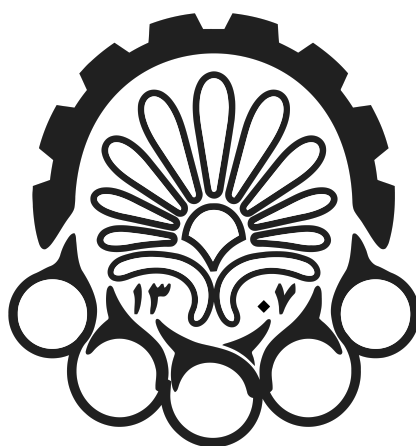


معماری افزاره‌های شبکه
دکتر صبا ئی



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری چهارم

۹ آذر ۱۴۰۳



دانشکده مهندسی کامپیوتر

معماری افزارهای شبکه

تمرین سری چهارم

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

سوال اول

در فرآیند طبقه‌بندی (classification) بسته‌ها:

۱. تشخیص جریان‌های ترافیکی چگونه انجام می‌شود؟

پاسخ

تشخیص جریان‌های ترافیکی (Traffic Flows Identification) در فرآیند طبقه‌بندی بسته‌ها به شناسایی و گروه‌بندی بسته‌هایی که به یک جریان خاص تعلق دارند، اشاره دارد. این جریان‌ها معمولاً بر اساس اطلاعات موجود در سربرگ بسته‌ها (Packet Headers) تعریف می‌شوند. به عنوان مثال، یک جریان می‌تواند شامل تمامی بسته‌هایی باشد که:

- دارای آدرس مبدأ (Source Address) و مقصد (Destination Address) یکسان هستند.

- از یک پروتکل مشخص مانند TCP یا UDP استفاده می‌کنند.

- دارای شماره پورت‌های مشخص مبدأ و مقصد هستند.

برای تشخیص این جریان‌ها، معمولاً از روش‌هایی مانند:

- تطبیق فیلدهای سربرگ (Header Field Matching)

- جداول حالت جریان (Flow State Tables)

استفاده می‌شود که امکان نگهداری اطلاعات مرتبط با هر جریان را فراهم می‌کند.

۲. انواع روش‌های طبقه‌بندی بسته‌ها را با ذکر ویژگی‌های کلی بیان کنید.

پاسخ

روش‌های طبقه‌بندی بسته‌ها را می‌توان به دو دسته کلی تقسیم کرد:

(آ) روش‌های مبتنی بر Header:

- این روش‌ها بسته‌ها را بر اساس اطلاعات موجود در Header مانند آدرس‌های IP، شماره پورت‌ها و پروتکل‌ها طبقه‌بندی می‌کنند.
- سریع هستند و به منابع پردازشی کمتری نیاز دارند.
- محدود به اطلاعات قابل مشاهده در سربرگ بوده و برای تحلیل داده‌های رمزنگاری شده مناسب نیستند.

(ب) روش‌های مبتنی بر Content:

- در این روش‌ها محتوای بسته‌ها (مانند داده‌های درون payload) مورد تحلیل قرار می‌گیرد.
- این روش‌ها دقت بالاتری در شناسایی نوع داده یا کاربرد خاص دارند.
- برای بسته‌های رمزنگاری شده یا حجم بالای داده‌ها، نیاز به پردازش سنگین و منابع بیشتر دارند.

(ج) روش‌های هندسی (Geometric Methods):

- این روش‌ها با استفاده از تکنیک‌های هندسی و تجزیه و تحلیل فضایی برای طبقه‌بندی بسته‌ها و جریان‌ها عمل می‌کنند.
- به عنوان مثال، از الگوریتم‌های مانند k-means clustering و SVM استفاده می‌شود که می‌توانند داده‌ها را به‌طور مؤثری در فضاها یا چندبعدی دسته‌بندی کنند.
- این روش‌ها معمولاً برای داده‌های با ویژگی‌های پیچیده و حجم بالای اطلاعات مناسب هستند.
- یکی از مزایای اصلی این روش‌ها، توانایی آن‌ها در شناسایی مرزهای غیرخطی بین کلاس‌هاست.
- این روش‌ها به محاسبات پیچیده‌تری نیاز دارند و در مقیاس‌های بزرگ می‌توانند منابع زیادی مصرف کنند.

