















和表名一一对应,输入表名后判断该表是否存在:若存在,则能成功获取匹配字段;否则报错.在操作正确的情况下,表的对应匹配字段会从文件中被提取出来.

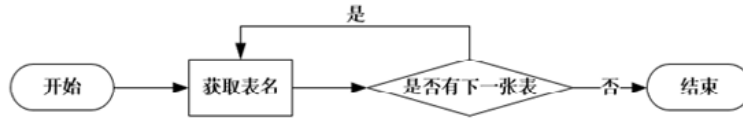


Fig.7 Parsing the table name process

图 7 解析表名流程

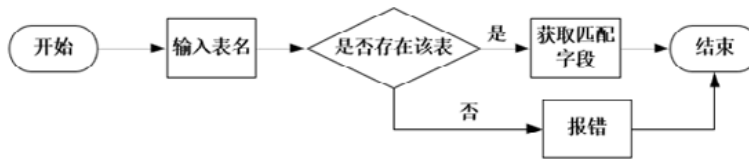


Fig.8 Procedure of extracting match field

图 8 解析匹配字段流程

### (3) 解析动作

动作名的层级关系为 actions→preamble→name.解析动作的流程如图 9 所示,根据表名获取该表的所有动作 ID 后,根据 ID 获取动作名,判断该 ID 是否为最后一个 ID:若不是,则继续获取 ID.表对应的所有动作名都会被从文件解析出来.

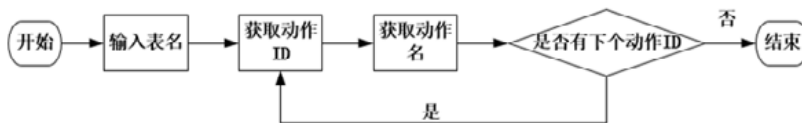


Fig.9 Procedure of extracting actions

图 9 解析动作流程

### (4) 解析动作参数

动作参数的层级关系为 actions→params→(id,name,bitwidth),解析动作参数的流程如图 10 所示,根据动作名判断该动作是否有参数:若有,则获取所有参数的信息;若没有,则结束.

每解析得到一个命令信息实体,命令信息实体即被插入命令信息实体列表.经过以上解析过程,最终生成测试用例所需的命令信息实体集合.

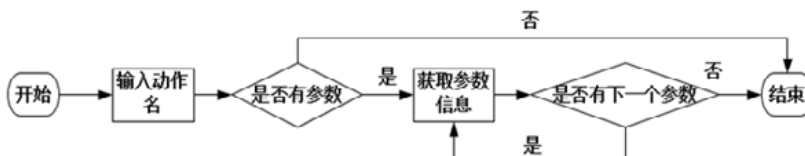


Fig.10 Procedure of extracting action parameters

图 10 解析动作参数流程

## 3.3 测试用例生成过程

前两节生成的命令信息实体集合为面向 P4 编程语言的 SDN 数据平面软件一致性测试用例的生成提供了数据基础,然而测试用例实体的生成需要一定的组织规范.本节不仅划分了测试用例类型,还分别设计了针对每种测试用例类型的测试用例构造方法.测试用例生成方法可分为生成单条测试用例和批量生成全部测试用例.



- 测试用例类型

本节依据测试用例结果类型,将最终生成的测试用例实体分为 5 种类型:命令合法、表名不合法、动作不合法、匹配域不合法、参数不合法。

- 构造测试用例

单条测试用例生成过程:用户依次选择测试用例类型、控制命令类型,再根据命令类型所需字段选择或编辑表名、动作名、匹配域、参数信息,最后生成单条测试用例。

批量生成全部测试用例集的过程为:以测试用例类型为组织单位构造测试用例,遍历解析过程生成的命令信息实体列表。若测试用例类型为命令合法,则直接依据每个命令信息实体的命令格式和字段信息构造测试用例;若测试用例类型为表名不合法,则将命令信息实体中的 TABLE\_NAME 字段修改为错误的 TABLE\_NAME 字段,再依据命令格式与其他字段信息构造测试用例;同理,动作不合法、匹配域不合法、参数不合法类型的测试用例构造方法与命令不合法类似。

