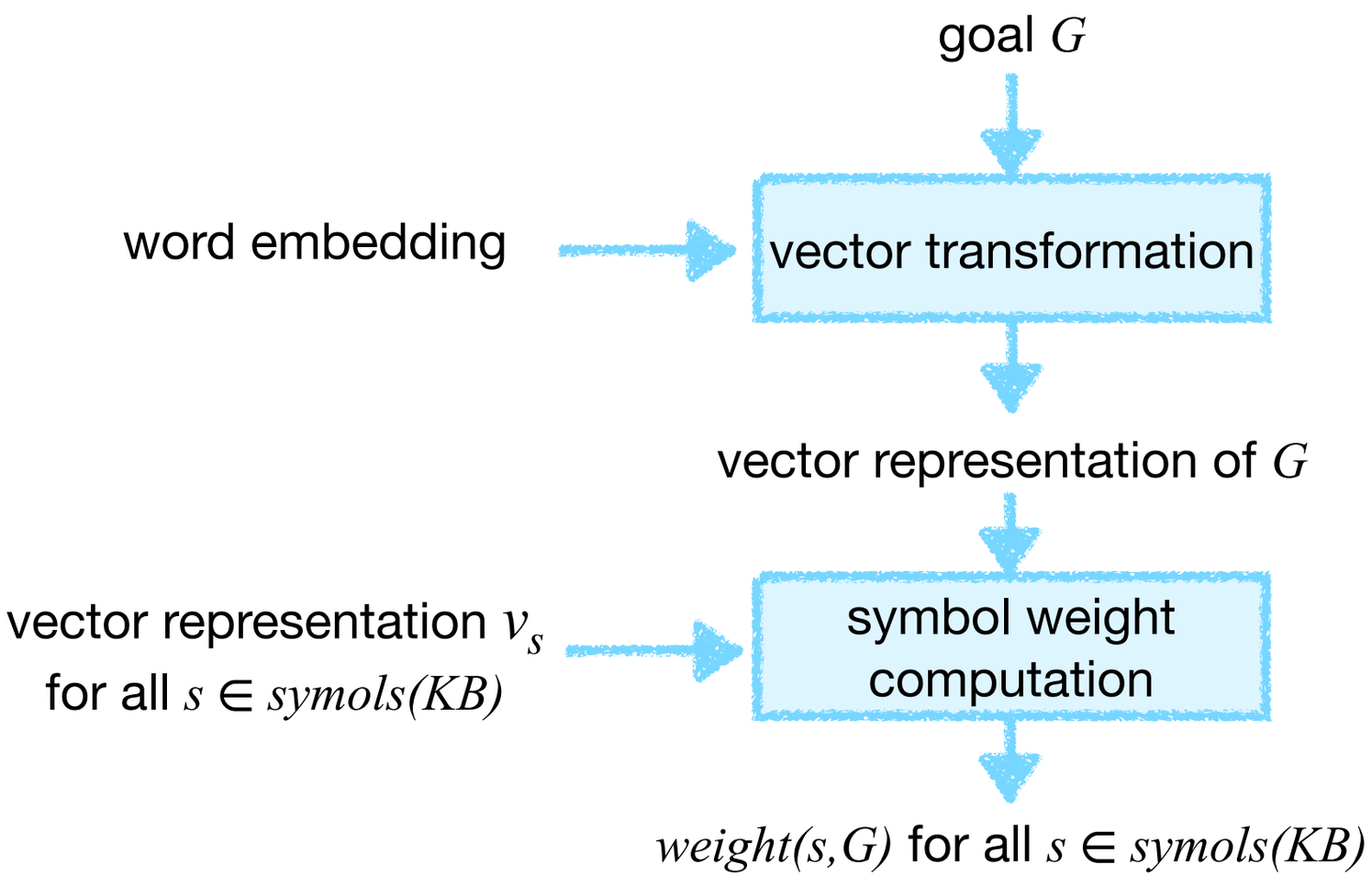
Symbol Name Heuristic

goal G

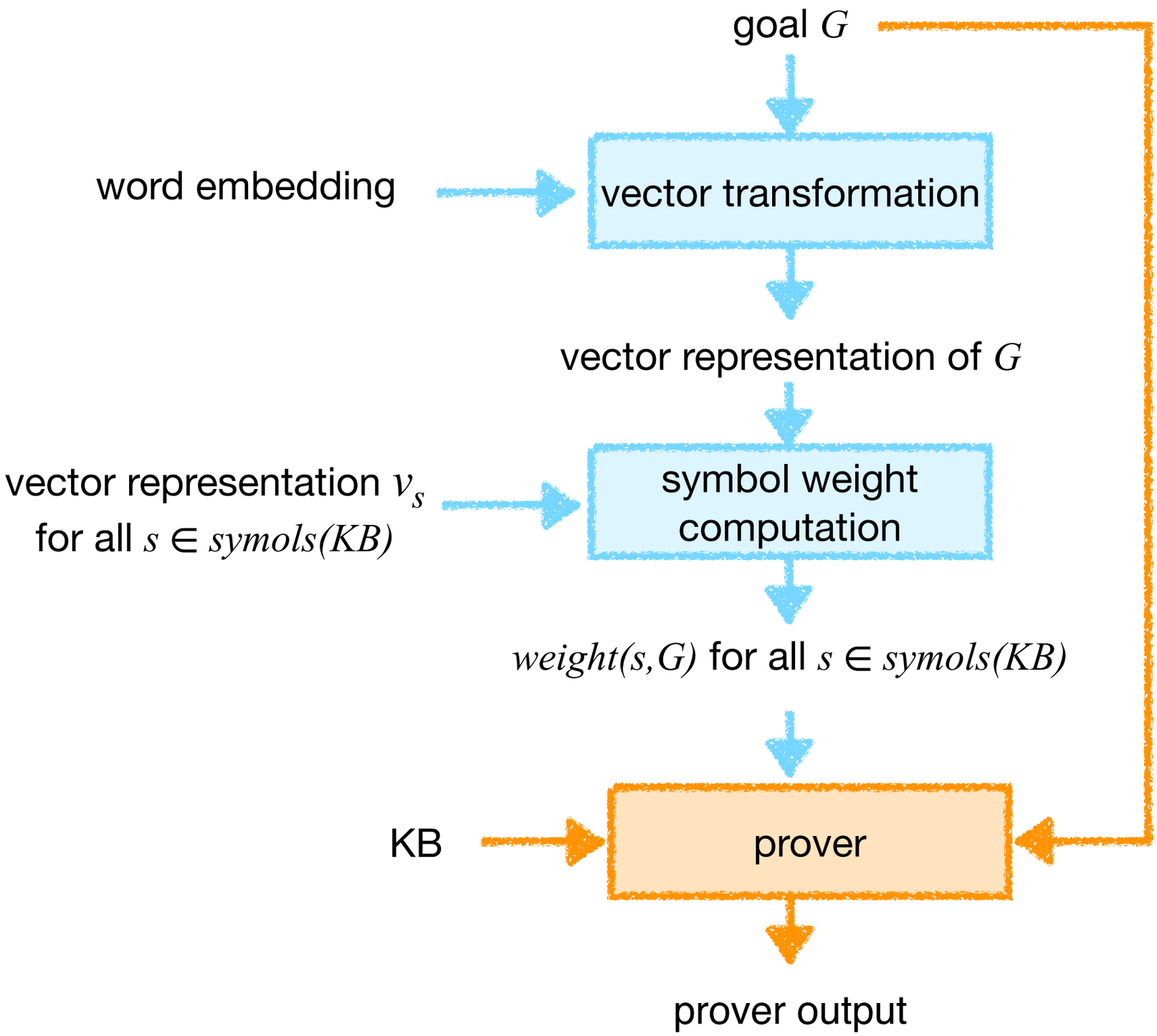
Symbol Name Heuristic



Symbol Name Heuristic



Symbol Name Heuristic



Representing Symbols and Formulae by Vectors

$symbols(F)$: the set of predicate and function symbols in formula F

Representing Symbols and Formulae by Vectors

$symbols(F)$: the set of predicate and function symbols in formula F

$f : Voc \to \mathbb{R}^n$ a word embedding with vocabulary Voc

Representing Symbols and Formulae by Vectors

$symbols(F)$: the set of predicate and function symbols in formula F

$f : Voc \to \mathbb{R}^n$ a word embedding with vocabulary Voc

assume $symbols(F) \subseteq Voc$

Representing Symbols and Formulae by Vectors

$symbols(F)$: the set of predicate and function symbols in formula F

$f : Voc \to \mathbb{R}^n$ a word embedding with vocabulary Voc

assume $symbols(F) \subseteq Voc$

The vector representation of a symbol s is

$$v_s = f(s)$$

Representing Symbols and Formulae by Vectors

$symbols(F)$: the set of predicate and function symbols in formula F

$f : Voc \to \mathbb{R}^n$ a word embedding with vocabulary Voc

assume $symbols(F) \subseteq Voc$

The vector representation of a symbol s is

$$v_s = f(s)$$

The vector representation of a formula F is

