

NPFL075 Practical Class 04

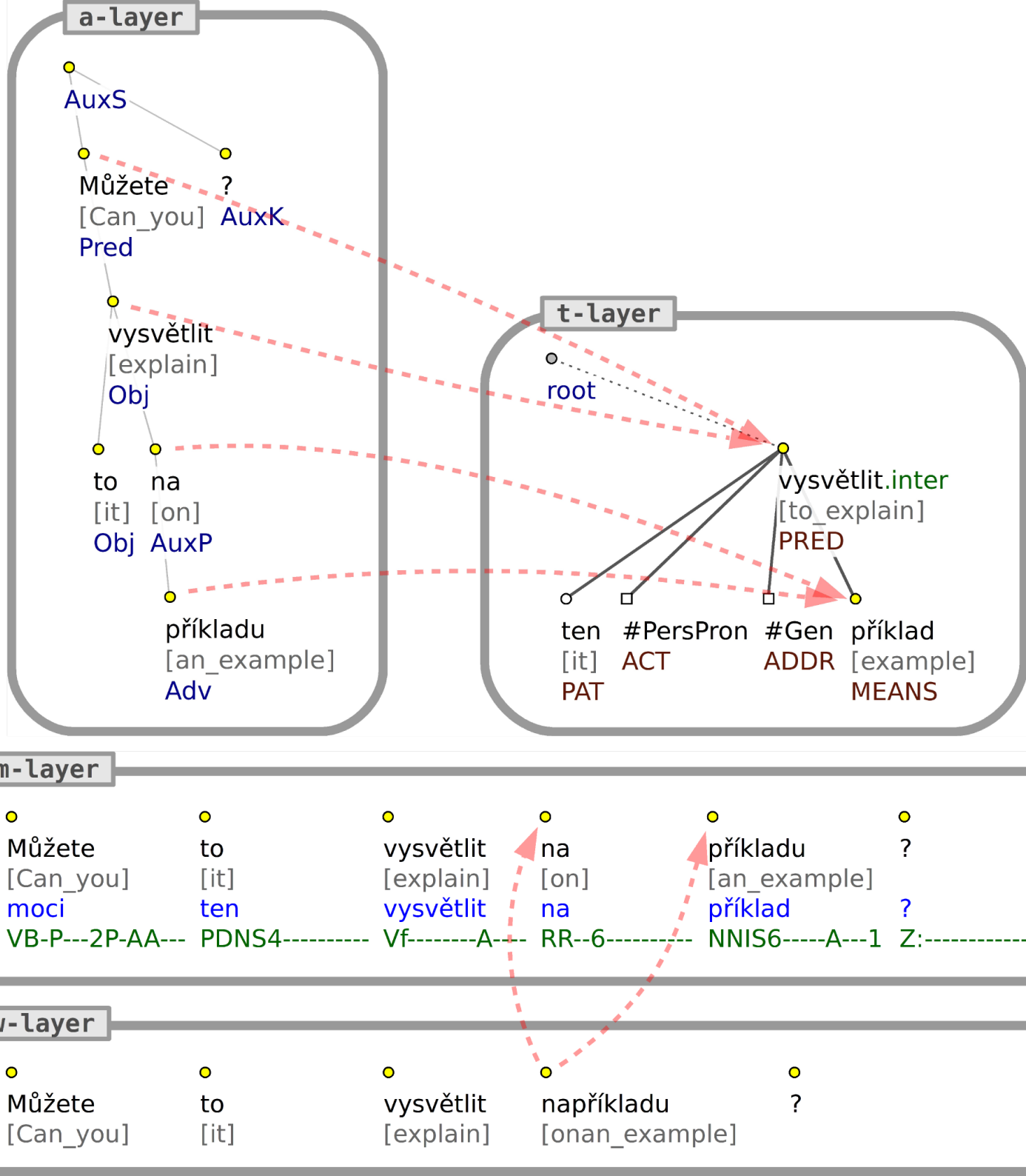
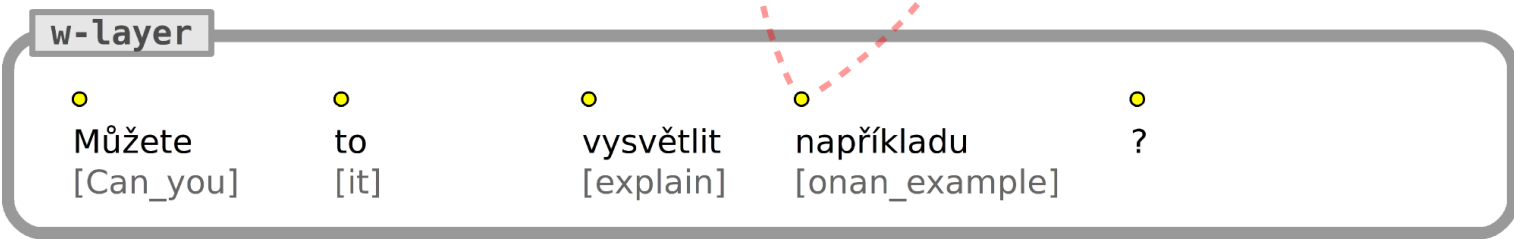