## 4 测试用例生成示例

本节以装载 P4 程序的 simple\_switch 虚拟交换机为测试对象,说明一致性测试用例生成过程。

### 4.1 解析控制命令集

simple\_switch 虚拟交换机的控制命令集如图 11 所示。

```
P4@ubuntu:/behavioral-model/targets/l2_switch ./runtime_CLI --thrift-port 9090
Obtaining JSON from switch...
Done
Control utility for runtime P4 table manipulation
RuntimeCmd: ?
Documented commands (type help <topic>):
=====
counter_read      show_tables
counter_reset    swap_config
help             switch_info
load_new_config_file table_add
mc_dump         table_delete
mc_mgrp_create  table_dump
mc_mgrp_destroy table_dump_2
mc_node_associate table_dump_entry
mc_node_create  table_dump_group
mc_node_destroy table_dump_member
mc_node_dissociate table_indirect_add
mc_node_update  table_indirect_add_member_to_group
mc_set_lag_membership table_indirect_add_with_group
meter_array_set_rates table_indirect_create_group
meter_set_rates  table_indirect_create_member
port_add         table_indirect_delete
port_remove     table_indirect_delete_group
register_read    table_indirect_delete_member
register_reset   table_indirect_modify_member
register_write   table_indirect_remove_member_from_group
reset_state     table_indirect_set_default
serialize_state table_indirect_set_default_with_group
set_crc16_parameters table_info
set_crc32_parameters table_modify
shell           table_set_default
show_actions    table_show_actions
show_ports     write_config_to_file

Undocumented commands:
=====
EOF greet
RuntimeCmd: █
```

Fig.11 simple\_switch virtual switch control command set

图 11 simple\_switch 虚拟交换机控制命令集

由于控制命令集包含许多功能类似的表项操作命令,并且不同的 P4 网络设备控制命令集并不完全相同,因此,本实验并没有对全部控制命令生成一致性测试用例,而是选取 simple\_switch 虚拟机中具有代表性的表操作控制命令 table\_add,table\_delete,table\_dump,table\_modify 来生成一致性测试用例。

### 4.2 控制命令类型及格式

- table\_add 表项增加命令

合法的增加表项的命令格式如下:

table\_add<table name><action name><match fields>=><action parameters>[priority].

插入一个表项需要确定表名、动作名、匹配域、动作参数。以命令“table\_add ipv4\_lpm drop 0/32 0/32=>0”

为测试用例进行说明.其中,ipv4\_lpm 指定了该命令是向 ipv4\_lpm 这张表插入表项,ipv4\_forward 指定了表项匹配成功时执行的动作为 ipv4\_forward,0xffffffff/32 指定了匹配字段是 16 进制的 ffffffff,⇒是命令的格式需要所定义的箭头形状,⇒后的 0x0 0x0 指定了 ipv4\_forward 动作的两个参数均为 0,命令中的动作需要几个参数,⇒符号后就应当给出几个参数.命令中/后的数字如 0/32 表示的是该数长度为 32 位.

- table\_delete 表项删除命令

合法的删除表项的命令格式如下:

table\_delete(table name)(entry handle).

删除一个表项需要确定表项所在表的表名以及该表项在表中的编号,在这里被称为 entry handle.以命令“table\_delete ipv4\_lpm 0”为测试用例进行说明.其中,ipv4\_lpm 指定了该命令删除的表项在表 ipv4\_lpm 中,0 说明了该命令会删除表中编号为 0 的表项.

- table\_dump 表项查询命令

合法的查询表项的命令格式如下:

table\_dump(table name).

查询表项需要确定需要查询的表项所在表的表名.以命令“table\_dump ipv4\_lpm”为测试用例进行说明.其中,ipv4\_lpm 指定了该命令查询的是名为 ipv4\_lpm 这张表中的所有表项.

- table\_modify 表项修改命令

合法的修改表项的命令格式如下:

table\_modify(table name)(action name)(entry handle)[action parameters].

