google/gemini-2.5-flash-lite — accuracy — prompt_0b276d34e8 (cnf_v1)

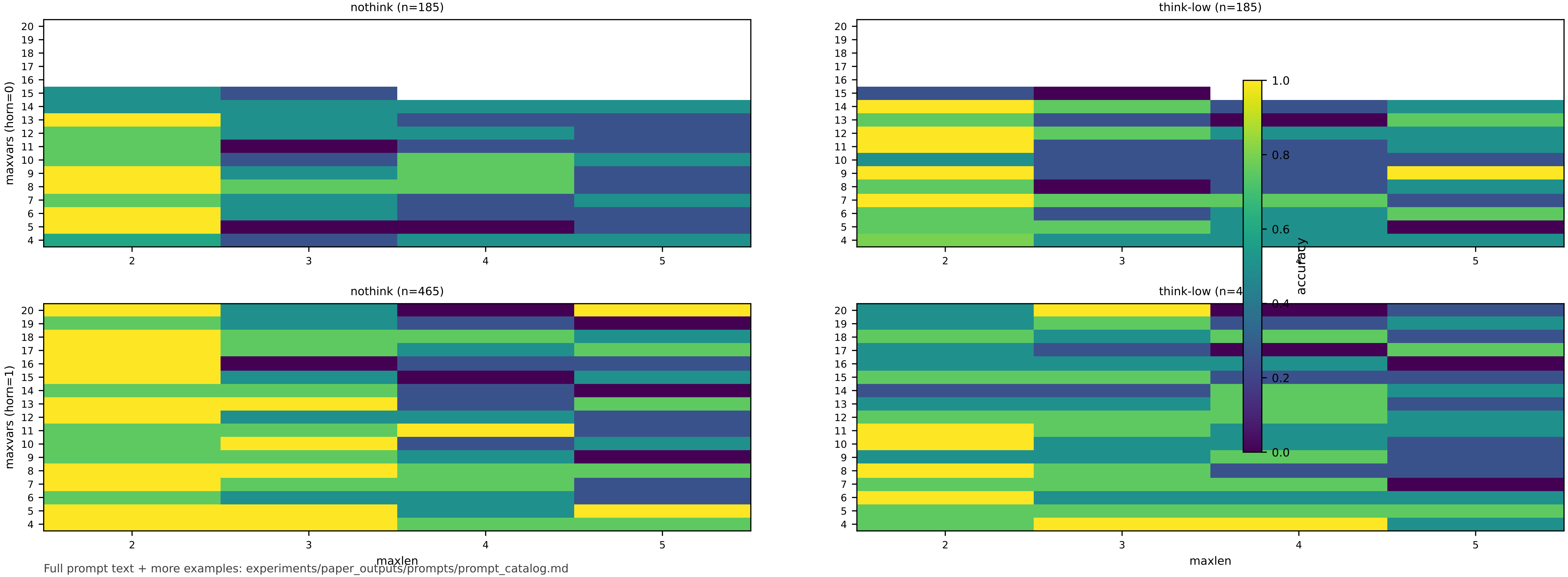
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
p4 is false.
p2 is true.
p3 is false or p1 is true.
p3 is false or p4 is true.
p2 is false or p1 is true.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_0b276d34e8 (cnf_v1)

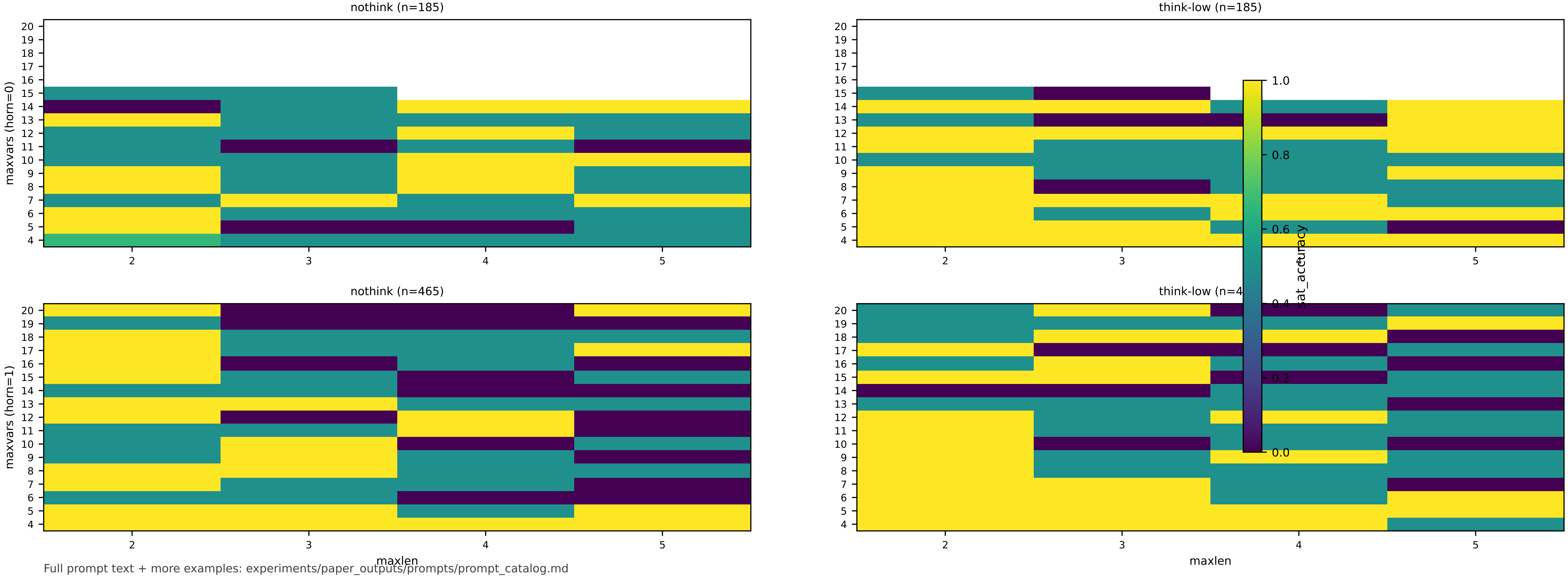
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
p4 is false.
p2 is true.
p3 is false or p1 is true.
p3 is false or p4 is true.
p2 is false or p1 is true.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_0b276d34e8 (cnf_v1)

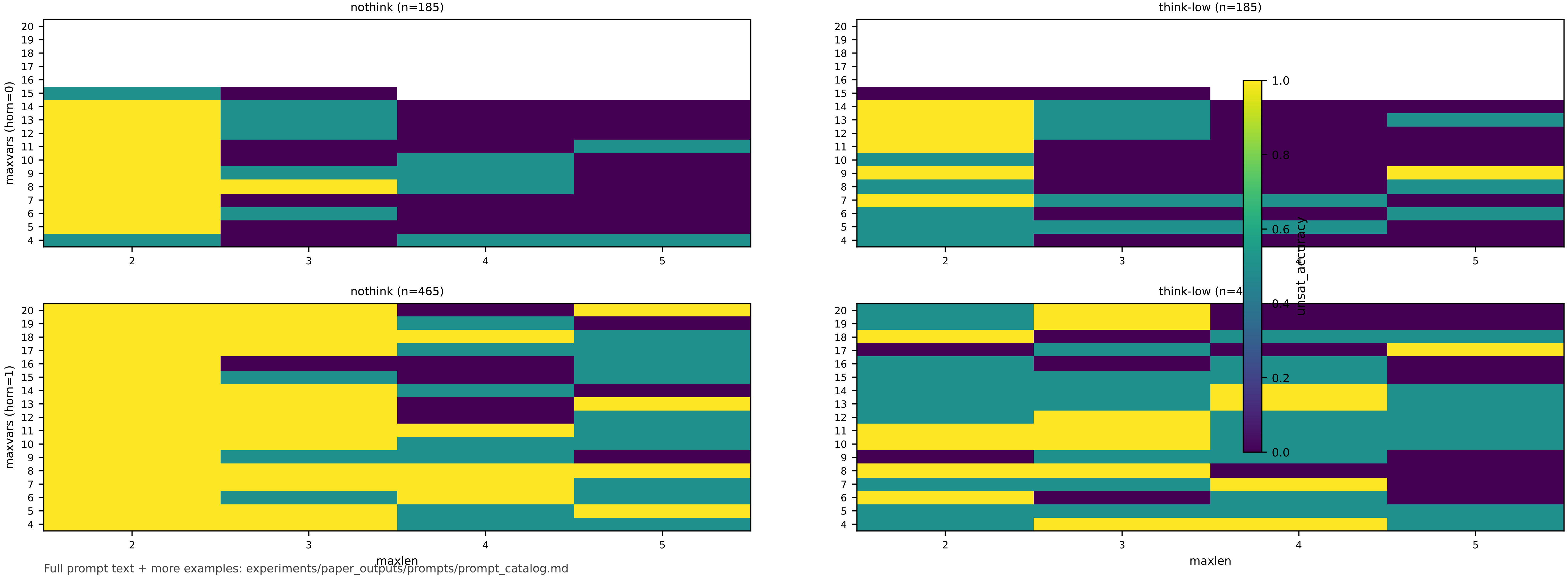
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
p4 is false.
p2 is true.
p3 is false or p1 is true.
p3 is false or p4 is true.
p2 is false or p1 is true.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_21889a86a3 (cnf_v1)

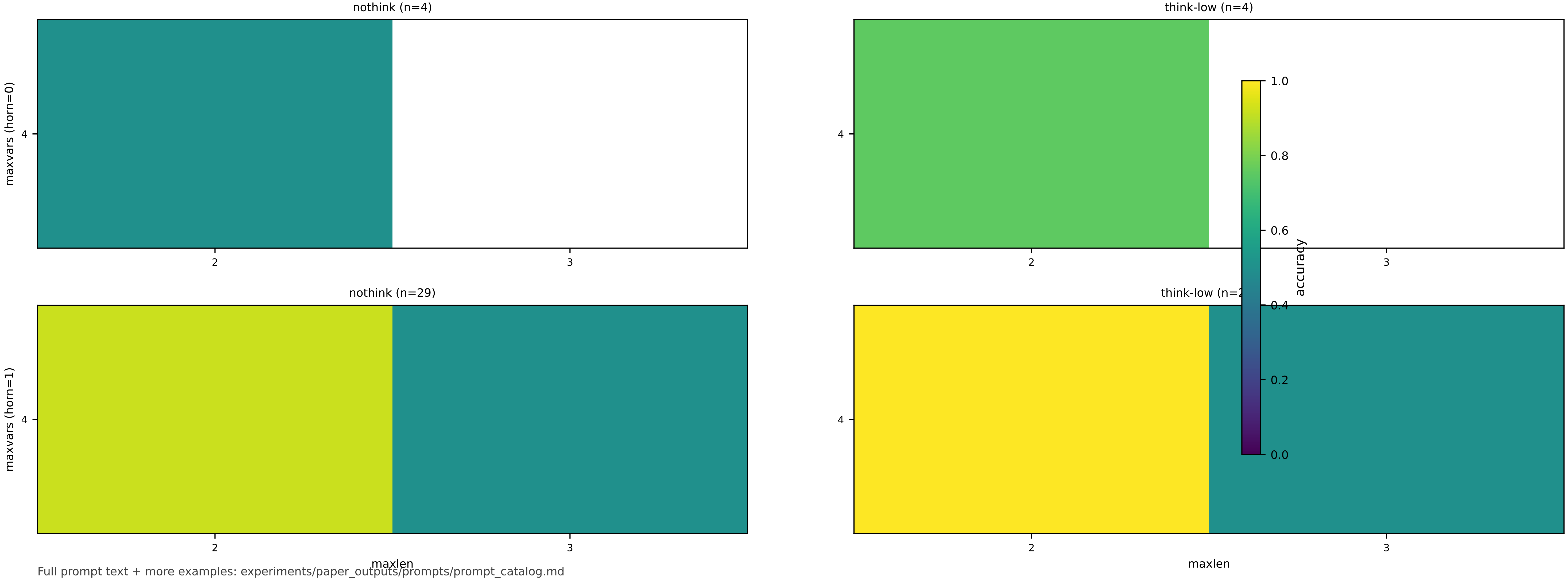
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
p4 is false.
p2 is true.
p3 is false or p1 is true.
p3 is false or p4 is true.
p2 is false or p1 is true.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_21889a86a3 (cnf_v1)