$$v_F = \frac{\sum_{s \in symbols(F)} v_s}{|symbols(F)|}$$

Symbol Weight Computation

$symbols(F)$: the set of predicate and function symbols in formula F

$symbols(KB)$: the set of predicate and function symbols occurring in KB

Symbol Weight Computation

$symbols(F)$: the set of predicate and function symbols in formula F

$symbols(KB)$: the set of predicate and function symbols occurring in KB

v_G the vector representation of goal G
v_s the vector representation of symbol s

Symbol Weight Computation

$symbols(F)$: the set of predicate and function symbols in formula F

$symbols(KB)$: the set of predicate and function symbols occurring in KB

v_G the vector representation of goal G
v_s the vector representation of symbol s

The weight of symbol s w.r.t goal G is defined as:

$$weight(s, G) = 1000 - 1000 \cdot cosine\_similarity(v_s, v_G)$$

Symbol Weight Computation

$symbols(F)$: the set of predicate and function symbols in formula F

$symbols(KB)$: the set of predicate and function symbols occurring in KB

v_G the vector representation of goal G
v_s the vector representation of symbol s

The weight of symbol s w.r.t goal G is defined as:

$$weight(s, G) = 1000 - 1000 \cdot cosine\_similarity(v_s, v_G)$$

Provers use clauses with symbols with a low weight first.

