

# NPFL075 Practical Class 03



**Jiří Mírovský**

Charles University

Institute of Formal and Applied Linguistics



# Perl Tips



```
my $tag=$this->attr('m/tag') // '';
```

# operator // meaning: or, if it is undefined, ...

# (by the way, operator || means: or, if it is zero, ...

```
my @children = grep {$_->attr('afun') =~ /^(Sb|Obj)$/} $verb->children
```

```
my @texts = map {$_ eq $this ? 'V' : $_->attr('afun')} @sorted_children;
```

# Perl Tips



```
my $tag=$this->attr('m/tag') // '';
```

# operator // meaning: or, if it is undefined, ...

# (by the way, operator || means: or, if it is zero, ...)

```
my @children = grep {$_->attr('afun') =~ /^(Sb|Obj)$/} $verb->children
```

```
my @texts = map {$_ eq $this ? 'V' : $_->attr('afun')} @sorted_children;
```

# Perl Tips



```
my $tag=$this->attr('m/tag') // '';
```

```
# operator // meaning: or, if it is undefined, ...
```

```
# (by the way, operator || means: or, if it is zero, ...)
```

```
my @children = grep {$_->attr('afun') =~ /^(Sb|Obj)$/} $verb->children
```

```
my @texts = map {$_ eq $this ? 'V' : $_->attr('afun')} @sorted_children;
```

# Perl Tips

```
my $text = join(' ', @elements)
```

```
my @elements = split(/ /, $text)  
my @elements = split(/\t/, $text)
```

# Perl Tips

```
my $text = join(' ', @elements)
```

```
my @elements = split(/ /, $text)
```

```
my @elements = split(/\t/, $text)
```

# Perl Tips



```
my $text = join(' ', @elements)
```

```
my @elements = split(/ /, $text)
```

```
my @elements = split(/\t/, $text)
```

# BTred Tips



```
my @texts = map {$_ eq $this ? 'V' : $_->attr('afun')} @sorted_children;
```

What about setting this first:

```
$this->set_attr('afun','V');
```

OK, as:

- nothing gets saved automatically
- you need to call `ChangingFile(1)` to make btred save changes
- you can even set non-existing attributes: `$this->set_attr('xyz','123')`, they will never be saved

# BTred Tips

```
my @texts = map {$_ eq $this ? 'V' : $_->attr('afun')} @sorted_children;
```

What about setting this first:

```
$this->set_attr('afun','V');
```

OK, as:

- nothing gets saved automatically
- you need to call `ChangingFile(1)` to make btred save changes
- you can even set non-existing attributes: `$this->set_attr('xyz','123')`, they will never be saved

# BTred Tips

```
my @texts = map {$_ eq $this ? 'V' : $_->attr('afun')} @sorted_children;
```

What about setting this first:

```
$this->set_attr('afun','V');
```

OK, as:

- nothing gets saved automatically
- you need to call `ChangingFile(1)` to make btred save changes
- you can even set non-existing attributes: `$this->set_attr('xyz','123')`, they will never be saved

# BTred Tips: Exit Hook



```
my %type_num;
...
if (exists $type_num{$type}) {
    $type_num{$type} += 1;
} else {
    $type_num{$type} = 1;
}
```

But:

Perl initializes hash elements to zero by default, so this is sufficient:

```
$type_num{$type}++;
```

```
sub exit_hook {
    foreach my $type (keys %type_num) {...}
}
```



# BTred Tips: Exit Hook

```
my %type_num;
...
if (exists $type_num{$type}) {
    $type_num{$type} += 1;
} else {
    $type_num{$type} = 1;
}
```

But:

Perl initializes hash elements to zero by default, so this is sufficient:

```
$type_num{$type}++;
```

```
sub exit_hook {
    foreach my $type (keys %type_num) {...}
}
```

# BTred Tips: Exit Hook

```
my %type_num;
...
if (exists $type_num{$type}) {
    $type_num{$type} += 1;
} else {
    $type_num{$type} = 1;
}
```

But:

Perl initializes hash elements to zero by default, so this is sufficient:

```
$type_num{$type}++;
```

```
sub exit_hook {
    foreach my $type (keys %type_num) {...}
}
```

# (B)Tred Tips: Filelists

my \$address = ThisAddress(\$node?)

vs.

FPosition(\$node?) # directly to stdout

Filelists – two options:

- file##tree\_number.node\_dfo
- file#node\_id # only if nodes have ids (not in UD)

If the filelist file name ends with .fl, the first line is interpreted as the filelist name.

