فرض کنید در زبانی مشابه SQL در جلو کلمه select لیستی از عبارات ریاضی و اسامی مستعار در قالب ناپایانه ItemList قرار گیرد. گرامر مربوط به این ناپایانه به صورت زیر است:

ItemList → Item | ItemList "**,**" Item
Item → Exp Alias
Alias → "***as***" Identifier | ε
Exp → Field | ***num***
Field → Identifier Tail
Tail → "**.**" Identifier | ε
Identifier → ***id***

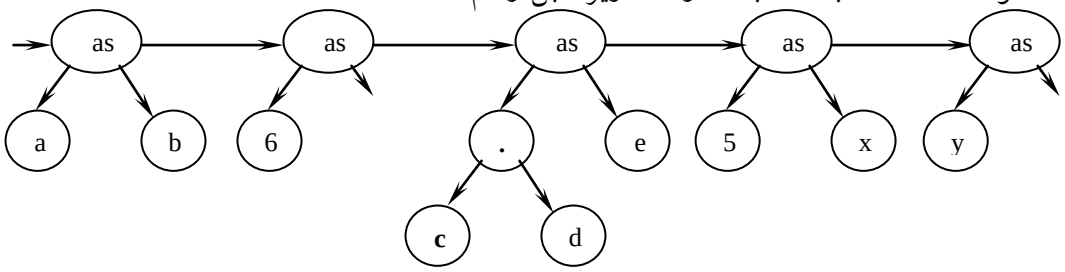
نمونه ای از جملات پذیرفته شده توسط گرامر فوق به صورت زیر است:

a as b , 6 , c.d as e , 5 as x , y

الف) یک گراف وضعیت منسجم و بدون ε معادل با گرامر فوق با حداقل گره بیان کنید.

ب) این گرامر را به فرم BNF منسجم در یک خط بیان کنید.

ج) می‌خواهیم در داخل گرامر فوق، حداکثر در ۶ محل گرامر کنش‌های مفهومی را طوری قرار دهیم که با اجرای آنها یک درخت معنا متناظر با تعریف های فوق ایجاد شود. مثلاً معادل با مثال بالا، درخت زیر قابل رسم است:



ساختار گره‌های درخت معنا (به زبان C++) به صورت زیر است:

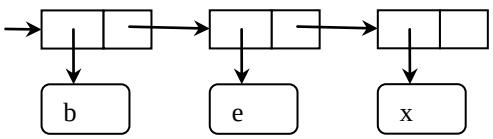
```
enum NodeType {_NUM, _ID, _POINT, _AS};
struct Node {
        NodeType  type;
        union {
                float  NumValue;
                char*  IdValue;
                struct {Node *Left, *Right;} s_point;  /* for _POINT*/
                struct {Node *Left, *Right, *Next;} s_as;  /* for _AS*/
        } u;
};
typedef  Node*  TREE;
```

برای تشکیل درخت معنا از یک پشته مفهومی با نام TreeStk استفاده می‌شود که فقط دو عمل push و pop دارد و برای شروع خالی است و پس از اعمال کلیه کنش ها، حاصل درخت معنا در داخل پشته قرار می‌گیرد. کنش های لازم را درج و پیاده سازی کنید.

د) می‌خواهیم درخت معنای ساخته شده را پردازش کنیم و لیستی از اسامی مستعار (id های نوشته شده جلو as ها) را به صورت یک لیست پیوندی با ساختار زیر بسازیم:

```
struct  ListNode  { char*  IdValue;   ListNode * Next; };
typedef  ListNode* LIST;
```

مثلاً لیست ساخته شده برای مثال بالا، شکل زیر را به خود خواهد گرفت:



برای این منظور تابعی بازگشتی و فاقد حلقه با نام Convert بنویسید که یک درخت معنا مربوط به ItemList را گرفته و لیست اسامی مستعار را تحویل دهد: LIST Convert(TREE t)