修改表项需要确定表项所在表的表名、表项中的动作名、表项的编号、动作的参数.以命令“table\_modify ipv4\_lpm ipv4\_forward 0 0x123456654321 1”为测试用例进行说明.其中,ipv4\_lpm 指定了该命令修改的是表 ipv4\_lpm 中的表项,ipv4\_forward 说明了该命令修改的表项中的动作为 ipv4\_forward,0 指定了该命令修改的是表 ipv4\_lpm 中编号为 0 的表项,0x123456654321 1 指定了该命令是将 ipv4\_forward 的两个参数改成 0x123456654321 1.命令要修改的表项中的动作有几个参数,就要给出几个参数.

### 4.3 生成测试用例集

如图 12 所示为测试用例生成的算法映射图,以测试用例类型命令合法、表名不合法、动作不合法、匹配域不合法、参数不合法为组织单位,以一个个命令信息实体为信息体,构造对应类型的 TABLE\_NAME, ACTION\_NAME, MATCH\_KEY, PARAMETERS 字段,生成 table\_add, table\_delete, table\_dump, table\_modify 命令.

如图 13 所示,本节以增删改查命令类型的测试用例生成过程为例进行说明.在完成了对 P4 信息文件(.p4info 文件)的解析后,以测试用例类型为组织单位,构造符合测试用例类型的字段,依据命令信息实体的命令格式,生成测试用例.表项的增加和修改动作需要用到表名、匹配字段、动作名和动作参数,而表项的查询和删除命令需要用到表名.本节实现的 SDN 数据平面软件一致性测试用例生成工具不直接依赖于 P4 程序,而是直接依赖于编译后的 P4 信息文件(.p4info 文件).

- 参数默认值的设定原则

命令中,用户需要自行填写的参数有匹配字段(match\_field)和动作参数(action\_params),这两个参数不能由.p4info 确定,需要用户根据从.p4info 文件中提取出参数的位数自行填入.程序可以识别十进制数并转换成十六进制,但是交换机中记录的是十六进制的格式,目前的 IP 地址格式是 32 位二进制数.为了方便也分成了 4 段,每段 8 位.考虑到以十进制数表示 IP 地址不直观也非当今主流,所以需要根据参数的位数给出一个默认的参数,以确定一个参数的格式.匹配字段中的参数往往是目的地址或者是源地址之类的 IP 地址.匹配字段的长度必须是 8bits 的整数倍.参考参数见表 1.

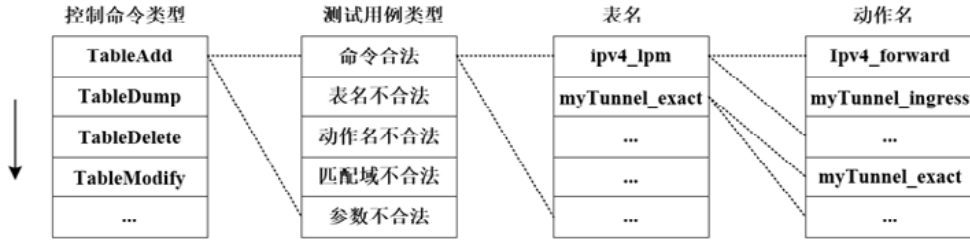


Fig.12 SDN data plane software conformance test case generation algorithm map for P4 programming language  
图 12 面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成算法映射图

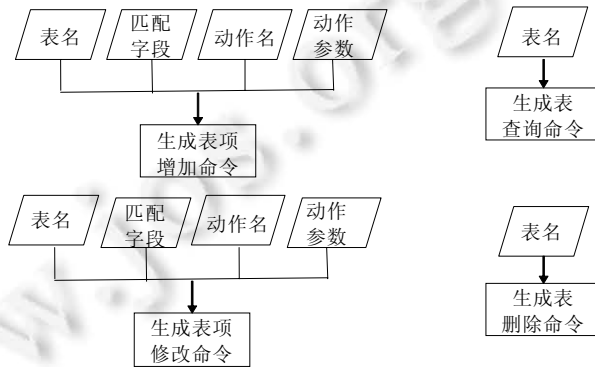


Fig.13 Procedure of generating commands  
图 13 命令生成流程

Table 1 Reference parameters

表 1 参考参数

参数	位数	默认参数
match_field	8	00
match_field	16	00:00
match_field	24	00:00:00
match_field	32	00:00:00:00
match_field	40	00:00:00:00:00
match_field	48	00:00:00:00:00:00
action_params	9	0

