

Domain Specific Language Project

Instructor : Suman Pandey

Due: 11/30 23:59 (submit on GIST LMS)

1. Goal

Your primary goal is to design and implement a Domain-Specific Language (DSL), provisionally named IntentLang, to simplify network configuration and management. This DSL must allow network operations to be expressed in a human-readable, high-level intent. The core objective is to implement a robust compiler pipeline that translates these high-level intents into equivalent, executable, low-level configurations for diverse networking platforms. The implementation must achieve the following translation pipeline:

$$\text{IntentLang} \longrightarrow \text{JSON Semantic Model} \longrightarrow \text{Target Configurations}$$

You must implement code generation for at least two of the following target configuration formats:

- P4/OpenFlow Table Entries (for Software-Defined Networking environments)
- Cisco Config (e.g., IOS/NX-OS syntax for ACLs, QoS, Routing)
- Linux Config (e.g., `iptables`, `tc` for traffic control, `ip route` commands)

2. Structure of the Project (Example)

You are given full freedom to define the input and output formats of your program. The primary objective is to maximize the utility, practicality, and extensibility of the proposed solution. The example of your project should be like this :

No.	IntentLang	JSON Semantic Model	P4/OpenFlow Table Entry	Cisco Config	Linux Config
1	allow tcp from A to B	<code>{"type":"acl","action":"allow","protocol":"tcp","src":"10.0.0.1","dst":"10.0.0.2"}</code>	<code>acl_table: match={"src":"10.0.0.1","dst":"10.0.0.2","proto":"tcp"}, action=allow</code>	<code>ip access-list extended ALLOW_TCP permit tcp host 10.0.0.1 host 10.0.0.2</code>	<code>iptables -A INPUT -p tcp -s 10.0.0.1 -d 10.0.0.2 -j ACCEPT</code>
2	block icmp	<code>{"type":"acl","action":"deny","protocol":"icmp"}</code>	<code>acl_table: match={"proto":"icmp"}, action=deny</code>	<code>ip access-list extended BLOCK_ICMP deny icmp any any</code>	<code>iptables -A INPUT -p icmp -j DROP</code>
3	limit bandwidth 10Mbps for hostA	<code>{"type":"meter","host":"10.0.0.1","rate":"10Mbps"}</code>	<code>meter_table: match={"src":"10.0.0.1"}, action={"set_rate":"10Mbps"}</code>	<code>class-map match-any HOSTA match ip address hostA ACLpolicy-map LIMIT_HOSTA class HOSTA police 10000000 conform-action transmit</code>	<code>tc qdisc add dev eth0 root handle 1: htb default 10tc class add dev eth0 parent 1: classid 1:1 htb rate 10mbit filter add dev eth0 protocol ip parent 1:0 prio 1 u32 match ip src 10.0.0.1 flowid 1:1</code>
4	assign qos high to vlan10	<code>{"type":"qos","vlan":"10","priority":"high"}</code>	<code>qos_table: match={"vlan":"10"}, action={"set_priority":"high"}</code>	<code>interface vlan10 priority-queue out mls qos trust cos</code>	<code>tc class add dev eth0 parent 1: classid 1:10 htb rate 100mbit prio 0tc filter add dev eth0 protocol 802.1Q parent 1:0 prio 1 u32 match vlan 10 flowid 1:10</code>
5	ensure connectivity between hostA and hostB	<code>{"type":"connectivity","src":"10.0.0.1","dst":"10.0.0.2"}</code>	<code>flow_table: match={"src":"10.0.0.1","dst":"10.0.0.2"}, action="forward", path=["SW1","SW3"]</code>	Controller installs static routes or ACLs	<code>ip route add 10.0.0.2/32 via 10.0.0.254</code>
6	create vlan 20 name Engineering	<code>{"type":"vlan","id":"20","name":"Engineering","action":"create"}</code>	VLAN setup via P4 metadata (optional)	<code>vlan 20 name Engineering</code>	<code>ip link add link eth0 name eth0.20 type vlan id 20</code>
7	delete vlan 10	<code>{"type":"vlan","id":"10","action":"delete"}</code>	Remove VLAN metadata in tables	<code>no vlan 10</code>	<code>ip link delete eth0.10</code>
8	set route 10.0.0.0/24 via 192.168.1.1	<code>{"type":"route","dst":"10.0.0.0/24","next_hop":"192.168.1.1"}</code>	Install flow_table entry for 10.0.0.0/24	<code>ip route 10.0.0.0 255.255.255.0 192.168.1.1</code>	<code>ip route add 10.0.0.0/24 via 192.168.1.1</code>
9	monitor latency between hostA and hostB	<code>{"type":"monitor","metric":"latency","src":"10.0.0.1","dst":"10.0.0.2"}</code>	Table with timestamp/counters to measure RTT	Use ICMP or controller probe	<code>ping -c 4 10.0.0.2</code>
10	backup configuration now	<code>{"type":"backup","action":"now"}</code>	Save controller switch state to JSON/YAML	<code>copy running-config startup-conf</code>	—

5. Submission

- **Due Date:** November 30th, 23:59
- **Submission File:** A .zip file containing the complete source code
- **File Format:** project1_studentID1_studentID2.zip

6. Evaluation Criteria (Total 100 points)

DSL syntax design:	20 points
Lexical and Grammar Correctness:	25 points
Semantic Analysis / Error Handling:	20 points
Interpreter or Code Generator:	25 points
Documentation and Sample Programs:	10 points

If you have any questions, please contact the TA: hyunwoo_lee_ug@gm.gist.ac.kr