Symbol Weight Computation

$symbols(F)$: the set of predicate and function symbols in formula F

$symbols(KB)$: the set of predicate and function symbols occurring in KB

v_G the vector representation of goal G
v_s the vector representation of symbol s

The weight of symbol s w.r.t goal G is defined as:

$$weight(s, G) = 1000 - 1000 \cdot cosine\_similarity(v_s, v_G)$$

Provers use clauses with symbols with a low weight first.

We use default weights of 1000 for all symbols.

Selection of Given-Clause

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$

$subClass(bed, furniture)$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses: $subClass(weapon, device)$ $subClass(bed, furniture)$

Cosine similarity to the goal:

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$

$subClass(bed, furniture)$

Cosine similarity to the goal:

0.1232

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\qquad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\qquad$ 0.1232 $\quad$ 0.0422

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\qquad$ $subClass(bed, furniture)$

Cosine similarity to the goal:

0.1232 $\quad$ 0.0422 $\quad$ 0.1933

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232 0.7261

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232 0.7261 0.3323

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232 0.7261 0.3323

Symbol weight:

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232 0.7261 0.3323

Symbol weight:

$$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232 0.7261 0.3323

Symbol weight: $\quad$ 876.8

$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $\quad$ $subClass(bed, furniture)$

Cosine similarity to the goal: $\quad$ 0.1232 0.0422 0.1933 $\quad$ 0.1232 0.7261 0.3323

Symbol weight: $\quad$ 876.8 957.8

$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $subClass(bed, furniture)$

Cosine similarity to the goal: 0.1232 0.0422 0.1933 0.1232 0.7261 0.3323

Symbol weight: 876.8 957.8 806.7

$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

$subClass(weapon, device)$ $subClass(bed, furniture)$

Cosine similarity to the goal: 0.1232 0.0422 0.1933 0.1232 0.7261 0.3323

Symbol weight: 876.8 957.8 806.7 876.8

$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

Cosine similarity to the goal:

Symbol weight:

$subClass(weapon, device)$			$subClass(bed, furniture)$		
0.1232	0.0422	0.1933	0.1232	0.7261	0.3323
876.8	957.8	806.7	876.8	273.9	

$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

Cosine similarity to the goal:

Symbol weight:

$subClass(weapon, device)$			$subClass(bed, furniture)$		
0.1232	0.0422	0.1933	0.1232	0.7261	0.3323
876.8	957.8	806.7	876.8	273.9	667.7

$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

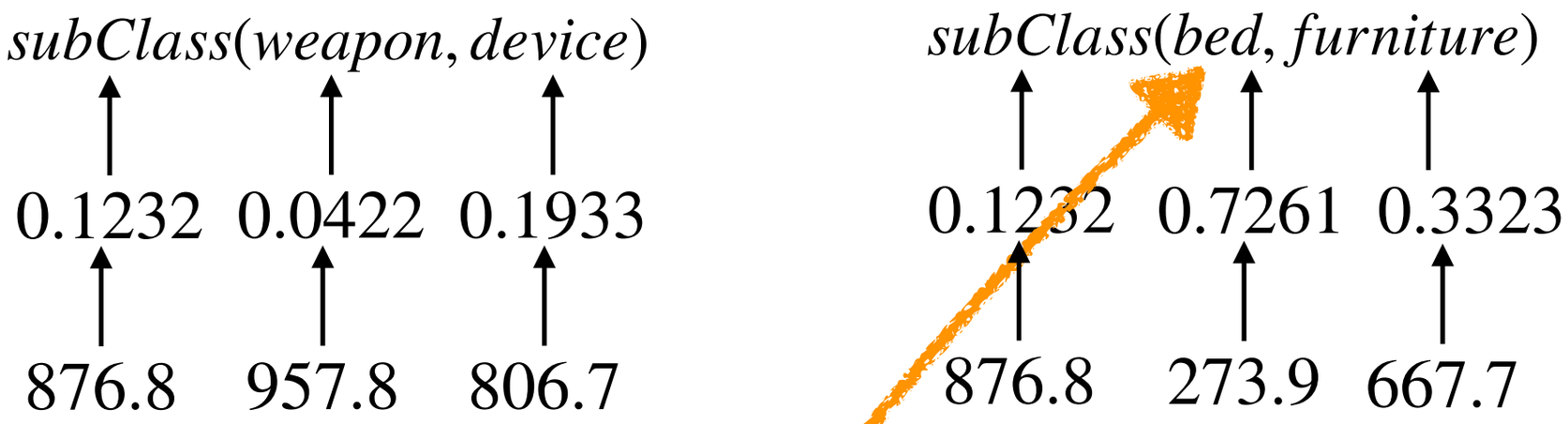
Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