• 单个测试用例示例

本节设计了测试用例实体结构,每个测试用例实体分为 4 部分:命令格式 CF(CommandFormat)、前提条件 PC(Precondition)、测试用例 TC(TestCase)、期望输出结果 EO(ExpectedOutput).设计该结构的目的在于规范测试用例实体,使测试用例实体信息更加清晰明了,方便面向 P4 编程语言的 SDN 数据平面软件一致性测试实际结果与预期结果比对过程.前提条件 PC 和期望输出结果 EO 为测试系统初期部署过程所作工作,本文不作赘述.

图 14 所示为一个测试用例实例,其中,

- 命令格式 CF 为“table\_add ipv4\_lpm myTunnel\_ingress 0xffffffff/32⇒0x0 0x0”;
- 前提条件 PC 为“The entity to be added should not be duplicate”;
- 测试用例 TC 为“table\_add ipv4\_lpm myTunnel\_ingress 0xffffffff/32⇒0x0 0x0”;
- 期望输出结果 EO 为“Entry has been added with handle N”.

TC 中,table\_add 和 ipv4\_lpm 代表向 ipv4\_lpm 表插入表项,0xffffffff/32 代表匹配字段是 32 位的 0xffffffff,匹配成功则向表中插入参数为 0x0 和 0x0 的动作 myTunnel\_ingress.

```

//测试用例实例:
CF_table_add(table name)<(action name)<(match fields)>=>(action parameters)[priority]
PC_The entity to be added should not be duplicate
TC_table_add ipv4_lpm myTunnel_ingress 0xfffffff/32=>0x0 0x0
EO_Entity has been added with handle N

```

Fig.14 Test case example diagram

图 14 测试用例实例图

- 测试用例集示例(如图 15 所示)

```

File Edit Search Options Help
table name:ipv4_da_lpm
action_name:set_l2ptr
action_name:my_drop

table name:mac_da
action_name:set_bd_dmac_intf
action_name:my_drop

table name:send_frame
action_name:rewrite_mac
action_name:my_drop

表名不合法
CF_add_ipv4_da_lpm_ipv4_set_l2ptr_hdr.ipv4.dstAddr_dstAddr-port_InvalidTN1
PC_The entity to be added should not be duplicate
TC_table_add ipv4_da_lpm_error set_l2ptr 0/32 => 255.255.255.0
EO_Invalid table name ipv4_da_lpm_error

CF_add_ipv4_da_lpm_ipv4_my_drop_hdr.ipv4.dstAddr_dstAddr-port_InvalidTN1
PC_The entity to be added should not be duplicate
TC_table_add ipv4_da_lpm_error my_drop 0/32 =>
EO_Invalid table name ipv4_da_lpm_error

CF_add_mac_da_ipv4_set_bd_dmac_intf_hdr.ipv4.dstAddr_dstAddr-port_InvalidTN1
PC_The entity to be added should not be duplicate
TC_table_add mac_da_error set_bd_dmac_intf 0/32 => 255.255.255.0 00:00:00:00:00:00 7
EO_Invalid table name mac_da_error

CF_add_mac_da_ipv4_my_drop_hdr.ipv4.dstAddr_dstAddr-port_InvalidTN1
PC_The entity to be added should not be duplicate
TC_table_add mac_da_error my_drop 0/32 =>
EO_Invalid table name mac_da_error

```

Fig.15 Test case set

图 15 测试用例集

#### 4.4 测试用例自动生成工具测试

本节对我们已经实现的 SDN 数据平面软件一致性测试用例自动生成工具进行一致性测试用例生成测试实验,该实验选取了 3 个 P4 程序作为测试工具的输入,通过分析生成测试用例覆盖程度验证其有效性。

