

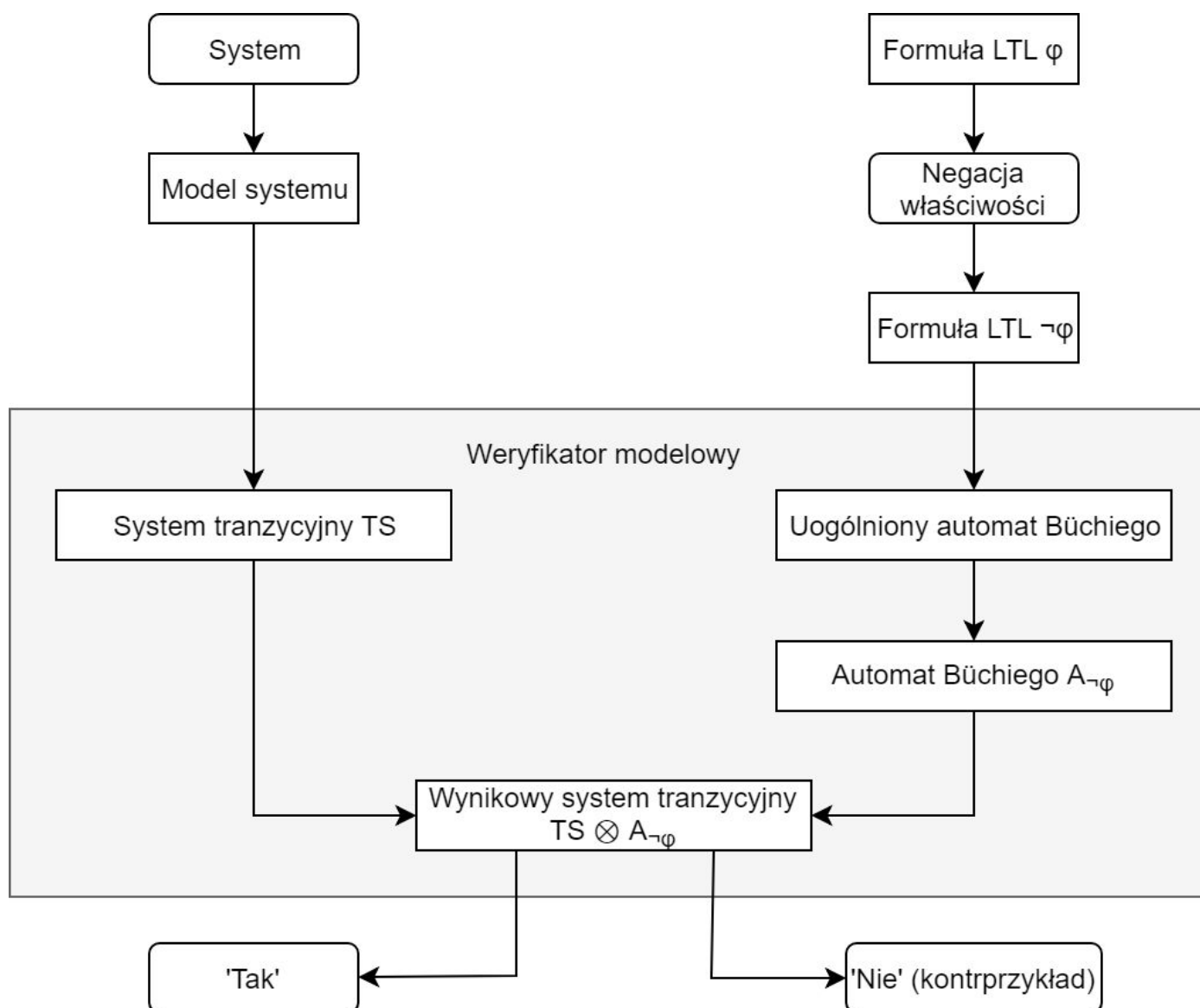
# Implementacja algorytmu weryfikacji modelowej własności LTL w środowisku rozproszonym

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE  
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY



Wojciech Kumoń

# Weryfikacja modelowa LTL



# Zastosowany algorytm

**on-the-fly OWCTY = OWCTY + heurystyka MAP**

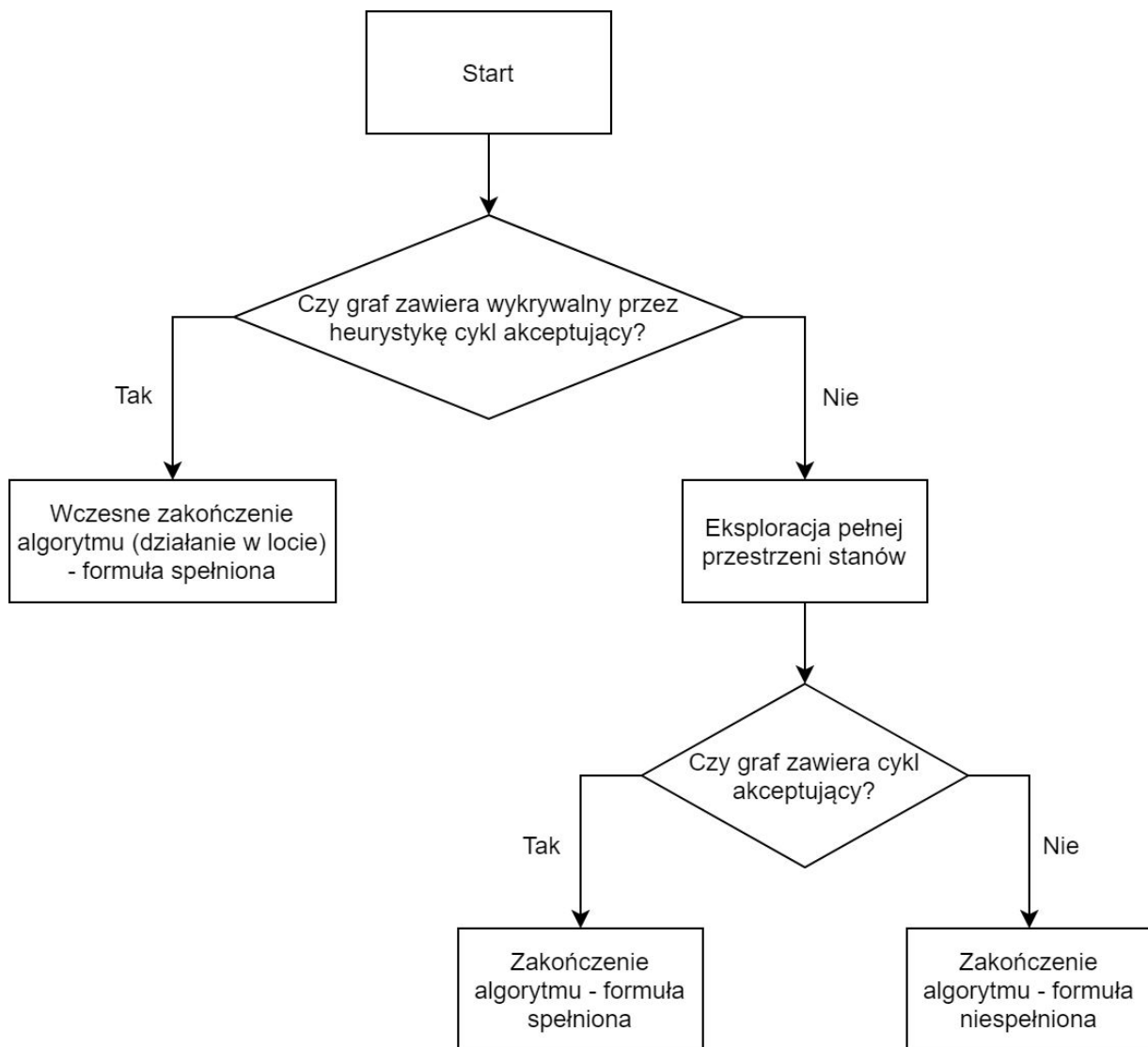
OWCTY: BFS + brak działania w locie

MAP: DFS + działanie w locie poziom 2

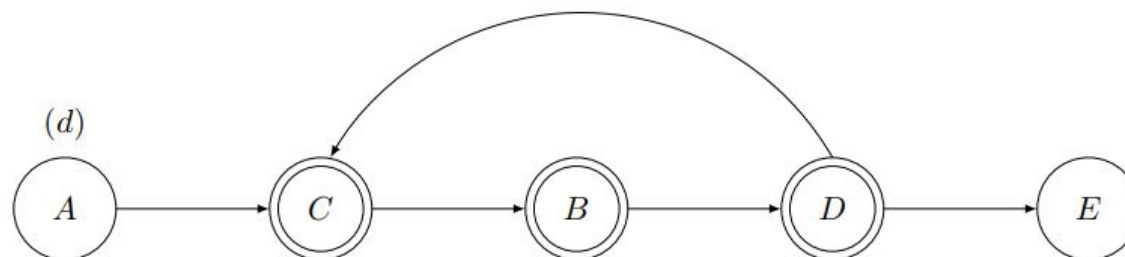
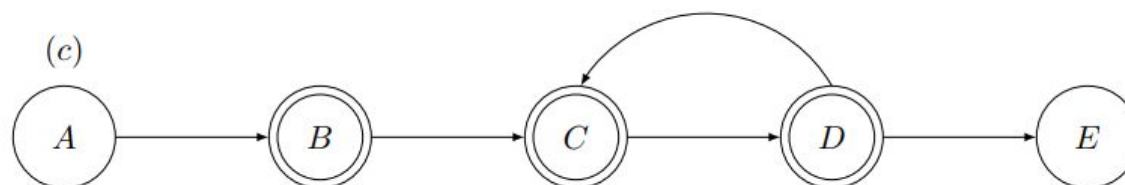
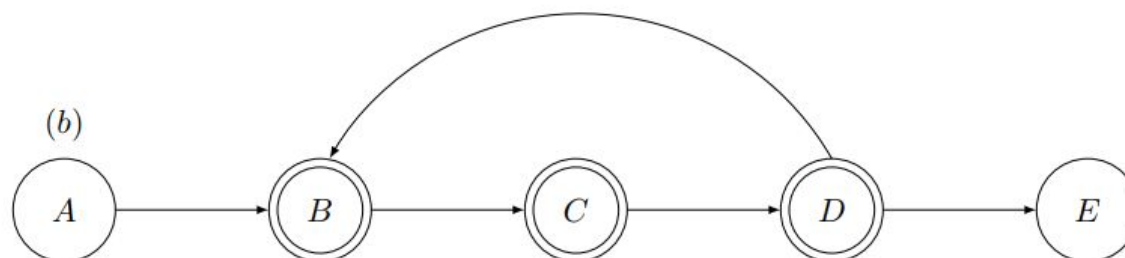
