

PR #32525 完整报告

sgl-project/sglang

fix(sampling): reject conflicting structural tag constraints

合并时间: 2026-08-03 11:24

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32525>

执行摘要

- 一句话: 将 `structural_tag` 纳入语法互斥校验, 拒绝冲突请求
- 推荐动作: 这是一个值得精读的小型 bugfix, 适合作为理解 SGLang 采样参数校验与 `grammar_manager` 约束选择机制的入口。重点阅读 `sampling_params.py` 中的 `verify()` 与测试文件中通过 `GRAMMAR_VALUES` 驱动的参数化测试写法。也可结合 Issue #31680 中的 root cause 分析 (blame 追溯) 学习如何定位跨模块的静默逻辑缺陷。

功能与动机

Issue #31680 指出 `SamplingParams.verify()` 只对 `json_schema`、`regex`、`ebnf` 做互斥校验, 遗漏 `structural_tag`。 `grammar_manager.py` 中 `structural_tag` 与其他三种约束在 `if/elif` 链中只能选一个, 且它排在最后, 因此同时设置时会被静默丢弃, 调用方得不到任何错误或警告, 输出可能被错误的语法约束。该问题可通过 `/v1/completions` 或 `/v1/chat/completions` 公共 API 直接触达。PR body 明确说明修改动机是让冲突请求在预处理阶段被拒绝, 使行为显式化。

实现拆解

实现分为三步:

1. 在 `python/sglang/srt/sampling/sampling_params.py` 的 `SamplingParams.verify()` 方法中, 将 `self.structural_tag` 加入 `grammars` 列表, 使其参与互斥计数判断; 同时将错误消息更新为 `Only one of json_schema, regex, ebnf, or structural_tag can be set.`, 与 `grammar_manager` 的四种约束类型保持一致。
2. 重构 `test/registered/unit/sampling/test_sampling_params.py` 中的相关测试: 新增 `GRAMMAR_VALUES` 字典统一四种约束的合法取值; 将 `test_multiple_grammars_raises` 改为遍历 `structural_tag` 与其余三种约束的两两组合; 将 `test_single_grammar_valid` 改为遍历四种约束验证各自单独设置合法; 删除了 `test_all_three_grammars_set_raises`。
3. 第二个提交 (由维护者 hnyls2002 完成) 将测试进一步精简为只覆盖 `structural_tag` 组合, 并使用 `assertRaisesRegex(ValueError, "Only one of")` 匹配错误消息前缀, 降低对错误文案的精确耦合。配套验证: 通过 `CI /rerun-test test/registered/unit/sampling/test_sampling_params.py` 在 `ubuntu-latest` 上运行通过; 本地 `pytest` 显示 72 个用例与 10 个 `subtest` 通过。

关键文件:

- python/sclang/srt/sampling/sampling_params.py (模块 采样参数; 类别 source; 类型 core-logic; 符号 verify) : 修复的核心文件: 在 verify() 方法中把 structural_tag 加入语法互斥校验列表, 使冲突请求在校验阶段即被拒绝, 并同步更新错误消息。
- test/registered/unit/sampling/test_sampling_params.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 test_multiple_grammars_raises, test_single_grammar_valid, GRAMMAR_VALUES) : 测试配套: 引入 GRAMMAR_VALUES 字典参数化覆盖所有两两组合和单约束合法场景, 删除不再必要的三约束叠加用例, 并通过 subTest 明确失败维度。

关键符号: SamplingParams.verify, test_multiple_grammars_raises, test_single_grammar_valid

关键源码片段

python/sclang/srt/sampling/sampling_params.py

修复的核心文件: 在 verify() 方法中把 structural_tag 加入语法互斥校验列表, 使冲突请求在校验阶段即被拒绝, 并同步更新错误消息。

```
# 位于 SamplingParams.verify(vocab_size) 方法内的语法互斥校验分支
# 背景: json_schema / regex / ebnf 与 structural_tag 在 grammar_manager 中由固定 if/elif 链挑选,
# 只能同时启用一个; 此前 structural_tag 未纳入本校验, 导致它被静默丢弃 (issue #31680) 。
grammars = [
    self.json_schema,
    self.regex,
    self.ebnf,
    self.structural_tag, # 修复点: 补上第四个约束
] # 四种约束互斥, 最多只能有一个不为 None
if sum(x is not None for x in grammars) > 1:
    raise ValueError(
        "Only one of json_schema, regex, ebnf, or structural_tag can be set."
    )
```

