



میثاق و کوروش

میثاق و کوروش، دو دانشجوی رشته کامپیوتر، توی کافه دانشگاه نشسته بودند و داشتند قهوه می‌خوردند که یهو میثاق با حالت چندان‌آوری قهوه‌اش رو روی میز گذاشت.

میثاق: "لعنت! این چه قهوه‌ایه آخه؟! نه طعم درست و حسابی داره، نه کیفیت! آب جوش طعم دار!"

کوروش: "خب چی کار کنیم؟ همینه که هست دیگه!"

میثاق: "نه داداش، من به فکر بهتر دارم. بیا به شرکت قهوه‌ساز راه بندازیم!"

کوروش: "جدی؟ شرکت قهوه‌ساز؟! فکر کنم زیادی قهوه خوردی!"

میثاق: "نه جدی میگم! می‌تونیم به شرکت بزیم و قهوه‌های با کیفیت به انسان‌ها بدهیم، بذار برات تعریف کنم..."

داستان شرکت MK

میثاق و کوروش می‌خواهند به شرکت بزرگ تو زمینه دستگاه‌های قهوه‌ساز بسازند. این شرکت با کلی مشتری طرفه که هر کدوم به جور قهوه‌ساز می‌خوان. یکی میگه فقط قهوه بده، یکی میگه چای هم باشه، یکی میگه خامه ایرلندی می‌خوام، یکی دیگه میگه نه همون خامه معمولی کافیه!

حالا میثاق و کوروش می‌خواهند به سیستم هوشمند طراحی کنند که بتونه سفارش‌های مختلف مشتری‌ها رو مدیریت کنه. برای همین اومدن به درخت ویژگی (Feature Tree) طراحی کردن که همه قابلیت‌های ممکن توی به دستگاه قهوه‌ساز رو نشون میده.

ساختار درخت ویژگی‌ها

این مدل شامل سلسله مراتبی از ویژگی‌های موجود در یک دستگاه قهوه است که به صورت یک درخت ریشه‌دار سازماندهی شده‌اند. ریشه همیشه نشان‌دهنده کل محصول دستگاه قهوه است، در حالی که گره‌های دیگر نشان‌دهنده ویژگی‌های دستگاه قهوه هستند.

روابط بین یک ویژگی والد و ویژگی‌های فرزند آن به صورت زیر دسته‌بندی می‌شوند:

- اجباری - ویژگی فرزند الزامی است
 - اختیاری - ویژگی فرزند اختیاری است
 - OR - حداقل یکی از زیر ویژگی‌ها باید انتخاب شود
 - XOR (انتخاب انحصاری) - دقیقاً یکی از زیرویژگی‌ها باید انتخاب شود
- با استفاده از یک مدل ویژگی، مشتری می‌تواند پیکربندی خاصی را درخواست کند و سیستم ما باید بررسی کند که معتبر هست یا نه.

قوانین پیکربندی معتبر:

- باید شامل ویژگی ریشه باشد
- ویژگی اجباری باید در صورت حضور والدش گنجانده شود
- از مجموعه ویژگی‌های دارای رابطه 'or' با والد حاضر، حداقل یکی باید حضور داشته باشد
- از مجموعه ویژگی‌های دارای رابطه 'xor' با والد حاضر، دقیقاً یکی باید حضور داشته باشد
- اگر فرزند در پیکربندی معتبر گنجانده شده، والدش نیز باید حضور داشته باشد
- هر ویژگی فقط یک بار در درخت مدل ویژگی ظاهر می‌شود

ساختار کلی ورودی

ورودی برنامه شامل چندین مورد تست است. هر مورد تست از دو بخش تشکیل شده:

1. توصیف درخت ویژگی‌ها

2. لیست پیکربندی‌هایی که باید اعتبارسنجی شوند

بخش اول: توصیف درخت ویژگی‌ها

هر خط از توصیف درخت به یکی از سه شکل زیر است:

```
F = F1 + F2 + ... + Fn
F = F1 | F2 | ... | Fn
F = F1 ^ F2 ^ ... ^ Fn
```

قوانین نام‌گذاری

- نام هر ویژگی فقط می‌تواند شامل حروف انگلیسی (بزرگ و کوچک) باشد
- بین حروف نام نمی‌تواند فاصله وجود داشته باشد
- هر ویژگی فقط یک بار در کل درخت می‌تواند ظاهر شود
- تعداد کل ویژگی‌ها حداکثر 1000 است

قوانین فرمت‌بندی

- می‌تواند قبل و بعد از علامت‌های $=$ ، $+$ ، $|$ ، $^$ و $?$ تعداد دلخواهی فاصله خالی وجود داشته باشد
- ابتدا و انتهای هر خط می‌تواند تعداد دلخواهی فاصله خالی داشته باشد
- اولین خط همیشه ریشه درخت را تعریف می‌کند (که باید در تمام پیکربندی‌های معتبر وجود داشته باشد)

معنای علامت‌ها

- $+$: ترکیب ویژگی‌های اجباری و اختیاری
- ویژگی‌های بدون $?$: اجباری هستند
- ویژگی‌های با $?$: اختیاری هستند
- $|$: رابطه OR (حداقل یکی از فرزندان باید انتخاب شود)
- $^$: رابطه XOR/Alternative (دقیقا یکی از فرزندان باید انتخاب شود)

