**calendar\_SMT\_** **Llama-3.1-8B-Instruct (5% correct within 5 iterations of evaluation)**

|  |  |
| --- | --- |
| **Number of Iterations** | **Count** |
| 1 | 1 |
| 2 | 2 |
| 4 | 2 |
| 5 | 95 |
| **Grand Total** | **100** |

pass 1:

44 errors (syntax and runtime), 55 wrong plan (semantics)

Pass 2:

48 errors (syntax and runtime), 49 wrong plan (semantics)

Pass 3:

47 errors (syntax and runtime), 50 wrong plan (semantics)

Pass 4:

44 errors (syntax and runtime), 51 wrong plan (semantics)

Pass 5:

41 errors (syntax and runtime), 54 wrong plan (semantics)

**task\_SMT\_** **Llama-3.1-8B-Instruct (5% correct within 5 iterations of evaluation)**

|  |  |
| --- | --- |
| **Number of Iterations** | **Count** |
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 96 |
| **Grand Total** | **100** |

pass 1:

85 errors (syntax and runtime), 14 wrong plan (semantics)

Pass 2:

78 errors (syntax and runtime), 20 wrong plan (semantics)

Pass 3:

67 errors (syntax and runtime), 30 wrong plan (semantics)

Pass 4:

68 errors (syntax and runtime), 28 wrong plan (semantics)

Pass 5:

51 errors (syntax and runtime), 45 wrong plan (semantics)

wrong use of method, data type unconscious (wrong casting, wrong grammar, etc.), unmatched “(“, invalid syntax (wrong grammar), index out of range,

Make sure to surround your final code with ```python\nYOUR\_CODE\n``` This is not a very good prompt for llama 8B. There are wrong code wrapping errors like:

{

"role": "assistant",

"content": "```python\nYOUR\_CODE\nfrom z3 import \*\n\n# Define the days of the week\ndays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday']\n\n# Define the start and end times\nstart\_times = ['09:00', '09:30', '10:00', '10:30', '11:00', '11:30', '12:00', '12:30', '13:00', '13:30', '14:00', '14:30', '15:00', '15:30', '16:00', '16:30', '17:00']\nend\_times = ['09:30', '10:00', '10:30', '11:00', '11:30', '12:00', '12:30', '13:00', '13:30', '14:00', '14:30', '15:00', '15:30', '16:00', '16:30', '17:00', '17:30']\n\n# Convert start and end times to minutes\nstart\_times\_minutes = [int(t[:2]) \* 60 + int(t[3:]) for t in start\_times]\nend\_times\_minutes = [int(t[:2]) \* 60 + int(t[3:]) for t in end\_times]\n\n# Define the meeting duration in minutes\nmeeting\_duration = 30\n\n# Define the existing schedules for Mary and Alexis\nmary\_schedule = {\n 'Monday': [],\n 'Tuesday': [start\_times\_minutes[2], start\_times\_minutes[13], start\_times\_minutes[14]],\n 'Wednesday': [start\_times\_minutes[1], start\_times\_minutes[13]],\n 'Thursday': [start\_times\_minutes[0], start\_times\_minutes[5]]\n}\n\nalexis\_schedule = {\n 'Monday': [start\_times\_minutes[0], start\_times\_minutes[2], start\_times\_minutes[8], start\_times\_minutes[14]],\n 'Tuesday': [start\_times\_minutes[0], start\_times\_minutes[2], start\_times\_minutes[5], start\_times\_minutes[7], start\_times\_minutes[13], start\_times\_minutes[16]],\n 'Wednesday': [start\_times\_minutes[0], start\_times\_minutes[2], start\_times\_minutes[11]],\n 'Thursday': [start\_times\_minutes[4], start\_times\_minutes[10], start\_times\_minutes[13], start\_times\_minutes[14], start\_times\_minutes[16]]\n}\n\n# Define the solver\nsolver = Solver()\n\n# Define the variables\nday = [Bool(f'day\_{i}') for i in range(len(days))]\nstart\_time = [Bool(f'start\_time\_{i}') for i in range(len(start\_times))]\nend\_time = [Bool(f'end\_time\_{i}') for i in range(len(end\_times))]\n\n# Define the constraints\nfor i, d in enumerate(days):\n solver.add(Or([day[i]]))\n for t in mary\_schedule[d]:\n if t in start\_times\_minutes:\n solver.add(Not(And([day[i], start\_time[start\_times\_minutes.index(t)]])))\n for t in alexis\_schedule[d]:\n if t in start\_times\_minutes:\n solver.add(Not(And([day[i], start\_time[start\_times\_minutes.index(t)]])))\n solver.add(Implies(day[i], Or([start\_time[t] for t in range(len(start\_times))])))\n solver.add(Implies(day[i], Or([end\_time[t] for t in range(len(end\_times))])))\n solver.add(Implies(day[i], start\_time[0] == Or([start\_time[t] for t in range(1, len(start\_times))])))\n solver.add(Implies(day[i], end\_time[-1] == Or([end\_time[t] for t in range(len(end\_times) - 1)])))\n for t in range(len(start\_times)):\n solver.add(Implies(day[i], And([start\_time[t], end\_time[t] == start\_time[t] + meeting\_duration])))\n solver.add(Implies(day[i], And([end\_time[t], start\_time[t] == end\_time[t] - meeting\_duration])))\n\n# Check the solution\nif solver.check() == sat:\n model = solver.model()\n day\_index = [i for i, d in enumerate(days) if model.evaluate(day[i]).as\_bool() == True][0]\n start\_time\_index = [i for i, t in enumerate(start\_times) if model.evaluate(start\_time[i]).as\_bool() == True][0]\n end\_time\_index = [i for i, t in enumerate(end\_times) if model.evaluate(end\_time[i]).as\_bool() == True][0]\n print(f\"SOLUTION:\")\n print(f\"Day: {days[day\_index]}\")\n print(f\"Start Time: {start\_times[start\_time\_index]}\")\n print(f\"End Time: {end\_times[end\_time\_index]}\")\nelse:\n print(\"No solution exists.\")\n```\n\nThe issue in the previous code was that the `YOUR\_CODE` string was not defined. This is because it was supposed to be a placeholder for the actual code. To fix this, we simply remove the `YOUR\_CODE` string and surround the code with triple backticks instead."

},

{

"role": "user",

"content": "The previous code had the following error:\nTraceback (most recent call last):\n File \"/home/ql98/calendar-planning/source/../output/SMT/meta-llama/Llama-3.1-8B-Instruct/calendar/n\_pass/calendar\_scheduling\_example\_812/3\_pass/solution.py\", line 1, in <module>\n YOUR\_CODE\nNameError: name 'YOUR\_CODE' is not defined\n\n\nPlease fix the code and provide a corrected version. Make sure to surround your final code with ```python\nYOUR\_CODE\n```."

},

1. calendar\_scheduling\_example\_114 correct
2. calendar\_scheduling\_example\_206 correct
3. calendar\_scheduling\_example\_579 error, super long weird repetitive code, cannot be extracted
4. calendar\_scheduling\_example\_594 correct
5. calendar\_scheduling\_example\_773 correct
6. calendar\_scheduling\_example\_812 error, wrong code wrapping, cannot extract code
7. calendar\_scheduling\_example\_910 correct
8. calendar\_scheduling\_example\_985 error, wrong code wrapping, cannot extract code