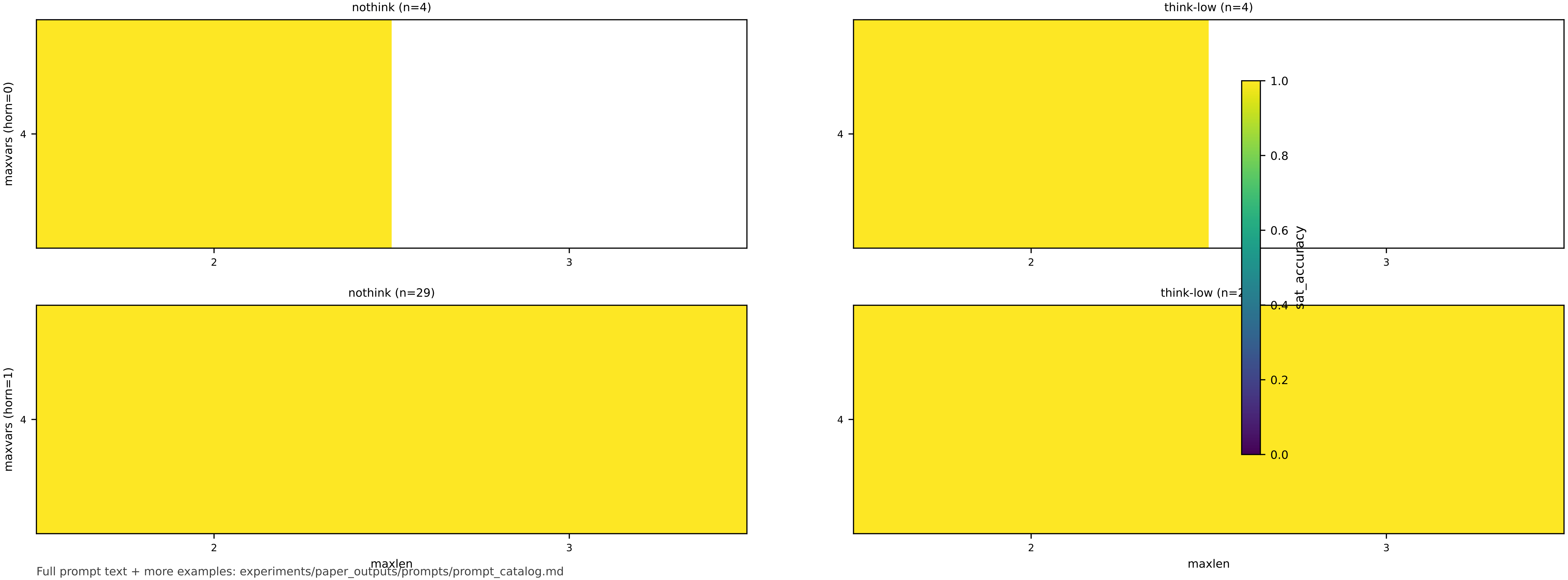
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
p4 is false.
p2 is true.
p3 is false or p1 is true.
p3 is false or p4 is true.
p2 is false or p1 is true.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_21889a86a3 (cnf_v1)

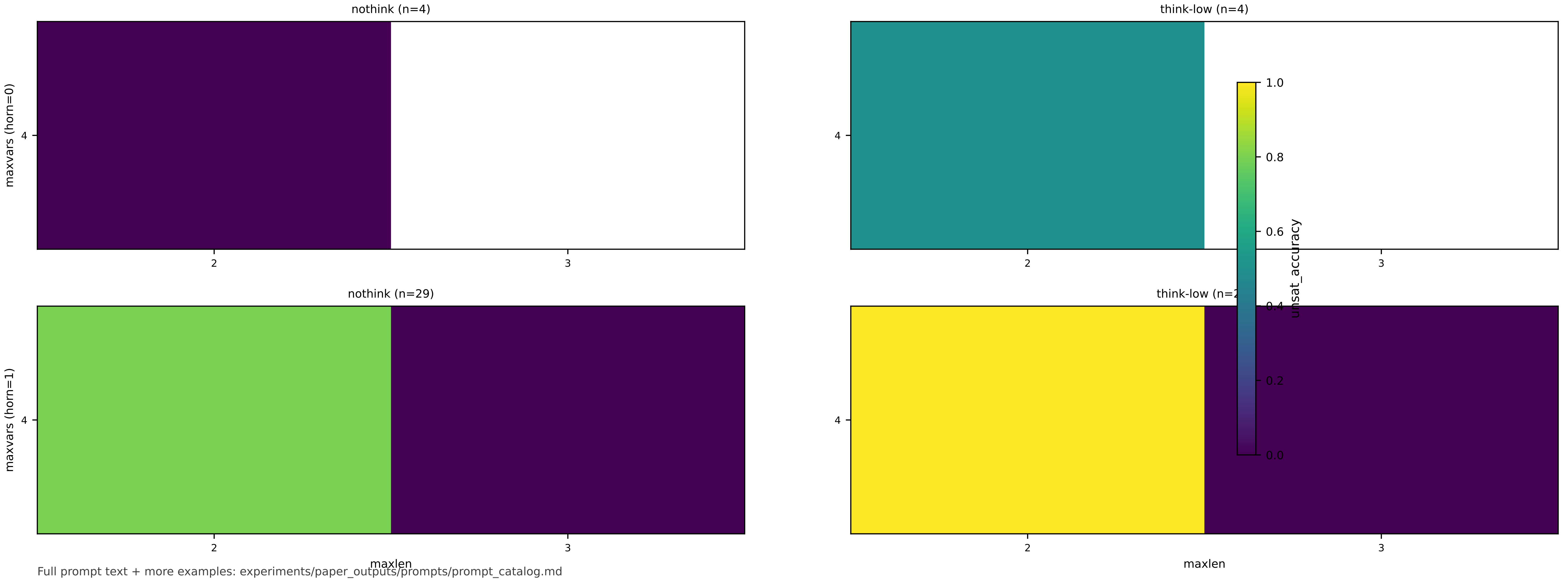
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
p4 is false.
p2 is true.
p3 is false or p1 is true.
p3 is false or p4 is true.
p2 is false or p1 is true.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_2376d1fca7 (horn_if_then)

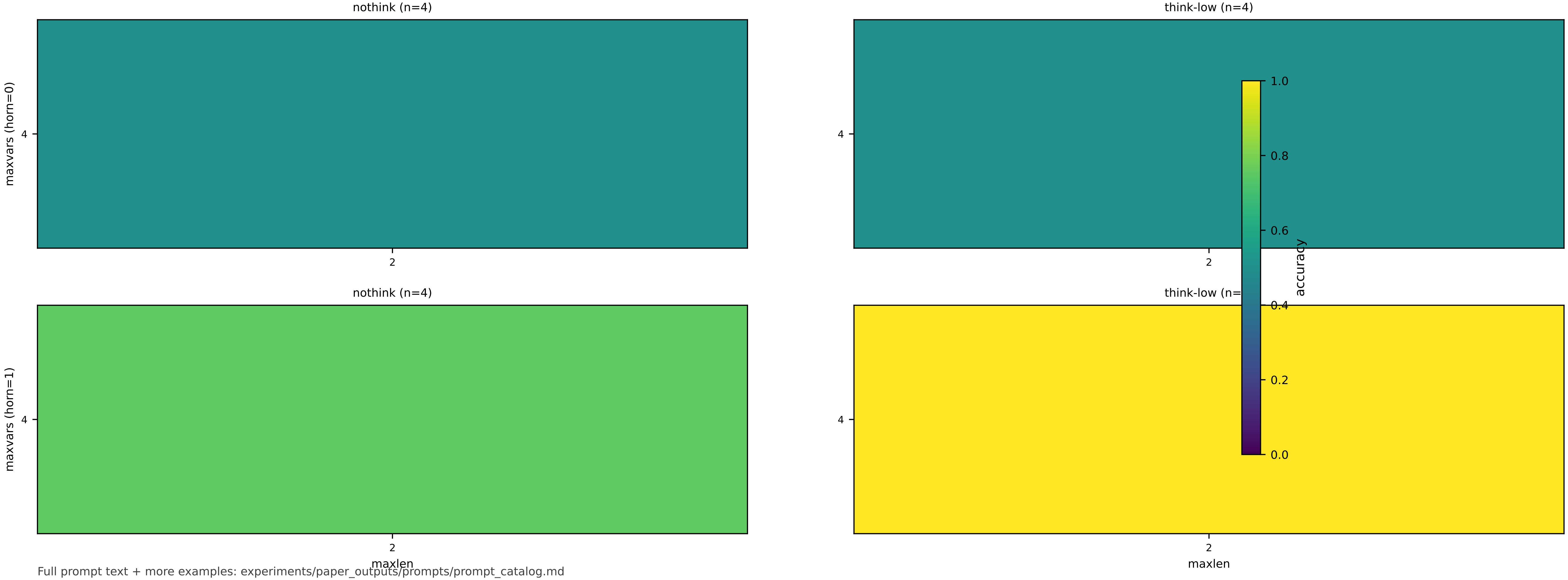
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Unified answer rule (mixed cases)
- Regardless of how the statements are rendered, output only a final single
word: "yes" if p0 is derivable OR the set is a contradiction; otherwise "no".
Do not output any other words.
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_2376d1fca7 (horn_if_then)

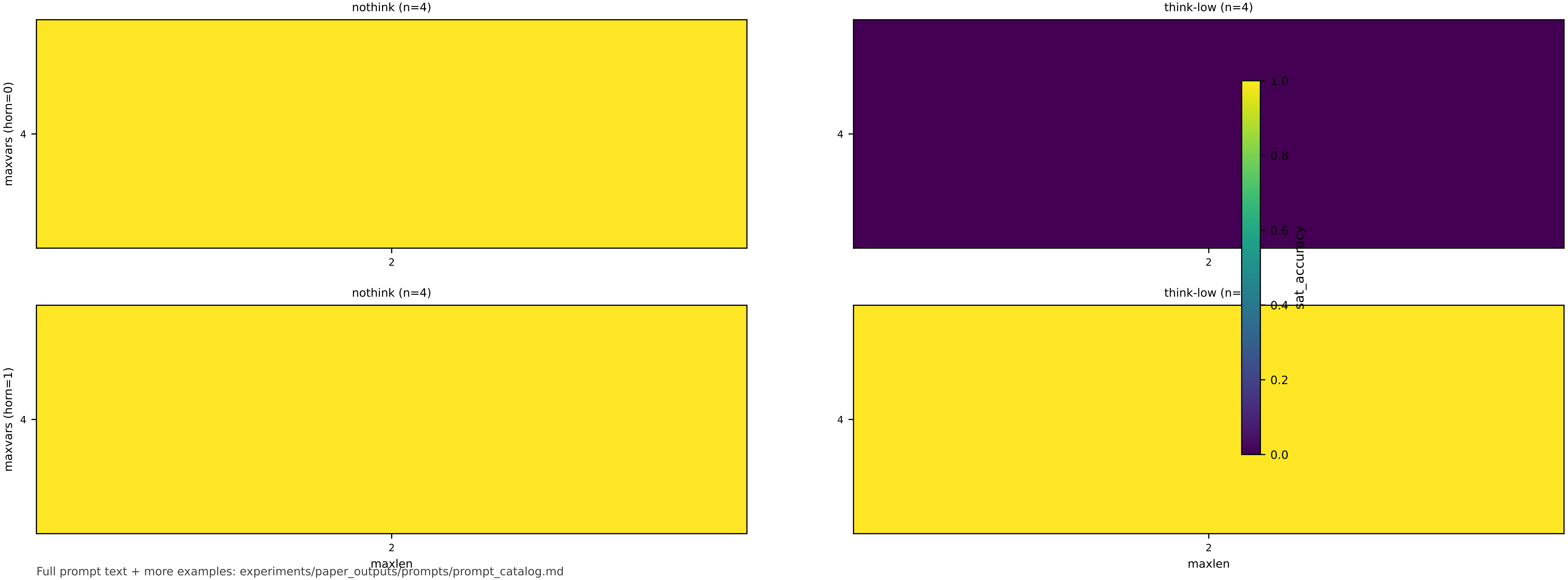
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Unified answer rule (mixed cases)
- Regardless of how the statements are rendered, output only a final single
word: "yes" if p0 is derivable OR the set is a contradiction; otherwise "no".
Do not output any other words.
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_2376d1fca7 (horn_if_then)

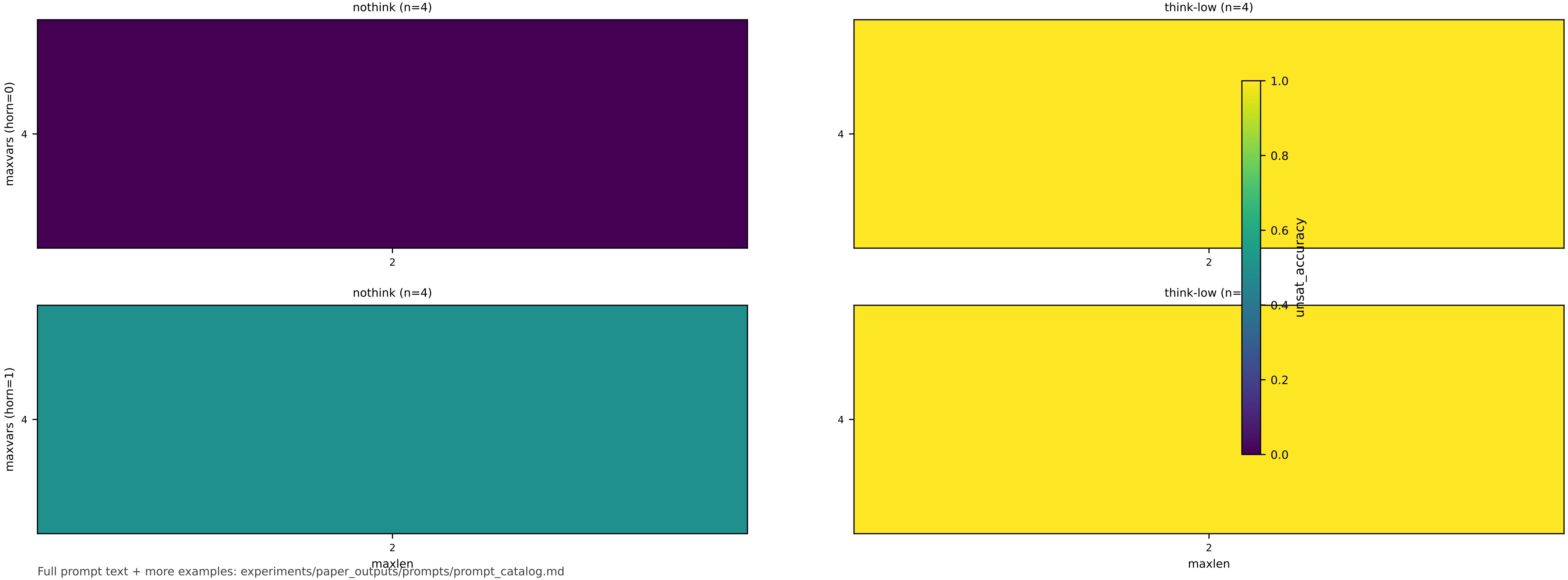
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Unified answer rule (mixed cases)
- Regardless of how the statements are rendered, output only a final single
word: "yes" if p0 is derivable OR the set is a contradiction; otherwise "no".
Do not output any other words.
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_2e9c5ccddf (cnf_v2)

