

PR #34778 完整报告

sgl-project/sglang

[Fix] Work around xgrammar 0.2.1 negative integer minimum in Kimi-K3 structural tags

合并时间: 2026-08-15 07:41

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34778>

执行摘要

- 一句话: 规避 xgrammar 0.2.1 负整数下限编译错误, 修复 Kimi-K3 工具调用
- 推荐动作: 值得精读, 可作为处理第三方依赖转换缺陷的典型 workaround 案例。关注点: 零拆分的 anyOf 策略如何保持语义等价; exclusiveMinimum 的边界转换; 以及 CI 中真实 xgrammar 编译测试的写法。

功能与动机

PR body 明确说明: Kimi-K3 工具参数 schema 若使用负整数下限, 在受约束解码下会得到结构性非法的工具调用。根因是 xgrammar 0.2.1 的 JSON-schema-to-grammar 转换对 `{"type": "integer", "minimum": -N}` 编译错误, 生成的文法接受不完整的字面量 `"-` 且拒绝所有合法负整数。

实现拆解

1. 定位转换入口: 在 `python/sglang/srt/function_call/kimik3_structural_tag.py` 的 `_value_format` 中拦截 `integer` 类型且 `schema` 为 `dict` 的情况。
2. 条件检测: 仅当 `schema` 不含 `maximum`、`exclusiveMaximum`、`multipleOf` 时, 收集 `minimum` 与 `exclusiveMinimum` (+1 后) 作为下界; 若 `max(lower_bounds) < 0` 则触发重写。
3. 零处拆分: 构造 `negative = {schema, maximum: -1}` 与 `nonnegative = {schema, minimum: 0}` (移除 `exclusiveMinimum`), 组成 `{"anyOf": [negative, nonnegative]}`。语义与原 `schema` 等价, 但每一分支都是 xgrammar 0.2.1 能正确编译的形式。
4. 配套测试: `test/registered/unit/function_call/test_kimik3_structural_tag.py` 新增 `test_strict_schema_handles_one_sided_negative_integer_minimum`, 跑在真实锁定的 xgrammar 上, 断言 `-1000000`、`-1`、`0`、`1000001` 被接受, 而 `"-`、`-1000001`、`1.5` 被拒绝; 既有测试 (覆盖 `maximum`、`multipleOf`、`additionalProperties` 默认等) 保持不变, 验证未触碰场景不受影响。

关键文件:

- `python/sglang/srt/function_call/kimik3_structural_tag.py` (模块 工具调用; 类别 `source`; 类型 `core-logic`; 符号 `_value_format`): 核心修复: 在 `_value_format` 中拦截负下限 `integer` `schema`, 按零拆分为 `anyOf` 双分支, 规避 xgrammar 0.2.1 编译缺陷, 同时保持 `schema` 语义。

- test/registered/unit/function_call/test_kimik3_structural_tag.py (模块 工具调用; 类别 test; 类型 test-coverage; 符号 test_strict_schema_handles_one_sided_negative_integer_minimum) : 新增 test_strict_schema_handles_one_sided_negative_integer_minimum, 使用真实锁定的 xgrammar 编译并断言正负边界值与非法值的接受 / 拒绝, 验证 workaround 语义等价。

关键符号: `_value_format`

关键源码片段

python/sclang/srt/function_call/kimik3_structural_tag.py

核心修复: 在 `_value_format` 中拦截负下限 integer schema, 按零拆分为 anyOf 双分支, 规避 xgrammar 0.2.1 编译缺陷, 同时保持 schema 语义。

```
def _value_format(
    schema: Union[bool, Dict[str, Any]],
    json_type: str,
    loose_string: bool = False,
) -> Format:
    if loose_string and json_type == "string":
        return AnyTextFormat()
    # XGrammar 0.2.1 对仅含负下限的 integer schema 编译错误:
    # {"type": "integer", "minimum": -N} 会接受不完整的 "-" 且拒绝所有负整数。
    # 这里把区间在零处拆成 anyOf 双分支, 语义不变但每支都能被正确编译。
    if (
        json_type == "integer"
        and isinstance(schema, dict)
        and "maximum" not in schema
        and "exclusiveMaximum" not in schema
        and "multipleOf" not in schema
    ):
        lower_bounds = []
        minimum = schema.get("minimum")
        if isinstance(minimum, int) and not isinstance(minimum, bool):
            lower_bounds.append(minimum)
        exclusive_minimum = schema.get("exclusiveMinimum")
        if isinstance(exclusive_minimum, int) and not isinstance(
            exclusive_minimum, bool
        ):
            lower_bounds.append(exclusive_minimum + 1) # 严格下界转为下界
        if lower_bounds and max(lower_bounds) < 0: # 只处理纯负下界场景
            negative = dict(schema)
            negative["maximum"] = -1
            nonnegative = dict(schema)
            nonnegative.pop("exclusiveMinimum", None)
            nonnegative["minimum"] = 0
            schema = {"anyOf": [negative, nonnegative]}
    return JSONSchemaFormat(
        json_schema=schema,
```

```
        style="qwen_xml" if json_type == "string" else "json",
    )
```

