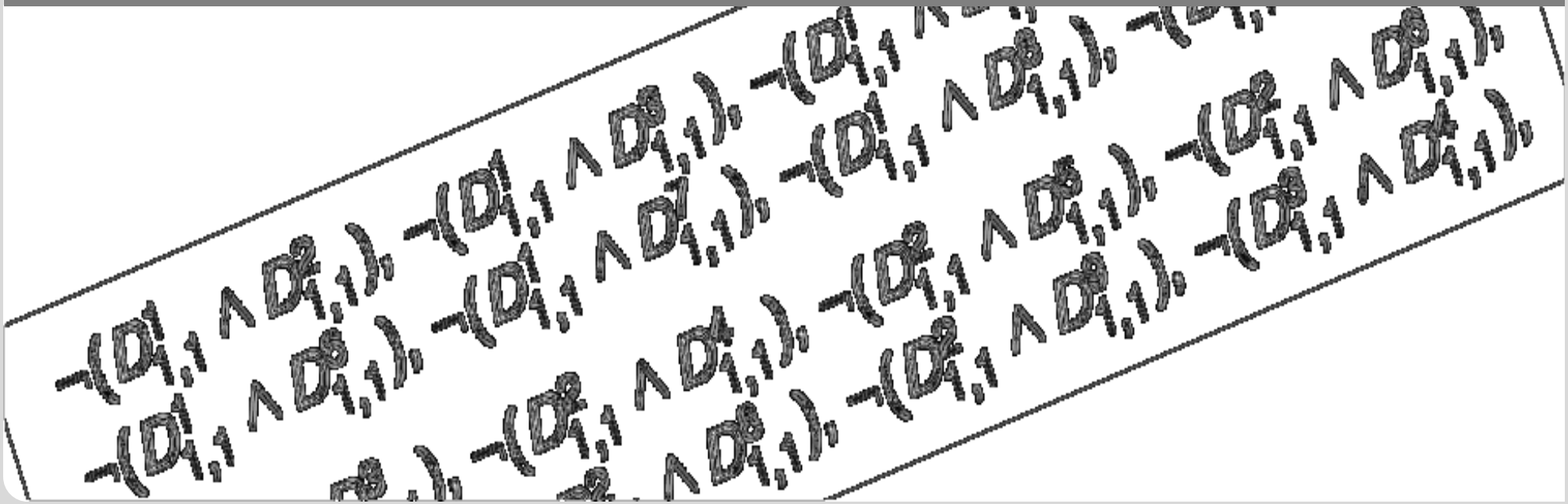
Karlsruhe Institute of Technology

Formale Systeme

Die Sprache PROMELA

Prof. Dr. Bernhard Beckert | WS 2009/2010

KIT – Institut für Theoretische Informatik



Beschreibung endlicher Automaten

Die Darstellung konkreter endlicher Automaten in graphischer Form ist nur für kleine Automaten möglich, für größere, wie sie in realistischen Anwendungen auftreten, ist das nicht praktikabel.

Wir betrachten als eine Alternative die Modellierungssprache *Promela*.

Promela

- Process meta language
- Modellierungssprache für indeterministische gekoppelte erweiterte endliche Automaten.
- Entwickelt von Gerard Holzmann seit 1980.
- Weit verbreitetes Verifikationswerkzeug SPIN (Simple Promela INterpreter)
- Angabe der zu verifizierenden Eigenschaft als LTL Formel oder Büchi-Automat.

Promela Beispiel

```
/* Peterson: mutual exclusion [1981] */
bool turn, flag[2];
byte ncrit;
active [2] proctype user()
{       assert(_pid == 0 || _pid == 1);
 again: flag[_pid] = 1;
        turn = _pid;
        (flag[1 - _pid] == 0 || turn == 1 - _pid);
        ncrit++;
        assert(ncrit == 1); /* critical section */
        ncrit--;
        flag[_pid] = 0;
        goto again}
```