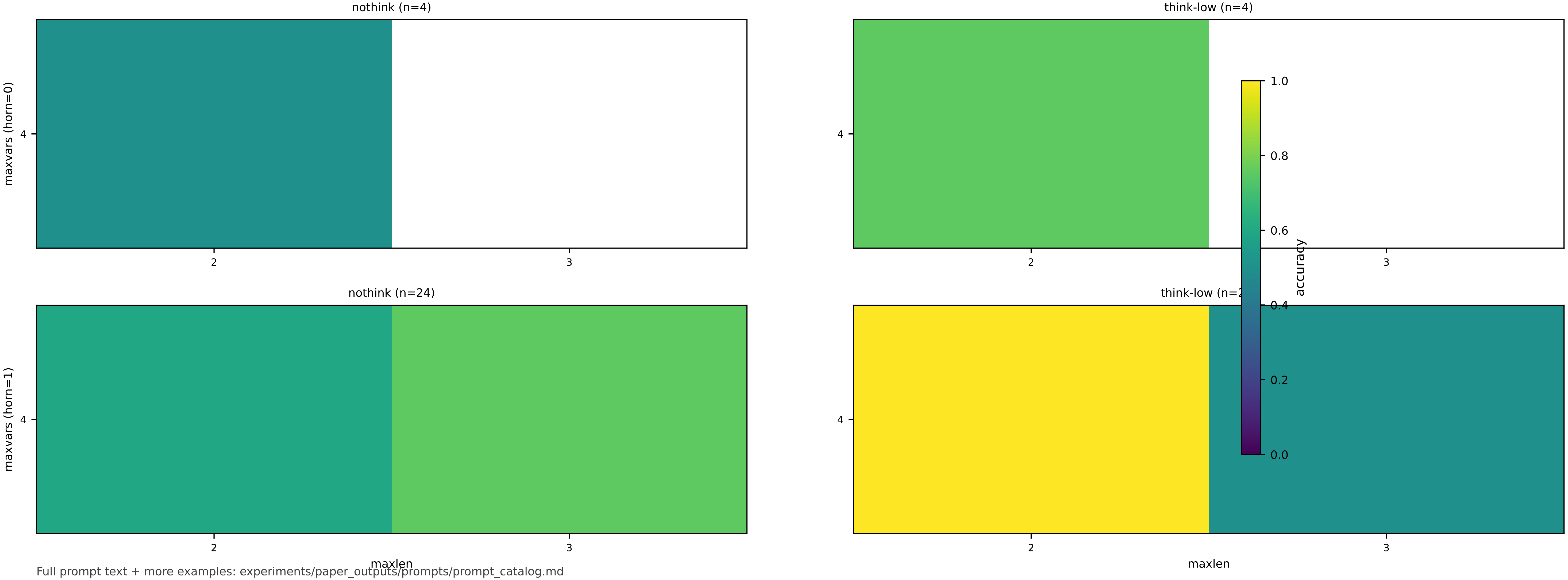
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
not(p4).
p2.
not(p3) or p1.
not(p3) or p4.
not(p2) or p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_2e9c5ccddf (cnf_v2)

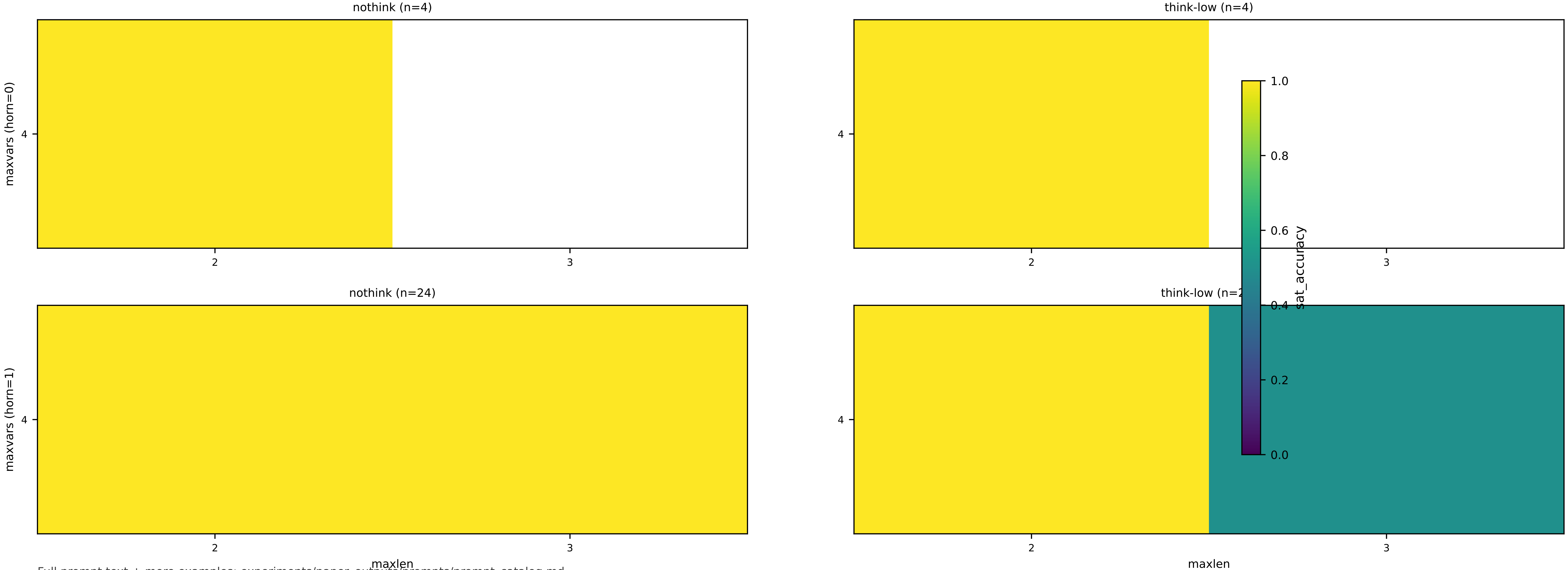
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
not(p4).
p2.
not(p3) or p1.
not(p3) or p4.
not(p2) or p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_2e9c5ccddf (cnf_v2)

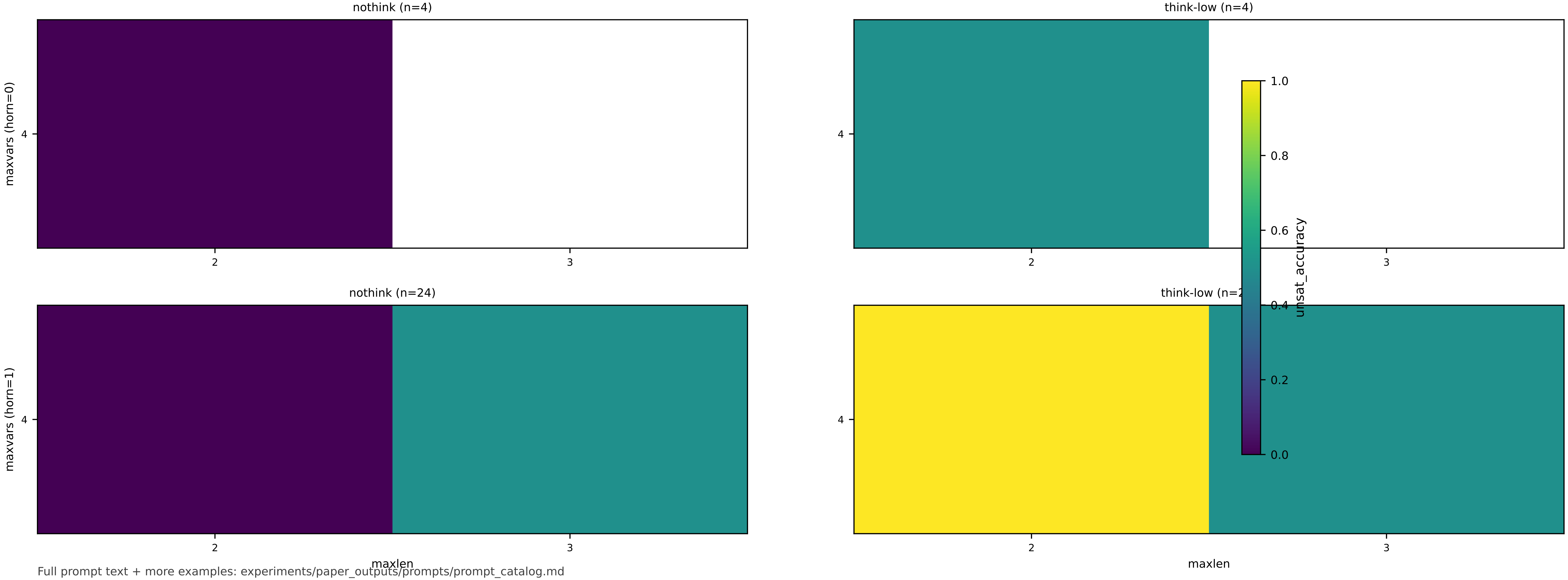
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
not(p4).
p2.
not(p3) or p1.
not(p3) or p4.
not(p2) or p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_62ba908560 (horn_if_then)

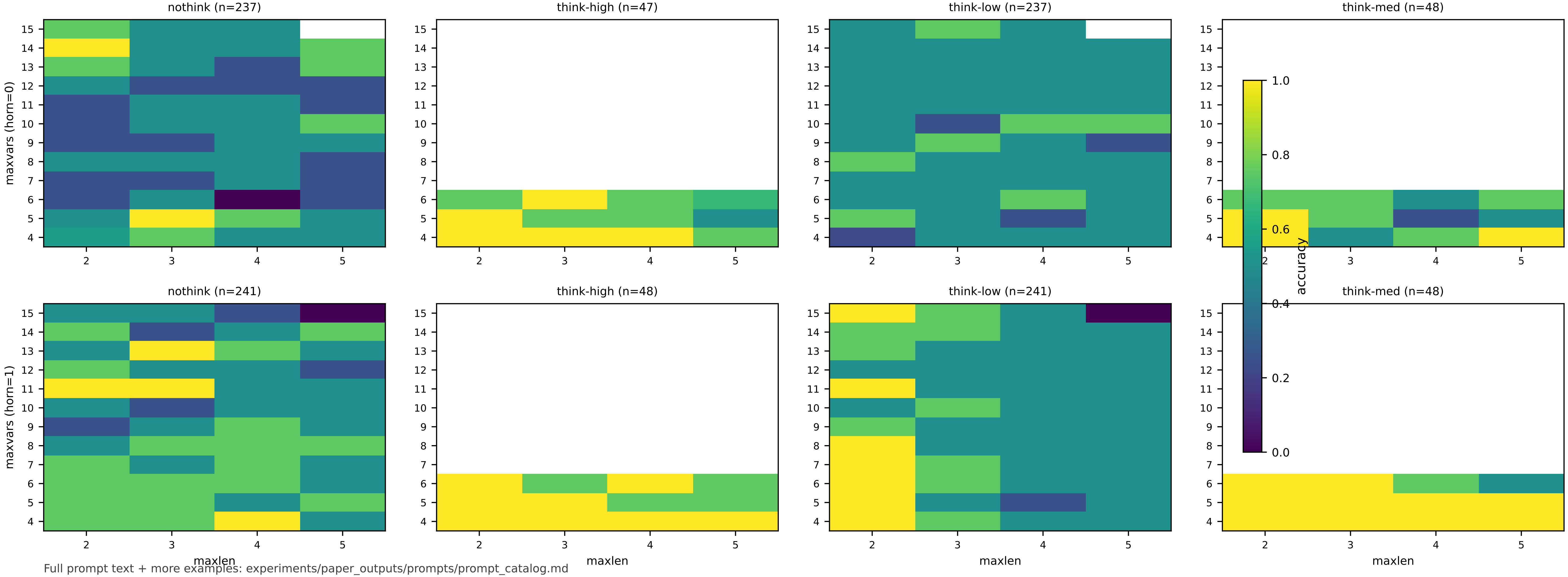
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Unified answer rule (mixed cases)
- Regardless of how the statements are rendered, output only a final single
word: "yes" if p0 is derivable OR the set is a contradiction; otherwise "no".
Do not output any other words.
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_62ba908560 (horn_if_then)