Cosine similarity to the goal:

Symbol weight:



$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

Given clause = clause with smallest similarity-clause-weight

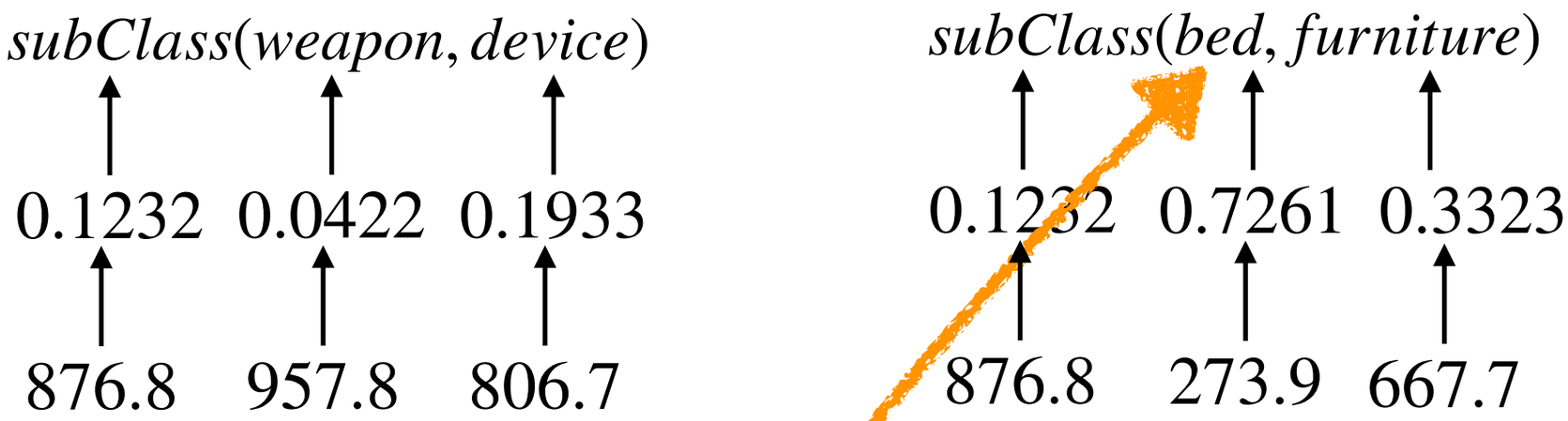
Selection of Given-Clause

Goal:
Are hammocks beds?
$\forall x(instance(x, hammock) \rightarrow instance(x, bed))$

Clauses:

Cosine similarity to the goal:

Symbol weight:

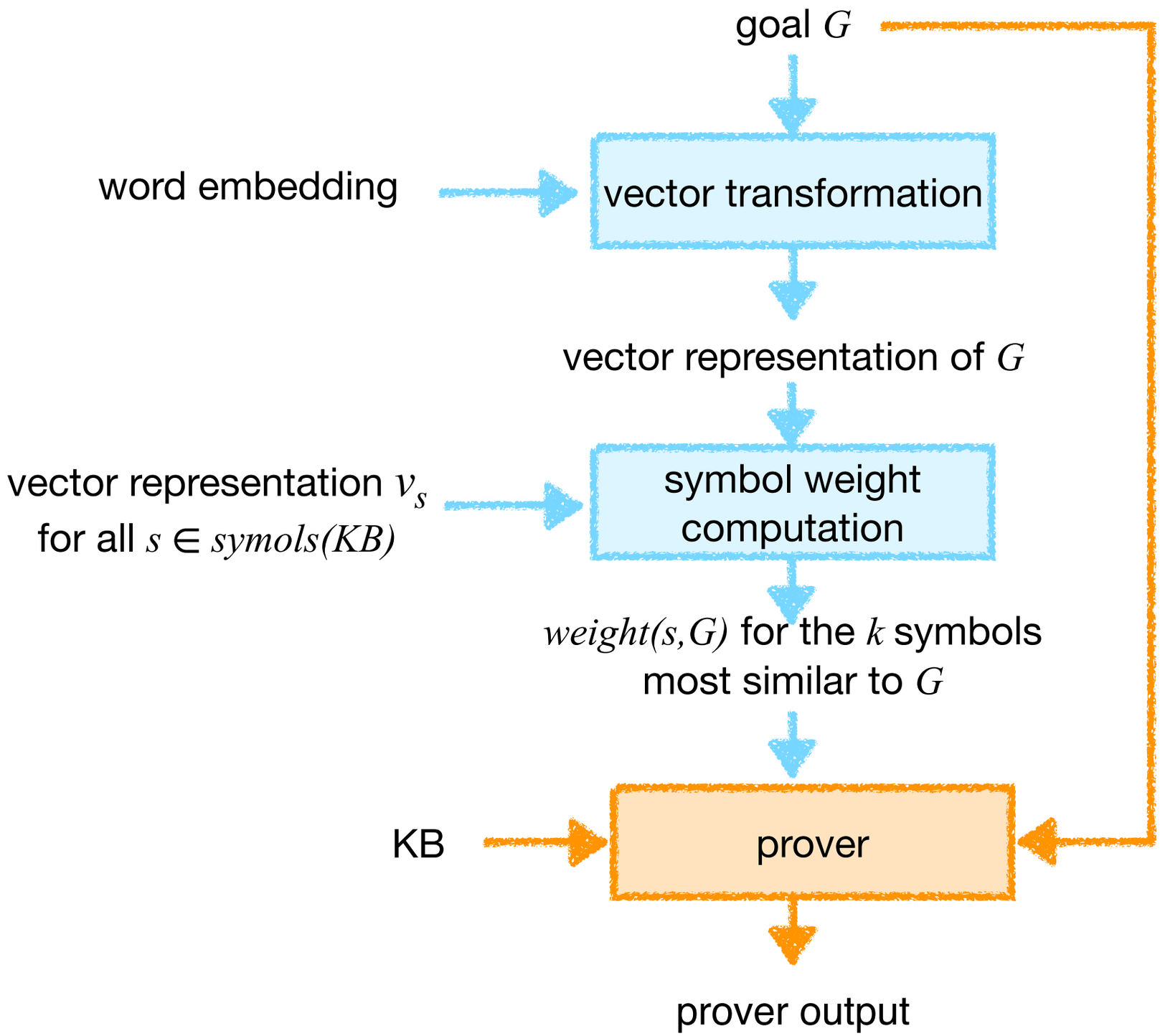


$1000 - 1000 \cdot cosine\_similarity(v_{symbol}, v_{goal})$

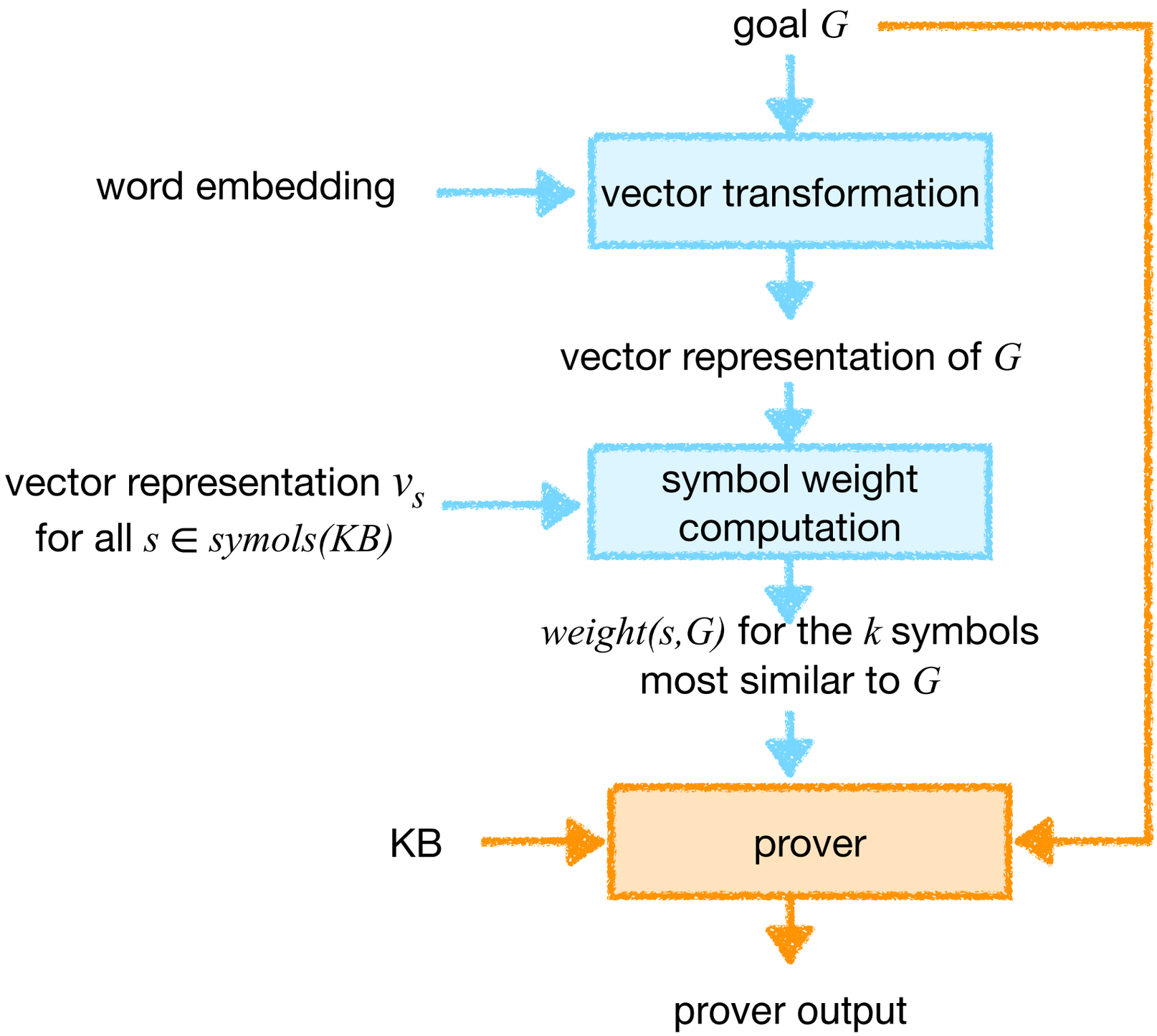
Given clause = clause with smallest similarity-clause-weight

