

**UZUPEŁNIENIA W DRUGIEJ KLAUZULI PROCEDURY plan
W CELU ZAIMPLEMENTOWANIA ITERACYJNIE ZWIĘKSZANEGO
OGRANICZENIA DŁUGOŚCI PLANU**

plan (InitState, Goals, **Limit**, Plan, FinalState) : -

```
; sprawdź limit
; generuj niedeterministycznie limit dla preplanu, LimitPre,
; jako liczbę z zakresu 0 - Limit

choose_goal (Goal, Goals, RestGoals, InitState),
achieves ( Goal, Action),
requires (Action, CondGoals(, Conditions),
plan (InitState, CondGoals, LimitPre , PrePlan, State1),
inst_action(Action, Conditions,State1, InstAction),
perform_action (State1, InstAction, State2),

; ustaw limit dla postplanu, LimitPost, jako Limit - LimitPre - 1

plan (State2, RestGoals, LimitPost , PostPlan, FinalState),
conc (PrePlan, [ InstAction | PostPlan ], Plan) .
```

**DODAJEMY UZUPEŁNIENIA W CELU ZAIMPLEMENTOWANIA
OCHRONY OSIĄGNIĘTYCH CELÓW**

plan (InitState, Goals, **AchievedGoals**, Limit, Plan, FinalState) : -

```
; sprawdź limit
; generuj niedeterministycznie limit dla preplanu, LimitPre,
; jako liczbę z zakresu 0 - Limit

choose_goal (Goal, Goals, RestGoals, InitState),
achieves ( Goal, Action),
requires (Action, CondGoals(, Conditions),
plan (InitState, CondGoals, LimitPre , PrePlan, State1),
inst_action(Action, Conditions,State1, InstAction),

; wywołaj procedurę sprawdzającą, czy zaplanowana akcja
; nie niszczy żadnego z osiągniętych wcześniej celów :

check_action(InstAction, AchievedGoals),

perform_action (State1, InstAction, State2),

; ustaw limit dla postplanu, LimitPost, jako Limit - LimitPre - 1

; dodaj do listy AchievedGoals ostatnio osiągnięty cel Goal,
; podaj do wywołania plan jako AchievedGoals1

plan (State2, RestGoals, AchievedGoals1, LimitPost , PostPlan, FinalState),
conc (PrePlan, [ InstAction | PostPlan ], Plan) .
```