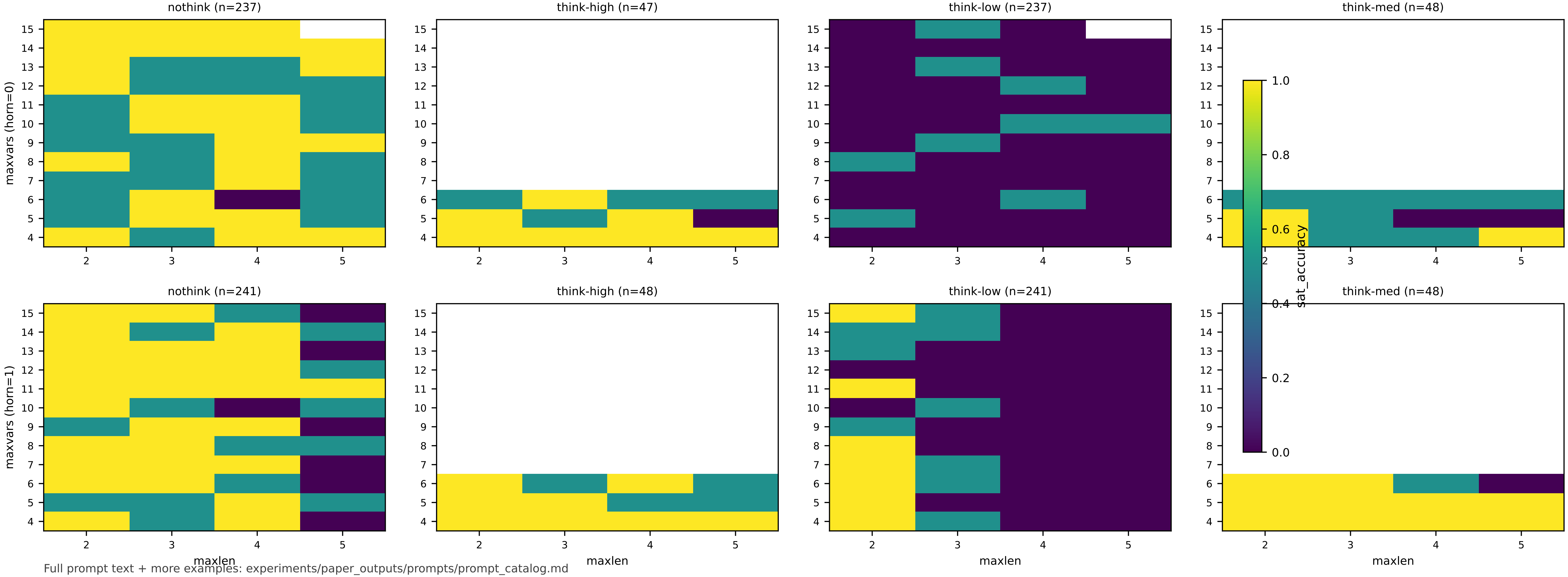
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Unified answer rule (mixed cases)
- Regardless of how the statements are rendered, output only a final single
word: "yes" if p0 is derivable OR the set is a contradiction; otherwise "no".
Do not output any other words.
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_62ba908560 (horn_if_then)

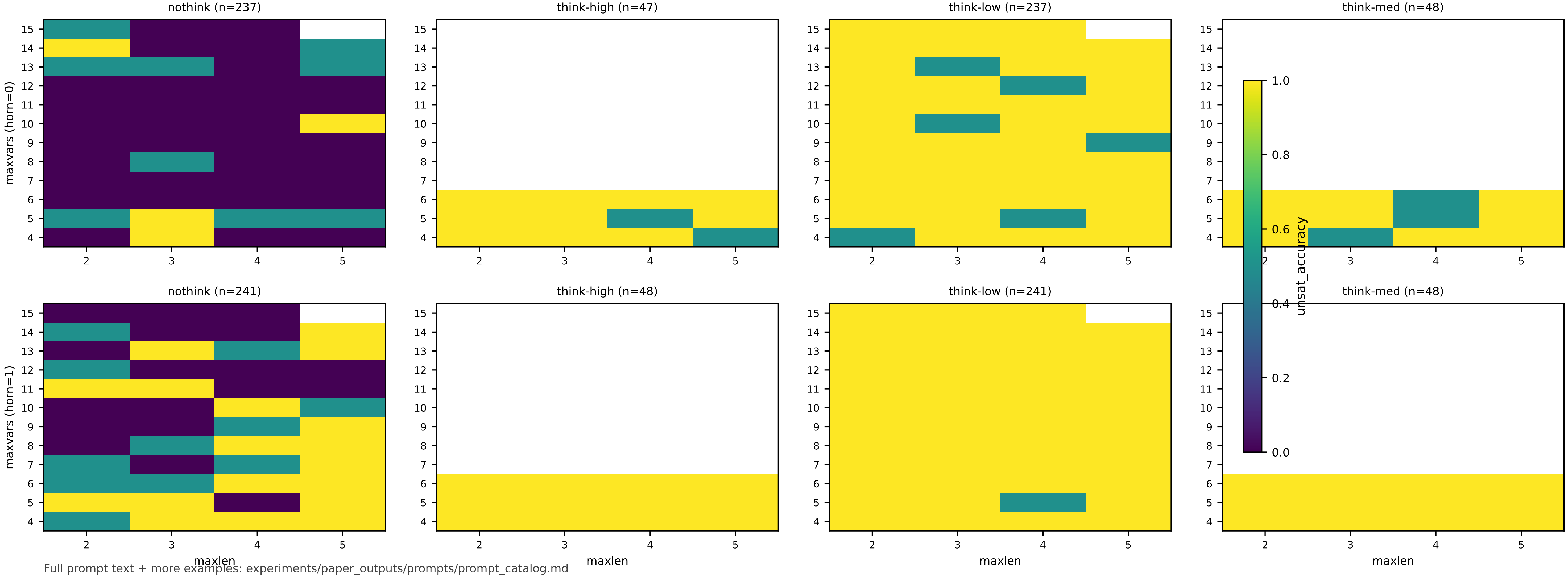
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Unified answer rule (mixed cases)
- Regardless of how the statements are rendered, output only a final single
word: "yes" if p0 is derivable OR the set is a contradiction; otherwise "no".
Do not output any other words.
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_73ecab0579 (horn_if_then)

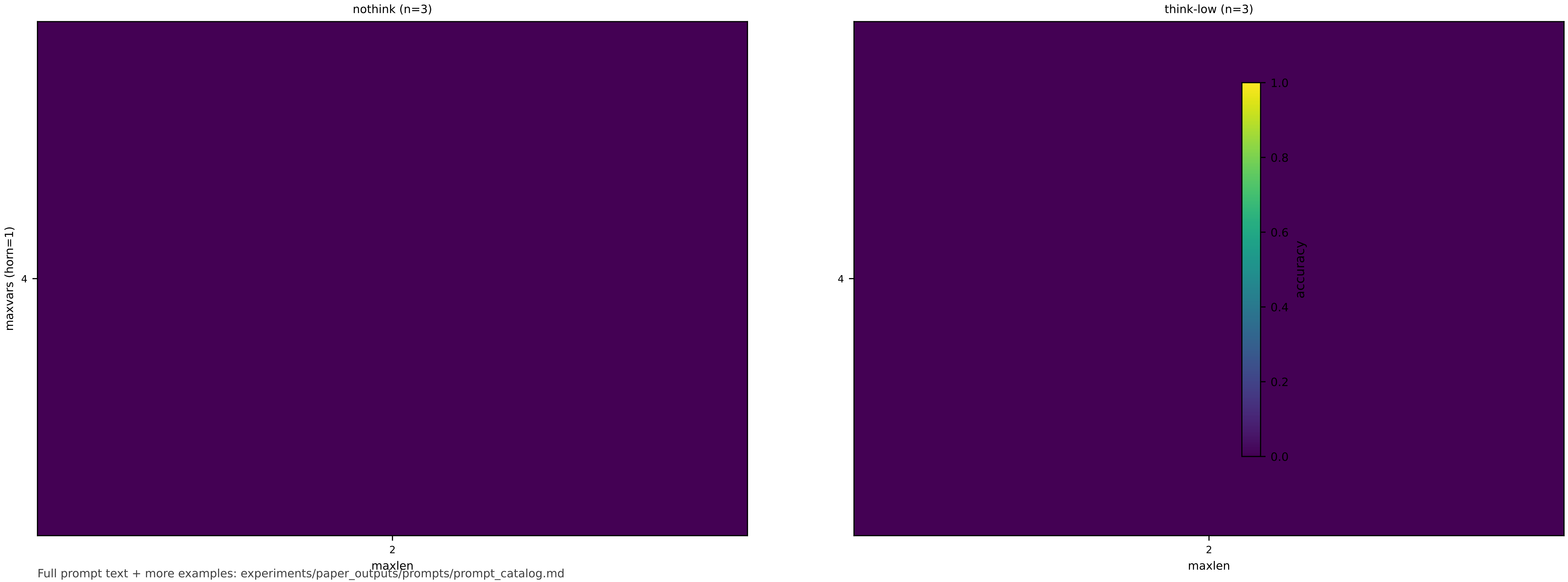
prompt_template= | parse_family=yes_no

Instruction excerpt:

(no instruction text found)

Example statements:

(no example statements found)



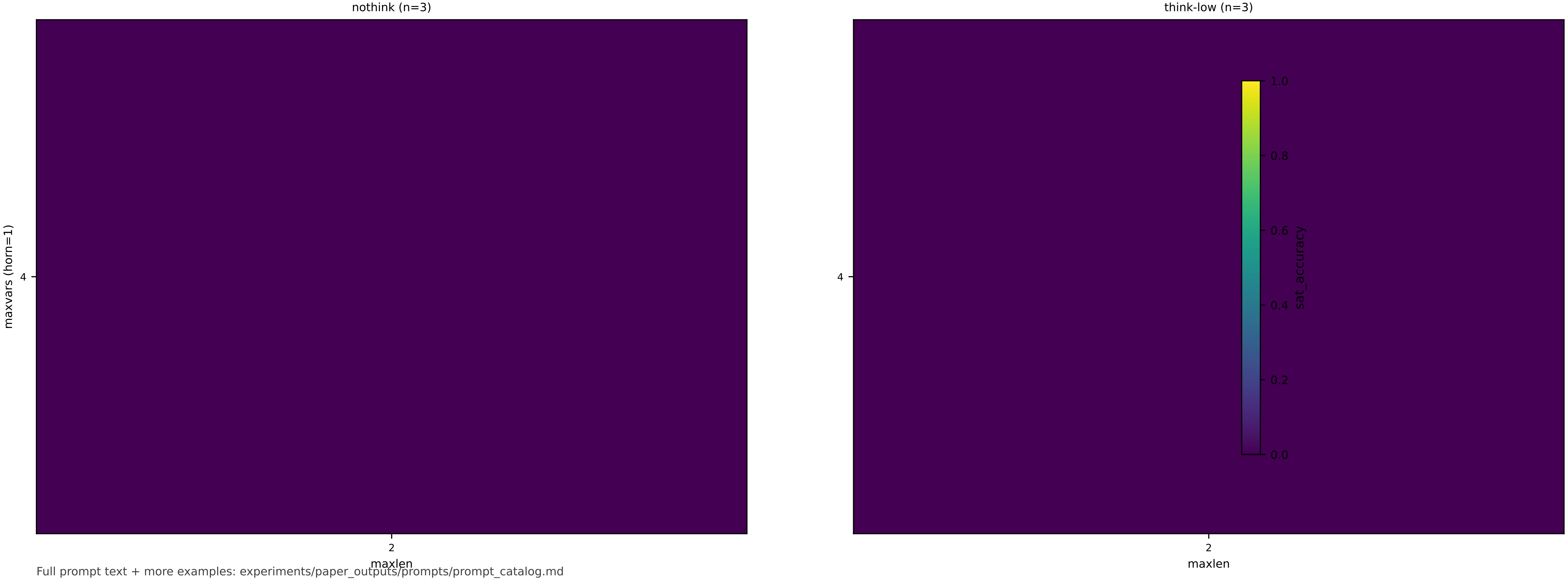
Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_73ecab0579 (horn_if_then)

prompt_template= | parse_family=yes_no

Instruction excerpt:
(no instruction text found)

Example statements:
(no example statements found)



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_73ecab0579 (horn_if_then)