We interleave the selection of the given clause by similarity-clause-weight with the selection of the oldest clause.

Symbol Name Heuristic



Symbol Name Heuristic



***k* is a parameter**

Experimental Results

- Adimen SUMO WhiteBoxTruthTests:

Experimental Results

- Adimen SUMO WhiteBoxTruthTests:
  - Collection of automatically generated tests

Experimental Results

- Adimen SUMO WhiteBoxTruthTests:
  - Collection of automatically generated tests
  - Each task: Goal for which it must be proved that it follows form the Adimen SUMO ontology

Experimental Results

- Adimen SUMO WhiteBoxTruthTests:
  - Collection of automatically generated tests
  - Each task: Goal for which it must be proved that it follows form the Adimen SUMO ontology
  - We randomly chose two disjoint sets of 1,000 of these proof tasks

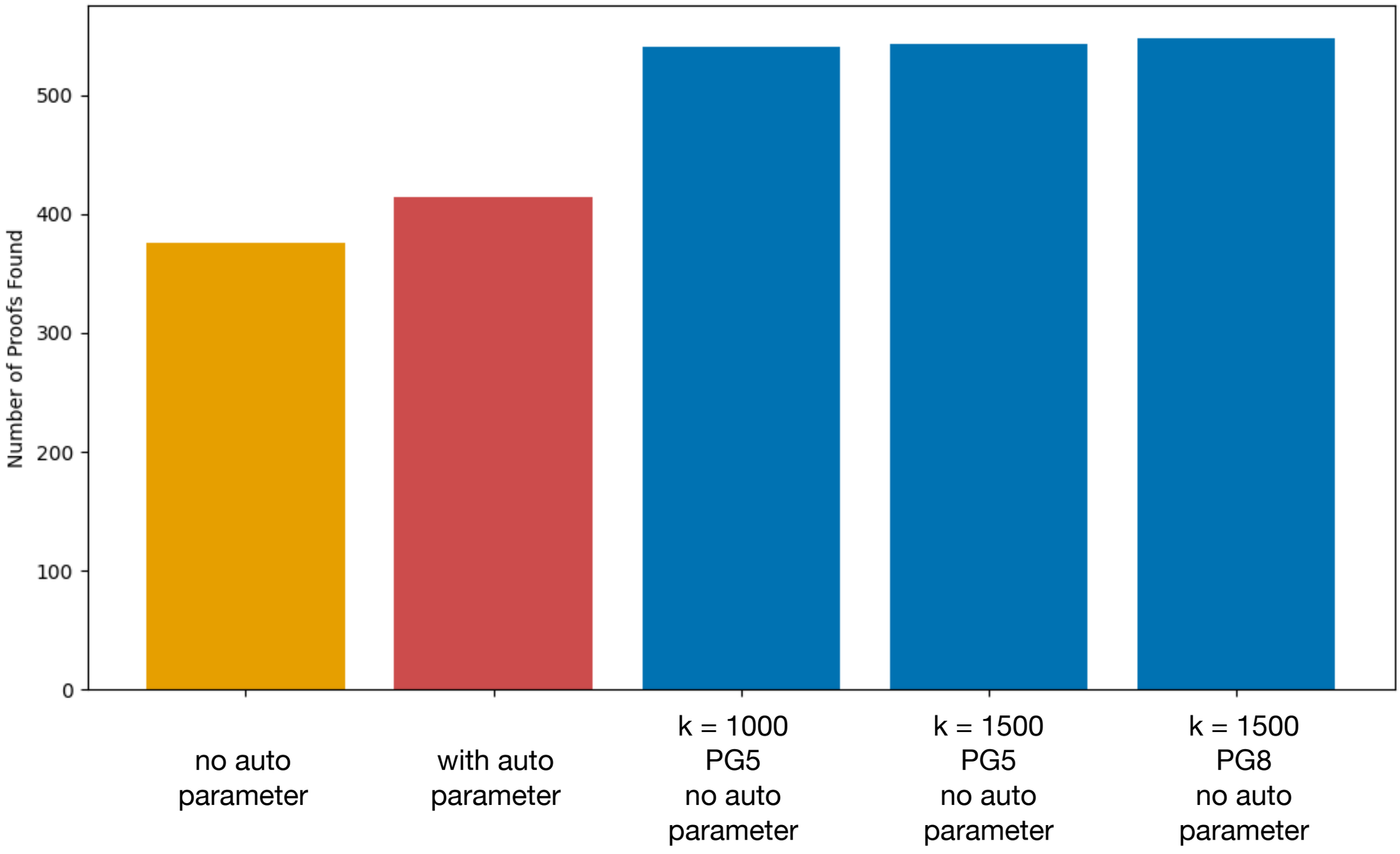