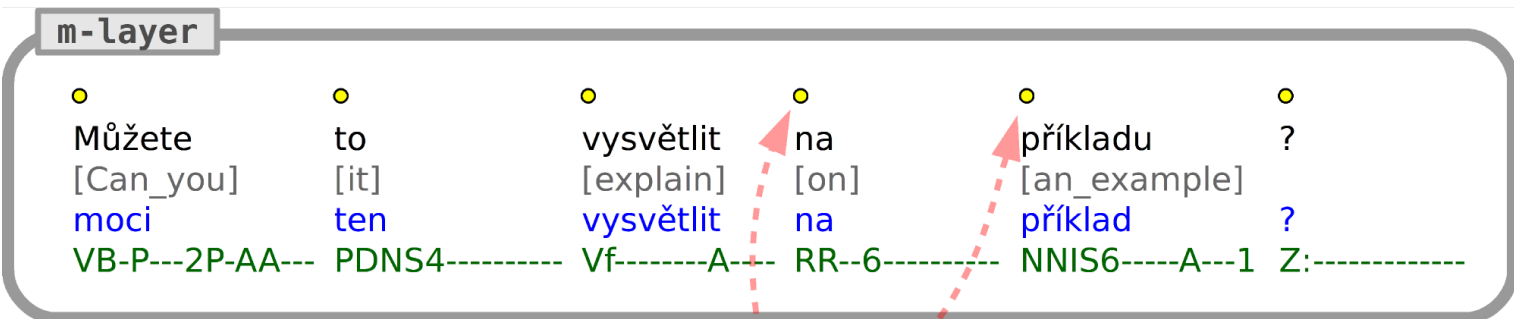
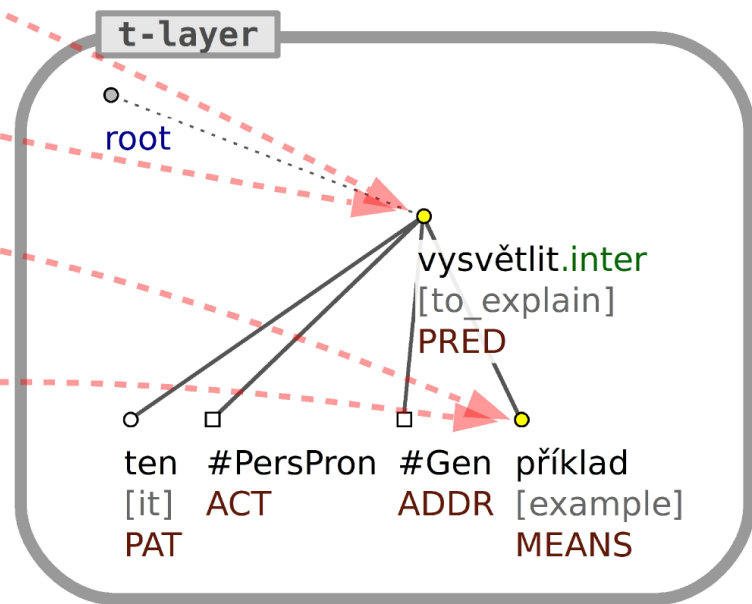
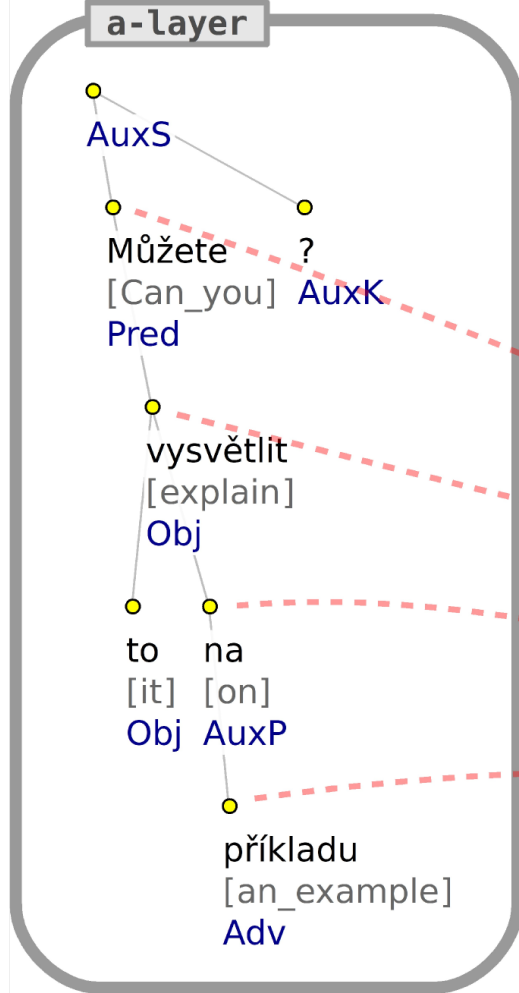