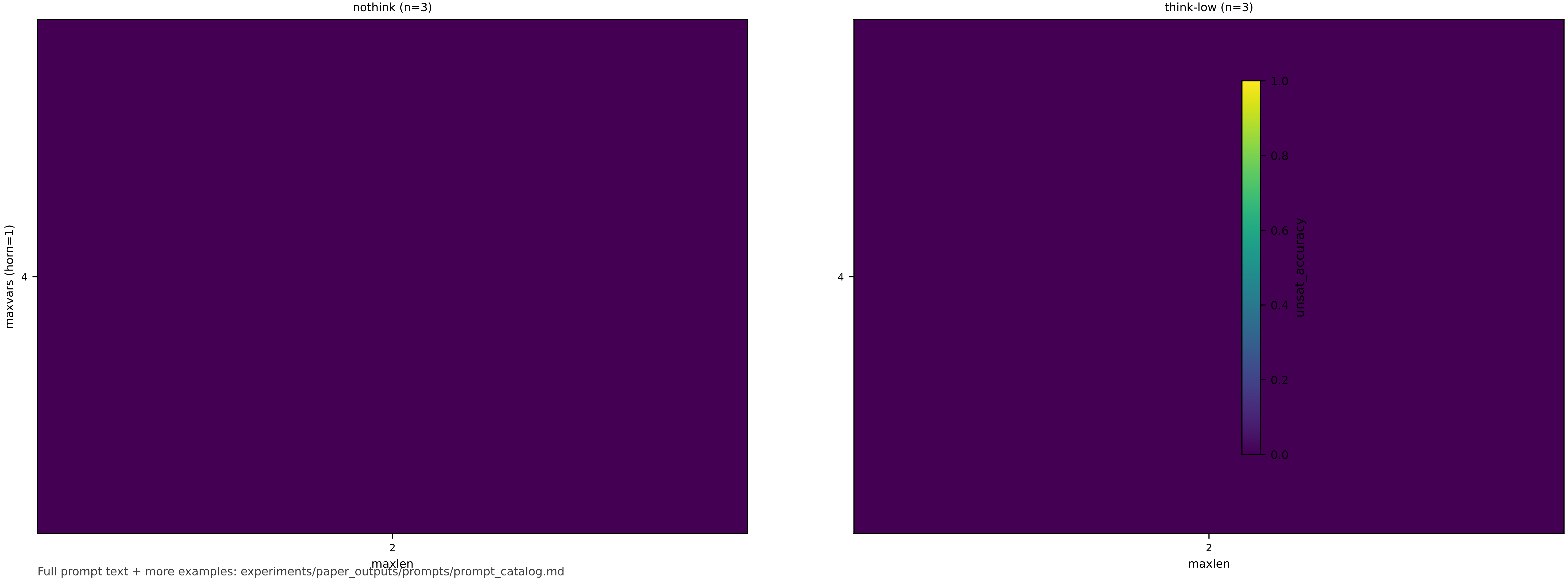
prompt_template= | parse_family=yes_no

Instruction excerpt:

```
(no instruction text found)
```

Example statements:

```
(no example statements found)
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_7b28aa32dc (horn_if_then)

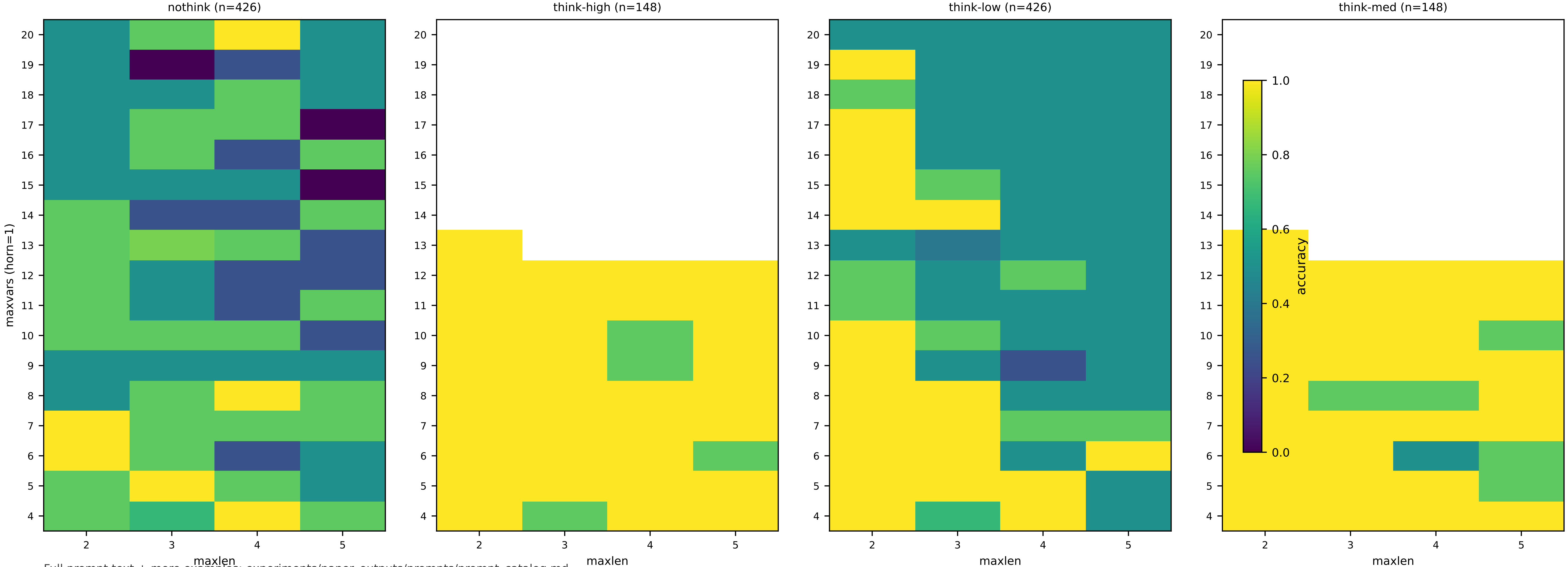
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Horn answer rule
- Output ONLY a single final word: "yes" if p0 is derivable, otherwise "no".
Do not output any other words.

...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_7b28aa32dc (horn_if_then)

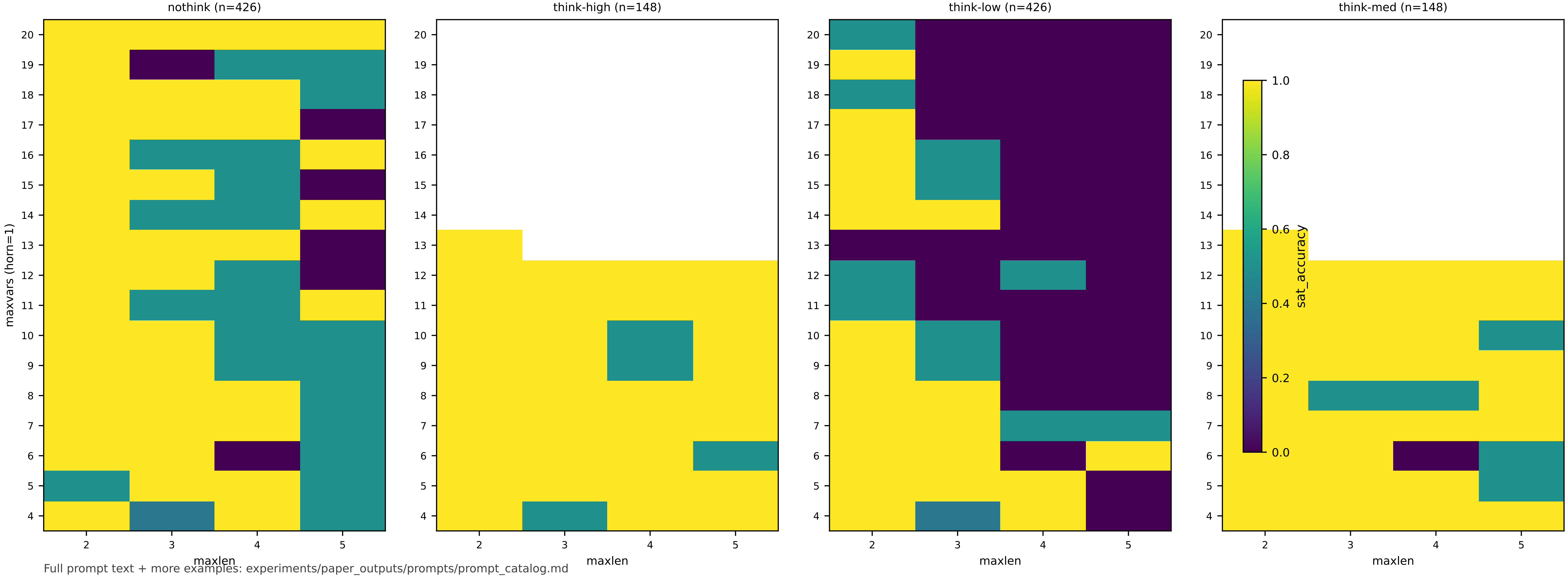
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Horn answer rule
- Output ONLY a single final word: "yes" if p0 is derivable, otherwise "no".
Do not output any other words.

...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_7b28aa32dc (horn_if_then)

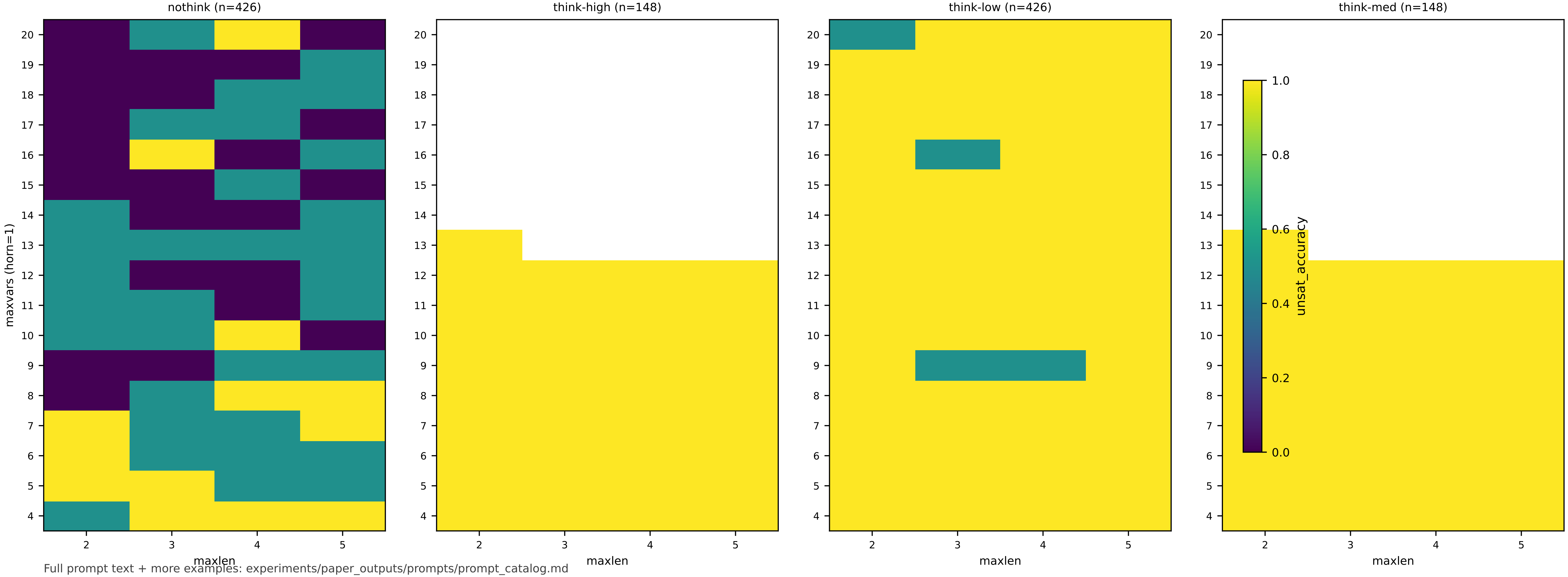
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.
Horn answer rule
- Output ONLY a single final word: "yes" if p0 is derivable, otherwise "no".
Do not output any other words.

...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_83b02d3a05 (cnf_v1)

prompt_template=prompts/exp1_cnf_v1_contradiction.j2 | parse_family=contradiction

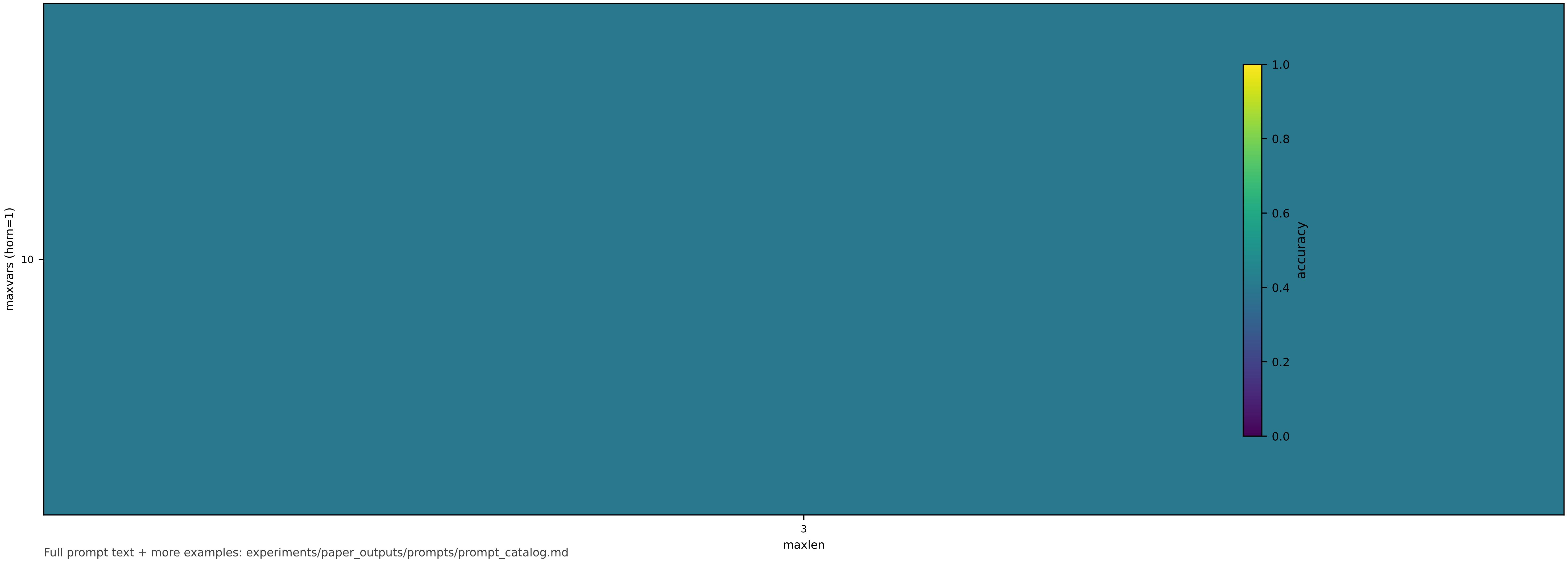
Instruction excerpt:

```
Your task is to solve a problem in propositional logic.
You will get a list of statements and have to determine whether the statements
form a logical contradiction or not.
If the statements form a contradiction, the last word of your answer should be
'contradiction',
otherwise the last word should be either 'satisfiable' or 'unknown'.