۳. معیارهای کارایی روش‌های طبقه‌بندی بسته‌ها به بیان کرده و به اختصار شرح دهید.

پاسخ

کارایی روش‌های طبقه‌بندی بسته‌ها بر اساس معیارهای زیر سنجیده می‌شود:

- **دقت:** میزان بسته‌هایی که به درستی طبقه‌بندی شده‌اند. دقت بالا برای جلوگیری از تداخل جریان‌ها و کاهش خطا اهمیت دارد.
- **سرعت:** سرعت طبقه‌بندی بسته‌ها، به خصوص در شبکه‌های پرظرفیت، باید بالا باشد. روش‌های سریع مانند Header-Based برای کاربردهای بلادرنگ مناسب‌تر هستند.
- **مصرف حافظه:** روشی که از حافظه کمتری استفاده کند، به خصوص در مقیاس‌های بزرگ، کارآمدتر است. بهینه‌سازی ساختار داده‌ها مانند استفاده از Trie یا Bitmap می‌تواند مفید باشد.
- **قابلیت مقیاس‌پذیری:** توانایی روش برای مدیریت حجم زیاد قوانین و جریان‌ها. روش‌هایی که از ساختارهای موازی و الگوریتم‌های موثر استفاده می‌کنند، مقیاس‌پذیرتر هستند.
- **انعطاف‌پذیری:** توانایی سازگاری روش با تغییرات در قوانین یا شرایط شبکه. روش‌های مبتنی بر یادگیری ماشین اغلب انعطاف‌پذیری بیشتری دارند.
- **پیچیدگی محاسباتی:** میزان منابع محاسباتی مورد نیاز، که در روش‌های پیچیده‌تر مانند تحلیل محتوا، بیشتر است.

این معیارها بسته به نوع کاربرد و نیاز شبکه ممکن است اولویت‌بندی متفاوتی داشته باشند.