Experimental Results

- Adimen SUMO WhiteBoxTruthTests:
  - Collection of automatically generated tests
  - Each task: Goal for which it must be proved that it follows form the Adimen SUMO ontology
  - We randomly chose two disjoint sets of 1,000 of these proof tasks
    - First set: parameter-tuning set to determine interesting values for k and the pick given ratio

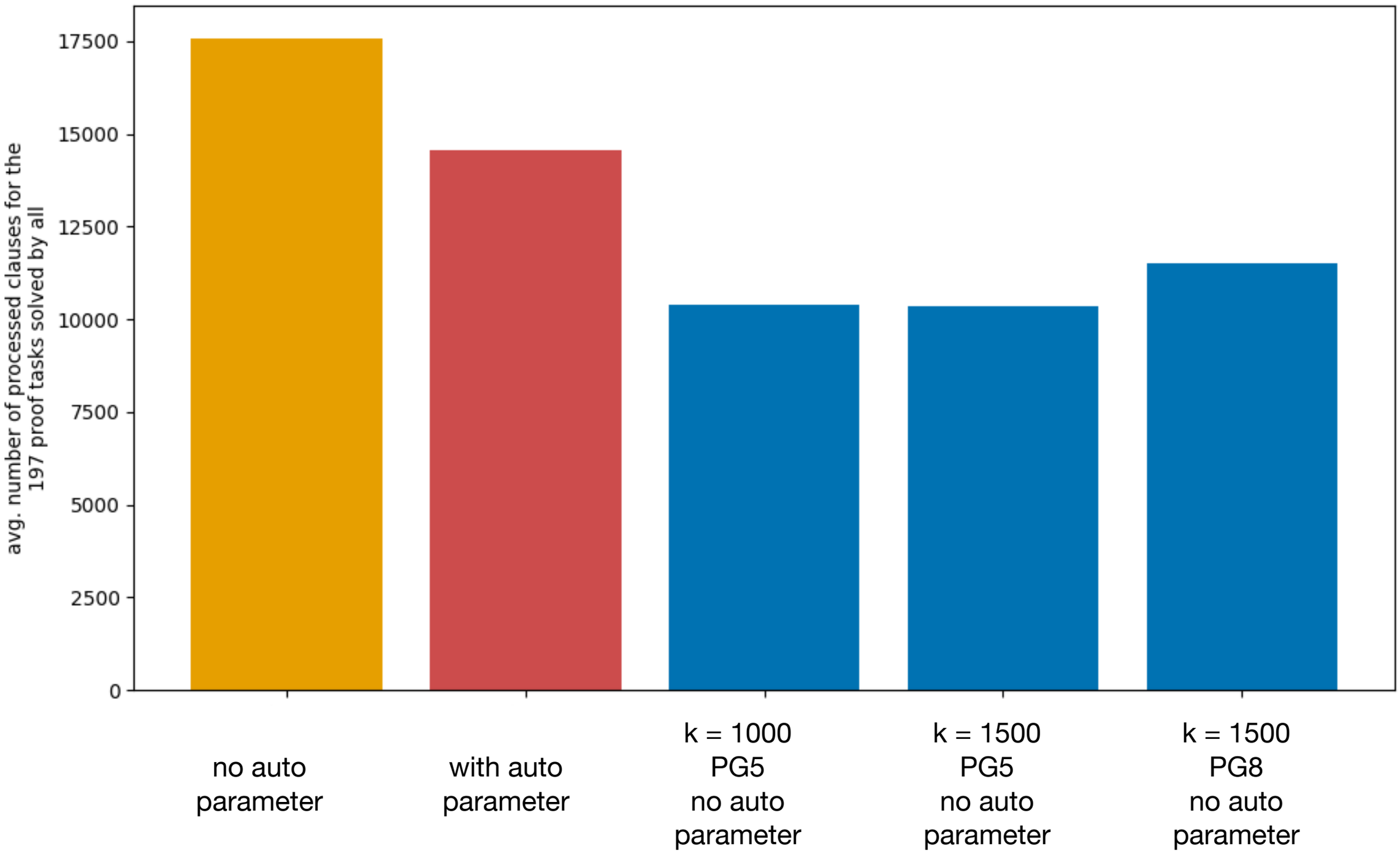
Experimental Results

- Adimen SUMO WhiteBoxTruthTests:
  - Collection of automatically generated tests
  - Each task: Goal for which it must be proved that it follows form the Adimen SUMO ontology
  - We randomly chose two disjoint sets of 1,000 of these proof tasks
    - First set: parameter-tuning set to determine interesting values for k and the pick given ratio
    - Second set: evaluation set
- We use the theorem prover E (timeout 10 sec.)