Propositional variables are represent as 'pN' where N is a number. They are
either true or false.
'X or Y' means that X is true or Y is true or both X and Y are true.
All the given statements are implicitly connected with 'and': they are all
claimed to be true.

...
```

Example (horn=1, low, maxvars=10, maxlen=3, satflag=1)

```
p1 is true. p1 is false or p2 is true. p2 is false. Answer: contradiction.
Example 2. Statements: p1 is true. p1 is true or p2 is true. p2 is false.
Answer: satisfiable.

Statements:
p2 is true.
p4 is true.
p10 is false or p2 is true.
p10 is false or p4 is true.
p9 is false or p1 is false.
p9 is false or p1 is true.
p8 is false or p2 is true.
p8 is false or p4 is true.
...
```

no_thinking_tag (n=5)



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_83b02d3a05 (cnf_v1)

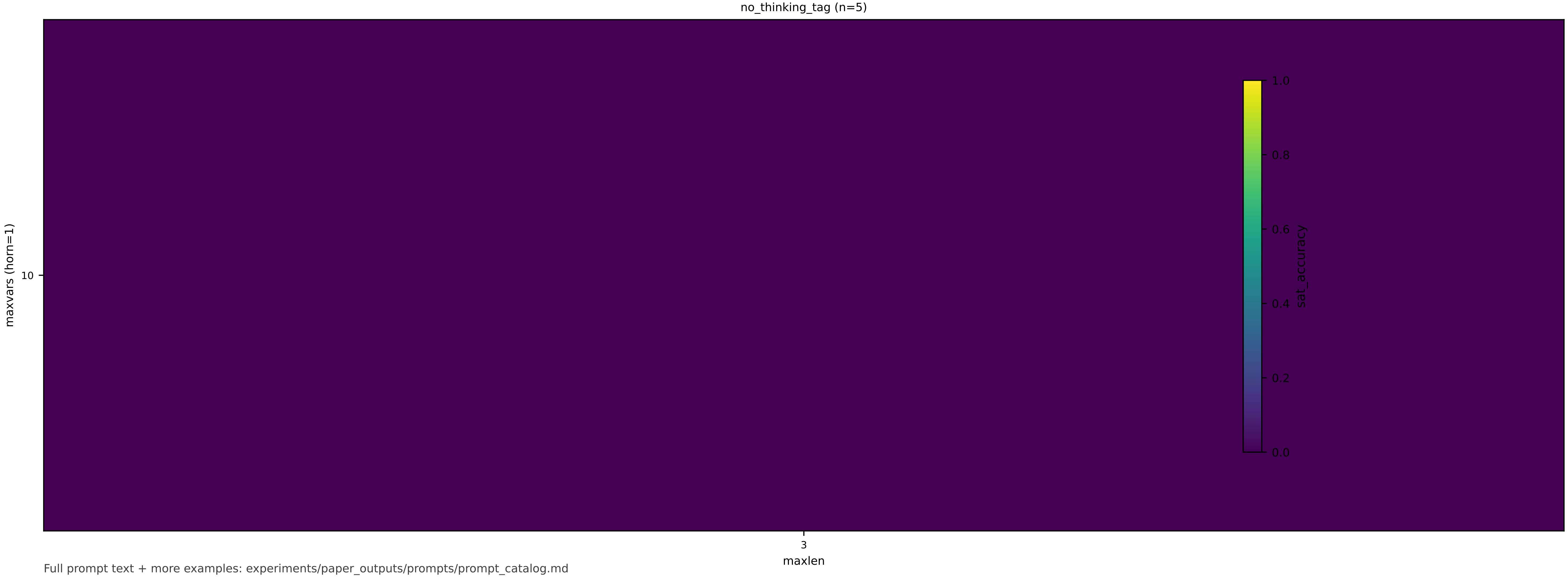
prompt_template=prompts/exp1_cnf_v1_contradiction.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a problem in propositional logic.
You will get a list of statements and have to determine whether the statements
form a logical contradiction or not.
If the statements form a contradiction, the last word of your answer should be
'contradiction',
otherwise the last word should be either 'satisfiable' or 'unknown'.

Propositional variables are represent as 'pN' where N is a number. They are
either true or false.
'X or Y' means that X is true or Y is true or both X and Y are true.
All the given statements are implicitly connected with 'and': they are all
claimed to be true.

...
```

Example (horn=1, low, maxvars=10, maxlen=3, satflag=1)

```
p1 is true. p1 is false or p2 is true. p2 is false. Answer: contradiction.
Example 2. Statements: p1 is true. p1 is true or p2 is true. p2 is false.
Answer: satisfiable.

Statements:
p2 is true.
p4 is true.
p10 is false or p2 is true.
p10 is false or p4 is true.
p9 is false or p1 is false.
p9 is false or p1 is true.
p8 is false or p2 is true.
p8 is false or p4 is true.
...
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_83b02d3a05 (cnf_v1)

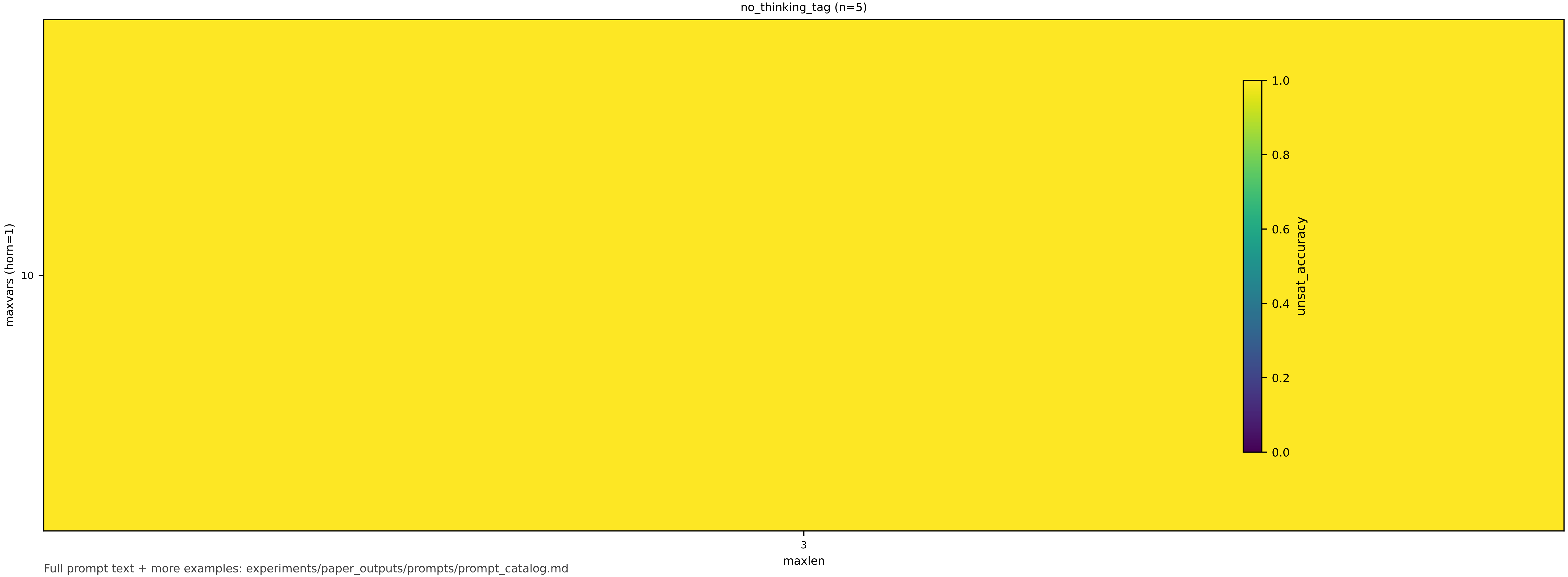
prompt_template=prompts/exp1_cnf_v1_contradiction.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a problem in propositional logic.
You will get a list of statements and have to determine whether the statements
form a logical contradiction or not.
If the statements form a contradiction, the last word of your answer should be
'contradiction',
otherwise the last word should be either 'satisfiable' or 'unknown'.

Propositional variables are represent as 'pN' where N is a number. They are
either true or false.
'X or Y' means that X is true or Y is true or both X and Y are true.
All the given statements are implicitly connected with 'and': they are all
claimed to be true.

...
```

