

Realizácia zotavenia (1)

1. Každú funkciu reprezentujúcu pravidlo EBNF je potrebné doplniť o parameter **keys** reprezentujúci množinu kľúčov (typu **KeySet**)

```
int Expr(KeySet keys) { ... }
```

2. Volanie **getsymbol()** nahradiť:

```
if (symbol == a) {  
    getsymbol();  
} else {  
    error(e, keys);  
}
```

Realizácia zotavenia (2)

3. Pri volaní procedúr zodpovedajúcich neterminálnym symbolom doplniť argument s aktuálnou hodnotou množiny kľúčov.

```
leftOp = Mul(allKeys);
```

4. Do každého bodu rozhodnutia (if+else, switch, while) umiestniť volanie check v tvare:

```
check(e, keys);
```

Zotavenie pri postupnosti neterminálov

- $A \rightarrow S_1 S_2 S_3 \dots S_i \dots S_n$

```
A(KeySet keys) {  
    S1 (H(S2) U H(S3) U ... U H(Sn) U keys);  
    S2 (H(S3) U ... U H(Sn) U keys);  
    ...  
    Si (H(Si+1) U ... U H(Sn) U keys);  
    ...  
    Sn (keys);  
}
```

Zotavenie pri vetvení

- $A \rightarrow (S_1 \mid S_2 \mid \dots \mid S_n)$

```
A(KeySet keys) {  
    KeySet allkeys = H(S1) U ... U H(Sn) U keys;  
    check(e, allkeys);  
    switch (symbol) {  
        case H(S1): S1(keys); break;  
        ...  
        case H(Sn): Sn(keys); break;  
        default: error(e, keys);  
    }  
}
```

Zotavenie pri cykle

- $A \rightarrow \{S_1 S_2 \dots S_n\}$

```
A(keys) {
```

```
    loopkeys = H(S1) U ... U H(Sn);
```

```
    allkeys = loopkeys U keys;
```

```
    check(e, allkeys);
```

```
    while (symbol ∈ loopkeys)
```

```
        S1(allkeys);
```

```
        S2(allkeys);
```

```
        ...
```

```
        Sn(allkeys);
```

```
    check(e, allkeys);
```

```
}
```

```
}
```