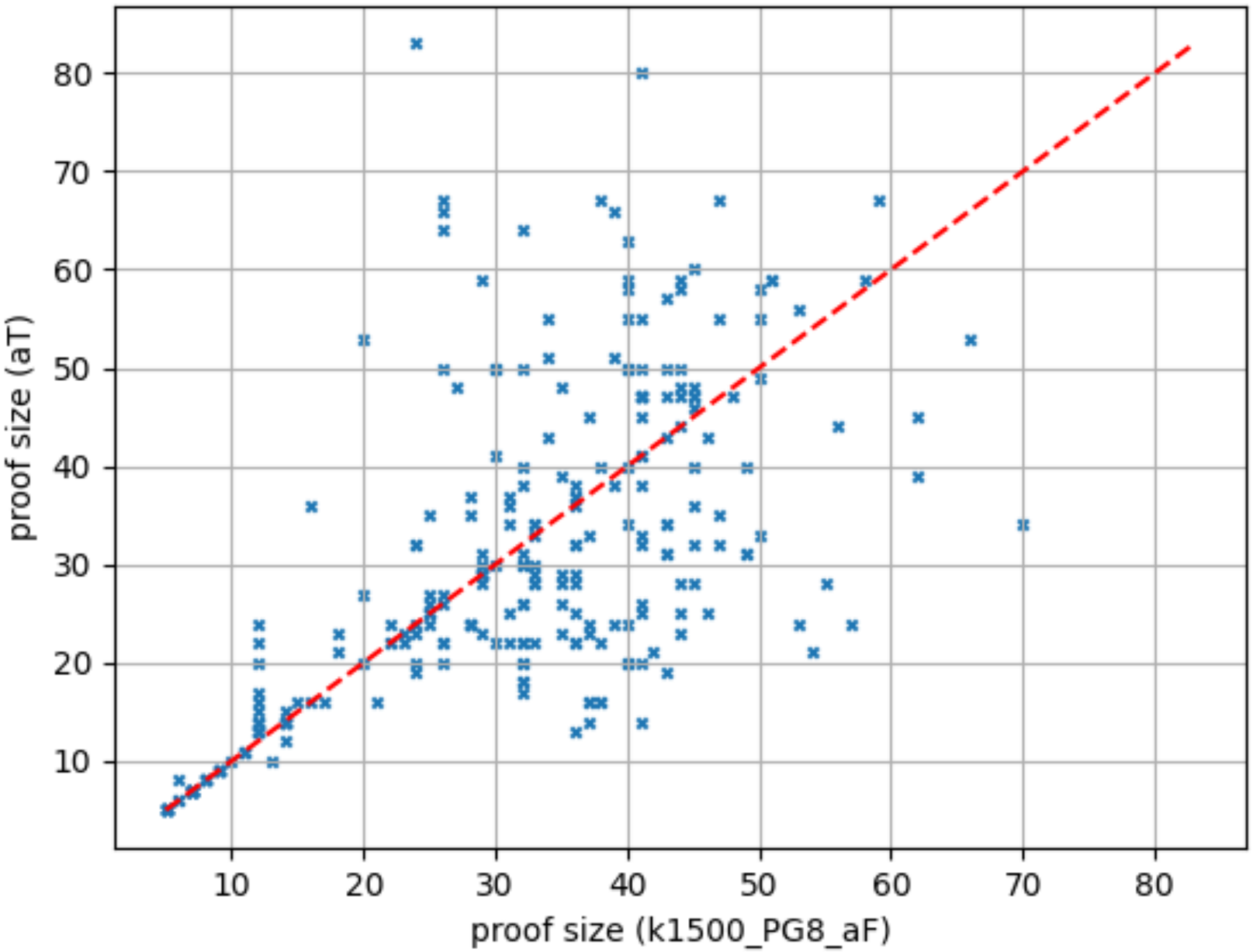
Evaluation Set: Number of Proofs Found



Evaluation Set: Average Number of Clauses Processed



Evaluation Set: Comparison of Proof Sizes



Conclusion and Future Work

Symbol names carry important information which:

- can be used to make the proof process more goal directed and

Image AI generated (DALL·E, OpenAI)

Conclusion and Future Work

Symbol names carry important information which:

- can be used to make the proof process more goal directed and
- helps to find more proofs.

Image AI generated (DALL·E, OpenAI)

Conclusion and Future Work

Symbol names carry important information which:

- can be used to make the proof process more goal directed and
- helps to find more proofs.

Ongoing and Future work:

- Investigate the difference of proofs found with our without the guidance by symbol names.
- Use the approach in other domains.

Image AI generated (DALL·E, OpenAI)

Conclusion and Future Work

Symbol names carry important information which:

- can be used to make the proof process more goal directed and
- helps to find more proofs.

Ongoing and Future work:

- Investigate the difference of proofs found with our without the guidance by symbol names.
- Use the approach in other domains.

Image AI generated (DALL·E, OpenAI)