پس از آخرین خط توصیف درخت، یک خط شامل فقط کاراکتر $\#$ می‌آید

بخش دوم: لیست پیکربندی‌ها

فرمت هر پیکربندی به شکل زیر است:

$\{F1, F2, \dots, Fn\}$

که در آن:

- $F1$ تا F_n : نام ویژگی‌هایی که در این پیکربندی انتخاب شده‌اند ($n > 0$)
- هر پیکربندی در یک خط جداگانه قرار می‌گیرد
- می‌تواند قبل و بعد از کاراکترهای $\{$ ، $\}$ و $,$ فاصله خالی وجود داشته باشد
- نام هر ویژگی فقط یک بار در هر پیکربندی می‌تواند ظاهر شود
- تمام ویژگی‌های ذکر شده باید در درخت تعریف شده باشند

پس از آخرین پیکربندی، یک خط شامل $\##$ می‌آید و پس از آخرین مورد تست، یک خط شامل $\###$ می‌آید.

خروجی

- برای هر پیکربندی در هر مورد تست، یک خط در خروجی چاپ می‌شود
- هر خط فقط شامل یکی از دو کلمه **Valid** یا **Invalid** است
- پس از پردازش هر مورد تست، یک خط شامل **+++** چاپ می‌شود

معیارهای اعتبارسنجی

یک پیکربندی زمانی معتبر است که تمام شرایط زیر را داشته باشد:

1. ریشه درخت در آن وجود داشته باشد
2. اگر یک ویژگی در پیکربندی وجود دارد، والدش هم باید وجود داشته باشد
3. اگر والد یک ویژگی اجباری در پیکربندی وجود دارد، آن ویژگی هم باید وجود داشته باشد
4. اگر والدی در پیکربندی وجود دارد که با فرزندانش رابطه OR دارد، حداقل یکی از فرزندانش باید در پیکربندی باشد
5. اگر والدی در پیکربندی وجود دارد که با فرزندانش رابطه XOR دارد، دقیقا یکی از فرزندانش باید در پیکربندی باشد

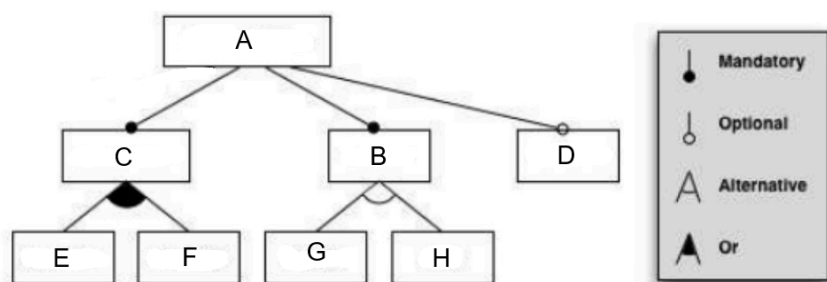
مثال کامل

ورودی

```
A = B + C + ?D
C = E | F
B = G ^ H
#
{A, B, C, E, D, G}
{A, B, E, F, C}
##
A=?B+C
B=D|E
#
{A,B,C,D,E}
{A,D,C}
##
###
```

خروجی

Valid
Invalid
+++
Valid
Invalid
+++



توضیح مثال اول

- پیکربندی اول معتبر است زیرا:
 - ریشه (A) را دارد
 - تمام ویژگی‌های اجباری (B, C) را دارد
 - برای B دقیقاً یک گزینه (G) را انتخاب کرده
 - برای C یک گزینه (E) را انتخاب کرده
 - D که اختیاری بوده را انتخاب کرده که مجاز است

- پیکربندی دوم نامعتبر است زیرا:
 - برای B (که در پیکربندی هست) باید دقیقاً یک فرزند (H یا G) انتخاب می‌شد

توضیح مثال دوم

- پیکربندی اول معتبر است زیرا:
 - A (ریشه) را دارد
 - C که اجباری است را دارد
 - B که اختیاری است را انتخاب کرده
 - برای B که انتخاب شده، حداقل یک فرزند (D و E) را دارد

- پیکربندی دوم نامعتبر است زیرا:
 - D را انتخاب کرده ولی والدش (B) را انتخاب نکرده

کوروش: "خب حالا این همه داستان تعریف کردی که چی بشه؟"
میثاق: "که بریم یه برنامه بنویسیم که بتونه این پیکربندی‌ها رو چک کنه دیگه! تازه می‌تونیم بعدا خودمون
یه شرکت قهوه‌ساز راه بندازیم، اسمش رو هم می‌ذاریم MK (میثاق و کوروش)!"
کوروش: "از دست تو! بزن بریم کد بزنیم..."

نکات و نحوه تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی zip با نام AP_Surname_SID.zip که SID شماره دانشجویی شماست در ریپلای به ایمیل پروژه ارسال کنید.
 - دقت کنید که نام پرونده قابل اجرای شما باید MK.out باشد.
 - هدف از این پروژه انتخاب دستیاران آموزشی جدید آموزشی است، لطفا پروژه را خودتان انجام دهید. در صورت کشف تقلب از فرایند انتخاب حذف خواهید شد.
 - در صورتی که در جایی ابهامی وجود داشت فرضی انجام دادید به صورت کامنت در کد وارد کنید، در صورتی که مشکل خاصی وجود داشت می‌توانید با ما در ارتباط باشید.
- elahekhodavrdi@gmail.com
 - attarshahriar@gmail.com