# (B)Tred Tips: Filelists

my \$address = ThisAddress(\$node?)

vs.

FPosition(\$node?) # directly to stdout

Filelists – two options:

- file##tree\_number.node\_dfo
- file#node\_id # only if nodes have ids (not in UD)

If the filelist file name ends with .fl, the first line is interpreted as the filelist name.

# (B)Tred Tips: Filelists

my \$address = ThisAddress(\$node?)

vs.

FPosition(\$node?) # directly to stdout

Filelists – two options:

- file##tree\_number.node\_dfo
- file#node\_id # only if nodes have ids (not in UD)

If the filelist file name ends with .fl, the first line is interpreted as the filelist name.

# (B)Tred Tips: Filelists



## In Tred:

`tred -l filelist.fl`

or, via menu

## In BTred:

`btred -l filelist.fl` (the entries can only be file names, not incl. nodes)

# (B)Tred Tips: Filelists



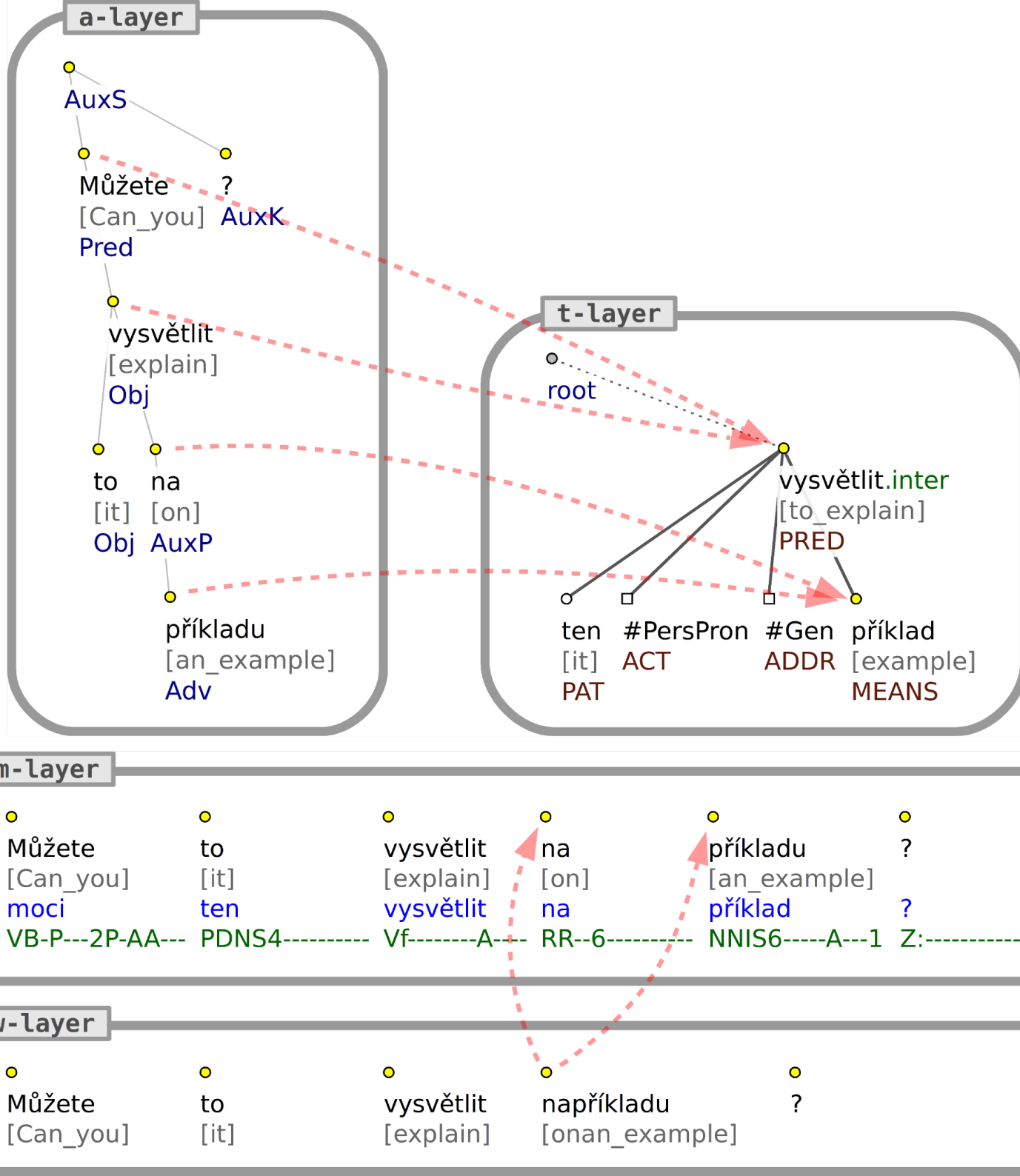
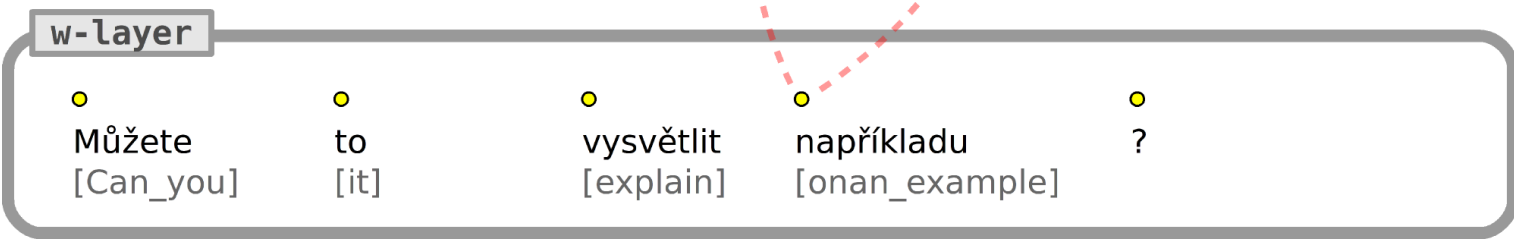
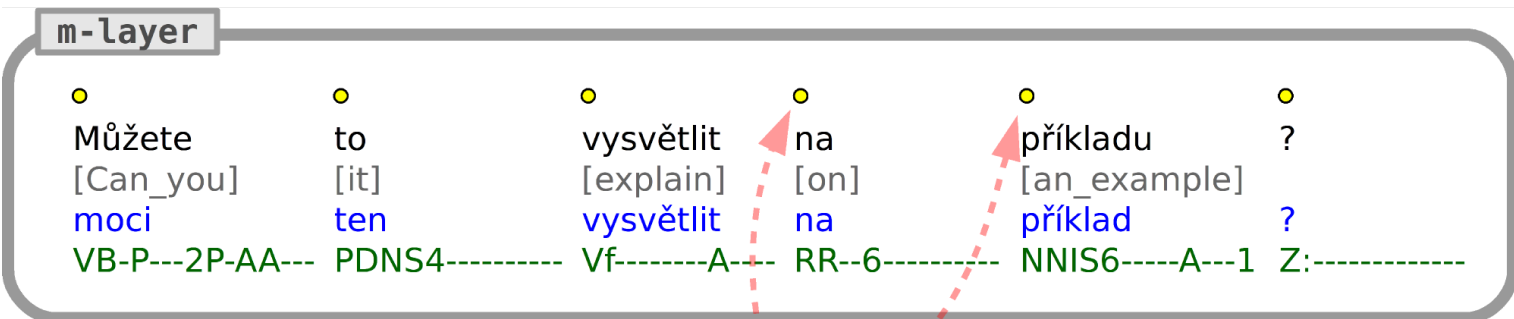
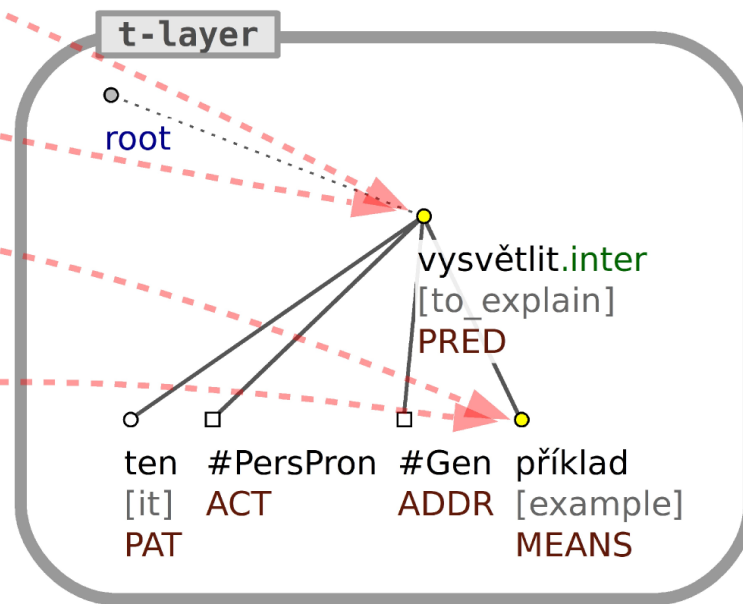
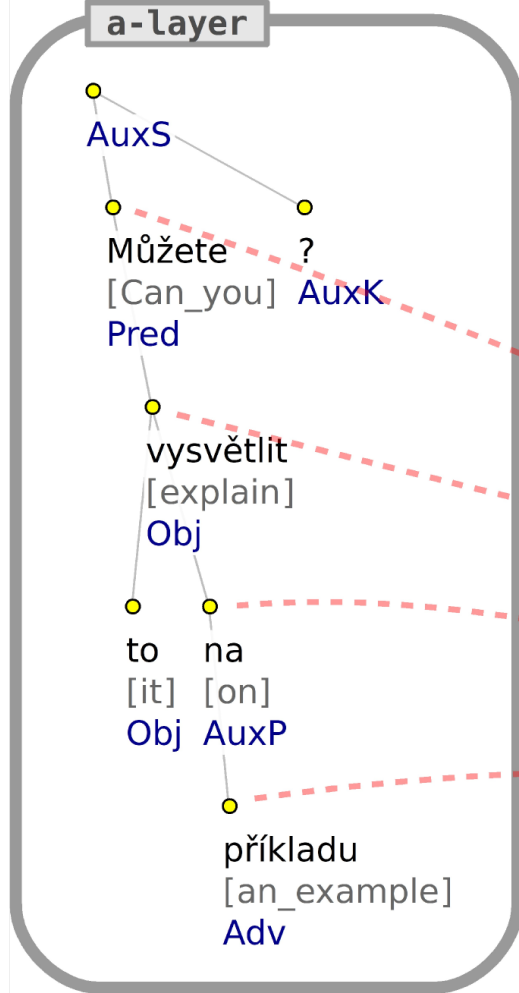
## In Tred:

`tred -l filelist.fl`

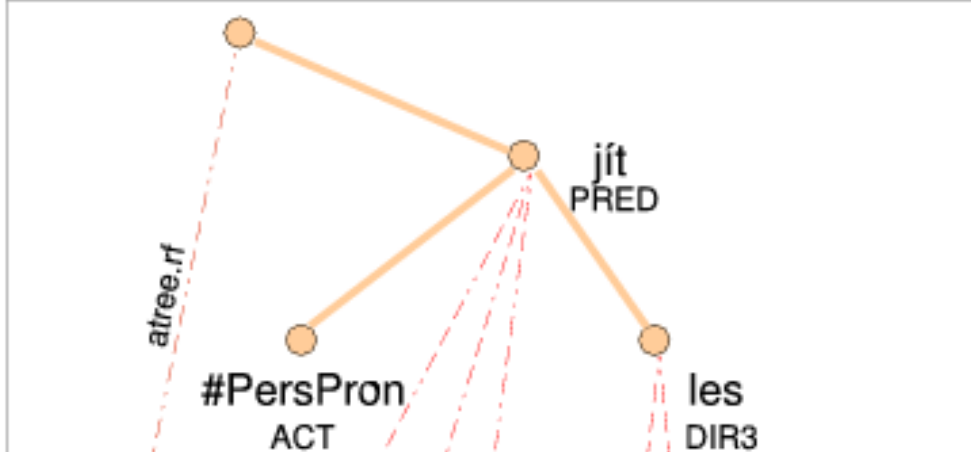
or, via menu

## In BTred:

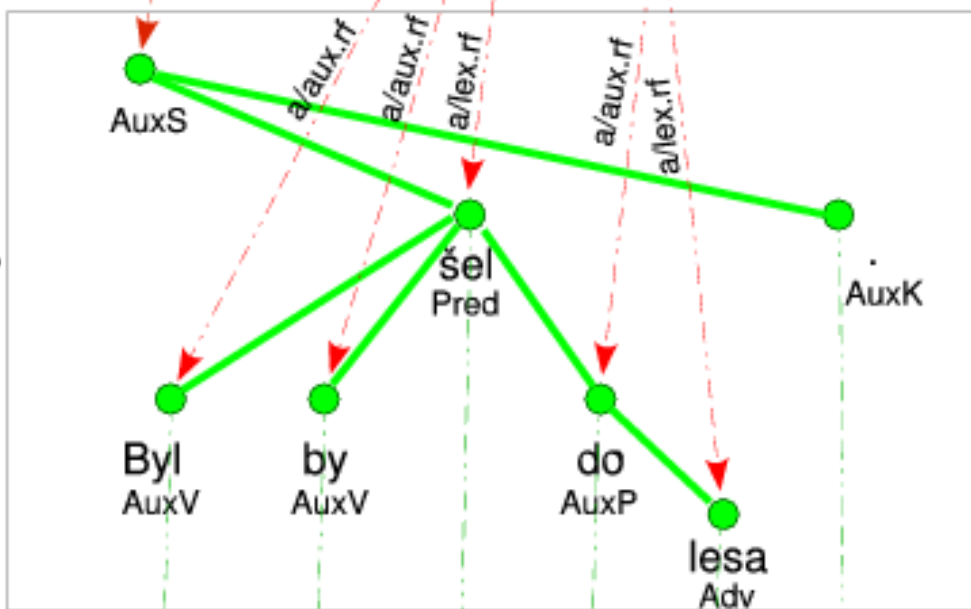
`btred -l filelist.fl` (the entries can only be file names, not incl. nodes)



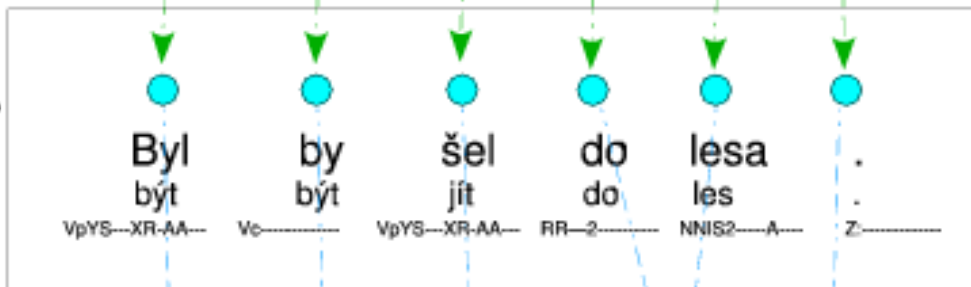
t - layer



a - layer



m - layer



w - layer



# BTred Tips: From T- to A-layer

```
my $lexref = $tnode->attr('a/lex.rf');  
my @auxrefs = ListV($tnode->attr('a/aur.rf');  
# and then PML_T::GetANodeByID(...);  
  
my @arefs = PML_T::GetANodeIDs($tnode);  
# and then PML_T::GetANodeByID(...);  
  
my $lex = PML_T::GetALexNode(...);  
my @auxs = PML_T::GetAAuxNodes(...);  
my @anodes = PML_T::GetANodes($tnode);  
# also works with $root
```