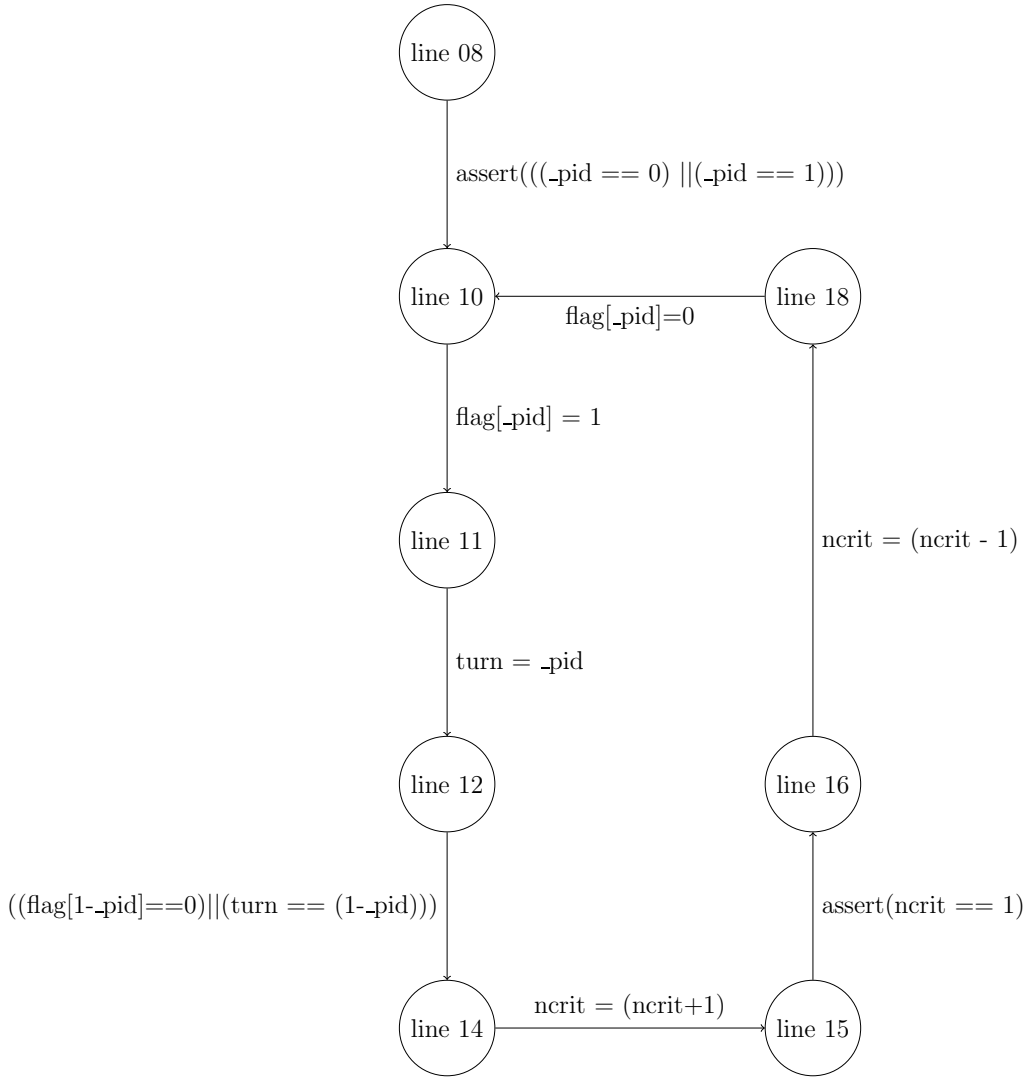
Promela Beispiel

```
bool turn, flag[2];
byte ncrit;
active [2] proctype user()
{       assert(_pid == 0 || _pid == 1);
again:  flag[_pid] = 1;
        turn = _pid;
        (flag[1 - _pid] == 0 || turn == 1 - _pid);
        ncrit++;
        assert(ncrit == 1); /* critical section */
        ncrit--;
         flag[_pid] = 0;
        goto again   }
```