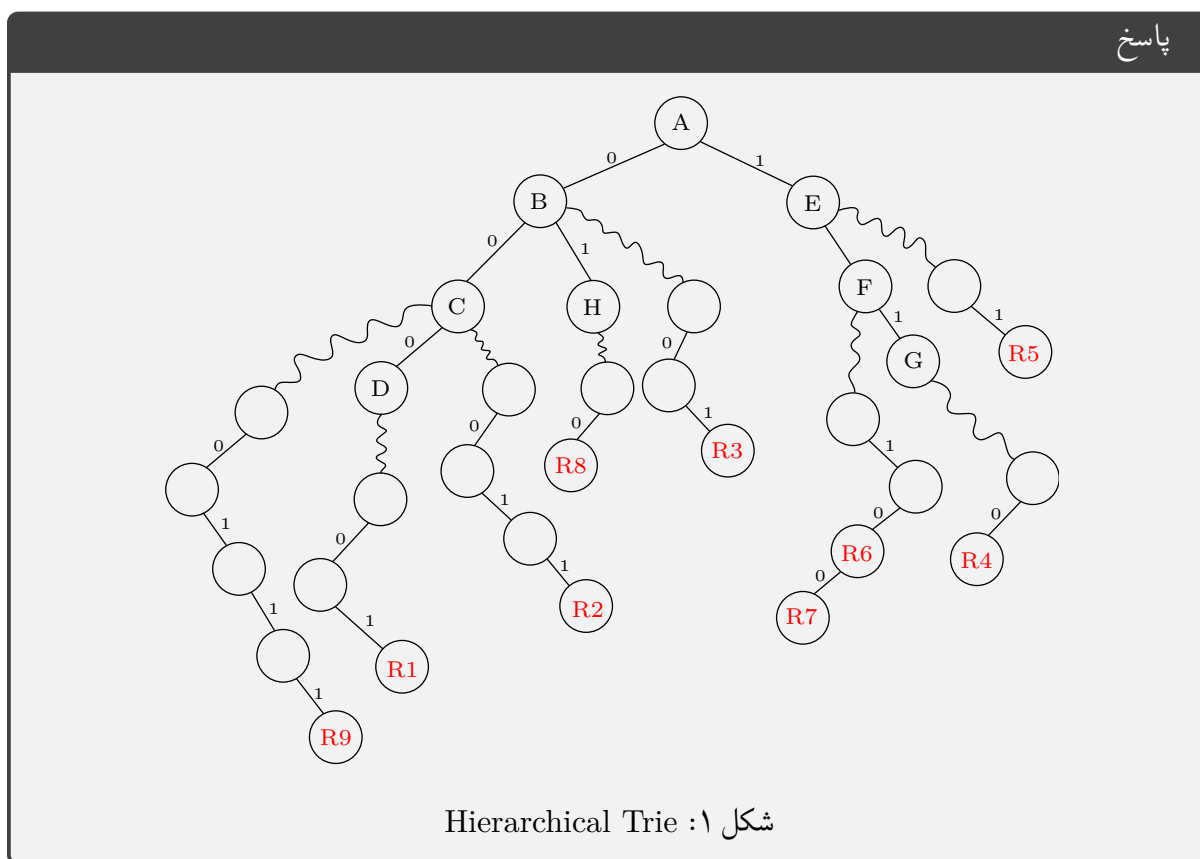
سوال دوم

جدول Classifier زیر را نظر بگیرید.

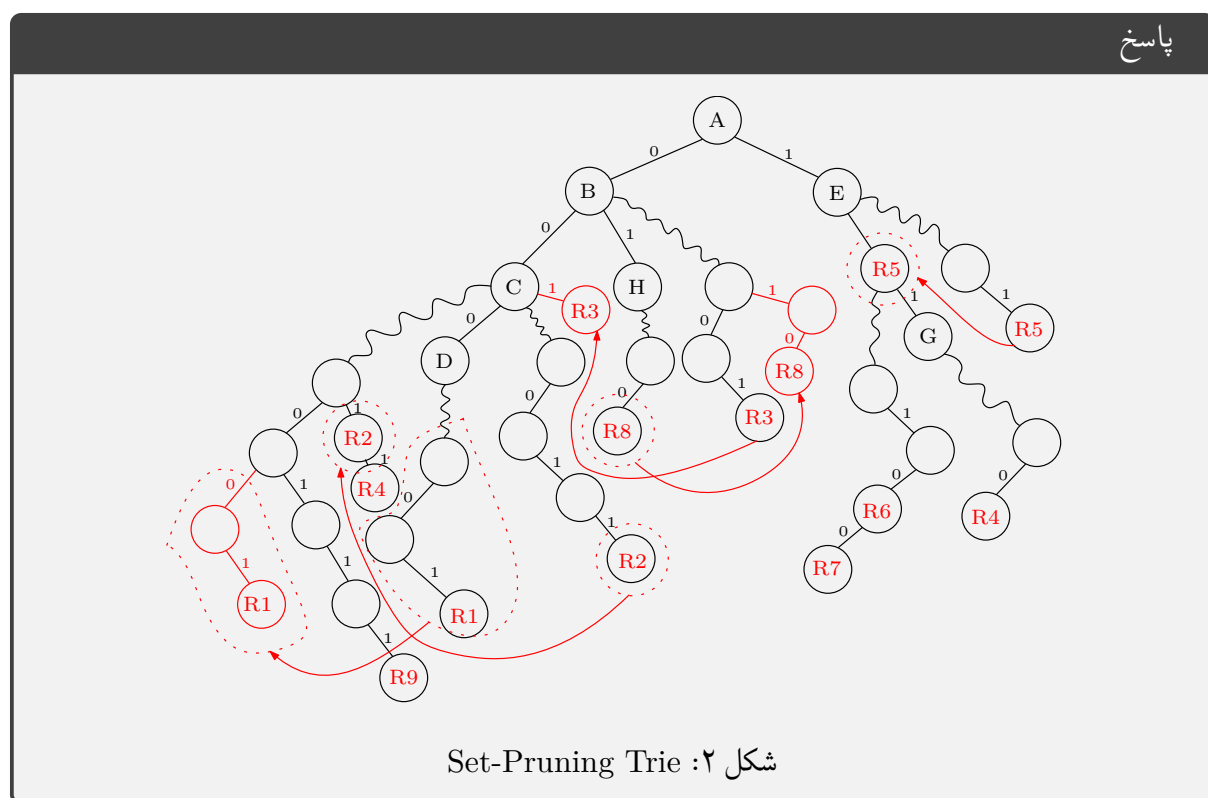
Rule	F1	F2	Action
R1	000*	01*	Act1
R2	00*	011*	Act2
R3	0*	01*	Act3
R4	111*	0*	Act4
R5	1*	1*	Act5
R6	11*	10*	Act6
R7	11*	1000*	Act7
R8	01*	0*	Act8
R9	0*	011*	Act9