test/registered/unit/function_call/test_kimik3_structural_tag.py

新增 test_strict_schema_handles_one_sided_negative_integer_minimum, 使用真实锁定的 xgrammar 编译并断言正负边界值与非法值的接受 / 拒绝, 验证 workaround 语义等价。

```
def test_strict_schema_handles_one_sided_negative_integer_minimum():
    tool = Tool(
        type="function",
        function=Function(
            name="submit",
            strict=True,
            parameters={
                "type": "object",
                "properties": {
                    "value": {
                        "type": "integer",
                        "minimum": -1000000,
                    }
                },
                "required": ["value"],
                "additionalProperties": False,
            },
        ),
    )
    grammar = _grammar([tool], tool_choice="required")
    # 合法负整数与零、正数都应被接受
    for value in ("-1000000", "-999999", "-1", "0", "1000001"):
        assert _accepts(
            grammar,
            _tools_section(_call("submit", 1, _argument("value", "number", value))),
        )
    # 不完整 "-", 越界负数、非整数都应被拒绝
    for value in ("-", "-1000001", "1.5"):
        assert not _accepts(
            grammar,
            _tools_section(_call("submit", 1, _argument("value", "number", value))),
        )
```

评论区精华

Review 无评论。PR 评论仅有 CI rerun 请求 (gongy 两次 /rerun-test, 第二次针对 test_json_schema_constraint.py), 官方 bot 均反馈通过。PR body 已明确这是 converter 侧 workaround, xgrammar 修复后应移除。

- 暂无高价值评论线程

风险与影响

- 风险:

1. 范围限制: 重写仅作用于 integer 类型且无 upper bound/multipleOf 的 schema, 语义等价性有测试覆盖; 但 number 类型仍可能受 xgrammar 缺陷影响, 需后续跟进。
2. anyOf 分支: schema 拆分后通过 JSONSchemaFormat 传给 xgrammar, 需确认 anyOf 在 qwen_xml style 下编译行为一致 (测试已覆盖 integer 场景)。
3. exclusiveMinimum 处理: exclusive_minimum 转 minimum+1 是严格下界语义, 若 xgrammar 对 exclusiveMinimum 支持不足, 该转换可能影响边界值, 应关注边界 -1/0 的行为。
4. 版本依赖: 测试跑在锁定的 xgrammar 0.2.1 上, 依赖升级后重写逻辑可能变成死代码, README 或代码注释应提醒清理。
 - 影响: 影响范围: Kimi-K3 工具调用语法编译路径, 只在 strict schema 且使用负整数下限时生效; 不影响其他模型与常规工具调用。
 - 影响程度: 修复了受约束解码下的结构性非法工具调用, 属于精度 / 可用性改进; 同时保留 schema 语义, 不引入 breaking change。
 - 对团队: 新增了针对 xgrammar 转换缺陷的 workaround 模式, 后续 pin 升级后需跟进清理。
 - 风险标记: 第三方依赖缺陷 workaround, 依赖升级后需清理

关联脉络

- PR #34777 [Fix] Require JSON booleans for response_format json_schema.strict: 同属 xgrammar/JSON Schema 约束解码入口的 bugfix, 且 /rerun-test 也跑过 test_protocol.py。
- PR #34781 fix(muse-glimmer): parse required/named tool calls natively: 同为 function_call 模块下的工具调用解析修复, 属于同一功能线。
- PR #34886 [Docs] Update Kimi-K3 installation options: 涉及 Kimi-K3 模型部署文档, 与本 PR 的模型特定工具调用修复同属 Kimi-K3 支持脉络。