Example (horn=1, low, maxvars=10, maxlen=3, satflag=1)

```
p1 is true. p1 is false or p2 is true. p2 is false. Answer: contradiction.
Example 2. Statements: p1 is true. p1 is true or p2 is true. p2 is false.
Answer: satisfiable.

Statements:
p2 is true.
p4 is true.
p10 is false or p2 is true.
p10 is false or p4 is true.
p9 is false or p1 is false.
p9 is false or p1 is true.
p8 is false or p2 is true.
p8 is false or p4 is true.
...
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_c012d6f2e6 (cnf_v2)

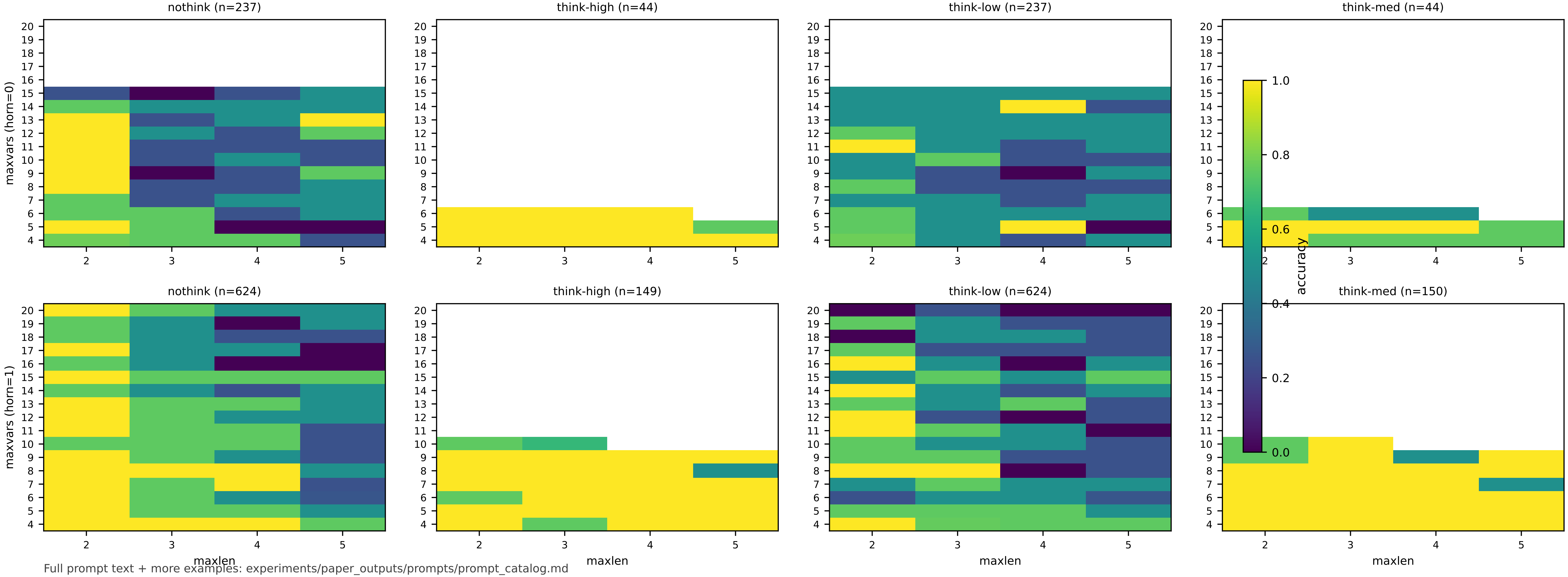
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
not(p4).
p2.
not(p3) or p1.
not(p3) or p4.
not(p2) or p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_c012d6f2e6 (cnf_v2)

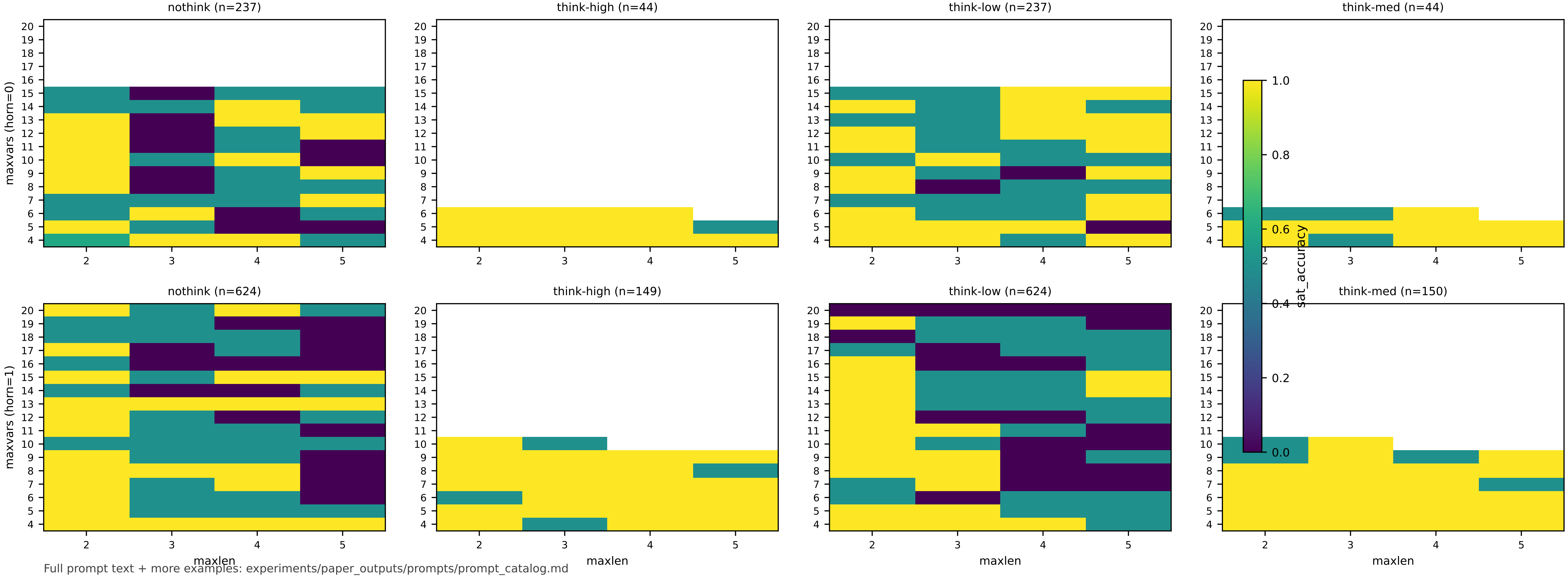
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
not(p4).
p2.
not(p3) or p1.
not(p3) or p4.
not(p2) or p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_c012d6f2e6 (cnf_v2)

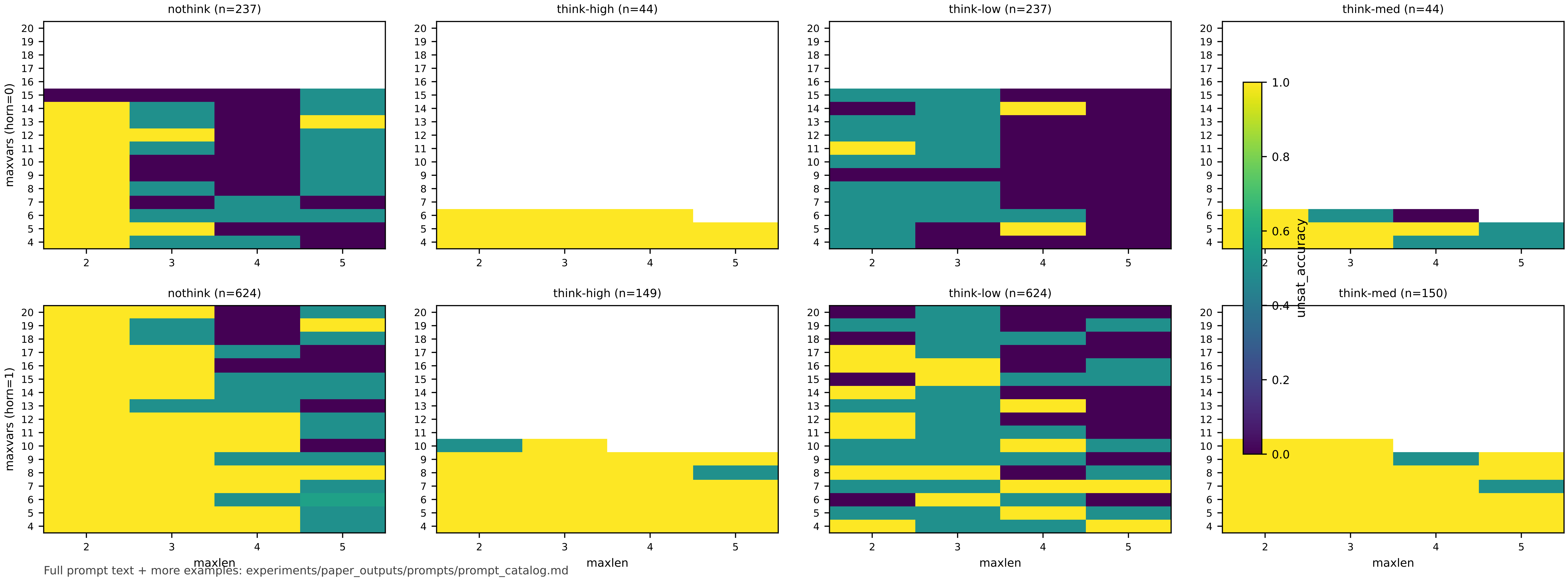
prompt_template=prompts/_template_unified.j2 | parse_family=contradiction

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
not(p4).
p2.
not(p3) or p1.
not(p3) or p4.
not(p2) or p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_c1b2be97aa (horn_if_then)

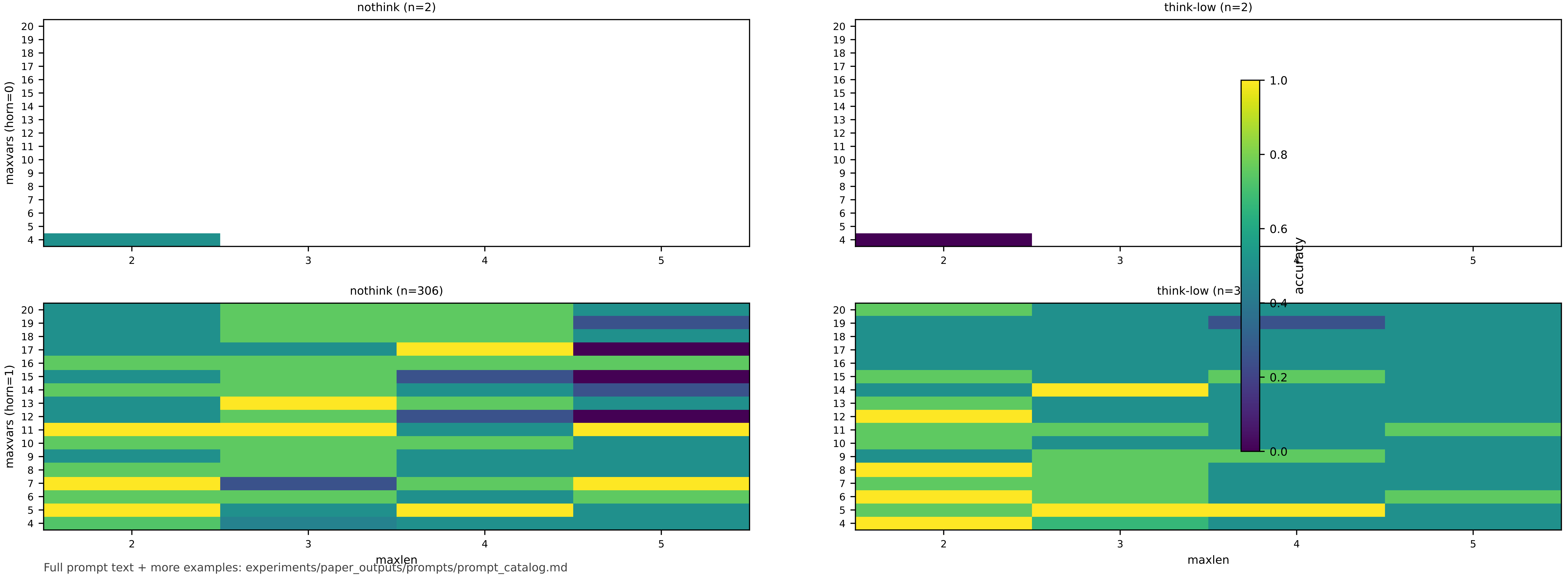
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_c1b2be97aa (horn_if_then)

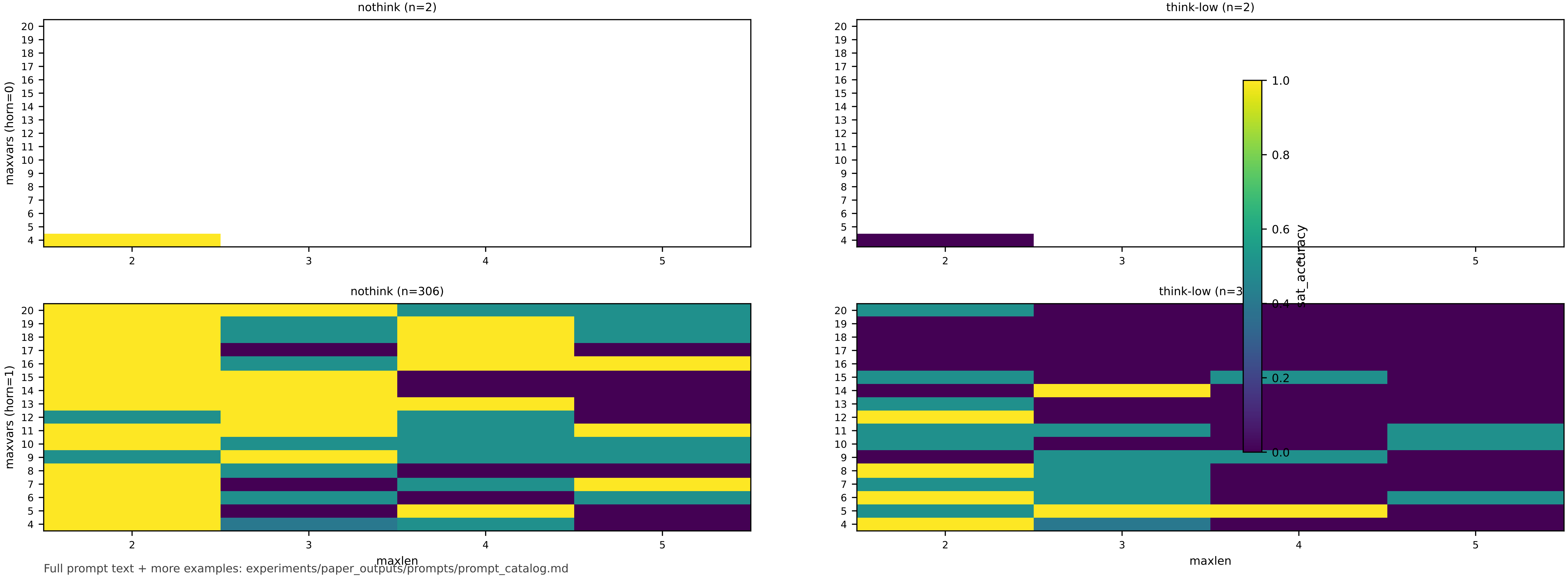
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_c1b2be97aa (horn_if_then)

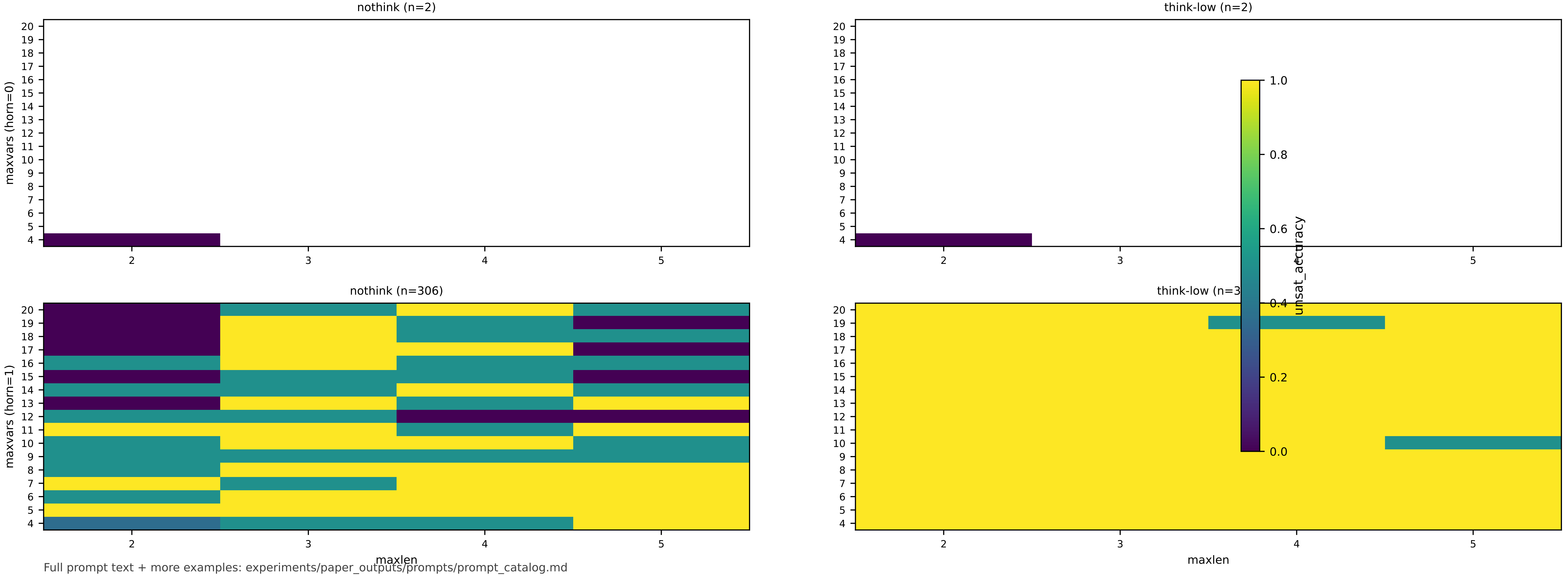
prompt_template=prompts/_template_unified.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a propositional logic problem.

Choose the appropriate interpretation based on how the statements are rendered
below.
- If you see facts like "p1." and rules like "if p2 and p3 then p4.", treat
them as Horn facts and implications, and determine whether p0 can be derived.
- If you see disjunctions like "p1 is true or p2 is false." or compact forms
like "p1 or not(p2).", treat them as CNF clauses, and determine whether the
set is a contradiction (unsatisfiable) or satisfiable.

Conventions
- Propositional variables are written as pN, where N is a number.
- All statements are jointly assumed true (conjoined).
...
```