- demo1 含有 1 张表,名为 ipv4\_lpm,表中有 3 个动作:ipv4\_forward,drop,NoAction,因此,预期生成 3 条表项的插入命令、1 条表项的查询命令、1 条表项的删除命令、3 条表项的修改命令;
- demo2 含有 3 张表,分别是 ipv4\_da\_lpm,mac\_da,send\_frame.其中,表 ipv4\_da\_lpm 中有 2 个动作:set\_l2ptr, my\_drop,表 mac\_da 中有 2 个动作:set\_bd\_dmac,my\_drop,表 send\_frame 中有 2 个动作:rewrite\_mac, mydrop.预期生成 6 条表项的插入命令、3 条表项的查询命令、3 条表项的删除命令、6 条表项的修改命令;
- demo3 含有 2 张表,分别是 ipv4\_lpm,myTunnel\_exact.其中,ipv4\_lpm 表中有 4 个动作:ipv4\_forward, myTunnel\_ingress,drop,NoAction,表 myTunnel\_exact 中有 3 个动作:myTunnel\_forward,myTunnel\_egress,drop.预期生成 7 条表项的插入命令、2 条表项的查询命令、2 条表项的删除命令、7 条表项的修改命令。

如表 2 所示,3 个 P4 程序共 12 条被测控制命令通过一致性测试用例自动生成工具进行一致性测试用例生成测试.我们发现:生成测试用例的实际结果与预期结果完全一致,每条被测命令生成的测试用例集都覆盖了该控制命令字段所有可能出现的情况,实现了全覆盖.则以上实验验证了面向 P4 编程语言的 SDN 数据平面软件一致性测试用例自动生成工具具有有效性。

**Table 2** Test case automatic generation tool test data table

表 2 测试用例自动生成工具测试数据表

测试用例		预期结果					实际结果				
		命令 合法	表名 不合法	动作 不合法	匹配域 不合法	参数 不合法	命令 合法	表名 不合法	动作 不合法	匹配域 不合法	参数 不合法
demo1	table_add	3	3	3	3	3	3	3	3	3	
	table_delete	1	1	1	1	1	1	1	1	1	
	table_modify	3	3	3	3	3	3	3	3	3	
	table_dump	1	1	1	1	1	1	1	1	1	
demo2	table_add	6	6	6	6	6	6	6	6	6	
	table_delete	3	3	3	3	3	3	3	3	3	
	table_modify	6	6	6	6	6	6	6	6	6	
	table_dump	3	3	3	3	3	3	3	3	3	
demo3	table_add	7	7	7	7	7	7	7	7	7	
	table_delete	2	2	2	2	2	2	2	2	2	
	table_modify	7	7	7	7	7	7	7	7	7	
	table_dump	2	2	2	2	2	2	2	2	2	

#### 4.5 测试工具界面

测试用例生成工具的界面如图 18 所示,在用户选定.p4info 文件后,点击 Write In 按钮,则生成.p4info 文件对应的全部测试用例.在选定.p4info 文件后,依次选择 table、actions 下拉框,随后读取 match\_field 和 action\_params,点击 Generate entry 按钮,即在下方空白处显示表项的信息.