Jiří Mírovský

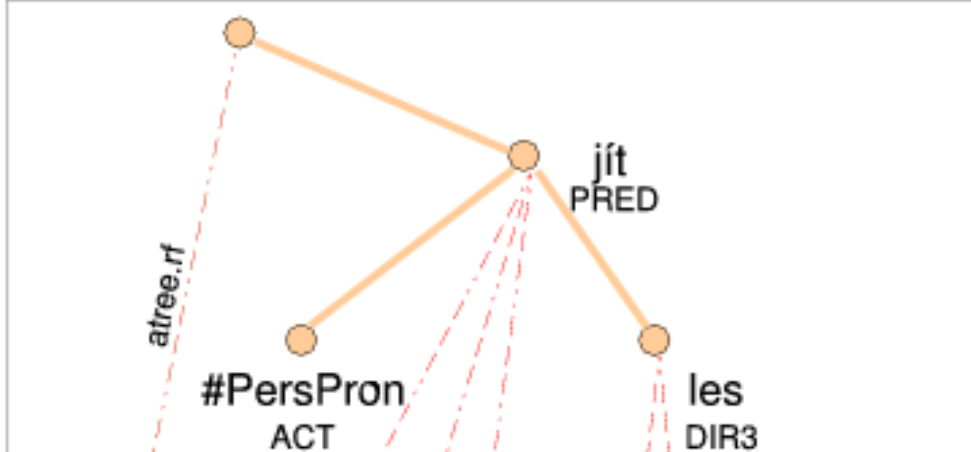
Charles University

Institute of Formal and Applied Linguistics

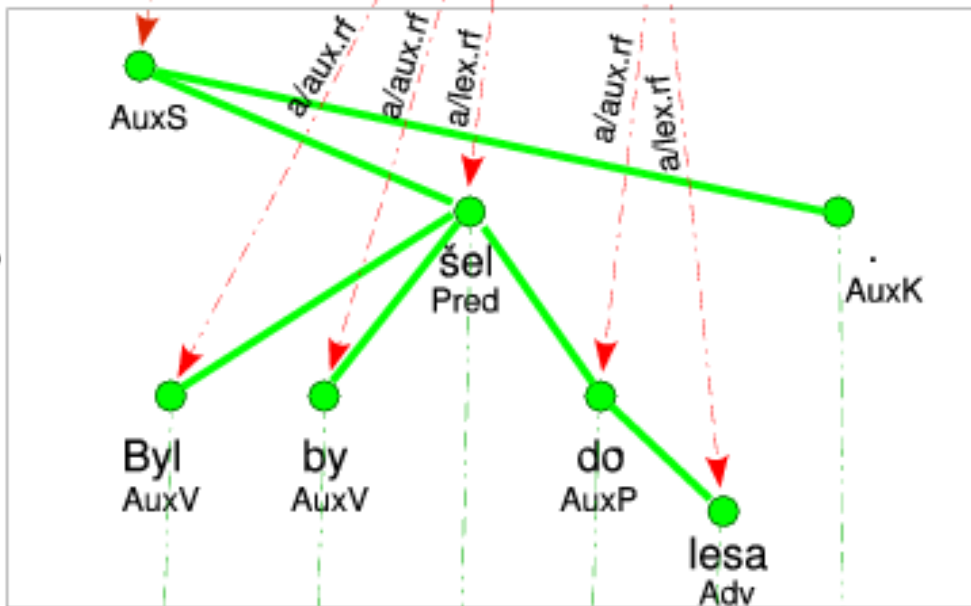




t - layer



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/aux.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