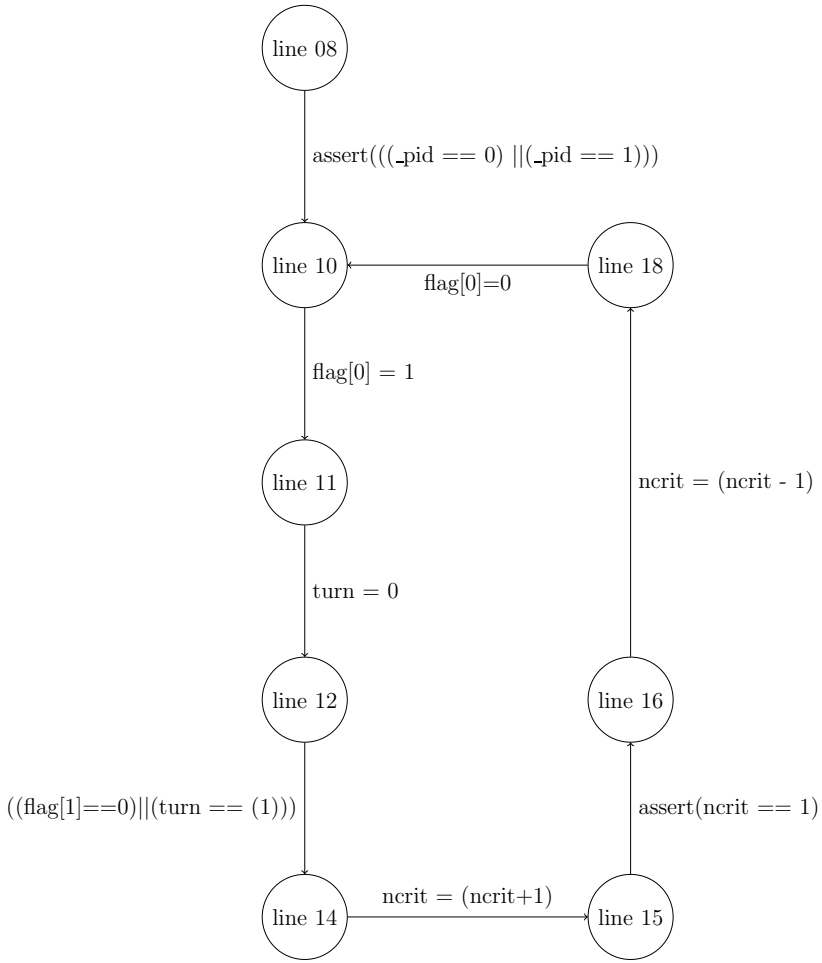
Promela beschreibt Zustandsübergangssysteme durch die Definition von Prozesstypen (process types).

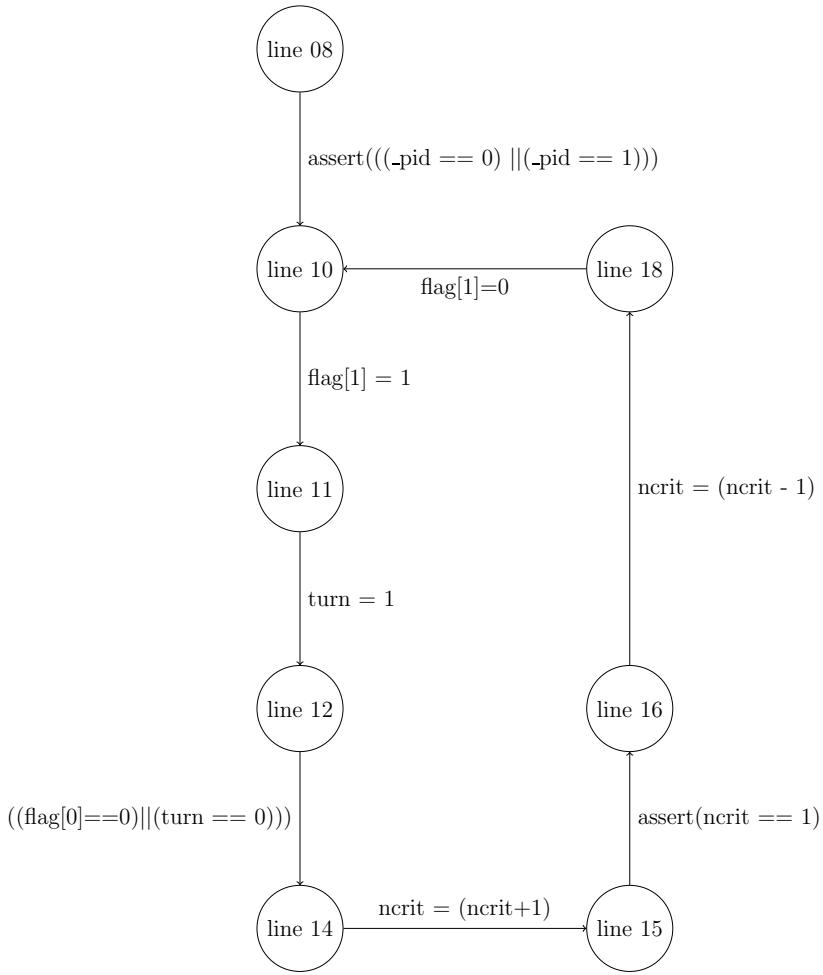
In diesem Beispiel wird der Prozesstyp user() deklariert.

Generischer Automat zum *user* Prozess



Instanzen des generischen Automaten





Approximation Produktautomat