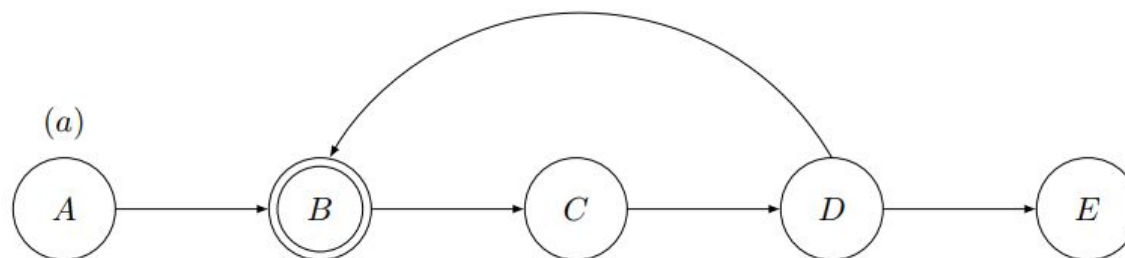
on-the-fly OWCTY: BFS + działanie w locie poziom 1

```
function detectAcceptingCycle() {  
    let initialStates = getInitialStates()  
    let result = tryOnTheFlyUsingMapHeuristic(initialStates)  
    if (result.acceptingCycleFound()) {  
        return result  
    } else {  
        return continueOwctyUsingAlreadyGeneratedStates()  
    }  
}
```