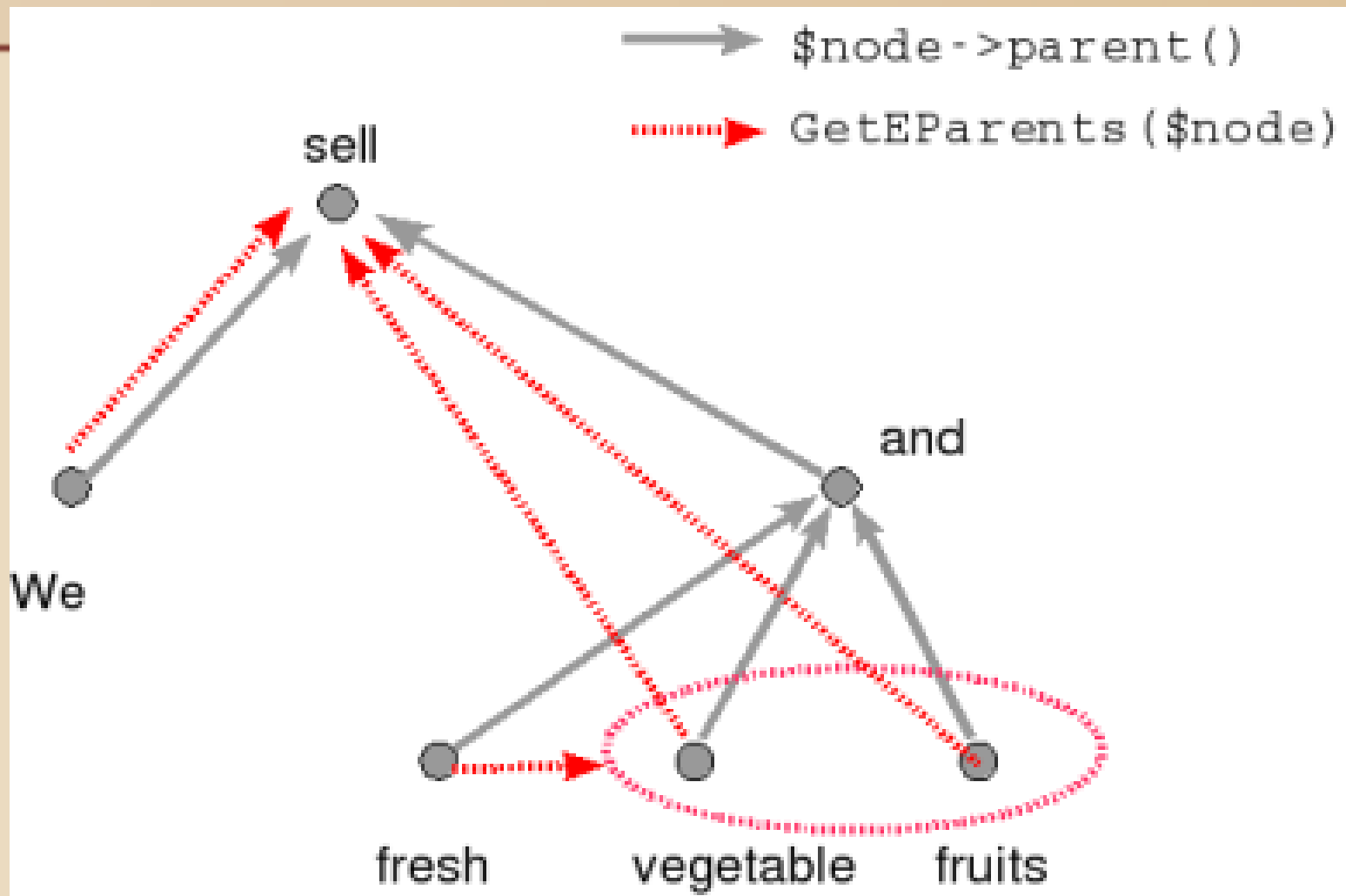
```
my $lexref = $tnode->attr('a/lex.rf');  
my @auxrefs = ListV($tnode->attr('a/aux.rf');  
  # and then PML_T::GetANodeByID(...);  
  
my @arefs = PML_T::GetANodeIDs($tnode);  
  # and then PML_T::GetANodeByID(...);  
  
my $lex = PML_T::GetALexNode(...);  
my @auxs = PML_T::GetAAuxNodes(...);  
my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



BTred and Universal Dependencies

ListV - If the value of a node attribute is a list (i.e., a Treex::PML::List object), return all its values as an array

In UD, e.g., `my @feats = ListV($this->attr('feats'));`

and then: `foreach my $feat (@feats) { # $feat is a pointer to a hash
 my $name = $feat->{'name'};
 my $value = $feat->{'value'};
}`

Attribute **misc** does not have such a strict format, therefore:

```
my @miscs = ListV($this->attr('miscs'));  
foreach my $misc (@miscs) {  
    if ($misc eq 'SpaceAfter=No') {  
        ...  
    }  
}
```