مطلوب است، رسم ساختار جست و جوی طبقه بندی بسته ها بر اساس:

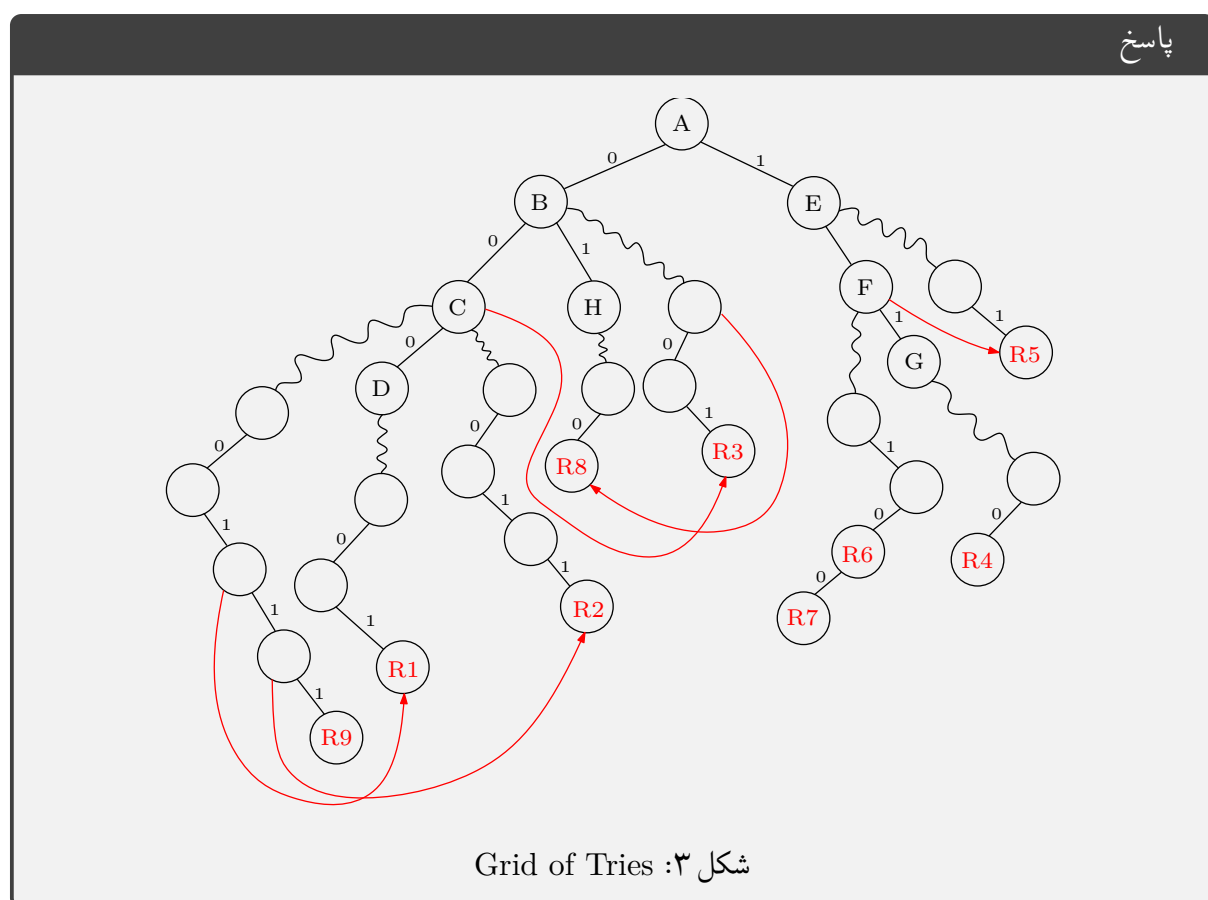
۱. Hierarchical Trie



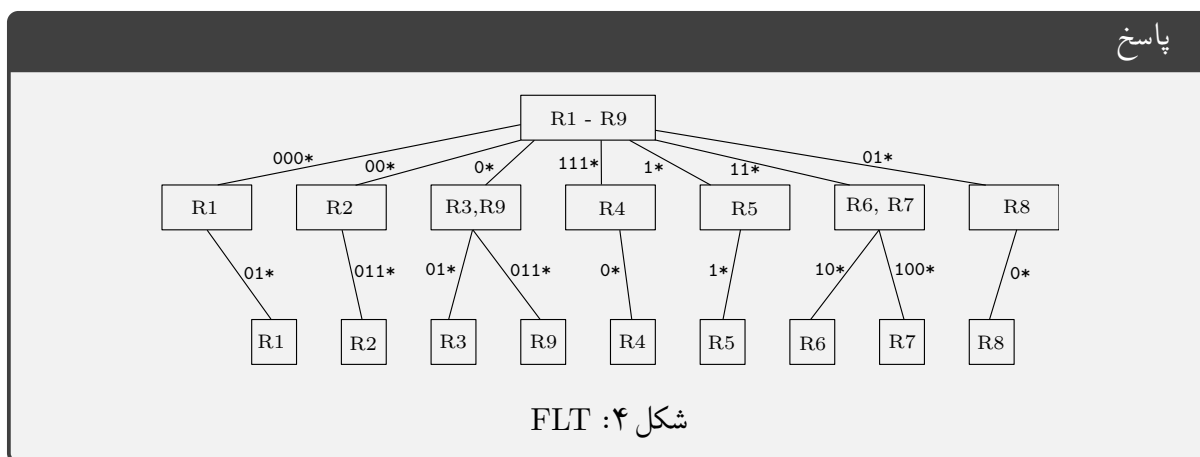
۲. Set-Pruning Trie



۳. Grid of Tries



۴. Field-Level Trie (FLT)

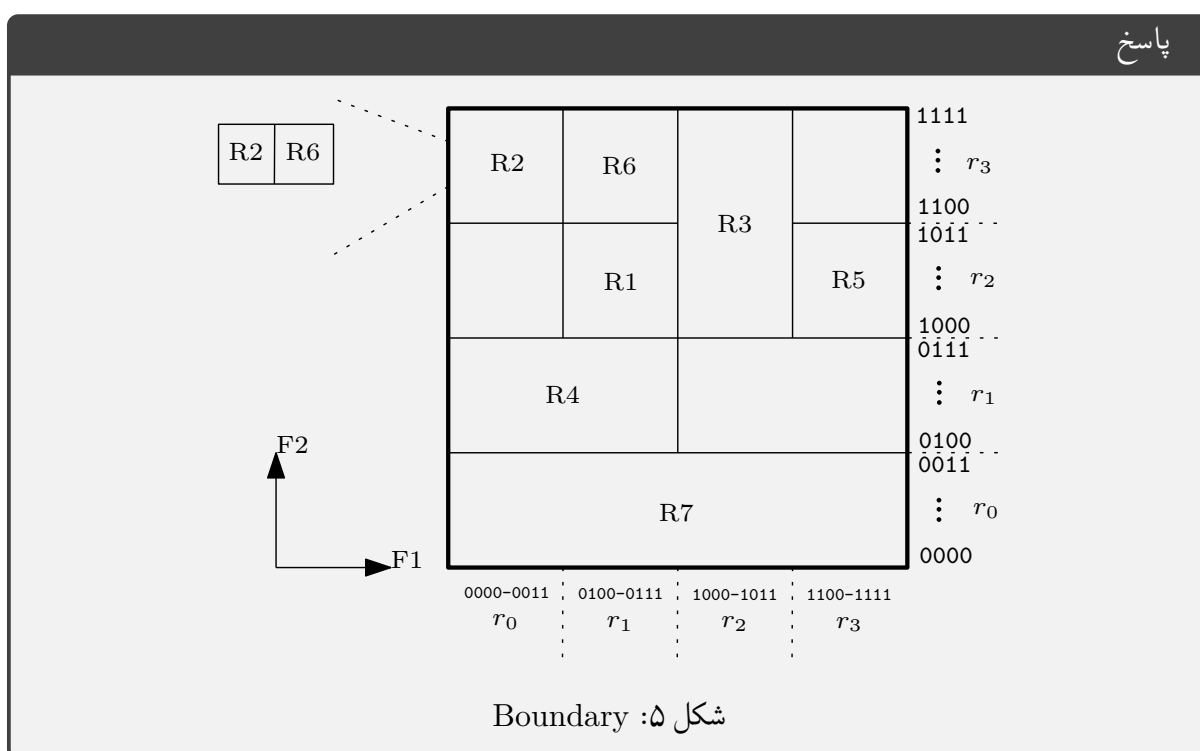


سوال سوم

جدول Classifier زیر را نظر بگیرید.

Rule	F1	F2	F3	F4	Action
R1	01*	10*	5	(7, 12)	Act0
R2	00*	11*	8	(6, 9)	Act1
R3	10*	1*	9	(4, 6)	Act2
R4	0*	01*	3	(10, 14)	Act1
R5	11*	10*	7	(6, 8)	Act0
R6	0*	11*	6	(11, 13)	Act3
R7	*	00*	7	(8, 12)	Act1

۱. بر اساس فیلدهای F1 و F2 فضای دو بعدی هندسی را رسم کنید و هر قانون (R1 تا R7) را به ناحیه‌های مربوطه بر اساس مقادیر F1 و F2 نگاشت کنید و هر ناحیه را در نمودار با برچسب مربوط به قانون مشخص نمایید.



۲. با استفاده از الگوریتم Cross-Producing ماتریس تصمیم‌گیری را ایجاد کنید.

پاسخ

۳. توضیح دهید که ستون‌های F3 و F4 چگونه بر روی Actions در هر ناحیه تأثیر می‌گذارند.

پاسخ

سوال چهارم

جدول Classifier زیر را در نظر بگیرید که در آن هر قانون با مجموعه‌ای از Fields همراه است.

