

Es gibt unendlich viele Wege in  $G_1$  von Ecke 1 nach Ecke 3, da man z.B. die Kante  $a$  bei Knoten 1 beliebig oft nehmen kann, bevor man zu Ecke 3 geht. Die einfachen Wege sind:  $(d)$ ,  $(b, c)$ ,  $(b, e)$

$G_1$  hat die einfachen Zykel  $(a)$ , die 2 möglichen Zykel die durch die Kanten  $(c, e)$  entstehen, die 6 möglichen Zykel die durch die Kanten  $(b, c, d)$  entstehen (je 3 verschiedene Startknoten, und 2 mögliche Richtungen), gleich für die Kanten  $(b, e, d)$ . Insgesamt hat  $G_1$  also 15 einfache Zykel.

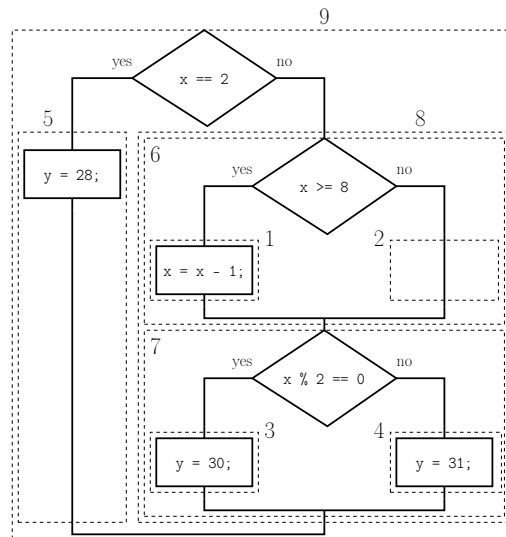
$G_2$  hat 3 möglichen einfachen Zykel und  $G_3$  hat keine Zykel.

$G_1$  und  $G_3$  sind zusammenhängend.

$G_1$  ist nicht zyklensfrei also weder ein Baum noch ein Wald.  $G_2$  ist kein ungerichteter Multi-graph, also ist die Definition nicht direkt anwendbar.  $G_3$  ist ein Baum und ein Wald.

□

5) Lösung. Wie der Flussgraph induktiv erstellt werden kann wird durch die nummerierten gestrichelten Boxen in



gezeigt. Boxen 1, 3, 4 und 5 sind Ausdrucks-Basisfälle, Box 2 ist ein leerer Basisfall. Wir haben dann Auswahlzusammensetzungsboxen 6 (aus 1 und 2), sowie 7 (aus 3 und 4), welche zu einer sequentiellen Zusammensetzungsbox 8 (aus 6 und 7) kombiniert werden. Schlussendlich

werden 5 und 8 zu einer Auswahlzusammensetzungbox 9 kombiniert, welche das gesamte Flussdiagramm darstellt.

Wir verwenden rekursive Inferenz (siehe Einführung in die Theoretische Informatik) um das Programm abzuleiten:

Wort	Variable	Regel	Rekursion
1 <code>x == 2</code>	$T$	Test	
2 <code>y = 28</code>	$S$	Ausdruck	
3 <code>y = 28;</code>	$P$	$P \rightarrow S;$	2
4 <code>x &gt;= 8</code>	$T$	Test	
5 <code>x = x - 1</code>	$S$	Ausdruck	
6 <code>x = x - 1;</code>	$P$	$P \rightarrow S;$	5
7 <code>;</code>	$P$	$P \rightarrow ;$	
8 <code>x % 2 == 0</code>	$T$	Test	
9 <code>y = 30</code>	$S$	Ausdruck	
10 <code>y = 30;</code>	$P$	$P \rightarrow S;$	9
11 <code>y = 31</code>	$S$	Ausdruck	
12 <code>y = 31;</code>	$P$	$P \rightarrow S;$	11
13 <code>if (x &gt;= 8) {x = x - 1;} else {;}</code>	$P$	$P \rightarrow \text{if}(T) \{P\} \text{ else } \{P\}$	4, 6, 7
14 <code>if (x % 2 == 0) {y=30;} else {y=31;}</code>	$P$	$P \rightarrow \text{if}(T) \{P\} \text{ else } \{P\}$	8, 10, 12
15 sequentielle Zusammenfügung zweier Programme	$P$	$P \rightarrow P P$	13, 14
16 komplettes Programm aus Angabe	$P$	$P \rightarrow \text{if}(T) \{P\} \text{ else } \{P\}$	1, 3, 15

□