Example (horn=1, low, maxvars=4, maxlen=2, satflag=1)

```
if p4 then p0.
p2.
if p3 then p1.
if p3 then p4.
if p2 then p1.
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — accuracy — prompt_c6875730a1 (horn_if_then)

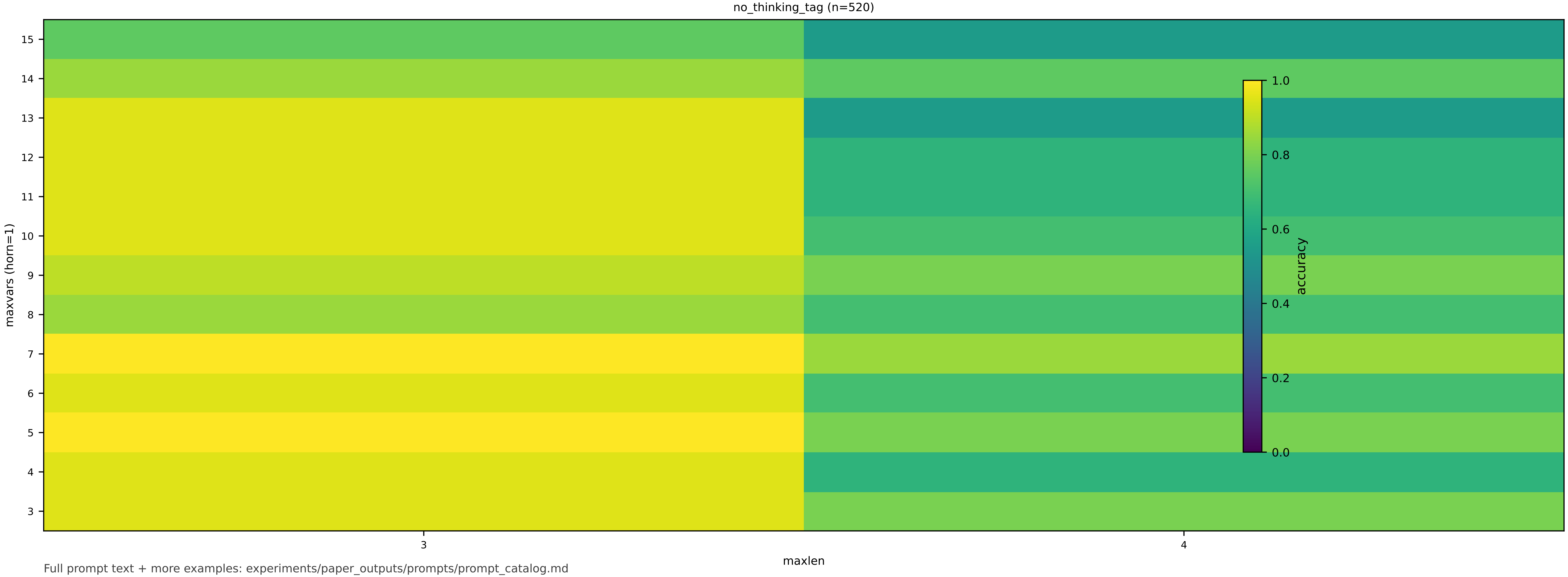
prompt_template=prompts/exp6_horn_yesno.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a problem in propositional logic containing both facts
and if-then rules.
You will get a list of facts and if-then rules and have to determine whether a
fact p0 can be derived from this list.
If a fact p0 can be derived, the last word of your answer should be 'yes',
otherwise the last word should be 'no'.

Facts are represented as 'pN' where N is a number.
All the statements are either facts or if-then rules allowing to derive a
single fact.
All the given statements are implicitly connected with 'and': they are all
claimed to be true.

...
```

Example (horn=1, low, maxvars=3, maxlen=3, satflag=1)

```
p1. p2. if p1 then p0. Answer: yes.
Example 2. Statements: p1. p2. if p1 then p9. Answer: no.
Example 3. Statements: p1. if p1 then p2. if p2 then p0. Answer: yes.
Example 4. Statements: p1. if p1 then p3. if p2 and p1 then p0. Answer: no.
Example 5. Statements: p1. if p1 then p2. if p2 then p3. if p3 then p0.
Answer: yes.
Example 6. Statements: p1. if p1 then p2. if p2 then p1. if p3 then p0.
Answer: no.
Example 7. Statements: p1. p3. if p1 then p2. if p2 and p3 then p4. if p4 then
p0. Answer: yes.
Example 8. Statements: p1. if p1 then p2. if p2 and p3 then p4. if p4 then p0.
Answer: no.
Example 9.  Statements: p6. p3. if p3 then p1. if p3 then p1. if p4 and p5
...
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — sat_accuracy — prompt_c6875730a1 (horn_if_then)

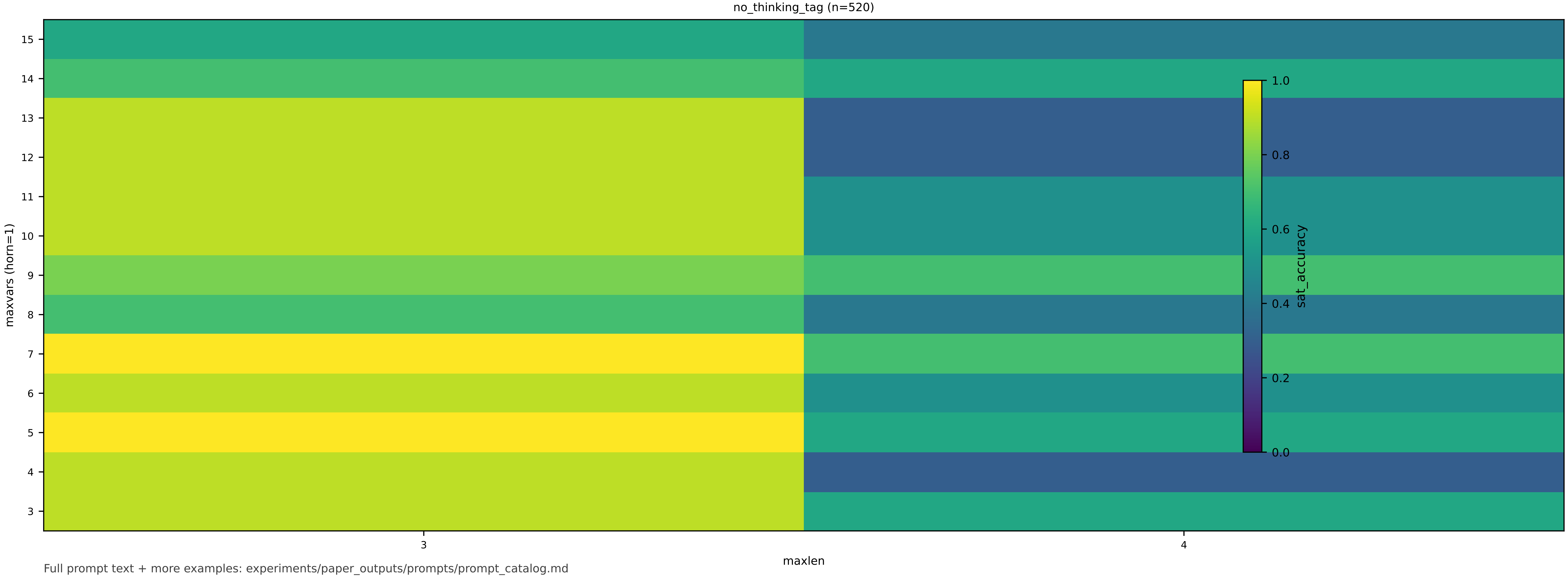
prompt_template=prompts/exp6_horn_yesno.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a problem in propositional logic containing both facts
and if-then rules.
You will get a list of facts and if-then rules and have to determine whether a
fact p0 can be derived from this list.
If a fact p0 can be derived, the last word of your answer should be 'yes',
otherwise the last word should be 'no'.

Facts are represented as 'pN' where N is a number.
All the statements are either facts or if-then rules allowing to derive a
single fact.
All the given statements are implicitly connected with 'and': they are all
claimed to be true.

...
```

Example (horn=1, low, maxvars=3, maxlen=3, satflag=1)

```
p1. p2. if p1 then p0. Answer: yes.
Example 2. Statements: p1. p2. if p1 then p9. Answer: no.
Example 3. Statements: p1. if p1 then p2. if p2 then p0. Answer: yes.
Example 4. Statements: p1. if p1 then p3. if p2 and p1 then p0. Answer: no.
Example 5. Statements: p1. if p1 then p2. if p2 then p3. if p3 then p0.
Answer: yes.
Example 6. Statements: p1. if p1 then p2. if p2 then p1. if p3 then p0.
Answer: no.
Example 7. Statements: p1. p3. if p1 then p2. if p2 and p3 then p4. if p4 then
p0. Answer: yes.
Example 8. Statements: p1. if p1 then p2. if p2 and p3 then p4. if p4 then p0.
Answer: no.
Example 9.  Statements: p6. p3. if p3 then p1. if p3 then p1. if p4 and p5
...
```



Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md

google/gemini-2.5-flash-lite — unsat_accuracy — prompt_c6875730a1 (horn_if_then)

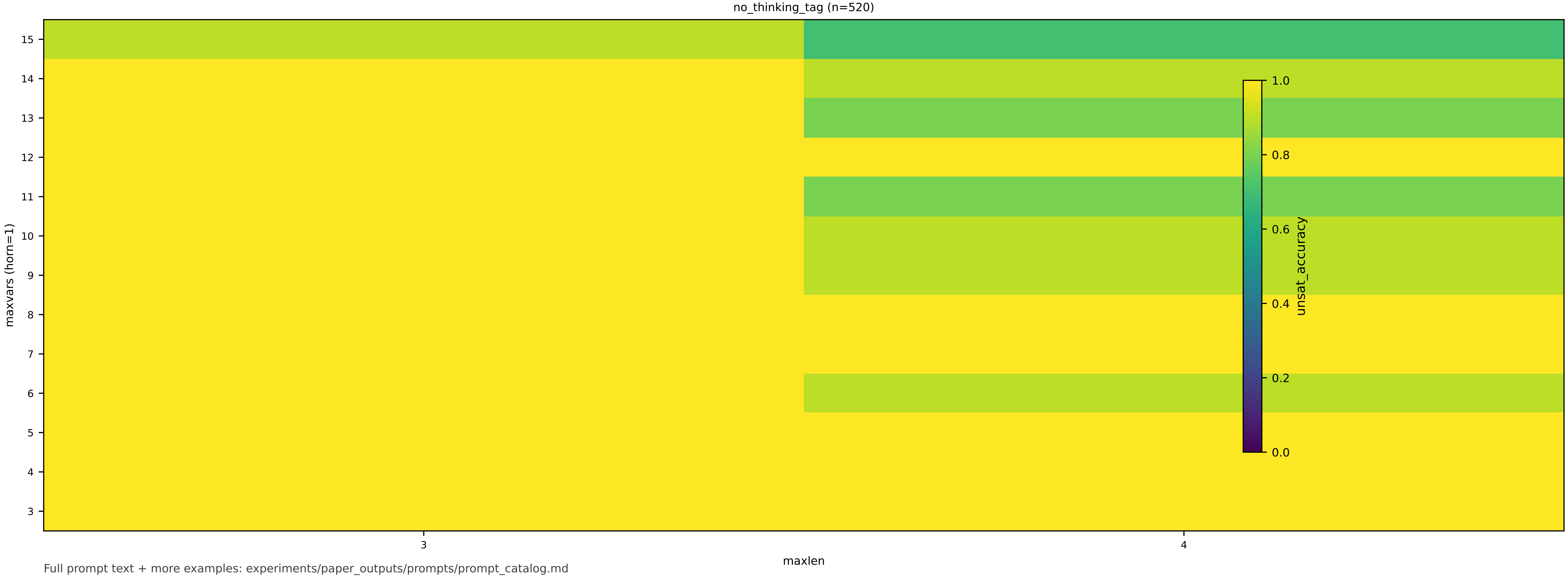
prompt_template=prompts/exp6_horn_yesno.j2 | parse_family=yes_no

Instruction excerpt:

```
Your task is to solve a problem in propositional logic containing both facts
and if-then rules.
You will get a list of facts and if-then rules and have to determine whether a
fact p0 can be derived from this list.
If a fact p0 can be derived, the last word of your answer should be 'yes',
otherwise the last word should be 'no'.

Facts are represented as 'pN' where N is a number.
All the statements are either facts or if-then rules allowing to derive a
single fact.
All the given statements are implicitly connected with 'and': they are all
claimed to be true.

...
```

Example (horn=1, low, maxvars=3, maxlen=3, satflag=1)

```
p1. p2. if p1 then p0. Answer: yes.
Example 2. Statements: p1. p2. if p1 then p9. Answer: no.
Example 3. Statements: p1. if p1 then p2. if p2 then p0. Answer: yes.
Example 4. Statements: p1. if p1 then p3. if p2 and p1 then p0. Answer: no.
Example 5. Statements: p1. if p1 then p2. if p2 then p3. if p3 then p0.
Answer: yes.
Example 6. Statements: p1. if p1 then p2. if p2 then p1. if p3 then p0.
Answer: no.
Example 7. Statements: p1. p3. if p1 then p2. if p2 and p3 then p4. if p4 then
p0. Answer: yes.
Example 8. Statements: p1. if p1 then p2. if p2 and p3 then p4. if p4 then p0.
Answer: no.
Example 9.  Statements: p6. p3. if p3 then p1. if p3 then p1. if p4 and p5
...
```


no_thinking_tag (n=520)

Full prompt text + more examples: experiments/paper_outputs/prompts/prompt_catalog.md