Rule	B1	B2	B3	B4	Action
R1	110*	01*	1*	101*	Act1
R2	10*	0*	11*	10*	Act2
R3	1*	11*	10*	1*	Act3
R4	11*	01*	1*	10*	Act4

۱. روش Cross-Producing را برای قوانین داده‌شده اجرا کنید و تمامی مقادیر منحصر به فرد برای هر بعد فیلد را شناسایی کنید.

پاسخ

- (آ) اگر ایندکس‌های ما $[1, 2][2]$, $arr[2][1][2]$, Rule=R1 می‌شود
- (ب) اگر ایندکس‌های ما $[1, 2]$, $arr[1][0, 1][2]$ باشد، Rule=R2 می‌شود
- (ج) اگر ایندکس‌های ما $[1, 2, 3][3][1][1, 2, 3]$ باشد، Rule=R3 می‌شود
- (د) اگر ایندکس‌های ما $[1, 2][1, 2][1, 2]$, $arr[2, 3][1]$ باشد، Rule=R4 می‌شود
- همچنین بازه‌های فیلدهای ما به صورت زیر به دست می‌آید:

F1	0000* - 0111*
	1000* - 1011*
	1100* - 1101*
	1110* - 1111*

F2	0000* - 0011*
	0100* - 0111*
	1000* - 1011*
	1100* - 1111*

F3	0000* - 0111*
	1000* - 1011*
	1100* - 1111*

F4	0000* - 0111*
	1010* - 1011*
	1000* - 1001*
	1100* - 1111*

۲. به اختصار چگونگی Cross-Producing در تطبیق قانون برای طبقه‌بندی بسته در مقیاس بزرگ را شرح دهید، به‌ویژه با تمرکز بر کارایی حافظه و سرعت جست و جو.

پاسخ

روش Cross-Producing در تطبیق قوانین برای طبقه‌بندی بسته به‌طور خاص در مقیاس بزرگ، با هدف بهبود کارایی حافظه و سرعت جست‌وجو طراحی شده است. این روش به جای بررسی تک‌تک قوانین به صورت جداگانه، از ترکیب مقادیر در فیلدهای مختلف برای تولید یک فضای اشتراکی از مقادیر استفاده می‌کند. در ادامه، به اختصار ویژگی‌ها و مزایای این روش بررسی می‌شود:

(آ) کاهش فضای جست‌وجو:

با استفاده از ترکیب مقادیر از فیلدهای مختلف (به‌جای پردازش تک‌تک قوانین)، فضای جست‌وجوی قوانین به شدت کاهش می‌یابد. این موضوع منجر به کاهش تعداد مقایسه‌های لازم برای یافتن یک تطبیق می‌شود.

(ب) استفاده بهینه از حافظه:

روش Cross-Producing به گونه‌ای طراحی شده است که از ساختارهای داده‌ای مانند bitmaps یا جداول فشرده برای ذخیره مقادیر استفاده می‌کند. این ساختارها فضای حافظه مورد نیاز را بهینه می‌کنند، زیرا فقط ترکیبات مرتبط ذخیره می‌شوند و از ذخیره داده‌های غیرضروری جلوگیری می‌شود.

(ج) سرعت بالا در جست‌وجو:

با ترکیب مقادیر و تولید کلیدهای یکتا برای هر ترکیب، جست‌وجو در یک مرحله انجام می‌شود. این موضوع به دلیل کاهش مقایسه‌های مکرر در میان قوانین بهبود محسوسی در زمان جست‌وجو ایجاد می‌کند.

(د) مقیاس‌پذیری بالا:

در شبکه‌های بزرگ با تعداد زیاد قوانین، این روش به دلیل کاهش پیچیدگی زمانی و استفاده بهینه از منابع، مقیاس‌پذیری بسیار خوبی دارد. با افزایش تعداد قوانین، رشد مصرف منابع به صورت خطی یا کمتر از آن است.