test/registered/unit/sampling/test_sampling_params.py

测试配套: 引入 GRAMMAR_VALUES 字典参数化覆盖所有两两组合和单约束合法场景, 删除不再必要的三约束叠加用例, 并通过 subTest 明确失败维度。

```
# 四种语法约束的合法取值统一放在 GRAMMAR_VALUES 中, 便于参数化验证
GRAMMAR_VALUES = {
    "json_schema": '{"type":"object"}',
    "regex": "abc",
    "ebnf": 'root ::= "abc"',
    "structural_tag": '{"structures":[],"triggers":[]}',
}

def test_multiple_grammars_raises(self):
    """structural_tag 与任一其他语法约束组合时都应被拒绝。"""
    # 约束选择在 grammar_manager 中是固定 if/elif 链, 漏掉任一约束会静默丢弃
    for other in ("json_schema", "regex", "ebnf"):
```

```

with self.subTest(other=other):
    sp = self._make(
        structural_tag=self.GRAMMAR_VALUES["structural_tag"],
        **{other: self.GRAMMAR_VALUES[other]},
    )
    # 用正则匹配错误消息前缀，避免过度耦合具体文案
    with self.assertRaisesRegex(ValueError, "Only one of"):
        sp.verify(self.VOCAB_SIZE)

def test_single_grammar_valid(self):
    """每个约束单独设置时都应合法。"""
    for grammar, value in self.GRAMMAR_VALUES.items():
        with self.subTest(grammar=grammar):
            self._make(**{grammar: value}).verify(self.VOCAB_SIZE)

```

评论区精华

Review 阶段无公开评论，主要讨论发生在 Issue #31680 及其评论中：

- 作者 gitover22 两次 @ hnyls2002 请求 review 并启用 run-ci 标签，说明 CI 红色仅因缺少 run-ci 标签。
- 维护者通过 /rerun-test 触发焦点测试，首次因 test_sampling_params.py 文件名歧义（匹配到两个文件）失败，改用完整路径后成功运行。
- 第二个提交显示维护者在合并前直接精简了测试范围，将全组合测试收敛到 `structural_tag` 两两组合，说明对测试目标有明确取舍。
- 请求 review 与启用 run-ci (other): 维护者通过 /rerun-test 触发焦点测试，验证通过后合入。
- /rerun-test 文件名歧义 (other): 改用完整路径 test/registered/unit/sampling/test_sampling_params.py 后成功运行。

风险与影响

- 风险：风险较低，但需注意以下几点：
 1. sampling_params.py 的校验逻辑变更意味着曾经能通过校验的请求（同时设置 structural_tag 与其他语法）现在会被拒绝，属于 API 行为变更，存量客户端若依赖静默丢弃行为将收到 400 错误。
 2. 错误消息字符串被改写，若外部代码或测试依赖旧文案（Only one of regex, json_schema, or ebnf can be set.）可能失效；本次测试已改用正则前缀匹配以降低该风险。
 3. structural_tag 的取值在测试中使用了简化的 JSON，但实际运行时的非 None 判断不依赖具体内容，因此行为一致。
 4. 该修复不影响推理路径，仅作用于请求预处理，无性能回归。- 影响：影响范围限于请求校验路径：所有走 SamplingParams.verify() 的入口（包括 /v1/completions 与 /v1/chat/completions，以及直接使用 SamplingParams 的调用方）都会获得更严格的校验。对合法请求无影响，但对之前错误地同时设置多种语法约束的请求会由静默降级变

为显式报错，有助于尽早暴露客户端配置问题。团队方面，维护者介入调整测试使覆盖更聚焦，体现了小型 bugfix 的协作流程。- 风险标记：请求校验行为变更，错误消息文案变更，潜在破坏存量客户端请求

关联脉络

- PR #33025 [Kimi K3] Add reasoning, tool-call, and OpenAI serving support: 该 PR 扩展了 `function_call` / `structural_tag` 相关能力，与本次修复的 `structural_tag` 校验位于同一功能线上，后续新增约束时需注意与采样参数互斥校验保持同步。