

sequentially wait-for all

<child> ; <child>

sequentially wait-for one wait-for-optional-  
subplans=yes

```
var child1-state = inactive, ..., childn-state
= inactive in
begin
  child1#(; ..., child1-state);
  ...
  childn#(; ..., childn-state);
  if child1-state = completed or ... or
childn-state = completed
  then parent-state := completed
  else parent-state := aborted
end

no break - all children are done
```

any-order wait-for all

<child1> ||a <child2>

Don't care for child state - parent waits.

any-order wait-for one

```
var child1-state = inactive, child2-state =
inactive in
begin
  break
  (: plan body :)
    child1#(; ..., child1-state)
  ||a child2#(; ..., child2-state)
  if (: continuation condition: wait for one
  :)
    child1-state = completed
    or child2-state = completed;
  (: Derive parent state :)
  if child1-state = completed or child2-state
= competed
  then parent-state := completed
  else parent-state := aborted
end
```

unordered wait-for none

```
<child> ||s <child> ||s skip;
```

```
parent-state := completed
not correct !!!
```

unordered retry-aborted-subplans

```
while child1-state unequal completed
  and child2-state unequal completed
do
  if    child1-state unequal completed
  then child1#(;time, child1-state)
||s if    child2-state unequal completed
  then child2#(;time, child2-state)
```

*or - better :*

```
while child1-state unequal completed
do child1#(;time, child1-state)
||s while child2-state unequal completed
do child2#(;time, child2-state)
```