(ه) کاربرد در سخت‌افزار:

این روش برای پیاده‌سازی در سخت‌افزارهایی مانند TCAM (تطبیق سریع محتوا) نیز مناسب است، زیرا ساختار ساده و قابل پیش‌بینی آن، امکان استفاده از منابع سخت‌افزاری برای پردازش موازی را فراهم می‌کند.

۳. مزایا و معایب استفاده از Cross-Producing را برای این جدول طبقه‌بندی شرح دهید.

پاسخ

(آ) مزایا:

- **بهینه‌سازی فضای جست‌وجو:** روش Cross-Producing به کمک ترکیب مقادیر فیلدها (مانند B1 تا B4) می‌تواند فضای جست‌وجو را کاهش دهد. در این جدول، با ترکیب مقادیر ممکن از هر فیلد، مجموعه‌ای یکتا از کلیدهای ترکیبی ایجاد می‌شود که جست‌وجو را سریع‌تر می‌کند.
- **افزایش کارایی حافظه:** برای جدول‌های کوچک مانند جدول فوق، ذخیره ترکیبات در قالب bitmaps باعث کاهش مصرف حافظه می‌شود، زیرا فقط ترکیبات ضروری ذخیره می‌گردند.

پاسخ

- **سرعت جست‌وجوی بالا:** در روش Cross-Producing، پس از تولید ترکیبات از فیلدها (مانند ترکیب B1 با B2 و ...)، جست‌وجو برای تطبیق قانون به صورت مستقیم روی کلیدهای تولیدشده انجام می‌شود، که نسبت به روش‌های سنتی مقایسه خط به خط، سرعت بیشتری دارد.
- **مقیاس‌پذیری:** در صورتی که تعداد قوانین (مانند R1 تا R4) یا تعداد فیلدها افزایش یابد، روش Cross-Producing به دلیل ساختار منظم، می‌تواند عملکرد مطلوبی ارائه دهد.
- **کاهش سربار محاسباتی:** ترکیب‌های از پیش محاسبه‌شده، نیاز به مقایسه تک‌تک فیلدها را در زمان جست‌وجو از بین می‌برد.

(ب) معایب:

- **پیچیدگی پیش‌پردازش:** تولید ترکیبات تمام فیلدها (به‌ویژه در جدول‌هایی با فیلدها یا قوانین زیاد) می‌تواند زمان‌بر و پرهزینه باشد. مثلاً برای جدول فوق، باید تمام مقادیر ممکن برای B1، B2، B3، و B4 تولید شوند که می‌تواند در جداول پیچیده‌تر مشکل‌ساز شود.
- **افزایش نیاز به حافظه در مقیاس بزرگ:** اگرچه روش Cross-Producing حافظه را برای جداول کوچک بهینه می‌کند، در جداول بزرگ با فیلدهای بیشتر و ترکیبات پیچیده‌تر، تعداد کلیدهای تولیدشده ممکن است بسیار زیاد شود و به مصرف بیش از حد حافظه منجر شود.
- **عدم انعطاف‌پذیری در تغییر قوانین:** اگر قوانین موجود (مانند تغییر در R1 یا اضافه‌شدن قوانین جدید) تغییر کنند، کل ترکیبات باید مجدداً محاسبه شوند، که باعث کاهش کارایی در جداول دینامیک می‌شود.
- **احتمال برخورد ترکیبات (Collision):** در موارد خاص که فیلدها الگوهای مشترک داشته باشند (مانند * در قوانین جدول فوق)، ترکیبات ممکن است به نتایج تکراری یا اشتباه منجر شوند که نیاز به الگوریتم‌های رفع برخورد دارد.
- **عدم تطابق با جداول بسیار کوچک:** برای جداول با تعداد قوانین و فیلدهای کم (مانند جدول بالا)، روش Cross-Producing ممکن است از نظر پیچیدگی پردازش اولیه نسبت به جست‌وجوی خط به خط معمولی کارایی کمتری داشته باشد.