# BTred Tips: From T- to A-layer

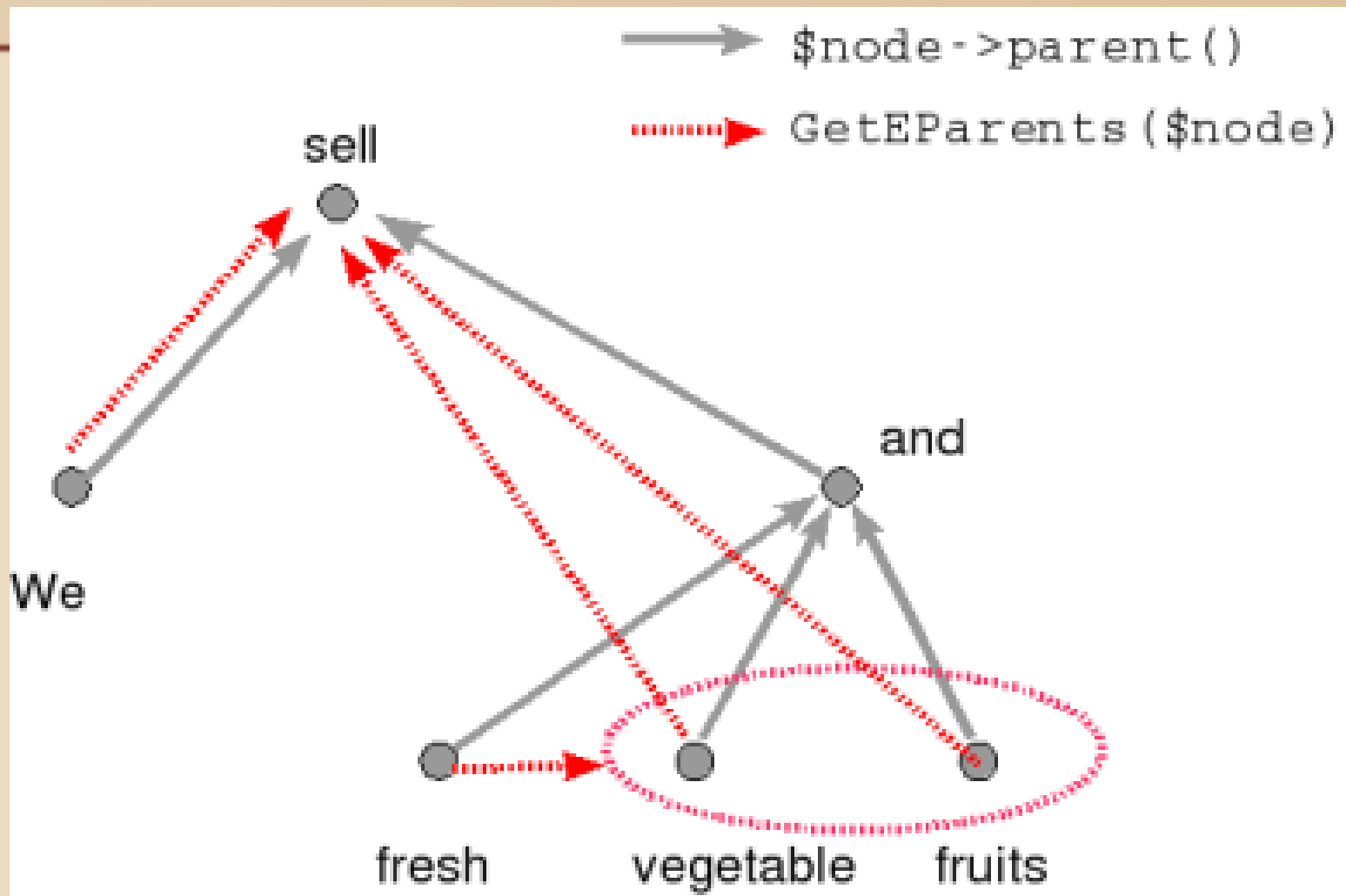
```
my $lexref = $tnode->attr('a/lex.rf');  
my @auxrefs = ListV($tnode->attr('a/aur.rf');  
# and then PML_T::GetANodeByID(...);  
  
my @arefs = PML_T::GetANodeIDs($tnode);  
# and then PML_T::GetANodeByID(...);  
  
my $lex = PML_T::GetALexNode(...);  
my @auxs = PML_T::GetAAuxNodes(...);  
my @anodes = PML_T::GetANodes($tnode);  
# also works with $root
```

# BTred Tips: From T- to A-layer

```
my $lexref = $tnode->attr('a/lex.rf');  
my @auxrefs = ListV($tnode->attr('a/aur.rf');  
  # and then PML_T::GetANodeByID(...);  
  
my @arefs = PML_T::GetANodeIDs($tnode);  
  # and then PML_T::GetANodeByID(...);  
  
my $lex = PML_T::GetALexNode(...);  
my @auxs = PML_T::GetAAuxNodes(...);  
my @anodes = PML_T::GetANodes($tnode);  
  # also works with $root
```

# BTred Tips: From T- to A-layer

```
my $lexref = $tnode->attr('a/lex.rf');  
my @auxrefs = ListV($tnode->attr('a/aur.rf');  
# and then PML_T::GetANodeByID(...);  
  
my @arefs = PML_T::GetANodeIDs($tnode);  
# and then PML_T::GetANodeByID(...);  
  
my $lex = PML_T::GetALexNode(...);  
my @auxs = PML_T::GetAAuxNodes(...);  
my @anodes = PML_T::GetANodes($tnode);  
# also works with $root
```



# BTred Tips: Effective Parentage



PML\_A::GetEChildren(\$a\_node)

PML\_A::GetEParents(\$a\_node)

PML\_T::GetEChildren(\$t\_node)

PML\_T::GetEParents(\$t\_node)