# Diagram wczesnego zakończenia algorytmu

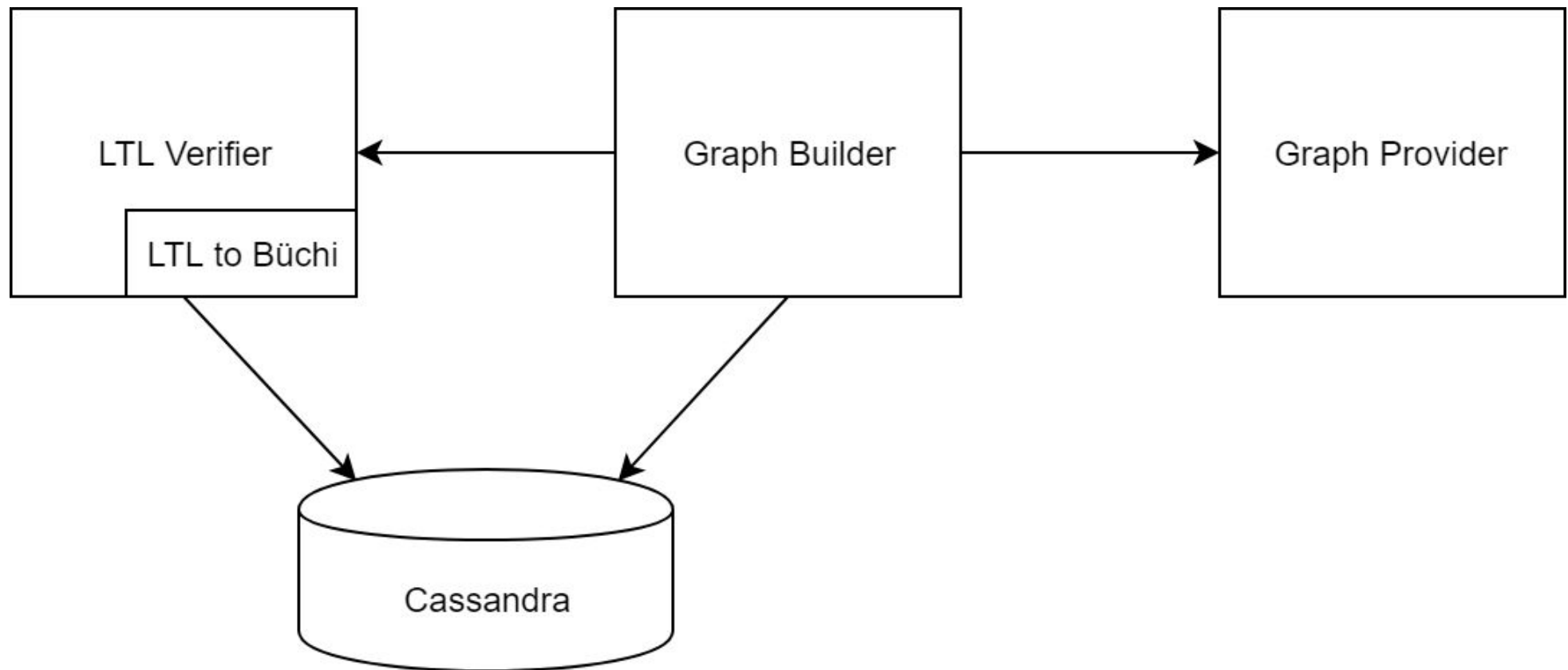


# Działanie heurystyki algorytmu

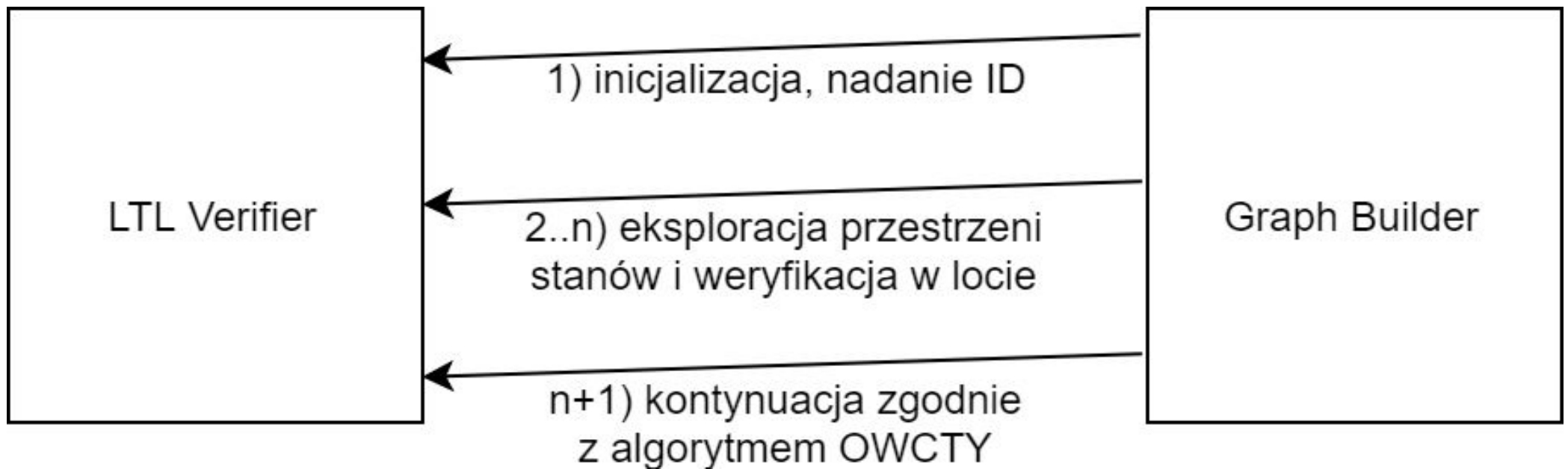


$A > B > C > D > E$

# Schemat ogólny systemu



# Typy zapytań



# Przykładowa formuła LTL w języku Kotlin

```
val bfsPcEq1 = proposition(hasProperty<SystemState>("bfsHeuristicState",  
    hasProperty<Int>("pc", equalTo(1))))  
val bfsPcGt1 = proposition(hasProperty<SystemState>("bfsHeuristicState",  
    hasProperty<Int>("pc", greaterThan(1))))  
val formula = globally(bfsPcEq1 implies finally(bfsPcGt1))
```