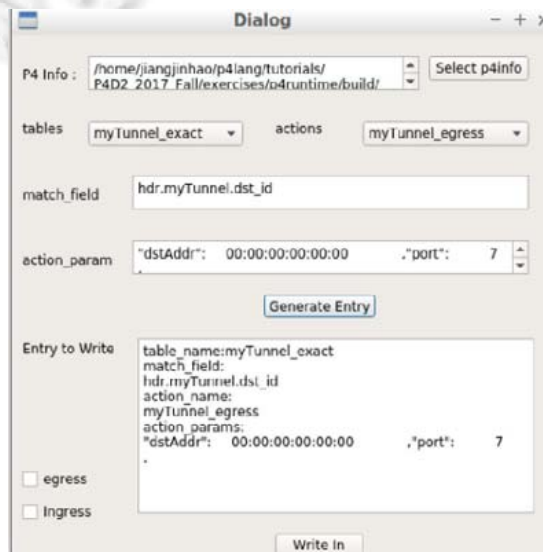


Fig.18 Test case generation tool interface

图 18 测试用例生成工具界面

## 5 总结

面向 P4 编程语言的一致性测试本质上是在将测试用例发送给 P4 网络设备,通过比较实际输出和预期输出的一致程度,保证了 P4 网络设备在错综复杂的网络环境中的表现与预期表现一致性.测试用例是执行一致性测试的载体,传统的人工构造测试用例是一项繁琐耗时费力的工作.本文介绍了 SDN 架构、P4 编程语言特性,调研了 SDN 领域一致性测试现有工作情况,设计了用于面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成方法,实现了面向 P4 编程语言的 SDN 数据平面软件一致性测试用例自动生成工具.实验结果表明:该测试工具能够由用户自己选择或编辑生成单条测试用例,也可以按照预期生成任意 P4 程序的测试用例集.

除此之外,在实际应用中,本文工作还有进一步提升空间。

本文是面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成研究工作,在本文工作的基础上,可以针对 P4 编程语言的 SDN 数据平面软件进行真实网络环境下的一致性测试工作研究;

面向 P4 编程语言的 SDN 数据平面软件一致性测试过程需要包括测试环境搭建、测试系统初期部署、一致性测试用例生成、一致性测试等阶段.后续工作可以设计面向 P4 编程语言的 SDN 数据平面软件一致性测试一体化过程,可以基于 gRPC 框架对真实网络环境下的测试用例的发送、测试服务调用和测试结果接收过程进行研究。

交换机操作系统 Stratum 支持 P4 API、P4Runtime API, 可以基于本文提出的面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成方法进行 Stratum 交换机一致性测试的研究。

## References:

- [1] Zhang CK, Cui Y, Tang W, Wu JP. Research progress of software defined network (SDN). Ruan Jian Xue Bao/Journal of Software, 2015,26(1):62–81 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4701.htm>. [doi: 10.13328/j.cnki.jos.004701]
- [2] Casado M, McKeown N, Shenker S. From ethane to SDN and beyond. ACM SIGCOMM Computer Communication Review. 2019, 49(5):92–95.
- [3] Shenker S. The future of networking, and the past of protocols. In: Proc. of the Open Networking Summit. 2011.
- [4] 1st P4 Workshop. P4 Official Website. Retrieved 1 August 2019.
- [5] Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G, Walker D. P4: Programming protocol-independent packet processors. Computer Communication Review, 2014,44(3):87–95.
- [6] OpenNetworking foundation. <https://www.opennetworking.org/>
- [7] Zhou Y, Bi J. ClickP4: Towards modular programming of P4. In: Proc. of the SIGCOMM Posters and Demos. Los Angeles, 2017.
- [8] Fan CZ, Bi J, Zhou Y. NS4: A P4-driven network simulator. In: Proc. of the SIGCOMM Posters and Demos. Los Angeles, 2017.
- [9] ONF Connect (NG-SDN Track).<https://www.opennetworking.org/wp-content/uploads/2019/09/2.00pm-Brian-OConnor-Stratum.pdf>
- [10] p4pktgen: Automated test case generation for P4 programs. [https://p4.org/assets/P4WS\\_2018/12\\_p4pktgen\\_Abstract.pdf](https://p4.org/assets/P4WS_2018/12_p4pktgen_Abstract.pdf).

## 附中文参考文献:

- [1] 张朝昆,崔勇. 软件定义网络(SDN)研究进展. 软件学报,2015,26(1):62–81. <http://www.jos.org.cn/1000-9825/4701.htm> [doi: 10.13328/j.cnki.jos.004701]



张雯雯(1995—),女,学士,主要研究领域为软件定义网络。



章玥(1981—),女,博士,副教授,CCF 专业会员,主要研究领域为软件定义网络,网络管理,通信软件。



许天予(1995—),男,学士,主要研究领域为软件定义网络。



郑孝遥(1981—),男,博士,副教授,CCF 专业会员,主要研究领域为信息安全,个性化推荐。